

MAHATMA GANDHI MEMORIAL COLLEGE
UDUPI - 576102



PROJECT REPORT ON

“FURNITURE SHOP MANAGEMENT SYSTEM”

DEVELOPED BY

Bhakthi Prabhu

Regno: 185761911

Vaishnavi K Nayak

Regno: 185761943

Under the guidance of

“Ms. Ramya”

Dept. of Computer Science

**Submitted to the Mangalore University in partial fulfillment
Of the Award Bachelor in Computer Application**

MANGALORE UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

**MAHATMA GANDHI MEMORIAL COLLEGE
UDUPI-576102**

MAHATMA GANDHI MEMORIAL COLLEGE

UDUPI - 576102



DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that the Project entitled “Furniture Shop Management System” has been carried out by:

Bhakthi Prabhu

Regno:185761911

Vaishnavi K Nayak

Regno:185761943

students of B.C.A. under the supervision and guidance and submitted as partial fulfillment of the requirement for the award of Bachelor in Computer Application of the Mangalore University during the academic year 2020-2021.

They have carried out this project during the period May 2021 to October 2021 in our Lab. Further, it is certified to the best of our knowledge that the matter embodied in this work has not been submitted for the award of any other degree.

Ms.Ramya

Prof. Dr.Devidas Naik

Mr. M. Vishwanath Pai

(Project Guide)

(Principal)

(Head of the Department)

Submitted to the Mangalore University Practical Examination held on _____ at
Mahatma Gandhi Memorial College, Udupi.

Examiners:

1) _____

2) _____

TO WHOMEVER IT MAY CONCERN

This is to certify that project entitled "**Furniture Shop Management System**" has been carried out by

Bhakthi Prabhu

Regno:185761911

Vaishnavi K Nayak

Regno:185761943

Students of BCA, Mahatma Gandhi Memorial College, Udupi, under our guidance and supervision submitted the partial fulfillment of requirement for the award for Bachelor of computer Application of Mangalore University during the period of 2020-2021. It is certified to the best of our Knowledge.

Project Guide

Manager

Date:

Place:

ACKNOWLEDGEMENT

Behind every achievement, there is a sea of gratitude to those who have activated this project. The magnitude of this project demanded the co-operation, guidance and help of many people. We have been fortunate enough to have this in the entire task of completion of our project on “Furniture Shop Management System”.

We would like to thank our principal Prof.Dr.Devidas Naik for giving us opportunity to carry out our project.

We thank Mr.M.Vishwanath Pai, a source of inspiration and encouragement, head of the Department of Computer Science, Mahatma Gandhi Memorial College, Udupi for having permitted us to carry out our project work.

We are extremely grateful to express our overwhelming gratitude to our guide Ms.Ramya, Lecturer of Computer Science Department, Mahatma Gandhi Memorial College Udupi for giving us valuable guidance to undertake this project.

Last but not the least; we are indebted to all teaching and non-teaching staff members, Mahatma Gandhi Memorial College Udupi for making this project successful.

Thanking you

Vaishnavi K Nayak

Bhakthi Prabhu

Furniture Shop Management System

Renovate Your Home

INDEX

1. INTRODUCTION

1.1 Synopsis

1.1.1 Project Title

1.1.2 Introduction and Objective

- 1.1.3 Project Category
- 1.1.4 Tools and Platforms
- 1.1.5 Software and Hardware Requirements
- specification
- 1.1.6 Structure of the Project
 - 1.1.6.1 Modules
 - 1.1.6.2 Modules Description
- 2. CFD
- 3. DFD
- 4. ER Diagram
- 5. Database Design
- 6. Table Structure
- 7. Source Code
- 8. User Manuals
- 9. Testing
- 10. Limitations and the Future Scope
- 11. Definitions, Acronyms, Abbreviation
- 12. Bibliography

1. INTRODUCTION

In this project we provide an automated process of furniture management. The records are inserted for the automated result from the software. Various records are handled in this software like employee records, customer records, purchase and sales records, inventory details and many more.

Synopsis

Title of the project:

Furniture Shop Management System

Group number: 8

Team members: Bhakthi Prabhu-185761911

Vaishnavi K Nayak-185761943

Introduction:

The software is being developed for the automation of the furniture management by reducing the manual interferences. Since there are many modules which need to be maintained in the right order, computerization of all the details is necessary.

The user can find it easy to manipulate and retrieve data when it is automated. Thus it will help to maintain the data in an organized manner retaining the efficiency and accuracy of the data.

Objectives:

- To maintain detailed records of the customer, employees and suppliers.
- To generate invoice, pay slip to the employees and generates purchase and sales report.
- To provide an error-free and user-friendly environment.

Project category:

This project is based on a client-server database application running on a single system.

Tools/Platform:

Programming tools:

- Front end: Microsoft Visual C#
- Back end: Microsoft SQL Server 2005

Tools used:

1. Required tools:
 - Front end: Microsoft Visual Studio 2005
 - Back end: MS SQL Server 2005
2. Hardware required:
 - RAM: 512MB or more.
 - Hard disk: 1 GB or more.
 - 32/64 bit processor
3. Software required:

- Windows XP or higher version.
- Microsoft Visual Studio 2005
- Microsoft SQL Server 2005

Structure of the project

Modules used:

- Login
- Customer details
- Suppliers details
- Stock/Inventory
- Sales
 - Sales bill
 - Sales order
- Service
 - Service bill
- Purchase
 - Purchase order
 - Purchase bill
 - Purchase return
- Employee
 - Employee details
 - Attendance
 - Pay slip

Module description

• Login

This module allows the user to type a user name and password to log in. This module is used for security purposes. This facility allows login to administrators and employees.

• Customer Details

This module maintains all the details about the customers like name, address, phone number, email-id etc. We can add, delete and display the customer information.

• Supplier Details

This module maintains all the details of the suppliers who supply items to the shop. It allows the manipulation of supplier data to obtain details.

• Inventory/Stock

This module allows us to maintain and display information about the available stock. In the stock database the information gets updated as the stock is being sold or purchased.

- **Sales**

This module contains the information about the stock which is sold.

Various sub modules are:

- Sales order: This sub module maintains the information about the orders placed by the customer for a particular item.
- Sales Bill: A sales bill generates the invoice for the stocks sold to the customers and stores the information.

- **Service**

This module contains the information about the service provided to the customer after the purchase of an item.

- Service Bill: This is an invoice containing the information about the service provided to the customer.

- **Purchase**

This module maintains the details about the items purchased from the suppliers.

Various sub modules are:

- Purchase Order: This sub module maintains the information about the orders placed for the items from the suppliers.
- Purchase Bill: This is an invoice containing the information about the purchased items from the supplier.
- Purchase Return: This sub module is used to maintain details about the faulty items returned to the supplier.

- **Employee**

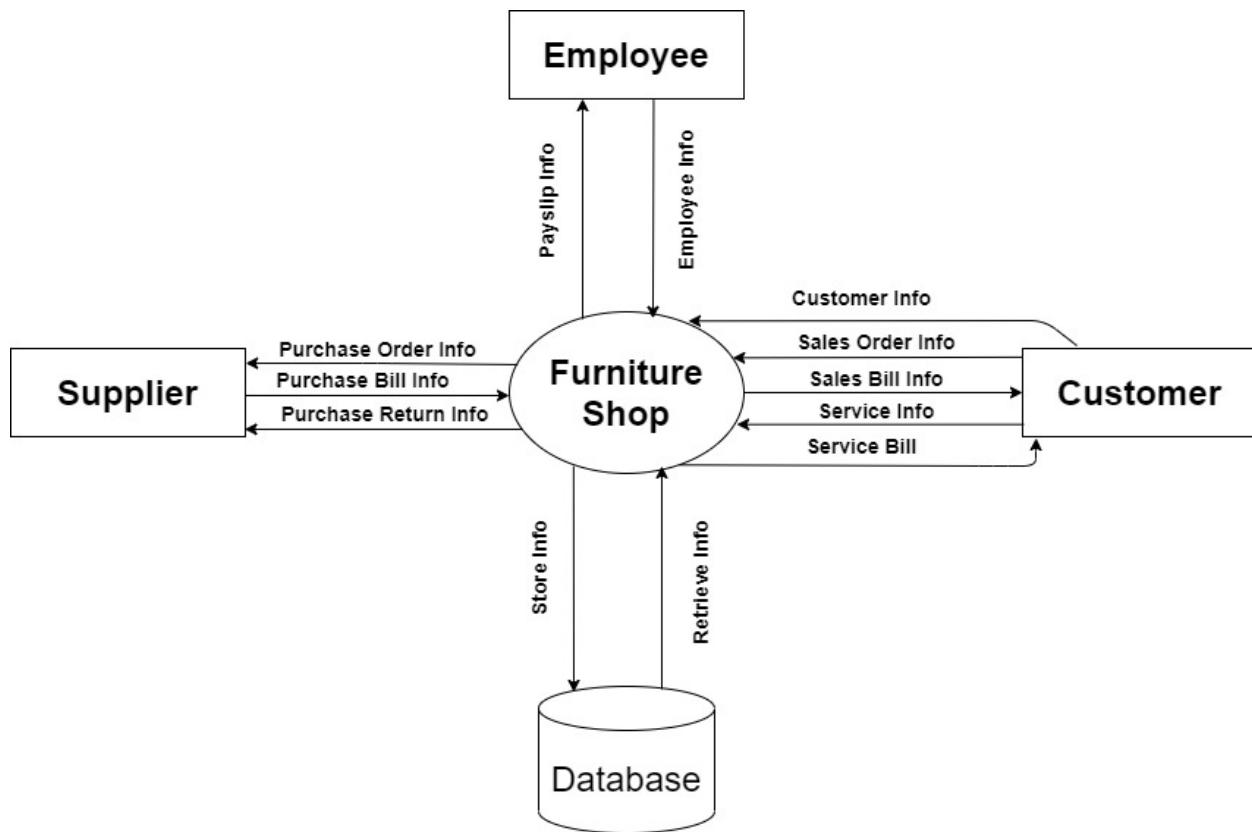
This module maintains the details about the employees in the shop.

Various sub modules are:

- Employee Details: This sub module maintains details about the employees working in the shop.
- Attendance: This sub module tracks the attendance of the employees in the shop.
- Salary: This maintains the salary details of the employees, which is to be given based on their attendance count.

2. Context Flow Diagram (CFD):

- Data Flow Diagram (DFD) start with a top level DFD called the Context Flow Diagram, which lists all the major inputs and outputs for the system. A CFD is another name for a program flowchart and will show the flow of control through the different parts of the program.
- CFD are used to describe the detailed logic of a business process or business rule.



3. Data Flow Diagrams (DFD):

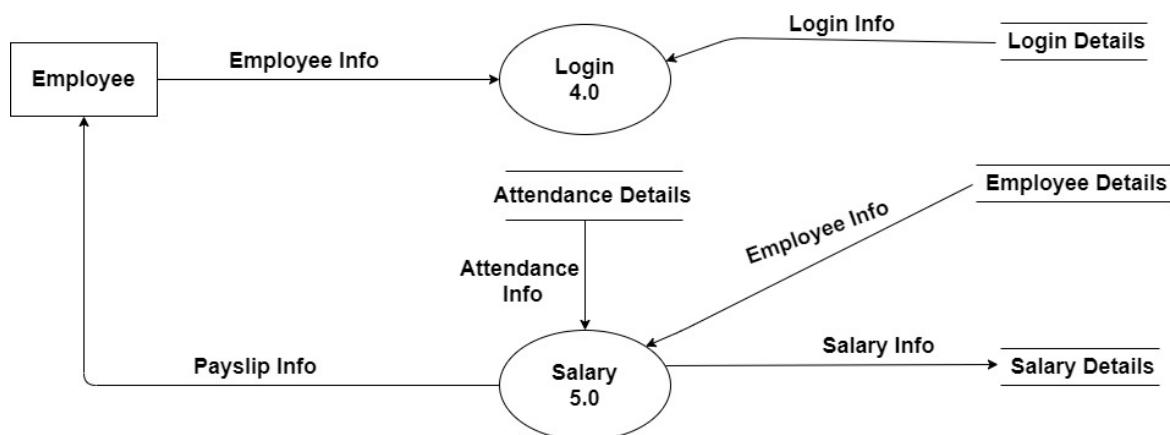
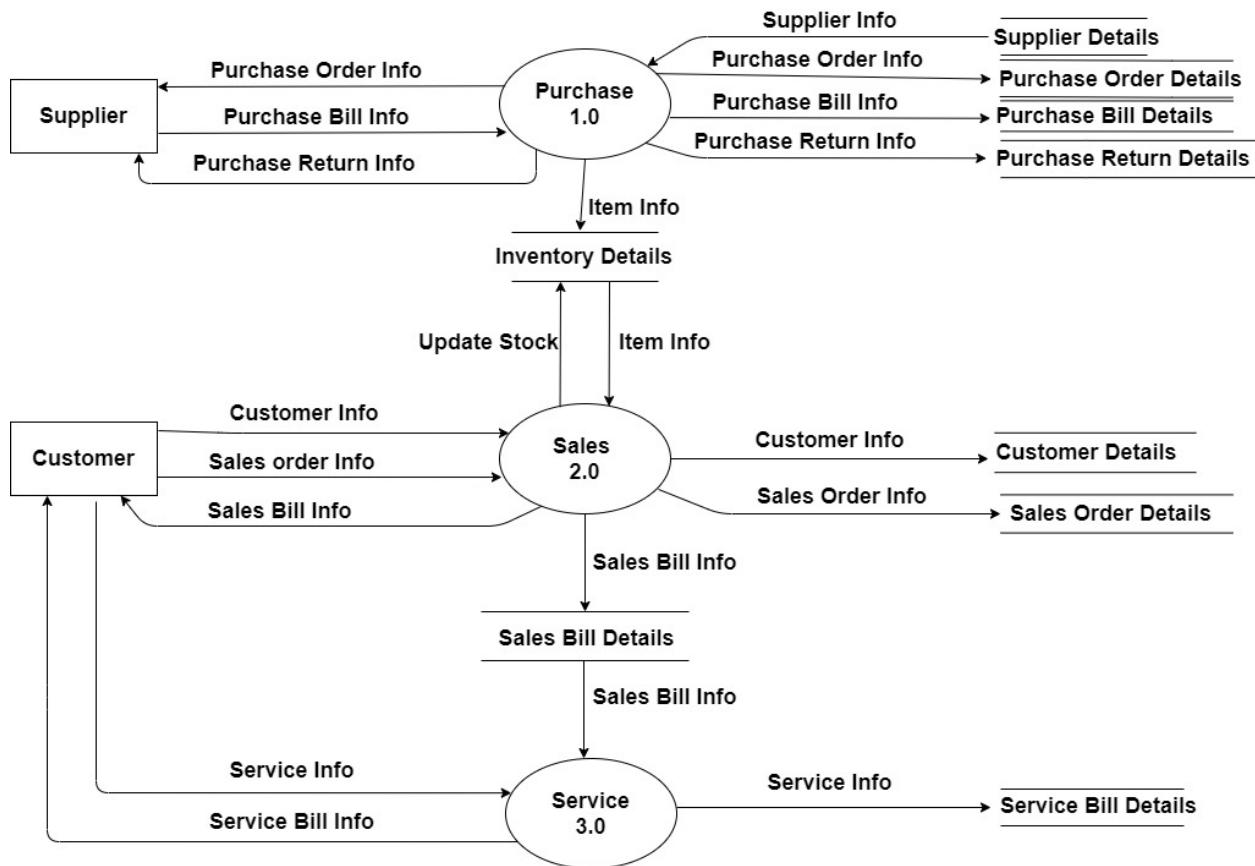
The data flow diagram describes the flow of data with the help of various levels of crystal-clearway.

The DFD serves to purpose:

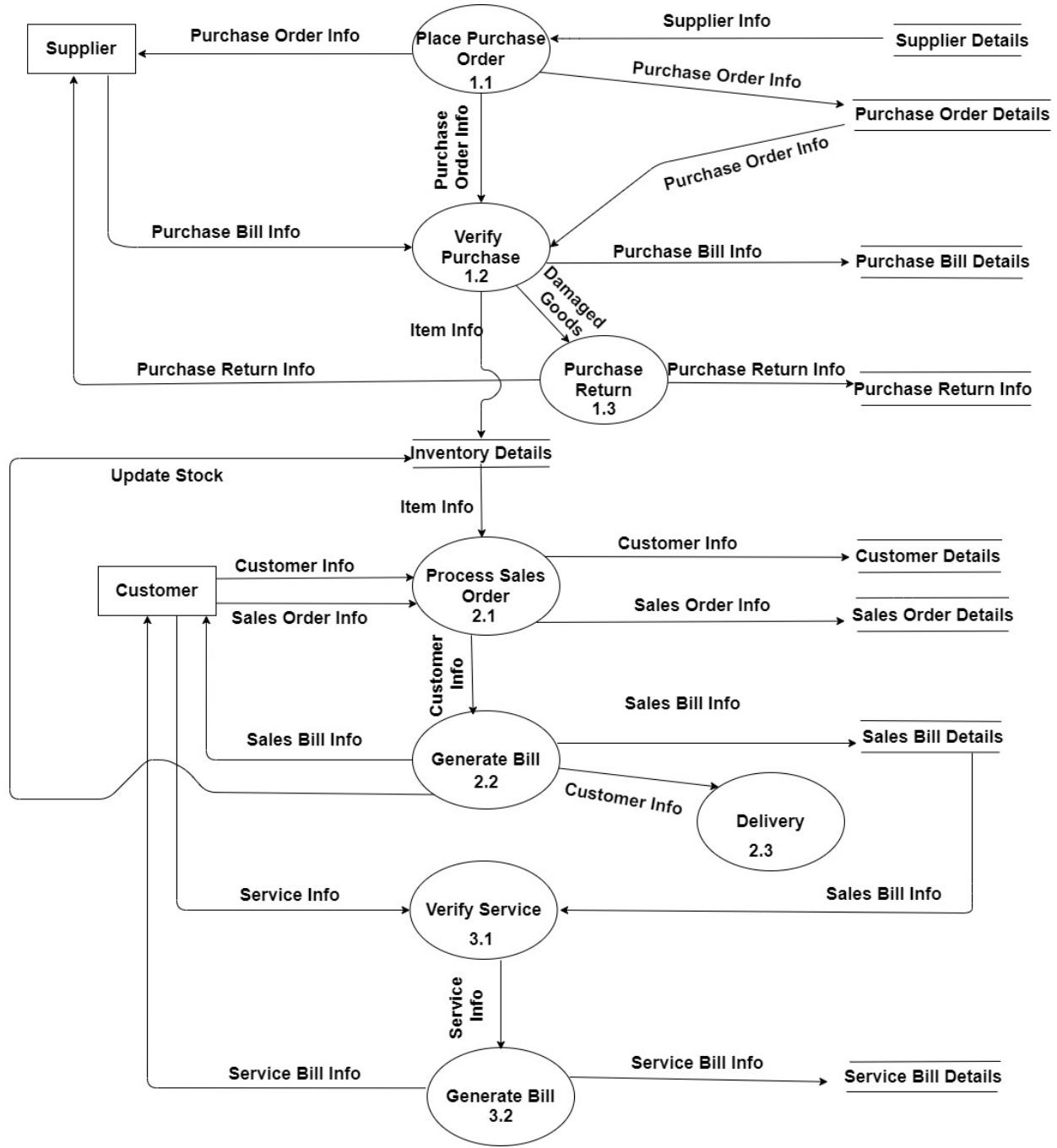
- To provide an indication of flow, data are transferred as they move through the system.
- To depict the function that performs the data flow.
- It provides additional information that is used during the analysis of the information domain and serves as the basic form modeling function.
- The DFD is also known as data flow diagram or bubble chart.

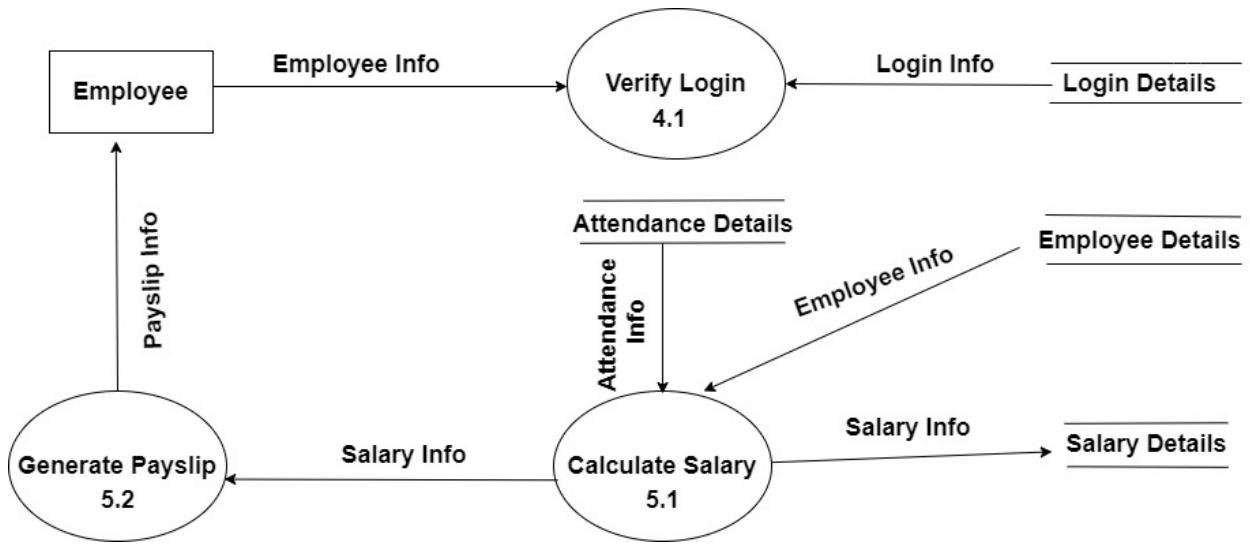
- The DFD may be used to represent a system of software of any level of abstraction. In fact DFD may be partitioned into levels that represent increasing information flow and the function details.
- There are DFD provides mechanism for functional modeling as well as information flow modeling.
- A level of DFD is also called a functional system model represents the entire software elements as a single bubble with input and output indicated incoming and outgoing arrows respectively.

Notation	Description
	Process: A circle represents a process or transforms that is applied to data or control and changes in some way.
	Dataflow: An arrow represents one or more data items or data objects.
	Source or sink: A rectangle is used to represent, system that produces, receives information produced by the software
	File or Database: The open box represents data stores, stored information used by the software.



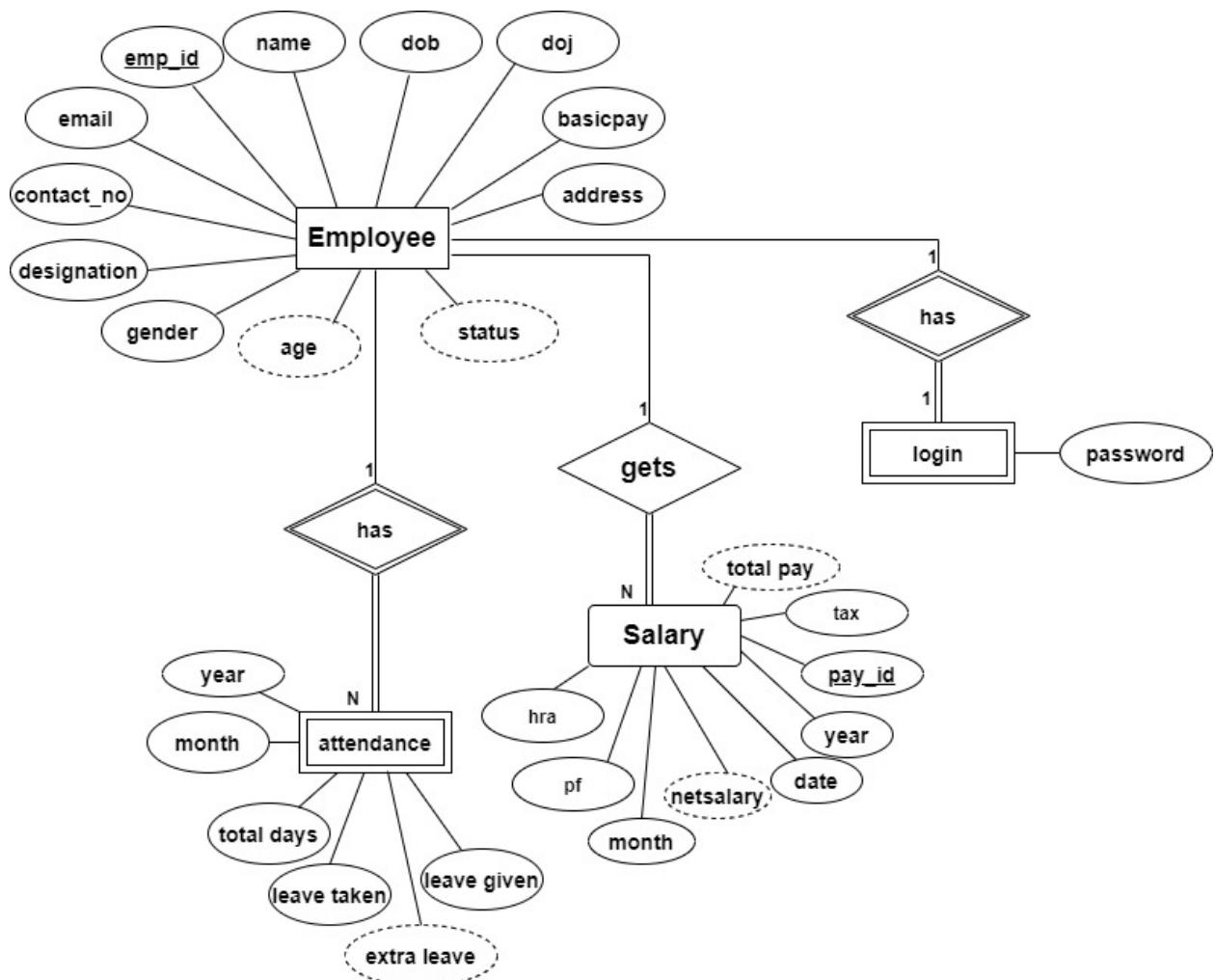
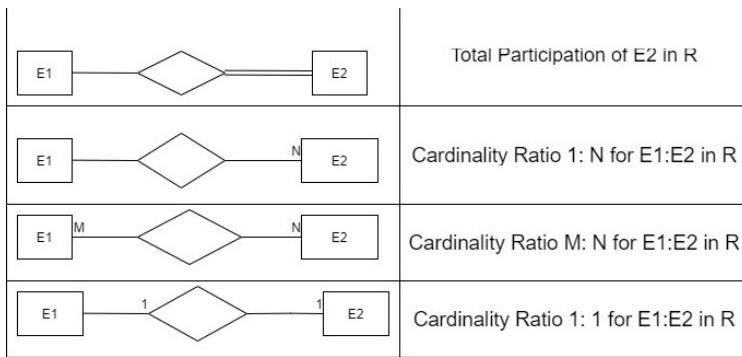
DFD Level 2:

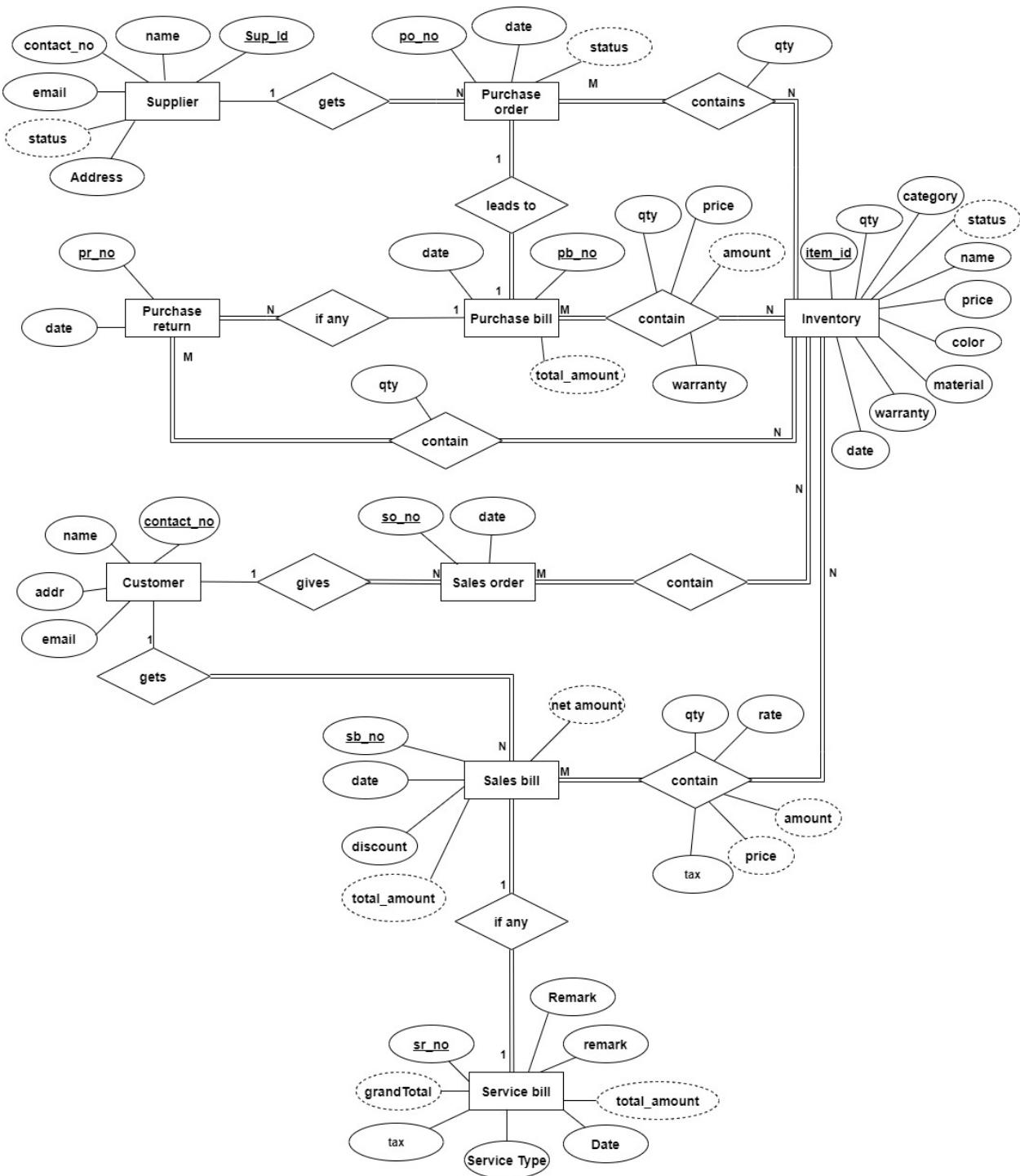




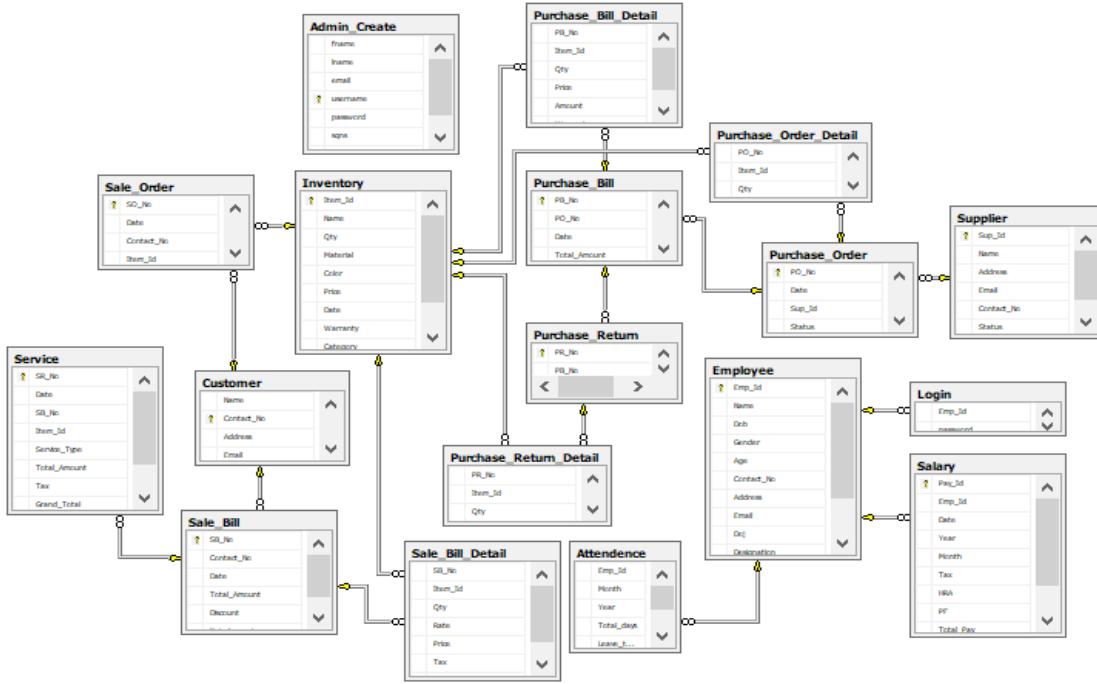
Entity Relationship Diagram (ERD):

Symbol	Meaning
rectangle	Entity
rectangle with a diagonal line	Weak Entity
line connecting an entity to an oval	Attribute
line connecting an entity to an oval with a minus sign	Key Attribute
line connecting two entities	Relationship
line connecting two entities with a diamond symbol	Identifying Relationship
entity connected to multiple other entities	Composite Attribute
entity with a dashed line	Derived Attribute
oval	Multivalued Attribute





DATABASE DESIGN:



DATA DICTIONARY:

1.Login:

Login information=@Emp_Id+ password

Emp_Id= emp+{0-9}

Password= {Legal Characters}

2.Admin Create:

Admin_Create information= @username +fname+lname+email+password+sqns+sans

username={a-z|0-9}

fname={a-z}

lname={a-z}
email={A-Z|a-z|0-9|_+@+.}
password= {Legal Characters}
sqns= {Legal Characters}
sans= {Legal Characters}

3.Employee:

Employee
Information=@Emp_Id+Name+Gender+Address+Contact_No+Email
+Designation+Basic_Pay+Dob+Doj+Age+Status

Emp_Id= emp+{0-9}

Name= {A-Z|a-z}

Gender= {A-Z|a-z}

Age={0-9}

Address= {Legal Characters}

Contact_No= {0-9}

Email={A-Z|a-z|0-9|_+@+.}

Designation={A-Z|a-z}

Basic_Pay={0-9}

Dob= {Valid Date}

Doj= {Valid Date}

Status={A-Z|a-z}

4.Attendance:

Attendance
Information=Emp_Id+Month+Year+Total_days+Extra_leave+Leave_
given+Leave_taken

Emp_Id= emp+{0-9}

Total_days= {0-9}

Year= {Valid Date}

Month= {Valid Date}

Leave_given= {0-9}

Leave_taken={0-9}

Extra_leave={0-9}

5.Inventory:

Inventory

information:@Item_Id+Name+Qty+Material+Color+Price+Date+Warranty+Category+Status

Item_Id= item+{0-9}

Name={A-Z|a-z}

Qty= {0-9}

Category={A-Z|a-z}

Date= {Valid Date}

Price= {0-9}

Material={A-Z|a-z}

Color={A-Z|a-z}

Warranty= {0-9}

Status={A-Z|a-z}

6.Supplier:

Supplierinformation=@Sup_Id+Name+Address+Email+Contact_No+Status

Sup_Id=sup+{0-9}

Name={A-Z|a-z}

Address={ Legal Characters}

Email={A-Z|a-z|0-9|_+@+.}

Contact_No={0-9}

Status={A-Z|a-z}

7.Customer:

customer information=@Contact_No+Address+Email+Name

Name={a|z}

Address = {Legal Characters}

Email={A-Z|a-z|0-9|_+@+.}

Contact_No= {0-9}

8.Purchase Bill:

Purchase billInformation=@PB_No+PO_No+Date+Total_Amount

PB_No=pb+{0-9}

PO_No=po+{0-9}

Date={ Valid Date }

Total_Amount={0-9}

9.Purchase Bill Detail:

Purchase Bill Detail

Information=PB_No+Item_Id+Qty+Price+Amount+Warranty

PB_No=pb+{0-9}

Item_Id=item+{0-9}

Qty={0-9}

Price={0-9}

Amount={0-9}

Warranty={0-9}

10.Purchase Order:

Purchase Order Information=@PO_No+Date+Sup_Id+Status

PO_Id=po+{0-9}

Date={Valid date}

Sup_Id=sup+{0-9}

Status={0-9}

11.Purchase Order Detail:

Purchase Order Detail Information:PO_No+Item_Id+Qty

PO_Id=po+{0-9}

Qty={0-9}

Item_Id=item+{0-9}

12.Purchase Return:

Purchase Return Information=@PR_No+PB_No+Date

PR_No=pr+{0-9}

PB_No=pb+{0-9}

Date={Valid date}

13.Purchase Return Detail:

Purchase Return Detail Information=PR_No+Item_Id+Qty

PR_No=pr+{0-9}

Item_Id=item+{0-9}

Qty={0-9}

14.Salary:

Salary

Information:@Pay_Id+Emp_Id+Date+Year+Month+Tax+HRA+PF+Total_Pay+Net_Pay

Pay_Id=pay+{0-9}

Emp_Id=emp+{0-9}

Date={Valid date}

Year={Valid date}

Month={Valid date}

Tax={0-9}

HRA={0-9}

PF={0-9}

Total_Pay={0-9}

Net_Pay={0-9}

15.Sales Bill:

Sales Bill

Information=@SB_No+Contact_No+Date+Total_Amount+Discount+Net_Amount

SB_No=sb+{0-9}

Contact_No={0-9}

Date={Valid Date}

Total_Amount={0-9}

Discount={0-9}

Net_Amount={0-9}

16.Sales Bill Detail:

Sales Bill Detail

Information=SB_No+Item_Id+Qty+Rate+Price+Tax+Amount

SB_No=sb+{0-9}

Item_Id=item+{0-9}

Qty={0-9}

Rate={0-9}

Price={0-9}

Tax={0-9}

Amount={0-9}

17.Sales Order:

Sales Order Information=@SO_No+Date+Cotact_No+Item_Id

SO_No= so+{0-9}

Date={Valid Date}

Contact_No={0-9}

Item_Id= item+{0-9}

18.Service:

Service

Information=@SR_No+Date+SB_No+Item_Id+Service_Type+Total_Amount+Tax+Grand_Total+Remark

SR_No= sr+{0-9}

Date={Valid Date}

Item_Id= item+{0-9}

SB_No= sb+{0-9}

Service_Type={Legal Characters}

Total_Amount={0-9}

Tax={0-9}

Grand_Total={0-9}

Remark={Legal Characters}

DATA TABLE:

1. Supplier

Field Name	Data Type	Size	Constraints
Sup_Id	nvarchar	50	Primary Key
Name	nvarchar	50	Allow Nulls
Address	nvarchar	MAX	Allow Nulls
Email	nvarchar	50	Allow Nulls
Contact_No	char	10	Allow Nulls
Status	nvarchar	50	Allow Nulls

2. Purchase Order

Field Name	Data Type	Size	Constraints
Po_No	nvarchar	50	Primary Key
Date	datetime		Allow Nulls
Sup_Id	nvarchar	50	Foreign Key
Status	nvarchar	50	Allows Null

3. Purchase Order Detail

Field Name	Data Type	Size	Constraints
Po_No	nvarchar	50	Foreign Key
Qty	int		Allows Null
Item_Id	nvarchar	50	Foreign Key

4. Inventory

Field Name	Data Type	Size	Constraints
<u>Item_Id</u>	nvarchar	50	Primary Key
Name	nvarchar	50	Allow Nulls
Qty	int		Allow Nulls
Material	nvarchar	50	Allow Nulls
Color	nvarchar	50	Allow Nulls
Price	decimal	20,2	Allow Nulls
Date	datetime		Allow Nulls
Warranty	decimal	18,1	Allow Nulls
Category	nvarchar	50	Allow Nulls
Status	nvarchar	50	Allow Nulls

5. Purchase Bill

Field Name	Data Type	Size	Constraints
<u>PB_No</u>	nvarchar	50	Primary Key
PO_No	nvarchar	50	Foreign Key
Date	datetime		Allow Nulls
Total_Amount	decimal	10,2	Allow Nulls

6. Purchase Bill Detail

Field Name	Data Type	Size	Constraints
<u>PB_No</u>	nvarchar	50	Foreign Key
<u>Item_Id</u>	nvarchar	50	Foreign Key
Qty	int		Allows Null
Price	decimal	18,2	Allows Null

Amount	decimal	18,2	Allows Null
Warranty	decimal	18,2	Allows Null

7. Purchase Return

Field Name	Data Type	Size	Constraints
<u>PR_No</u>	nvarchar	50	Primary Key
PB_No	nvarchar	50	Foreign Key
Date	datetime		Allow Nulls

8. Purchase Return Detail

Field Name	Data Type	Size	Constraints
<u>PR_No</u>	nvarchar	50	Foreign Key
Item_Id	nvarchar	50	Foreign Key
Qty	int		Allow Nulls

9. Customer

Field Name	Data Type	Size	Constraints
<u>Contact_No</u>	char	10	Primary Key
Name	nvarchar	50	Allows Null
Address	nvarchar	MAX	Allows Null
Email	nvarchar	50	Allows Null

10. Sales Order

Field Name	Data Type	Size	Constraints
<u>SO_No</u>	nvarchar	50	Primary Key
Date	datetime		Allow Nulls
Contact_No	char	10	Foreign Key
Item_Id	nvarchar	50	Foriegn Key

11. Sales Bill

Field Name	Data Type	Size	Constraints
<u>SB_No</u>	nvarchar	50	Primary Key
Date	datetime		Allow Nulls
Contact_No	char	10	Allow Nulls
Total_Amount	decimal	18,2	Allow Nulls
Discount	decimal	5,2	Allow Nulls
Net_Amount	decimal	18,2	Allow Nulls

12. Sales Bill Detail

Field Name	Data Type	Size	Constraints
SB_No	nvarchar	50	Foreign Key
Qty	int		Allow Nulls
Item_Id	nvarchar	50	Foreign Key
Rate	decimal	18,2	Allow Nulls
Price	decimal	18,2	Allow Nulls
Tax	decimal	18,2	Allow Nulls
Amount	decimal	18,2	Allow Nulls

13. Service

Field Name	Data Type	Size	Constraints
<u>SR_No</u>	nvarchar	50	Primary Key
SB_No	nvarchar	50	Foreign Key
Item_Id	nvarchar	50	Foreign Key
Date	datetime		Allow Nulls
Total_Amount	decimal	18,2	Allow Nulls
Service_Type	nvarchar	50	Allow Nulls

Tax	decimal	18,2	Allow Nulls
Grand_Total	decimal	18,2	Allow Nulls
Remark	nvarchar	50	Allow Nulls

14. Employee

Field Name	Data Type	Size	Constraints
<u>Emp_Id</u>	nvarchar	50	Primary Key
Name	nvarchar	50	Allow Nulls
Dob	datetime		Allow Nulls
Age	int		Allow Nulls
Contact_No	char	10	Allow Nulls
Gender	nvarchar	50	Allow Nulls
Status	nvarchar	50	Allow Nulls
Address	nvarchar	MAX	Allow Nulls
Doj	datetime		Allow Nulls
Email	nvarchar	MAX	Allow Nulls
Designation	nvarchar	50	Allow Nulls
Basic_Pay	decimal	20,2	Allow Nulls

15. Salary

Field Name	Data Type	Size	Constraints
<u>Pay_Id</u>	nvarchar	50	Primary Key
Year	datetime		Allows Null
Emp_Id	nvarchar	50	Primary Key
Date	datetime		Allows Null
Month	datetime		Allows Null
Tax	decimal	18,2	Allow Nulls

HRA	decimal	18,2	Allow Nulls
PF	decimal	18,2	Allow Nulls
Total_Pay	decimal	18,2	Allow Nulls
Net_Pay	decimal	18,2	Allow Nulls

16. Attendance

Field Name	Data Type	Size	Constraints
Emp_Id	nvarchar	50	Foreign Key
Month	datetime		Allow Nulls
Year	datetime		Allow Nulls
Total_days	int		Allow Nulls
Leave_taken	int		Allow Nulls
Leave_given	int		Allow Nulls
Extra_leave	int		Allow Nulls

17. Admin Create

Field Name	Data Type	Size	Constraints
fname	nvarchar	50	Allow Nulls
lname	nvarchar	50	Allow Nulls
email	nvarchar	50	Allow Nulls
<u>username</u>	nvarchar	50	Primary Key
password	nvarchar	50	Allow Nulls
sqns	nvarchar	50	Allow Nulls
sans	nvarchar	50	Allow Nulls

18. Login

Field Name	Data Type	Size	Constraints
Emp_Id	nvarchar	50	Foreign Key
password	nvarchar	50	Allow Nulls

Coding:

Form 1 :

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace WindowsApplication
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            //page load
            this.timer1.Start();
            progressBar1.Style = ProgressBarStyle.Continuous;
            progressBar1.Step = 10;
        }

        private void Loadnext()
        {
            //loadnext function
            Connect c = new Connect();
            c.cmd.CommandText = "select username from Admin_Create";
            string username = Convert.ToString(c.cmd.ExecuteScalar());
            if (username == "")
                new Admin_Create().Show();
            else
                new Login_User().Show();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
    
```

```

        //timer tick
        this.progressBar1.Increment(1);
        loading.Text = "Loading..." + progressBar1.Value + "%";
        if (this.progressBar1.Value == 100)
        {
            timer1.Stop();
            this.Hide();
            Loadnext();
        }
    }

    private void Form1_FormClosing(object sender, FormClosingEventArgs e)
    {
        //close button
        Application.Exit();
    }
}

```

Admin Create:

```

using System.Windows.Forms;
using System.Net.Mail;

namespace WindowsApplication
{
    public partial class Admin_Create : Form
    {
        public Admin_Create()
        {
            InitializeComponent();
        }

        private void Admin_Create_Load(object sender, EventArgs e)
        {
            this.Text = "Admin Creation";
        }

        private void temail_TextChanged(object sender, EventArgs e)
        {
            //email text changed
            errorProvider1.Clear();
        }

        private void tRetype_KeyUp(object sender, KeyEventArgs e)
        {
            //Retypepassword_KeyUp
            if (tpassword.Text == tRetype.Text)
            {
                msg.Visible = true;
                msg.ForeColor = Color.Lime;
                msg.Text = "Matched!";
            }
            else
            {
                msg.Visible = true;

```

```

        msg.ForeColor = Color.Red;
        msg.Text = "Not Matched!";
    }
}

private void tpassword_TextChanged(object sender, EventArgs e)
{
    if (tpassword.Text == tRetype.Text)
    {
        msg.Visible = true;
        msg.ForeColor = Color.Lime;
        msg.Text = "Matched!";
    }
    else
    {
        msg.Visible = true;
        msg.ForeColor = Color.Red;
        msg.Text = "Not Matched!";
    }
}

private void signup_Click(object sender, EventArgs e)
{
    //signup
    try
    {
        if (comboBox1.SelectedItem.ToString() == "" || tsans.Text == ""
            || tlname.Text == "" || tfname.Text == "" || temail.Text == "" ||
            tpassword.Text == "" || tRetype.Text == "" || comboBox1.Text == "" ||
            tsans.Text == "" || comboBox1.Text == "Select")
            throw new ArgumentException();
        else
        {
            new MailAddress(this.temail.Text);
            if (tpassword.Text.Length <= 7)
                MessageBox.Show("Password Must be atleast 8 characters
long!", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            else if (tpassword.Text != tRetype.Text)
            {
                msg.Visible = true;
                msg.ForeColor = Color.Red;
                msg.Text = "Not Matched!";
            }
            else
            {
                Connect c = new Connect();
                c.cmd.CommandText = "insert into Admin_Create
values('" + tfname.Text + "','" + tlname.Text + "','" + temail.Text + "','" +
tusername.Text + "','" + tpassword.Text + "','" + comboBox1.SelectedItem.ToString() + "','" + tsans.Text + "')";
                c.cmd.ExecuteNonQuery();
                this.Hide();
                new Login_User().Show();
            }
        }
    }
    catch (FormatException)
}

```

```

        {
            errorProvider1.SetError(temail, "Enter valid format Email");
        }
        catch (ArgumentException)
        {
            MessageBox.Show("Enter all the details", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
        catch (NullReferenceException)
        {
            MessageBox.Show("Enter all the details", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }

    private void clear_Click(object sender, EventArgs e)
    {
        //clear
        tfname.Text = "";
        tlname.Text = "";
        temail.Text = "";
        tusername.Text = "";
        tpassword.Text = "";
        tRetype.Text = "";
        tsans.Text = "";
        comboBox1.ResetText();
        comboBox1.Text = "Select";
        msg.Visible = false;
        msg.ForeColor = Color.Red;
        tusername.Text = "admin";
        tusername.ReadOnly = true;
    }

    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        msg.Visible = false;
    }

    private void tfname_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar == (char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || e.KeyChar == '.');
    }

    private void tlname_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar == (char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || e.KeyChar == '.');
    }
}

```

Login User:

```

using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsApplication
{
    public partial class Login_User : Form
    {
        Connect c = new Connect();
        String username, password, usern, userp;
        int cnt = 1;
        public Login_User()
        {
            InitializeComponent();
        }

        private void Login_Admin_Load(object sender, EventArgs e)
        {
            //page load
            this.Text = "Login";
            textBox2.Text = "PASSWORD";
            textBox1.Text = "USERNAME";
        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {
            //password textchanged
            errorProvider2.Clear();
            label2.Text = "";
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            //username textchanged
            errorProvider1.Clear();
            label2.Text = "";
        }

        private void textBox1_Click(object sender, EventArgs e)
        {
            //username click
            if (textBox1.Text == "USERNAME")
                textBox1.Text = "";
            textBox1.ForeColor = Color.Black;
        }

        private void textBox2_Click(object sender, EventArgs e)
        {
            //password click
            if (textBox2.Text == "PASSWORD")
                textBox2.Text = "";
            eye.Visible = true;
            hide.Visible = false;
            textBox2.TabStop = true;
            textBox2.ForeColor = Color.Black;
            textBox2.PasswordChar = '*';
        }
    }
}

```

```

        }

    private void hide_Click(object sender, EventArgs e)
    {
        //hide password
        textBox2.PasswordChar = '.';
        eye.Visible = true;
        hide.Visible = false;
    }

    private void eye_Click(object sender, EventArgs e)
    {
        //show password
        textBox2.PasswordChar = '\0';
        eye.Visible = false;
        hide.Visible = true;
    }

    private void login_MouseHover(object sender, EventArgs e)
    {
        //login_MouseHover
        toolTip1.SetToolTip(login, "Login");
    }

    private void notyou_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        //not you
        textBox1.Clear();
        textBox2.Clear();
        textBox2.Text = "PASSWORD";
        textBox1.Text = "USERNAME";
        textBox1.ForeColor = Color.Gray;
        textBox2.ForeColor = Color.Gray;
        textBox2.PasswordChar = '\0';
    }

    public void info()
    {
        //info function
        c.cmd.CommandText = "Select username from Admin_Create";
        username = c.cmd.ExecuteScalar().ToString();

        c.cmd.CommandText = "Select password from Admin_Create";
        password = c.cmd.ExecuteScalar().ToString();
    }

    private void login_Click(object sender, EventArgs e)
    {
        info();

        SqlCommand cmd = new SqlCommand("Select * from Login where
Emp_Id=' " + textBox1.Text + "' and password=' " + textBox2.Text + " ', c.cn");
        SqlDataAdapter adp = new SqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        adp.Fill(dt);
    }
}

```

```

        SqlCommand cmd1 = new SqlCommand("Select * from Employee where
Emp_Id=' " + textBox1.Text + " ' and Status='Locked' or Emp_Id=' " +
textBox1.Text + " ' and Status='Inactive'", c.cn);
        SqlDataAdapter adp1 = new SqlDataAdapter(cmd1);
        DataTable dt1 = new DataTable();
        adp1.Fill(dt1);

        if (textBox1.Text == "" || textBox1.Text == "USERNAME")
            errorProvider1.SetError(textBox1, "Required!");
        else if (textBox2.Text == "" || textBox2.Text == "PASSWORD")
            errorProvider2.SetError(textBox2, "Required!");
        else if (textBox2.Text == password && textBox1.Text == username)
        {
            this.Hide();
            new Nav_Admin(textBox1.Text).Show();
        }
        else if (dt.Rows.Count > 0)
        {
            for (int i = 0; i < dt.Rows.Count; i++)
            {
                usern = dt.Rows[i]["Emp_Id"].ToString();
                userp = dt.Rows[i]["password"].ToString();
                if (textBox2.Text == userp && textBox1.Text == usern)
                {
                    if (dt1.Rows.Count > 0)
                    {
                        MessageBox.Show("You cannot login here because
your account is locked. \nContact admin for more!", "", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                    else
                    {
                        this.Hide();
                        new Nav_User(usern).Show();
                    }
                }
            }
        }
        else
        {
            c.cmd.CommandText = "Select Emp_Id from Login where
Emp_Id=' "+textBox1.Text+" '";
            try
            {
                if (textBox1.Text == c.cmd.ExecuteScalar().ToString())
                {
                    cnt++;
                    if (dt1.Rows.Count > 0)
                    {
                        MessageBox.Show("You cannot login here because
your account is locked. \nContact admin for more!", "", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                    else if (cnt > 3)
                    {
                        c.cmd.CommandText = "update Employee set
status='Locked' where Emp_Id=' " + textBox1.Text + " '";

```

```

                c.cmd.ExecuteNonQuery();
                MessageBox.Show("Account has been locked.\nContact
the Administrator.", "", MessageBoxButtons.OK, MessageBoxIcon.Stop);
            }
            else
            {
                label2.Visible = true;
                label2.Text = "Invalid Username / Password!";
            }
        }
        else
        {
            label2.Visible = true;
            label2.Text = "Invalid Username / Password!";
        }
    }
    catch (Exception)
    {
        label2.Visible = true;
        label2.Text = "Invalid Username / Password!";
    }
}
}

private void forgot_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    try
    {
        info();
        c.cmd.CommandText = "Select Emp_Id from Login where Emp_Id='"
+ textBox1.Text + "'";
        if (textBox1.Text == "" || textBox1.Text == "USERNAME")
            errorProvider1.SetError(textBox1, "Required!");
        else if (textBox1.Text == username)
        {
            this.Hide();
            new Password_Reset().Show();
        }
        else if (textBox1.Text == c.cmd.ExecuteScalar().ToString())
        {
            MessageBox.Show("Contact Admin If you forgot your
password", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            label2.Visible = true;
            label2.Text = "Invalid Username / Password!";
        }
    }
    catch (Exception)
    {
        label2.Visible = true;
        label2.Text = "Invalid Username / Password!";
    }
}
}

```

```
}
```

Admin Dash:

```
using System.Windows.Forms;

namespace WindowsApplication
{
    public partial class Admin_Dash : Form
    {
        Connect c = new Connect();
        public Admin_Dash()
        {
            InitializeComponent();
        }

        private void Admin_Dash_Load(object sender, EventArgs e)
        {

            try
            {
                c.cmd.CommandText = "Select count(*) from Customer";
                tcust.Text = c.cmd.ExecuteScalar().ToString();
                c.cmd.CommandText = "Select count(*) from Supplier where
Status='Active'";
                tsup.Text = c.cmd.ExecuteScalar().ToString();
                c.cmd.CommandText = "Select count(*) from Employee where
Status='Active'";
                temp.Text = c.cmd.ExecuteScalar().ToString();
                c.cmd.CommandText = "Select count(*) from Sale_Bill";
                int s = Convert.ToInt32(c.cmd.ExecuteScalar());
                c.cmd.CommandText = "Select count(*) from Purchase_Order where
Status='Invoiced'";
                int p = Convert.ToInt32(c.cmd.ExecuteScalar());
                tsales.Text = s.ToString();
                ttransactions.Text = (s + p).ToString();
                c.cmd.CommandText = "select Qty from Inventory where
Status='Active' and Qty is not null";
                c.cmd.CommandText = "Select sum(Qty) from Inventory";
                titems.Text = c.cmd.ExecuteScalar().ToString();
                c.cmd.CommandText = "Select count(*) from Sale_Order";
                tsalesorder.Text = c.cmd.ExecuteScalar().ToString();
                c.cmd.CommandText = "Select sum(Total_Amount) from Sale_Bill
where Net_Amount is not null";
                netrevenue.Text = c.cmd.ExecuteScalar().ToString();
                if (netrevenue.Text == "")
                    netrevenue.Text = "0";
                double nr, pr;
                c.cmd.CommandText = "Select sum(Net_Amount) from Sale_Bill
where Net_Amount is not null";
                nr = Convert.ToDouble(c.cmd.ExecuteScalar());
                c.cmd.CommandText = "Select sum(Rate) from Sale_Bill_Detail
where Rate is not null";
                pr = Convert.ToDouble(c.cmd.ExecuteScalar());
                double tt = nr - pr;
                if (tt < 0)
                    netprofit.Text = "0";
            }
        }
    }
}
```

```

        else
            netprofit.Text = tt.ToString();

    }
    catch (Exception)
    {
    }
}

}
}

```

Nav Admin:

```

using System.Windows.Forms;

namespace WindowsApplication
{
    public partial class Nav_Admin : Form
    {
        string username;
        public Nav_Admin(string str)
        {
            username = str;
            InitializeComponent();
        }

        private void Nav_Admin_Load(object sender, EventArgs e)
        {
            //page load
            this.Size = new Size(960, 720);
            panelmain.Location = new Point(104, 46);
            panelleft.BackColor = Color.FromArgb(46,46,46);
            panelbottom.Visible = false;
            panelsale.Visible = false;
            this.Text = "Admin Dashboard";
            Admin_Dash frm = new Admin_Dash();
            frm.TopLevel = false;
            this.panelmain.Controls.Add(frm);
            frm.Show();
        }

        private void Nav_Admin_FormClosing(object sender, FormClosingEventArgs e)
        {
            //close button
            if (MessageBox.Show("Are you sure want to Exit?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
                e.Cancel = false;
            else
                e.Cancel = true;
        }

        private void expand_Click(object sender, EventArgs e)
        {

```

```

//expand
if (panelsub.Visible == true)
{
    panelsub.Visible = false;
}
else
{
    panelsub.Visible = true;
    panelsub.BringToFront();
}
}

private void profile_Click(object sender, EventArgs e)
{
    //profile
    this.Size = new Size(960, 720);
    panelmain.Location = new Point(104, 46);
    panelsub.Visible = false;
    panelbottom.Visible = false;
    panelsale.Visible = false;
    btnsupp.BackColor = Color.Transparent;
    btn_sales.BackColor = Color.Transparent;
    btn_pur.BackColor = Color.Transparent;
    btn_emp.BackColor = Color.Transparent;
    btn_display.BackColor = Color.Transparent;
    btn_service.BackColor = Color.Transparent;
    btncust.BackColor = Color.Transparent;
    btnstock.BackColor = Color.Transparent;
    btn_logout.BackColor = Color.Transparent;
    this.Text = "Profile Page";

    panelleft.Cursor = Cursors.WaitCursor;
    this.panelmain.Controls.Clear();
    Profile frm = new Profile();
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
    panelleft.Cursor = Cursors.Default;
}

private void btn_logout_Click(object sender, EventArgs e)
{
    //logout
    this.Hide();
    new Login_User().Show();
}

private void fadmindash()
{
    //admin dashboard
    this.Size = new Size(960, 720);
    panelmain.Location = new Point(104, 46);
    this.Text = "Admin Dashboard";
    panelsub.Visible = false;
    panelsale.Visible = false;
    panelbottom.Visible = false;
    btnsupp.BackColor = Color.Transparent;
}

```

```

        btn_sales.BackColor = Color.Transparent;
        btn_pur.BackColor = Color.Transparent;
        btn_emp.BackColor = Color.Transparent;
        btncust.BackColor = Color.Transparent;
        btnstock.BackColor = Color.Transparent;
        btn_logout.BackColor = Color.Transparent;
        btn_display.BackColor = Color.Transparent;

        this.Text = "Admin Dashboard";
        panelleft.Cursor = Cursors.WaitCursor;
        this.panelmain.Controls.Clear();
        Admin_Dash frm = new Admin_Dash();
        frm.TopLevel = false;
        this.panelmain.Controls.Add(frm);
        frm.Show();
        panelleft.Cursor = Cursors.Default;
    }

private void fsupplier()
{
    //supplier
    this.Size = new Size(960, 720);
    panelmain.Location = new Point(104, 46);
    panelsub.Visible = false;
    panelbottom.Visible = false;
    panelsale.Visible = false;
    btnsupp.BackColor = Color.FromArgb(33, 33, 33);
    btn_sales.BackColor = Color.Transparent;
    btn_pur.BackColor = Color.Transparent;
    btn_emp.BackColor = Color.Transparent;
    btn_service.BackColor = Color.Transparent;
    btnstock.BackColor = Color.Transparent;
    btn_logout.BackColor = Color.Transparent;
    btn_display.BackColor = Color.Transparent;
    this.Text = "Supplier Page";

    panelleft.Cursor = Cursors.WaitCursor;
    this.panelmain.Controls.Clear();
    Supplier frm = new Supplier();
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
    panelleft.Cursor = Cursors.Default;
}

private void femployee()
{
    //employee page
    this.Size = new Size(1136, 720);
    panelmain.Location = new Point(104, 46);
    panelmain.Size = new Size(851, 645);
    panelsub.Visible = false;
    panelsale.Visible = false;
    panelbottom.Visible = false;
    btnsupp.BackColor = Color.Transparent;
    btn_sales.BackColor = Color.Transparent;
    btn_pur.BackColor = Color.Transparent;
}

```

```

        btn_emp.BackColor = Color.FromArgb(33, 33, 33);
        btn_service.BackColor = Color.Transparent;
        btncust.BackColor = Color.Transparent;
        btnstock.BackColor = Color.Transparent;
        btn_display.BackColor = Color.Transparent;
        btn_logout.BackColor = Color.Transparent;
        this.Text ="Employee Page - Add New";

        panelleft.Cursor = Cursors.WaitCursor;
        this.panelmain.Controls.Clear();
        Emp_Add_New frm = new Emp_Add_New();
        frm.TopLevel = false;
        this.panelmain.Controls.Add(frm);
        frm.Show();
        panelleft.Cursor = Cursors.Default;
    }

private void fcustomer()
{
    //customer page
    this.Size = new Size(960, 720);
    panelmain.Location = new Point(104, 46);
    panelsub.Visible = false;
    panelsale.Visible = false;
    panelbottom.Visible = false;
    btnsupp.BackColor = Color.Transparent;
    btn_sales.BackColor = Color.Transparent;
    btn_pur.BackColor = Color.Transparent;
    btn_emp.BackColor = Color.Transparent;
    btn_service.BackColor = Color.Transparent;
    btncust.BackColor = Color.FromArgb(33, 33, 33);
    btnstock.BackColor = Color.Transparent;
    btn_logout.BackColor = Color.Transparent;
    btn_display.BackColor = Color.Transparent;
    this.Text = "Customer Page";

    panelleft.Cursor = Cursors.WaitCursor;
    this.panelmain.Controls.Clear();
    Customer frm = new Customer();
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
    panelleft.Cursor = Cursors.Default;
}

private void fservice()
{
    //service page
    this.Size = new Size(960, 720);
    panelmain.Location = new Point(104, 46);
    panelmain.Size = new Size(851, 645);
    panelsub.Visible = false;
    panelsale.Visible = false;
    panelbottom.Visible = false;
    btnsupp.BackColor = Color.Transparent;
    btn_sales.BackColor = Color.Transparent;
    btn_pur.BackColor = Color.Transparent;
}

```

```

        btn_emp.BackColor = Color.Transparent;
        btn_service.BackColor = Color.FromArgb(33, 33, 33);
        btncust.BackColor = Color.Transparent;
        btnstock.BackColor = Color.Transparent;
        btn_logout.BackColor = Color.Transparent;
        btn_display.BackColor = Color.Transparent;
        this.Text = "Service Page";

        panelleft.Cursor = Cursors.WaitCursor;
        this.panelmain.Controls.Clear();
        Service frm = new Service();
        frm.TopLevel = false;
        this.panelmain.Controls.Add(frm);
        frm.Show();
        panelleft.Cursor = Cursors.Default;
    }

    private void finventory()
    {
        //inventory page
        this.Size = new Size(960, 720);
        panelmain.Location = new Point(104, 46);
        panelsub.Visible = false;
        panelsale.Visible = false;
        panelbottom.Visible = false;
        btnsupp.BackColor = Color.Transparent;
        btn_sales.BackColor = Color.Transparent;
        btn_pur.BackColor = Color.Transparent;
        btn_emp.BackColor = Color.Transparent;
        btn_service.BackColor = Color.Transparent;
        btncust.BackColor = Color.Transparent;
        btnstock.BackColor = Color.FromArgb(33, 33, 33);
        btn_logout.BackColor = Color.Transparent;
        btn_display.BackColor = Color.Transparent;
        this.Text = "Inventory Page";

        panelleft.Cursor = Cursors.WaitCursor;
        this.panelmain.Controls.Clear();
        Inventory frm = new Inventory();
        frm.TopLevel = false;
        this.panelmain.Controls.Add(frm);
        frm.Show();
        panelleft.Cursor = Cursors.Default;
    }

    private void fpurchase()
    {
        //purchase page
        this.Size = new Size(1136, 720);
        panelmain.Location = new Point(104, 46);
        panelmain.Size = new Size(1031, 580);
        panelsub.Visible = false;
        panelsale.Visible = false;
        panelbottom.Visible = true;
        btnsupp.BackColor = Color.Transparent;
        btn_sales.BackColor = Color.Transparent;
        btn_pur.BackColor = Color.FromArgb(33, 33, 33);
    }
}

```

```

        btn_emp.BackColor = Color.Transparent;
        btn_service.BackColor = Color.Transparent;
        btncust.BackColor = Color.Transparent;
        btnstock.BackColor = Color.Transparent;
        btn_logout.BackColor = Color.Transparent;
        btn_display.BackColor = Color.Transparent;
        this.Text = "Purchase Page - Order";

        panelleft.Cursor = Cursors.WaitCursor;
        this.panelmain.Controls.Clear();
        Purchase_Order frm = new Purchase_Order();
        frm.TopLevel = false;
        this.panelmain.Controls.Add(frm);
        frm.Show();
        panelleft.Cursor = Cursors.Default;
    }

    private void fsale()
    {
        //sale page
        this.Size = new Size(1136, 720);
        panelmain.Location = new Point(104, 46);
        panelmain.Size = new Size(1031, 645);
        panelsub.Visible = false;
        panelbottom.Visible = false;
        panelsale.Visible = true;
        btnsupp.BackColor = Color.Transparent;
        btn_sales.BackColor = Color.FromArgb(33, 33, 33);
        btn_pur.BackColor = Color.Transparent;
        btn_emp.BackColor = Color.Transparent;
        btn_service.BackColor = Color.Transparent;
        btncust.BackColor = Color.Transparent;
        btnstock.BackColor = Color.Transparent;
        btn_logout.BackColor = Color.Transparent;
        btn_display.BackColor = Color.Transparent;
        this.Text = "Sale Page - Bill";

        panelleft.Cursor = Cursors.WaitCursor;
        this.panelmain.Controls.Clear();
        Sale_Bill frm = new Sale_Bill(username);
        frm.TopLevel = false;
        this.panelmain.Controls.Add(frm);
        frm.Show();
        panelleft.Cursor = Cursors.Default;
    }

    private void display()
    {
        //display page
        this.Size = new Size(960, 720);
        panelmain.Location = new Point(104, 46);
        panelmain.Size = new Size(851, 645);
        panelsub.Visible = false;
        panelsale.Visible = false;
        panelbottom.Visible = false;
        btnsupp.BackColor = Color.Transparent;
        btn_sales.BackColor = Color.Transparent;
    }

```

```

        btn_pur.BackColor = Color.Transparent;
        btn_emp.BackColor = Color.Transparent;
        btn_service.BackColor = Color.Transparent;
        btncust.BackColor = Color.Transparent;
        btnstock.BackColor = Color.Transparent;
        btn_logout.BackColor = Color.Transparent;
        btn_display.BackColor = Color.FromArgb(33, 33, 33);
        this.Text = "Display Page";

        panelleft.Cursor = Cursors.WaitCursor;
        this.panelmain.Controls.Clear();
        Display frm = new Display();
        frm.TopLevel = false;
        this.panelmain.Controls.Add(frm);
        frm.Show();
        panelleft.Cursor = Cursors.Default;
    }

private void admindashboard_Click(object sender, EventArgs e)
{
    fadmindash();
}

private void pbadmin_Click(object sender, EventArgs e)
{
    fadmindash();
}

private void btnsupp_Click(object sender, EventArgs e)
{
    fsupplier();
}

private void pbsup_Click(object sender, EventArgs e)
{
    fsupplier();
}

private void bsup_Click(object sender, EventArgs e)
{
    fsupplier();
}

private void btn_emp_Click(object sender, EventArgs e)
{
    femployee();
}

private void pbemp_Click(object sender, EventArgs e)
{
    femployee();
}

private void bemp_Click(object sender, EventArgs e)
{
    femployee();
}

```

```

private void btncust_Click(object sender, EventArgs e)
{
    fcustomer();
}

private void pbcust_Click(object sender, EventArgs e)
{
    fcustomer();
}

private void bcust_Click(object sender, EventArgs e)
{
    fcustomer();
}

private void plockorunlock_Click_1(object sender, EventArgs e)
{
    //manage user
    this.Text = "Employee Page - Manage Employee";
    this.panelmain.Controls.Clear();
    Emp_Manage_User frm = new Emp_Manage_User();
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
}

private void padd_Click(object sender, EventArgs e)
{
    //new employee add
    this.Text = "Employee Page - Add New";
    this.panelmain.Controls.Clear();
    Emp_Add_New frm = new Emp_Add_New();
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
}

private void pbstock_Click(object sender, EventArgs e)
{
    finventory();
}

private void bstock_Click(object sender, EventArgs e)
{
    finventory();
}

private void pattendance_Click(object sender, EventArgs e)
{
    //attendance
    this.Text = "Employee Page - Attendance";
    this.panelmain.Controls.Clear();
    Emp_Attendance frm = new Emp_Attendance();
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
}

```

```

}

private void psalary_Click(object sender, EventArgs e)
{
    //salary
    this.Text = "Employee Page - Salary Calculation";
    this.panelmain.Controls.Clear();
    btn_calculate frm = new btn_calculate();
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
}

private void ppur_Click(object sender, EventArgs e)
{
    fpurchase();
}

private void bpur_Click(object sender, EventArgs e)
{
    fpurchase();
}

private void btn_pur_Click(object sender, EventArgs e)
{
    fpurchase();
}

private void pbsale_Click(object sender, EventArgs e)
{
    fsale();
}

private void bsale_Click(object sender, EventArgs e)
{
    fsale();
}

private void btn_sales_Click(object sender, EventArgs e)
{
    fsale();
}

private void btnstock_Click(object sender, EventArgs e)
{
    finventory();
}

private void purchaseorder_Click(object sender, EventArgs e)
{
    //purchase order
    this.panelmain.Controls.Clear();
    Purchase_Order frm = new Purchase_Order();
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
    this.Text = "Purchase Page - Order";
}

```

```

}

private void purchasebill_Click(object sender, EventArgs e)
{
    //purchase bill
    this.panelmain.Controls.Clear();
    Purchase_Bill frm = new Purchase_Bill();
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
    this.Text = "Purchase Page - Bill";
}

private void purchasereturn_Click(object sender, EventArgs e)
{
    //purchase return
    this.panelmain.Controls.Clear();
    Purchase_Return frm = new Purchase_Return();
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
    this.Text = "Purchase Page - Return";
}

private void saleorder_Click(object sender, EventArgs e)
{
    //sale order
    this.panelmain.Controls.Clear();
    Sale_Order frm = new Sale_Order();
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
    this.Text = "Sale Page - Order";
}

private void salebill_Click(object sender, EventArgs e)
{
    //sale bill
    this.panelmain.Controls.Clear();
    Sale_Bill frm = new Sale_Bill(username);
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
    this.Text = "Sale Page - Bill";
}

private void pbservice_Click(object sender, EventArgs e)
{
    fservice();
}

private void bservice_Click(object sender, EventArgs e)
{
    fservice();
}

private void btn_service_Click(object sender, EventArgs e)

```

```

    {
        fservice();
    }

    private void pbdisplay_Click(object sender, EventArgs e)
    {
        display();
    }

    private void bdisplay_Click(object sender, EventArgs e)
    {
        display();
    }

    private void btn_display_Click(object sender, EventArgs e)
    {
        display();
    }
}

}

```

Nav User:

```

using System.Windows.Forms;

namespace WindowsApplication
{
    public partial class Nav_User : Form
    {
        String username;
        Connect c = new Connect();
        public Nav_User(String str)
        {
            InitializeComponent();
            username = str;
        }

        private void Nav_User_Load(object sender, EventArgs e)
        {
            //page load
            panelleft.BackColor = Color.FromArgb(46, 46, 46);
            this.Text = "Employee Dashboard";
            c.cmd.CommandText="Select Name from Employee where
Emp_Id='"+username+"'";
            label1.Text = c.cmd.ExecuteScalar().ToString();
            groupBox10.Visible = false;
            groupBox9.Visible = false;
        }

        private void Nav_User_FormClosing(object sender, FormClosingEventArgs
e)
        {
            //close button
            if (MessageBox.Show("Are you sure want to Exit?", "",

MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

```

```

        e.Cancel = false;
    else
        e.Cancel = true;
}

private void logoutpage()
{
    //logout
    this.Hide();
    new Login_User().Show();
}

private void pblogout_Click(object sender, EventArgs e)
{
    logoutpage();
}

private void blogout_Click(object sender, EventArgs e)
{
    logoutpage();
}

private void btn_logout_Click(object sender, EventArgs e)
{
    logoutpage();
}

private void fcustomer()
{
    //customer page
    this.Size = new Size(960, 720);
    panelmain.Location = new Point(104, 46);
    panelsale.Visible = false;
    btn_sales.BackColor = Color.Transparent;
    btn_service.BackColor = Color.Transparent;
    btncust.BackColor = Color.FromArgb(33, 33, 33);
    btn_logout.BackColor = Color.Transparent;
    this.Text = "Customer Page";

    panelleft.Cursor = Cursors.WaitCursor;
    this.panelmain.Controls.Clear();
    Customer frm = new Customer();
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
    panelleft.Cursor = Cursors.Default;
}

private void fservice()
{
    //service page
    this.Size = new Size(960, 720);
    panelmain.Location = new Point(104, 46);
    panelmain.Size = new Size(851, 645);
    panelsale.Visible = false;
    btn_sales.BackColor = Color.Transparent;
    btn_service.BackColor = Color.FromArgb(33, 33, 33);
}

```

```

btncust.BackColor = Color.Transparent;
btn_logout.BackColor = Color.Transparent;
this.Text = "Service Page";

panelleft.Cursor = Cursors.WaitCursor;
this.panelmain.Controls.Clear();
Service frm = new Service();
frm.TopLevel = false;
this.panelmain.Controls.Add(frm);
frm.Show();
panelleft.Cursor = Cursors.Default;
}

private void btncust_Click(object sender, EventArgs e)
{
    fcustomer();
}

private void pbcust_Click(object sender, EventArgs e)
{
    fcustomer();
}

private void bcust_Click(object sender, EventArgs e)
{
    fcustomer();
}

private void fsale()
{
    //sale page
    this.Size = new Size(1136, 720);
    panelmain.Location = new Point(104, 46);
    panelmain.Size = new Size(1031, 645);
    panelsale.Visible = true;
    groupBox10.Visible = true;
    groupBox9.Visible = true;
    btn_sales.BackColor = Color.FromArgb(33, 33, 33);
    btn_service.BackColor = Color.Transparent;
    btncust.BackColor = Color.Transparent;
    btn_logout.BackColor = Color.Transparent;
    this.Text = "Sale Page - Bill";

    panelleft.Cursor = Cursors.WaitCursor;
    this.panelmain.Controls.Clear();
    Sale_Bill frm = new Sale_Bill(username);
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
    panelleft.Cursor = Cursors.Default;
}

private void pbsale_Click(object sender, EventArgs e)
{
    fsale();
}

```

```

private void bsale_Click(object sender, EventArgs e)
{
    fsale();
}

private void btn_sales_Click(object sender, EventArgs e)
{
    fsale();
}

private void pbservice_Click(object sender, EventArgs e)
{
    fservice();
}

private void bservice_Click(object sender, EventArgs e)
{
    fservice();
}

private void btn_service_Click(object sender, EventArgs e)
{
    fservice();
}

private void saleorder_Click(object sender, EventArgs e)
{
    //sale order
    this.panelmain.Controls.Clear();
    Sale_Order frm = new Sale_Order();
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
    this.Text = "Sale Page - Order";
}

private void salebill_Click(object sender, EventArgs e)
{
    //sale bill
    this.panelmain.Controls.Clear();
    Sale_Bill frm = new Sale_Bill(username);
    frm.TopLevel = false;
    this.panelmain.Controls.Add(frm);
    frm.Show();
    this.Text = "Sale Page - Bill";
}

}

```

Connect.cs:

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Data.SqlClient;

```

```

namespace WindowsApplication
{
    class Connect
    {
        public SqlConnection cn = new SqlConnection();
        public SqlCommand cmd = new SqlCommand();
        public Connect()
        {
            cn.ConnectionString = "data source =DESKTOP-6U35LDM;initial
catalog = Project;integrated security = true";
            cn.Open();
            cmd.Connection = cn;
        }
    }
}

```

Profile:

```

using System.Windows.Forms;
using System.Net.Mail;
using System.Net;

namespace WindowsApplication
{
    public partial class Profile : Form
    {
        Connect c = new Connect();
        String password, otp;
        int num;
        public Profile()
        {
            InitializeComponent();
        }

        private void Profile_Load(object sender, EventArgs e)
        {
            c.cmd.CommandText = "select fname from Admin_Create";
            String fname = Convert.ToString(c.cmd.ExecuteScalar());
            c.cmd.CommandText = "select lname from Admin_Create";
            String lname = Convert.ToString(c.cmd.ExecuteScalar());
            tfullname.Text = fname + " " + lname;
            c.cmd.CommandText = "select email from Admin_Create";
            temail.Text = Convert.ToString(c.cmd.ExecuteScalar());
            c.cmd.CommandText = "select username from Admin_Create";
            tusername.Text = Convert.ToString(c.cmd.ExecuteScalar());
            c.cmd.CommandText = "select password from Admin_Create";
            password = Convert.ToString(c.cmd.ExecuteScalar());
        }

        private void edit_Click(object sender, EventArgs e)
        {
            //edit
            temail.ReadOnly = false;
            edit.Visible = false;
            pictureBox1.Visible = false;
        }
    }
}

```

```

        temail.Cursor = Cursors.IBeam;
        esave.Enabled = true;
        esave.BackColor = Color.Black;
    }

    private void temail_TextChanged(object sender, EventArgs e)
    {
        errorProvider4.Clear();
    }

    private void esave_Click(object sender, EventArgs e)
    {
        //save
        try
        {
            if (temail.Text == "")
                errorProvider4.SetError(temail, "Required!");
            else
            {
                new MailAddress(this.temail.Text);
                temail.ReadOnly = true;
                edit.Visible = true;
                pictureBox1.Visible = true;
                temail.Cursor = Cursors.No;
                esave.Enabled = false;
                esave.BackColor = Color.Gray;
                c.cmd.CommandText = "update Admin_Create set email='"
                temail.Text + "'";
                c.cmd.ExecuteNonQuery();
            }
        }
        catch (FormatException)
        {
            errorProvider4.SetError(temail, "Enter valid format Email");
        }
    }

    private void clearall()
    {
        tcurred.Text = "";
        errcpass.Visible = false;
        totp.Text = "";
        errotp.Visible = false;
        tnew.Text = "";
        tcnew.Text = "";
        errmsg.Visible = false;
    }

    private void changepassword_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        //change password
        groupBox2.Visible = true;
        pcurrentpass.Visible = true;
        groupBox2.Size = new Size(336, 155);
        clearall();
    }

```

```

private void ceye_MouseHover(object sender, EventArgs e)
{
    tcurr.UseSystemPasswordChar = false;
}

private void ceye_MouseLeave(object sender, EventArgs e)
{
    tcurr.UseSystemPasswordChar = true;
}

private void verifypass_Click(object sender, EventArgs e)
{
    //verify current password
    c.cmd.CommandText = "select password from Admin_Create";
    password = Convert.ToString(c.cmd.ExecuteScalar());
    if (tcurr.Text == "")
        errorProvider1.SetError(tcurr, "Required!");
    else if (tcurr.Text != password)
    {
        errcpass.Visible = true;
        errcpass.Text = "Invalid Password!";
        pcurrentpass.Cursor = Cursors.Default;
    }
    else if (tcurr.Text == password)
    {
        groupBox2.Size = new Size(330, 245);
        pcurrentpass.Visible = false;
        pnewpass.Visible = true;
        clearall();
    }
}

private void linkLabel1_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    //forgot current password
    pcurrentpass.Cursor = Cursors.WaitCursor;
    try
    {
        c.cmd.CommandText = "select email from Admin_Create";
        String email = Convert.ToString(c.cmd.ExecuteScalar());
        Random r = new Random();
        num = r.Next(0, 1000000);
        otp = num.ToString("000000");
        MailMessage mail = new MailMessage();
        SmtpClient smtpserver = new SmtpClient("smtp.gmail.com");
        mail.From = new MailAddress("bhakthiprabhu15@gmail.com");
        mail.To.Add(email);
        mail.Subject = "Verify Profile - OTP";
        mail.Body = "Please enter the OTP mentioned below to verify
your profile.\n\n"+otp+"\n\nPlease don't share your OTP with anyone.";
        smtpserver.Port = 587;
        smtpserver.Credentials = new
NetworkCredential("bhakthiprabhu15@gmail.com", "bhakthi15");
        smtpserver.EnableSsl = true;
        smtpserver.Send(mail);
    }
}

```

```

        potp.Visible = true;
        groupBox2.Size = new Size(336, 155);
        pcurrentpass.Visible = false;
    }
    catch (Exception)
    {
        errcpass.Visible = true;
        errcpass.Text = "No Internet Connection!";
        pcurrentpass.Cursor = Cursors.Default;
    }
}

private void totp_TextChanged(object sender, EventArgs e)
{
    errotp.Visible = false;
}

private void verifyotp_Click(object sender, EventArgs e)
{
    //verify otp
    potp.Cursor = Cursors.WaitCursor;
    if (totp.Text == otp)
    {
        groupBox2.Size = new Size(330, 245);
        potp.Visible = false;
        pnewpass.Visible = true;
        clearall();
    }
    else
    {
        errotp.Visible = true;
        errotp.Text = "Invalid OTP!";
        pcurrentpass.Cursor = Cursors.Default;
    }
}

private void tnew_Leave(object sender, EventArgs e)
{
    if (tnew.Text.Length <= 7)
    {
        errmsg.Visible = true;
        errmsg.Text = "Be atleast 8 characters!";
    }
}

private void tnew_TextChanged(object sender, EventArgs e)
{
    errorProvider2.Clear();
    errmsg.Visible = false;
}

private void neye_MouseHover(object sender, EventArgs e)
{
    tcnew.UseSystemPasswordChar = false;
}

private void neye_MouseLeave(object sender, EventArgs e)

```

```

{
    tcnew.UseSystemPasswordChar = true;
}

private void tcurr_TextChanged(object sender, EventArgs e)
{
    errpass.Visible = false;
    errorProvider1.Clear();
}

private void tcnew_TextChanged(object sender, EventArgs e)
{
    errorProvider3.Clear();
    errmsg.Visible = false;
}

private void update_Click(object sender, EventArgs e)
{
    //update password
    if (tnew.Text == "")
        errorProvider2.SetError(tnew, "Required!");
    else if (tcnew.Text == "")
        errorProvider3.SetError(tcnew, "Required!");
    else if (tnew.Text != tcnew.Text)
    {
        errmsg.Visible = true;
        errmsg.Text = "Password must match!";
    }
    else
    {
        c.cmd.CommandText = "update Admin_create set password=''" +
tcnew.Text + "'";
        c.cmd.ExecuteNonQuery();
        MessageBox.Show("Password Updated!", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        groupBox2.Visible = false;
        clearall();
    }
}

private void totp_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !
char.IsControl(e.KeyChar);
}
}
}

```

Password Reset:

```

using System.Windows.Forms;
namespace WindowsApplication
{
    public partial class Password_Reset : Form
    {

```

```

        Connect c = new Connect();
        string sans;
        int i = 0;
        public Password_Reset()
        {
            InitializeComponent();
        }

        private void Password_Reset_Load(object sender, EventArgs e)
        {
            this.Text = "Forgot Password";
            this.Location = new Point(500, 200);
            this.Size = new Size(320, 380);
            Connect c = new Connect();
            c.cmd.CommandText = "select sqns from Admin_Create";
            lsqns.Text = Convert.ToString(c.cmd.ExecuteScalar());
            c.cmd.CommandText = "select sans from Admin_Create";
            sans = Convert.ToString(c.cmd.ExecuteScalar());
        }

        private void back_MouseHover(object sender, EventArgs e)
        {
            toolTip1.SetToolTip(back, "Back");
        }

        private void tsans_TextChanged(object sender, EventArgs e)
        {
            errorProvider1.Clear();
        }

        private void fconfirm_Click(object sender, EventArgs e)
        {
            //confirm button
            if (tsans.Text == "")
                errorProvider1.SetError(tsans, "Required!");
            else if (tsans.Text == sans)
            {
                this.Text = "Reset Password";
                this.Size = new Size(659, 380);
                tsans.ReadOnly = true;
                tsans.Cursor = Cursors.No;
                fconfirm.Enabled = false;
                fconfirm.Cursor = Cursors.No;
                linkLabel1.Enabled = false;
                this.StartPosition = FormStartPosition.Manual;
                this.Location = new Point(350, 200);
            }
            else
            {
                i++;
                MessageBox.Show("Incorrect Answer", "", MessageBoxButtons.RetryCancel, MessageBoxIcon.Exclamation);
                if (i > 2)
                {
                    this.Hide();
                    new Login_User().Show();
                }
            }
        }
    
```

```
        }

    }

    private void linkLabel1_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    //forgot security answer
    linkLabel1.Cursor = Cursors.WaitCursor;
    Sans_Reset sr = new Sans_Reset();
    sr.Owner = this;
    sr.Show();
    linkLabel1.Cursor = Cursors.Default;
}

private void back_Click(object sender, EventArgs e)
{
    //back
    this.Hide();
    new Login_User().Show();
}

private void tnewpassword_TextChanged(object sender, EventArgs e)
{
    errorProvider2.Clear();
}

private void rconfirm_Click(object sender, EventArgs e)
{
    //confirm button
    c.cmd.CommandText = "select password from Admin_Create";
    String password = Convert.ToString(c.cmd.ExecuteScalar());
    if (tnewpassword.Text == "")
        errorProvider2.SetError(tnewpassword, "Required!");
    else
    {
        if (MessageBox.Show("Are you sure?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            c.cmd.CommandText = "update Admin_Create set password =" +
+ tnewpassword.Text + "'";
            c.cmd.ExecuteNonQuery();
            MessageBox.Show("Password Changed", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
            this.Hide();
            new Login_User().Show();
        }
    }
}
```

Inventory:

```
using System.Windows.Forms;
using System.Data.SqlClient;
```

```

namespace WindowsApplication
{
    public partial class Inventory : Form
    {
        Connect c = new Connect();
        string itemid, date;
        public Inventory()
        {
            InitializeComponent();
        }

        private void Inventory_Load(object sender, EventArgs e)
        {
            //page load
            Supplier s = new Supplier();
            s.datagridviewstyle(dataGridView1);
            dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;
            nupdate.Enabled = false;
            ndelete.Enabled = false;
            displayall();
            dateTimePicker1.Value = DateTime.Today;
        }

        private void displayall()
        {
            //display function
            if ((cbactive.Checked == true && cbinactive.Checked == true) ||
(cbactive.Checked == false && cbinactive.Checked == false))
            {
                SqlDataAdapter sda = new SqlDataAdapter("select * from
Inventory", c.cn);
                DataSet ds = new DataSet();
                sda.Fill(ds, "dt");
                dataGridView1.DataSource = ds.Tables["dt"];
            }
            else if (cbactive.Checked == true && cbinactive.Checked == false)
            {
                SqlDataAdapter sda = new SqlDataAdapter("select * from
Inventory where Status='Active'", c.cn);
                DataSet ds = new DataSet();
                sda.Fill(ds, "dt");
                dataGridView1.DataSource = ds.Tables["dt"];
            }
            else if (cbactive.Checked == false && cbinactive.Checked == true)
            {
                SqlDataAdapter sda = new SqlDataAdapter("select * from
Inventory where Status='Inactive'", c.cn);
                DataSet ds = new DataSet();
                sda.Fill(ds, "dt");
                dataGridView1.DataSource = ds.Tables["dt"];
            }
        }

        private void clearall()
        {
            //clear function

```

```

        titemname.Text = "";
        tqty.Text = "0";
        tprice.Text = "0.0";
        twarranty.Text = "0.0";
        tcolor.Text = "";
        tmaterial.Text = "";
        dateTimePicker1.Text = "";
        cbcategory.ResetText();
        cbcategory.Text = "Select";
        filter.ResetText();
        filter.Text = "Sort By";
        msg.Text = "";
        nupdate.Enabled = false;
        ndelete.Enabled = false;
        nadd.Enabled = true;
        dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
        searchitem.Text = "search...";
        searchitem.ForeColor = Color.Silver;
    }

private void tsaleprice_KeyPress(object sender, KeyPressEventArgs e)
{
    //accept only decimal numbers
    if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) &&
e.KeyChar != '.')
        e.Handled = true;
    else
        e.Handled = false;
}

private void tqty_KeyPress(object sender, KeyPressEventArgs e)
{
    //accept only numbers
    e.Handled = !char.IsDigit(e.KeyChar) && !
char.IsControl(e.KeyChar);
}

private void twarranty_KeyPress(object sender, KeyPressEventArgs e)
{
    //accept only decimal numbers
    if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) &&
e.KeyChar != '.')
        e.Handled = true;
    else
        e.Handled = false;
}

private void tprice_KeyPress(object sender, KeyPressEventArgs e)
{
    //accept only decimal numbers
    if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) &&
e.KeyChar != '.')
        e.Handled = true;
    else
        e.Handled = false;
}

```

```

private void nclear_Click(object sender, EventArgs e)
{
    //clear
    clearall();
    cbinactive.Checked = false;
    cbactive.Checked = false;
    displayall();
}

private void nadd_Click(object sender, EventArgs e)
{
    //add
    date = dateTimePicker1.Value.ToString("MM-dd-yyyy");
    try
    {
        if (titemname.Text == "" || tprice.Text == "" ||
twarranty.Text == "" || tcolor.Text == "" || tmaterial.Text == "" || cbccategory.Text == "Select")
            throw new ArgumentException();
        else
        {
            SqlCommand cmd = new SqlCommand("Select * from Inventory
where Name='" + titemname.Text + "' and Color='" + tcolor.Text + "' and
Material='" + tmaterial.Text + "'", c.cn);
            SqlDataAdapter adp = new SqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            adp.Fill(dt);
            if (dt.Rows.Count > 0)
                MessageBox.Show("Record Exists!\n Please Edit item if
you want to make changes!", "", MessageBoxButtons.OK, MessageBoxIcon.Error);
            else if (MessageBox.Show("Are You Sure?", "",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                c.cmd.CommandText = "select Count(*) from Inventory";
                int cnt = Convert.ToInt16(c.cmd.ExecuteScalar());
                cnt++;
                itemid = "item" + cnt;
                c.cmd.CommandText = "insert into Inventory (Item_Id)
values('" + itemid + "')";
                c.cmd.ExecuteNonQuery();
                c.cmd.CommandText = "update Inventory set Name='" +
titemname.Text + "',Qty='" + tqty.Text + "',Date='" + date + "',Category='"
+ cbccategory.SelectedItem.ToString() + "',Color='" + tcolor.Text +
"',Material='" + tmaterial.Text + "',Price='" + tprice.Text + "',Warranty='"
+ twarranty.Text + "' where Item_Id='" + itemid + "'";
                c.cmd.ExecuteNonQuery();
                c.cmd.CommandText = "Select Qty from Inventory where
Item_Id='" + itemid + "'";
                int qt = Convert.ToInt32(c.cmd.ExecuteScalar());
                if (qt <= 0)
                {
                    c.cmd.CommandText = "update Inventory set Status
='Inactive',Qty=0 where Item_Id='" + itemid + "'";
                    c.cmd.ExecuteNonQuery();
                }
            }
        }
    }
}

```

```

        {
            c.cmd.CommandText = "update Inventory set Qty =" +
qt + ",Status ='Active' where Item_Id=''' + itemid + '''";
            c.cmd.ExecuteNonQuery();
        }
        MessageBox.Show("Item Added Successfully!", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
        displayall();
        clearall();
    }
}
catch (ArgumentException)
{
    MessageBox.Show("Enter all the details", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}

private void searchitem_Click(object sender, EventArgs e)
{
    //search item Click
    if (searchitem.Text == "search...")
        searchitem.Text = "";
    searchitem.ForeColor = Color.FromArgb(20, 168, 0);
}

private void searchitem_Leave(object sender, EventArgs e)
{
    //search item leave
    if (searchitem.Text == "")
        searchitem.Text = "search...";
    searchitem.ForeColor = Color.Silver;
}

private void searchitem_KeyUp(object sender, KeyEventArgs e)
{
    //search item keyup
    filter.ResetText();
    filter.Text = "Sort By";
    cbinactive.Checked = false;
    cbactive.Checked = false;
    SqlDataAdapter sda = new SqlDataAdapter("select * from Inventory
where Name like '" + searchitem.Text + "%'", c.cn);
    DataSet ds = new DataSet();
    sda.Fill(ds, "dt");
    dataGridView1.DataSource = ds.Tables["dt"];
}

private void filter_SelectedIndexChanged(object sender, EventArgs e)
{
    //filter by
    searchitem.Text = "search...";
    cbactive.Checked = false;
    cbinactive.Checked = false;
    sortfunc(filter.SelectedItem.ToString());
}

```

```

private void sortfunc(String value)
{
    //sortfunc function
    SqlDataAdapter adp = new SqlDataAdapter();
    DataSet ds = new DataSet();
    c.cmd = new SqlCommand("select * from Inventory", c.cn);
    adp.SelectCommand = c.cmd;
    ds.Clear();
    adp.Fill(ds);
    dataGridView1.Sort(dataGridView1.Columns[value],
ListSortDirection.Ascending);
}

private void dgvinventoryd_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    //dataGridView1_CellContentClick
    msg.Visible = false;
    if (dataGridView1.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
    {
        nupdate.Enabled = true;
        ndelete.Enabled = true;
        nadd.Enabled = false;
        dataGridView1.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        dataGridView1.CurrentRow.Selected = true;
        itemid =
dataGridView1.Rows[e.RowIndex].Cells["Item_Id"].FormattedValue.ToString();
        titemname.Text =
dataGridView1.Rows[e.RowIndex].Cells["Name"].FormattedValue.ToString();
        tmaterial.Text =
dataGridView1.Rows[e.RowIndex].Cells["Material"].FormattedValue.ToString();
        tcolor.Text =
dataGridView1.Rows[e.RowIndex].Cells["Color"].FormattedValue.ToString();
        dateTimePicker1.Text =
dataGridView1.Rows[e.RowIndex].Cells["Date"].FormattedValue.ToString();
        cbccategory.Text =
dataGridView1.Rows[e.RowIndex].Cells["Category"].FormattedValue.ToString();
        tqty.Text =
dataGridView1.Rows[e.RowIndex].Cells["Qty"].FormattedValue.ToString();
        tprice.Text =
dataGridView1.Rows[e.RowIndex].Cells["Price"].FormattedValue.ToString();
        twarranty.Text =
dataGridView1.Rows[e.RowIndex].Cells["Warranty"].FormattedValue.ToString();
    }
}

private void nupdate_Click(object sender, EventArgs e)
{
    //update
    date = dateTimePicker1.Value.ToString("MM-dd-yyyy");
    try
{

```

```

        if (titemname.Text == "" || tprice.Text == "" || 
twarranty.Text == "" || tcolor.Text == "" || tmaterial.Text == "" || 
cbccategory.Text == "Select")
            throw new ArgumentException();
        else
        {
            if (MessageBox.Show("Are You Sure want to Update ?", "", 
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                c.cmd.CommandText = "update Inventory set Name='"
+ titemname.Text + "',Date='"+ date + "',Category='"
+ cbccategory.SelectedItem.ToString() + "',Color='"
+ tcolor.Text +
'',Material=''
+ tmaterial.Text + "',Price='"
+ tprice.Text + "',Warranty='"
+ twarranty.Text + "',Qty='"
+ tqty.Text + "' where Item_Id='"
+ itemid + "'";
                c.cmd.ExecuteNonQuery();
                c.cmd.CommandText = "Select Qty from Inventory where 
Item_Id='"
+ itemid + "'";
                int qt = Convert.ToInt32(c.cmd.ExecuteScalar());
                if (qt <= 0)
                {
                    c.cmd.CommandText = "update Inventory set Status 
='Inactive',Qty=0 where Item_Id='"
+ itemid + "'";
                    c.cmd.ExecuteNonQuery();
                }
                else
                {
                    c.cmd.CommandText = "update Inventory set Qty ="
+ qt + ",Status ='Active' where Item_Id='"
+ itemid + "'";
                    c.cmd.ExecuteNonQuery();
                }
                MessageBox.Show("Inventory Updated Successfully!", "", 
MessageBoxButtons.OK, MessageBoxIcon.Information);
                dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
                dataGridView1.Rows[0].Cells[0].Selected = true;
                displayall();
                clearall();
            }
        }
    }
    catch (ArgumentException)
    {
        MessageBox.Show("Enter all the details", "", 
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void ndelete_Click(object sender, EventArgs e)
{
    //delete
    c.cmd.CommandText = "select Status from Inventory where Item_Id='"
+ itemid + "'";
    if (Convert.ToString(c.cmd.ExecuteScalar()) == "Inactive")
    {
        msg.Visible = true;
        msg.Text = "Item already Inactivated!";
    }
}

```

```

        else
        {
            if (MessageBox.Show("Are You Sure want to Inactivate " +
titemname.Text + " ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.Yes)
            {
                c.cmd.CommandText = "update Inventory set
Status='Inactive',Qty=0 where Item_Id='" + itemid + "'";
                c.cmd.ExecuteNonQuery();
                MessageBox.Show("Item Inactivated!", "",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
                dataGridView1.Rows[0].Cells[0].Selected = true;
                displayall();
                clearall();
            }
        }
    }

private void cbactive_CheckedChanged(object sender, EventArgs e)
{
    //checkbox active
    searchitem.Text = "Search...";
    filter.ResetText();
    filter.Text = "Sort By";
    displayall();
}

private void cbinactive_CheckedChanged(object sender, EventArgs e)
{
    //checkbox inactive
    searchitem.Text = "Search...";
    filter.ResetText();
    filter.Text = "Sort By";
    displayall();
}

private void tcolor_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar));
}

private void tmaterial_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar));
}

private void searchitem_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back);
}

private void titemname_KeyPress(object sender, KeyPressEventArgs e)

```

```

        {
            e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || char.IsDigit(e.KeyChar));
        }
    }
}

```

Customer:

```

using System.Text.RegularExpressions;
using System.Net.Mail;
using System.Data.SqlClient;

namespace WindowsApplication
{
    public partial class Customer : Form
    {
        Connect c = new Connect();
        public Customer()
        {
            InitializeComponent();
        }

        private void Customer_Load(object sender, EventArgs e)
        {
            //page load
            nupdate.Enabled = false;
            displayall();
            Supplier s = new Supplier();
            s.datagridviewstyle(dataGridView1);
        }

        private void vsupname_KeyPress(object sender, KeyPressEventArgs e)
        {
            //accept only numbers
            e.Handled = !char.IsDigit(e.KeyChar) && !
char.IsControl(e.KeyChar);
        }

        private void clearall()
        {
            //clear function
            filter.ResetText();
            filter.Text = "Sort By";
            nadd.Enabled = true;
            dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
            searchcno.Text = "search...";
            searchcno.ForeColor = Color.Silver;
            taddr.Text = "";
            temail.Text = "";
            tname.Text = "";
            maskedTextBox2.Text = "";
            nupdate.Enabled = false;
        }
    }
}

```

```

private void displayall()
{
    //display function
    SqlDataAdapter sda = new SqlDataAdapter("select * from Customer",
c.cn);
    DataSet ds = new DataSet();
    sda.Fill(ds, "dt");
    dataGridView1.DataSource = ds.Tables["dt"];
}

private void nclear_Click(object sender, EventArgs e)
{
    //clear
    clearall();
    displayall();
}

private void nadd_Click(object sender, EventArgs e)
{
    //add
    try
    {
        Regex rx = new Regex("^[6-9][0-9]{9}$");
        if (tname.Text == "" || taddr.Text == "" || Convert.ToString(maskedTextBox2.Text) == "" || temail.Text == "")
            throw new ArgumentException();
        else if (!(rx.IsMatch(maskedTextBox2.Text)))
            errorProvider1.SetError(maskedTextBox2, "Enter valid mobile number");
        else
        {
            new MailAddress(temail.Text);
            SqlCommand cmd = new SqlCommand("Select * from Customer where Name='"
+ tname.Text + "' and Contact_No='"
+ maskedTextBox2.Text + "' and Email='"
+ temail.Text + "'", c.cn);
            SqlDataAdapter adp = new SqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            adp.Fill(dt);
            if (dt.Rows.Count > 0)
                MessageBox.Show("Record Exists!\n Please Edit Customer if you want to make changes!", "", MessageBoxButtons.OK, MessageBoxIcon.Error);
            else if (MessageBox.Show("Are You Sure?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                c.cmd.CommandText = "insert into Customer (Contact_No) values('"
+ Convert.ToString(maskedTextBox2.Text) + "')";
                c.cmd.ExecuteNonQuery();
                c.cmd.CommandText = "update Customer set Name='"
+ tname.Text + "',Address='"
+ taddr.Text + "',Email='"
+ temail.Text + "' where Contact_No='"
+ Convert.ToString(maskedTextBox2.Text)+ "'";
                c.cmd.ExecuteNonQuery();
                MessageBox.Show("Customer Added Successfully!", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
                displayall();
                clearall();
            }
        }
    }
}

```

```

        }
    }
    catch (FormatException)
    {
        errorProvider2.SetError(temail, "Enter valid format email");
    }
    catch (ArgumentException)
    {
        MessageBox.Show("Enter all the details", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void temail_TextChanged(object sender, EventArgs e)
{
    //email textchanged
    errorProvider2.Clear();
}

private void maskedTextBox2_TextChanged(object sender, EventArgs e)
{
    //contact no textchanged
    errorProvider1.Clear();
}

private void searchcno_Click(object sender, EventArgs e)
{
    //search contact no click
    if (searchcno.Text == "search...")
        searchcno.Text = "";
    searchcno.ForeColor = Color.FromArgb(20, 168, 0);
}

private void searchcno_Leave(object sender, EventArgs e)
{
    //search contact no leave
    if (searchcno.Text == "")
        searchcno.Text = "search...";
    searchcno.ForeColor = Color.Silver;
}

private void searchcno_KeyUp(object sender, KeyEventArgs e)
{
    //search contact no keyup
    filter.ResetText();
    filter.Text = "Sort By";
    SqlDataAdapter sda = new SqlDataAdapter("select * from Customer
where Contact_No like '" + searchcno.Text + "%'", c.cn);
    DataSet ds = new DataSet();
    sda.Fill(ds, "dt");
    dataGridView1.DataSource = ds.Tables["dt"];
}

private void filter_SelectedIndexChanged(object sender, EventArgs e)
{
    //filter by selected index changed
    searchcno.Text = "search...";
}

```

```

        sortfunc(filter.SelectedItem.ToString());
    }

    private void sortfunc(String value)
    {
        //sortfunc function
        SqlDataAdapter adp = new SqlDataAdapter();
        DataSet ds = new DataSet();
        c.cmd = new SqlCommand("select * from Customer", c.cn);
        adp.SelectCommand = c.cmd;
        ds.Clear();
        adp.Fill(ds);
        dataGridView1.Sort(dataGridView1.Columns[value],
ListSortDirection.Ascending);
    }

    private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    //dataGridView1_CellContentClick
    if (dataGridView1.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
    {
        nupdate.Enabled = true;
        nadd.Enabled = false;
        dataGridView1.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        dataGridView1.CurrentRow.Selected = true;
        tname.Text =
dataGridView1.Rows[e.RowIndex].Cells["Name"].FormattedValue.ToString();
        temail.Text =
dataGridView1.Rows[e.RowIndex].Cells["Email"].FormattedValue.ToString();
        maskedTextBox2.Text =
dataGridView1.Rows[e.RowIndex].Cells["Contact_No"].FormattedValue.ToString();
        taddr.Text =
dataGridView1.Rows[e.RowIndex].Cells["Address"].FormattedValue.ToString();
    }
}

private void nupdate_Click(object sender, EventArgs e)
{
    //update
    try
    {
        Regex rx = new Regex("^([6-9][0-9]{9})$");
        if (tname.Text == "" || taddr.Text == "" || Convert.ToString(maskedTextBox2.Text) == "" || temail.Text == "")
            throw new ArgumentException();
        else if (!(rx.IsMatch(maskedTextBox2.Text)))
            errorProvider1.SetError(maskedTextBox2, "Enter valid
mobile number");
        else
        {
            new MailAddress(temail.Text);
            if (MessageBox.Show("Are You Sure want to Update ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {

```

```

        c.cmd.CommandText = "update Customer set Name='" +
tname.Text + "',Address='"+ taddr.Text + "',Email='"+ temail.Text + "' where
Contact_No='"+ Convert.ToString(maskedTextBox2.Text) + "'";
c.cmd.ExecuteNonQuery();
MessageBox.Show("Customer Updated Successfully!", "",

MessageBoxButtons.OK, MessageBoxIcon.Information);
dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
dataGridView1.Rows[0].Cells[0].Selected = true;
displayall();
clearall();
}
}
}
catch (FormatException)
{
    errorProvider2.SetError(temail, "Enter valid format email");
}
catch (ArgumentException)
{
    MessageBox.Show("Enter all the details", "",

MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}

private void tname_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || e.KeyChar == '.');
}

private void taddr_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || char.IsDigit(e.KeyChar) ||
e.KeyChar == ',' || e.KeyChar == '-' || e.KeyChar == '&' || e.KeyChar == '.');
}
}
}

```

EMP Add New:

```

using System.Text.RegularExpressions;
using System.Net.Mail;
using System.Data.SqlClient;

namespace WindowsApplication
{
    public partial class Emp_Add_New : Form
    {
        Connect c = new Connect();
        DateTime dt1, dt2;
        String dob, gender, doj, empid;
        public Emp_Add_New()
        {
            InitializeComponent();

```

```

        }

    private void Employee_Load(object sender, EventArgs e)
    {
        //page load
        Supplier s = new Supplier();
        s.datagridviewstyle(dataGridView1);
        nupdate.Enabled = false;
        ndelete.Enabled = false;
        displayall();
        dateTimePicker1.Value = DateTime.Today;
    }

    private void maskedTextBox1_Leave(object sender, EventArgs e)
    {
        //dob leave
        try
        {
            dt1 = Convert.ToDateTime(maskedTextBox1.Text);
            dt2 = DateTime.Today.Date;
            Regex rx = new Regex("^(0-2)[0-9]|(3)[0-1])[-]([0][1-9]|1[0-2])[-]([0-9]{4})$");
            if (!(rx.IsMatch(maskedTextBox1.Text)))
                errorProvider3.SetError(maskedTextBox1, "Enter valid birth date in format dd/mm/yyyy");
            else if (dt2 < dt1)
                errorProvider3.SetError(maskedTextBox1, "Enter valid birth date in format dd/mm/yyyy");
            else
            {
                int d = (int)(dt2 - dt1).TotalDays;
                tage.Text = (d / 365).ToString();
                if(Convert.ToInt16(tage.Text)<18)
                    errorProvider3.SetError(maskedTextBox1,"Age should be atleast 18");
            }
        }
        catch (Exception)
        {
            errorProvider3.SetError(maskedTextBox1, "Enter valid birth date in format dd/mm/yyyy");
        }
    }

    private void maskedTextBox1_TextChanged(object sender, EventArgs e)
    {
        //dob textchanged
        errorProvider3.Clear();
    }

    private void tbasicpay_KeyPress(object sender, KeyPressEventArgs e)
    {
        //accept only decimal numbers
        if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) && e.KeyChar != '.')
            e.Handled = true;
        else
    }

```

```

        e.Handled = false;
    }

    private void displayall()
    {
        //display function
        if ((cbactive.Checked == true && cbinactive.Checked == true) ||
        (cbactive.Checked == false && cbinactive.Checked == false))
        {
            SqlDataAdapter sda = new SqlDataAdapter("select * from
Employee", c.cn);
            DataSet ds = new DataSet();
            sda.Fill(ds, "dt");
            dataGridView1.DataSource = ds.Tables["dt"];
        }
        else if (cbactive.Checked == true && cbinactive.Checked == false)
        {
            SqlDataAdapter sda = new SqlDataAdapter("select * from
Employee where Status='Active'", c.cn);
            DataSet ds = new DataSet();
            sda.Fill(ds, "dt");
            dataGridView1.DataSource = ds.Tables["dt"];
        }
        else if (cbactive.Checked == false && cbinactive.Checked == true)
        {
            SqlDataAdapter sda = new SqlDataAdapter("select * from
Employee where Status='Inactive'", c.cn);
            DataSet ds = new DataSet();
            sda.Fill(ds, "dt");
            dataGridView1.DataSource = ds.Tables["dt"];
        }
    }

    private void clearall()
    {
        //clear function
        tname.Text = "";
        taddr.Text = "";
        temail.Text = "";
        maskedTextBox1.Text = "";
        maskedTextBox2.Text = "";
        tage.Text = "";
        tdesgn.Text = "";
        tbasicpay.Text = "";
        searchemp.Text = "";
        radioButton1.Checked = false;
        radioButton2.Checked = false;
        radioButton3.Checked = false;
        dateTimePicker1.Text = "";
        radioButton1.Checked = true;
        msg.Text = "";
        filter.ResetText();
        filter.Text = "Sort By";
        msg.Text = "";
        nupdate.Enabled = false;
        ndelete.Enabled = false;
        nadd.Enabled = true;
    }
}

```

```

        dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
        searchemp.Text = "search...";
        searchemp.ForeColor = Color.Silver;
        errorProvider1.Clear();
        errorProvider2.Clear();
        errorProvider3.Clear();
        errorProvider4.Clear();
    }

    private void nadd_Click(object sender, EventArgs e)
{
    //register
    dob = dt1.ToString("MM-dd-yyyy");
    DOJ = dateTimePicker1.Value.ToString("MM-dd-yyyy");
    gender="";
    DateTime db = new
DateTime(Convert.ToInt16(dt1.ToString("yyyy")),Convert.ToInt16(dt1.ToString("M
M")),Convert.ToInt16(dt1.ToString("dd")),12,00,00);
    DateTime dj = new
DateTime(Convert.ToInt16(dateTimePicker1.Value.ToString("yyyy")),
Convert.ToInt16(dateTimePicker1.Value.ToString("MM")),
Convert.ToInt16(dateTimePicker1.Value.ToString("dd")), 12, 00, 00);
    if (radioButton1.Checked == true)
        gender = "Female";
    else if (radioButton2.Checked == true)
        gender = "Male";
    else if (radioButton3.Checked == true)
        gender = "Other";
    try
    {
        Regex rx1 = new Regex("^[6-9][0-9]{9}$");
        if (tname.Text == "" || taddr.Text == "" || Convert.ToString(maskedTextBox2.Text) == "" || temail.Text == "" || maskedTextBox2.Text == "" || tdesgn.Text == "" || tbasicpay.Text == "")
            throw new ArgumentException();
        else if (!(rx1.IsMatch(maskedTextBox2.Text)))
            errorProvider1.SetError(maskedTextBox2, "Enter valid
mobile number");
        else
        {
            new MailAddress(temail.Text);
            SqlCommand cmd = new SqlCommand("Select * from Employee
where Name='" + tname.Text + "' and Contact_No='" + maskedTextBox2.Text + "' and Email='"
+ temail.Text + "'", c.cn);
            SqlDataAdapter adp = new SqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            adp.Fill(dt);
            if (db > dj)
                errorProvider4.SetError(dateTimePicker1, "invalid DOJ\
n DOJ should be greater than dob");
            else if (dt.Rows.Count > 0)
                MessageBox.Show("Record Exists!\n Please Edit Employee
if you want to make changes!", "", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            else if (MessageBox.Show("Are You Sure?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

```

```

        {
            c.cmd.CommandText = "select Count(*) from Employee";
            int cnt = Convert.ToInt16(c.cmd.ExecuteScalar());
            cnt++;
            empid = "emp" + cnt;
            c.cmd.CommandText = "insert into Employee (Emp_Id)
values('" + empid + "')";
            c.cmd.ExecuteNonQuery();
            c.cmd.CommandText = "update Employee set Name='" +
tname.Text + "',Address='" + taddr.Text + "',Email='" + temail.Text +
"',Contact_No= '" + Convert.ToString(maskedTextBox2.Text) +
"',Status='Active',Dob='" + dob + "',Gender='" + gender + "',Age='" +
tage.Text + "',Doj='" + doj + "',Designation='" + tdesign.Text +
"',Basic_Pay='" + tbasicpay.Text + "' where Emp_Id='" + empid + "'";
            c.cmd.ExecuteNonQuery();
            //update employee login
            String password = empid + "123";
            c.cmd.CommandText = "insert into Login values('" +
empid + "','" + password + "')";
            c.cmd.ExecuteNonQuery();
            MessageBox.Show("Employee Added Successfully!", "", 
MessageBoxButtons.OK, MessageBoxIcon.Information);
            displayall();
            clearall();
        }
    }
}
catch (FormatException)
{
    errorProvider2.SetError(temail, "Enter valid format email");
}
catch (ArgumentException)
{
    MessageBox.Show("Enter all the details", "", 
MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
catch (SqlException)
{
    errorProvider5.SetError(tbasicpay, "Invalid Basic Pay");
}
}

private void maskedTextBox2_TextChanged(object sender, EventArgs e)
{
    //contact no textchanged
    errorProvider1.Clear();
}

private void temail_TextChanged(object sender, EventArgs e)
{
    //email textchanged
    errorProvider2.Clear();
}

private void nclear_Click(object sender, EventArgs e)
{
    //clear
}

```

```

        clearall();
        cbinactive.Checked = false;
        cbactive.Checked = false;
        displayall();
    }

    private void searchemp_Click(object sender, EventArgs e)
    {
        //search employee Click
        if (searchemp.Text == "search...")
            searchemp.Text = "";
        searchemp.ForeColor = Color.FromArgb(20, 168, 0);
    }

    private void searchemp_Leave(object sender, EventArgs e)
    {
        //search employee leave
        if (searchemp.Text == "")
            searchemp.Text = "search...";
        searchemp.ForeColor = Color.Silver;
    }

    private void searchemp_KeyUp(object sender, KeyEventArgs e)
    {
        //search employee keyup
        filter.ResetText();
        filter.Text = "Sort By";
        cbinactive.Checked = false;
        cbactive.Checked = false;
        SqlDataAdapter sda = new SqlDataAdapter("select * from Employee
where Name like '" + searchemp.Text + "%" + " or Emp_Id like '" + searchemp.Text +
"%" + "'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dataGridView1.DataSource = ds.Tables["dt"];
    }

    private void filter_SelectedIndexChanged(object sender, EventArgs e)
    {
        //filter by
        searchemp.Text = "search...";
        cbactive.Checked = false;
        cbinactive.Checked = false;
        sortfunc(filter.SelectedItem.ToString());
    }

    private void sortfunc(String value)
    {
        //sortfunc function
        SqlDataAdapter adp = new SqlDataAdapter();
        DataSet ds = new DataSet();
        c.cmd = new SqlCommand("select * from Employee", c.cn);
        adp.SelectCommand = c.cmd;
        ds.Clear();
        adp.Fill(ds);
        dataGridView1.Sort(dataGridView1.Columns[value],
ListSortDirection.Ascending);
    }

```

```

        }

        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    //dataGridView1_CellContentClick
    msg.Visible = false;
    if (dataGridView1.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
    {
        nupdate.Enabled = true;
        ndelete.Enabled = true;
        nadd.Enabled = false;
        dataGridView1.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        dataGridView1.CurrentRow.Selected = true;
        empid =
dataGridView1.Rows[e.RowIndex].Cells["Emp_Id"].FormattedValue.ToString();
        tname.Text =
dataGridView1.Rows[e.RowIndex].Cells["Name"].FormattedValue.ToString();
        temail.Text =
dataGridView1.Rows[e.RowIndex].Cells["Email"].FormattedValue.ToString();
        maskedTextBox2.Text =
dataGridView1.Rows[e.RowIndex].Cells["Contact_No"].FormattedValue.ToString();
        taddr.Text =
dataGridView1.Rows[e.RowIndex].Cells["Address"].FormattedValue.ToString();
        maskedTextBox1.Text =
dataGridView1.Rows[e.RowIndex].Cells["Dob"].FormattedValue.ToString();
        tage.Text =
dataGridView1.Rows[e.RowIndex].Cells["Age"].FormattedValue.ToString();
        dateTimePicker1.Text =
dataGridView1.Rows[e.RowIndex].Cells["Doj"].FormattedValue.ToString();
        tdesgn.Text =
dataGridView1.Rows[e.RowIndex].Cells["Designation"].FormattedValue.ToString();
        tbasicpay.Text =
dataGridView1.Rows[e.RowIndex].Cells["Basic_Pay"].FormattedValue.ToString();
        if
(dataGridView1.Rows[e.RowIndex].Cells["Gender"].FormattedValue.ToString() ==
"Female")
            radioButton1.Checked = true;
        else if
(dataGridView1.Rows[e.RowIndex].Cells["Gender"].FormattedValue.ToString() ==
"Male")
            radioButton2.Checked = true;
        else if
(dataGridView1.Rows[e.RowIndex].Cells["Gender"].FormattedValue.ToString() ==
"Other")
            radioButton3.Checked = true;
    }
}

private void nupdate_Click(object sender, EventArgs e)
{
    //update
    dt1 = Convert.ToDateTime(maskedTextBox1.Text);
    dob = dt1.ToString("MM-dd-yyyy");
    DOJ = dateTimePicker1.Value.ToString("MM-dd-yyyy");
}

```

```

        DateTime db = new DateTime(Convert.ToInt16(dt1.ToString("yyyy")),
Convert.ToInt16(dt1.ToString("MM")), Convert.ToInt16(dt1.ToString("dd")), 12,
00, 00);
        DateTime dj = new
DateTime(Convert.ToInt16(datePicker1.Value.ToString("yyyy")),
Convert.ToInt16(datePicker1.Value.ToString("MM")),
Convert.ToInt16(datePicker1.Value.ToString("dd")), 12, 00, 00);
        gender = "";
        if (radioButton1.Checked == true)
            gender = "Female";
        else if (radioButton2.Checked == true)
            gender = "Male";
        else if (radioButton3.Checked == true)
            gender = "Other";
    try
    {
        Regex rx = new Regex("^([6-9][0-9]{9})$");
        if (tname.Text == "" || taddr.Text == "" || Convert.ToString(maskedTextBox2.Text) == "" || temail.Text == "" || maskedTextBox2.Text == "" || tdesgn.Text == "" || tbasicpay.Text == "")
            throw new ArgumentException();
        else if (!(rx.IsMatch(maskedTextBox2.Text)))
            errorProvider1.SetError(maskedTextBox2, "Enter valid mobile number");
        else
        {
            new MailAddress(temail.Text);
            if (db > dj)
                errorProvider4.SetError(datePicker1, "invalid DOJ\\n DOJ should be greater than dob");
            else if (MessageBox.Show("Are You Sure want to Update ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                c.cmd.CommandText = "update Employee set Name='"
+ tname.Text + "',Address='"
+ taddr.Text + "',Email='"
+ temail.Text + "',Contact_No='"
+ Convert.ToString(maskedTextBox2.Text) + "',Dob='"
+ dob + "',Gender='"
+ gender + "',Age='"
+ tage.Text + "',Doj='"
+ DOJ + "',Designation='"
+ tdesgn.Text + "',Basic_Pay='"
+ tbasicpay.Text + "' where Emp_Id='"
+ empid + "'";
                c.cmd.ExecuteNonQuery();
                MessageBox.Show("Employee Updated Successfully!", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
                dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
                dataGridView1.Rows[0].Cells[0].Selected = true;
                displayall();
                clearall();
            }
        }
    }
    catch (FormatException)
    {
        errorProvider2.SetError(temail, "Enter valid format email");
    }
    catch (ArgumentException)
    {

```

```

        MessageBox.Show("Enter all the details", "", 
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    catch (SqlException)
    {
        errorProvider5.SetError(tbasicpay, "Invalid Basic Pay");
    }
}

private void ndelete_Click(object sender, EventArgs e)
{
    //delete
    c.cmd.CommandText = "select Status from Employee where Emp_Id='"
empid + "'";
    if (Convert.ToString(c.cmd.ExecuteScalar()) == "Inactive")
    {
        msg.Visible = true;
        msg.Text = "Employee already Inactivated!";
    }
    else
    {
        if (MessageBox.Show("Are You Sure want to Inactivate " +
tname.Text + " ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.Yes)
        {
            c.cmd.CommandText = "update Employee set Status='Inactive'
where Emp_Id='"
empid + "'";
            c.cmd.ExecuteNonQuery();
            MessageBox.Show("Employee Inactivated!", "", 
MessageBoxButtons.OK, MessageBoxIcon.Information);
            dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
            dataGridView1.Rows[0].Cells[0].Selected = true;
            displayall();
            clearall();
        }
    }
}

private void cbactive_CheckedChanged(object sender, EventArgs e)
{
    //checkbox active
    searchemp.Text = "Search...";
    filter.ResetText();
    filter.Text = "Sort By";
    displayall();
}

private void cbinactive_CheckedChanged(object sender, EventArgs e)
{
    //checkbox inactive
    searchemp.Text = "Search...";
    filter.ResetText();
    filter.Text = "Sort By";
    displayall();
}

```

```

        private void tname_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar == 
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || e.KeyChar == '.');
    }

        private void tdesign_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar == 
(char)Keys.Back);
    }

        private void maskedTextBox1_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !(char.IsDigit(e.KeyChar));
    }

        private void taddr_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar == 
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || char.IsDigit(e.KeyChar) || 
e.KeyChar == ',' || e.KeyChar == '-' || e.KeyChar == '&'amp; || e.KeyChar == '.');
    }

        private void searchemp_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar == 
(char)Keys.Back || char.IsDigit(e.KeyChar));
    }

        private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
    {
        errorProvider4.Clear();
    }

        private void tbasicpay_TextChanged(object sender, EventArgs e)
    {
        errorProvider5.Clear();
    }
}

```

Emp Attendance:

```

using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsApplication
{
    public partial class Emp_Attendance : Form
    {
        Connect c = new Connect();
        int totaldays;
        public Emp_Attendance()
        {

```

```

        InitializeComponent();
    }

    private void Attendance_Load(object sender, EventArgs e)
    {
        //page load
        Supplier s = new Supplier();
        s.datagridviewstyle(dataGridView1);
        displayall();
        SqlCommand sc = new SqlCommand("select * from Employee where
Status='Active'", c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(sc);
        DataSet ds = new DataSet();
        adp.Fill(ds);
        sc.ExecuteNonQuery();
        cbempid.DataSource = ds.Tables[0];
        cbempid.DisplayMember = "Emp_Id";
        cbempid.ValueMember = "Emp_Id";
        cbempid.Text = "Select";
        dateTimePicker2.MaxDate = DateTime.Today;
    }

    private void displayall()
    {
        dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
        SqlDataAdapter sda = new SqlDataAdapter("select * from
Attendance", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dataGridView1.DataSource = ds.Tables["dt"];
    }

    private void clearall()
    {
        cbempid.Text = "Select";
        cbmonth.Text = "Select";
        dateTimePicker2.Text = DateTime.Today.ToString();
        tname.Text = "";
        tltaken.Text = "";
        tlgiven.Text = "";
        searchemp.Text = "";
        filter.ResetText();
        filter.Text = "Sort By";
        dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
        searchemp.Text = "search...";
        nadd.Enabled = true;
        searchemp.ForeColor = Color.Silver;
        cbempid.Enabled = true;
        cbmonth.Enabled = true;
        dateTimePicker2.Enabled = true;
        nupdate.Enabled = false;
        nadd.Enabled = true;
    }

    private void nclear_Click(object sender, EventArgs e)

```

```

{
    //clear
    clearall();
    displayall();
}

private void nadd_Click(object sender, EventArgs e)
{
    //save
    try
    {
        if (cbempid.Text == "Select" || tname.Text == "" || 
tltaken.Text == "" || tlgiven.Text == "" || cbmonth.Text=="Select" || 
cbmonth.Text==".....")
            throw new ArgumentException();
        else
        {
            SqlCommand cmd = new SqlCommand("Select * from Attendence
where Emp_Id=''' + cbempid.Text + ''' and Month=''' +
cbmonth.SelectedItem.ToString() + ''' and Year=''' + dateTimePicker2.Text + '''
and Emp_Id=''' + cbempid.Text + '''", c.cn);
            SqlDataAdapter adp = new SqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            adp.Fill(dt);
            c.cmd.CommandText = "select Doj from Employee where
Emp_Id=''' + cbempid.Text + ''' ";
            DateTime DOJ = Convert.ToDateTime(c.cmd.ExecuteScalar());
            DateTime date = new
DateTime(Convert.ToInt16(dateTimePicker2.Text), cbmonth.SelectedIndex,
Convert.ToInt16(DateTime.Today.ToString("dd")), 12, 00, 00);
            if ((cbmonth.SelectedIndex >= DateTime.Now.Month &&
Convert.ToInt16(dateTimePicker2.Text) == DateTime.Now.Year) || DOJ>date)
                MessageBox.Show("Invalid Month!", "", 
MessageBoxButtons.OK, MessageBoxIcon.Error);
            else
            {
                totaldays =
DateTime.DaysInMonth(Convert.ToInt16(dateTimePicker2.Text),
cbmonth.SelectedIndex);
                if (Convert.ToInt16(tltaken.Text) > totaldays || 
Convert.ToInt16(tlgiven.Text) > totaldays)
                    MessageBox.Show("Please enter valid days.\nThe
days must be less than total number of days in month.", "", 
MessageBoxButtons.OK, MessageBoxIcon.Error);
                else if (dt.Rows.Count > 0)
                    MessageBox.Show("Attendance already exist!", "", 
MessageBoxButtons.OK, MessageBoxIcon.Error);
                else if (MessageBox.Show("Are You Sure?", "", 
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    int extral = 0;
                    int l = Convert.ToInt16(tlgiven.Text) -
Convert.ToInt16(tltaken.Text);
                    if (l > 0)
                        extral = 0;
                    else
                        extral = -1 * l;
                }
            }
        }
    }
}

```

```

        c.cmd.CommandText = "insert into Attendece
(Emp_Id,Month,Year,Total_days,Leave_taken,Leave_given,Extra_leave) values('" +
cbempid.Text + "','" + cbmonth.SelectedItem.ToString() + "','" +
dateTimePicker2.Text + "','" + totaldays + "','" + tltaken.Text + "','" +
tlgiven.Text + "','" + extral + ")";
        c.cmd.ExecuteNonQuery();
        MessageBox.Show("Attendece Details Saved!", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
        displayall();
        clearall();
    }
}
}
catch (ArgumentException)
{
    MessageBox.Show("Enter all the details", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}

private void searchemp_Click(object sender, EventArgs e)
{
    //search employee Click
    if (searchemp.Text == "search...")
        searchemp.Text = "";
    searchemp.ForeColor = Color.FromArgb(20, 168, 0);
}

private void searchemp_Leave(object sender, EventArgs e)
{
    //search employee leave
    if (searchemp.Text == "")
        searchemp.Text = "search...";
    searchemp.ForeColor = Color.Silver;
}

private void searchemp_KeyUp(object sender, KeyEventArgs e)
{
    //search employee keyup
    filter.ResetText();
    filter.Text = "Sort By";
    SqlDataAdapter sda = new SqlDataAdapter("select * from Attendece
where Emp_Id like '" + searchemp.Text + "%'", c.cn);
    DataSet ds = new DataSet();
    sda.Fill(ds, "dt");
    dataGridView1.DataSource = ds.Tables["dt"];
}

private void filter_SelectedIndexChanged(object sender, EventArgs e)
{
    //filter by
    searchemp.Text = "search...";
    sortfunc(filter.SelectedItem.ToString());
}

```

```

private void sortfunc(String value)
{
    //sortfunc function
    SqlDataAdapter adp = new SqlDataAdapter();
    DataSet ds = new DataSet();
    c.cmd = new SqlCommand("select * from Attendance", c.cn);
    adp.SelectCommand = c.cmd;
    ds.Clear();
    adp.Fill(ds);
    dataGridView1.Sort(dataGridView1.Columns[value],
ListSortDirection.Ascending);
}

private void cbempid_Leave(object sender, EventArgs e)
{
    if (!(cbempid.Text == "Select" || cbempid.Text==""))
    {
        SqlCommand cmd = new SqlCommand("select Name from Employee
where Emp_Id=' " + cbempid.Text + "'", c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        adp.Fill(dt);
        if (dt.Rows.Count > 0)
        {
            for (int i = 0; i < dt.Rows.Count; i++)
            {
                tname.Text = dt.Rows[i]["Name"].ToString();
            }
        }
    }
}

private void tltaken_KeyPress(object sender, KeyPressEventArgs e)
{
    //accept only numbers
    e.Handled = !char.IsDigit(e.KeyChar) && !
char.IsControl(e.KeyChar);
}

private void tlgiven_KeyPress(object sender, KeyPressEventArgs e)
{
    //accept only numbers
    e.Handled = !char.IsDigit(e.KeyChar) && !
char.IsControl(e.KeyChar);
}

private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if (dataGridView1.Rows[e.RowIndex].Cells[e.ColumnIndex].Value !=
null)
    {
        nupdate.Enabled = true;
        nadd.Enabled = false;
        cbempid.Enabled = false;
        cbmonth.Enabled = false;
        dateTimePicker2.Enabled = false;
    }
}

```

```

        dataGridView1.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        dataGridView1.CurrentRow.Selected = true;
        cbempid.Text =
dataGridView1.Rows[e.RowIndex].Cells["Emp_Id"].FormattedValue.ToString();
        cbmonth.Text =
dataGridView1.Rows[e.RowIndex].Cells["Month"].FormattedValue.ToString();
        String year =
dataGridView1.Rows[e.RowIndex].Cells["Year"].FormattedValue.ToString();
        dateTimePicker2.Text= DateTime.ParseExact(year,
"yyyy",null).ToShortDateString();
        tlgiven.Text =
dataGridView1.Rows[e.RowIndex].Cells["Leave_given"].FormattedValue.ToString();
        tltaken.Text =
dataGridView1.Rows[e.RowIndex].Cells["Leave_taken"].FormattedValue.ToString();
        c.cmd.CommandText = "select Name from Employee where Emp_Id='"
+ cbempid.Text + "'";
        tname.Text = c.cmd.ExecuteScalar().ToString();
    }
}

private void nupdate_Click(object sender, EventArgs e)
{
    try
    {
        if (tltaken.Text == "" || tlgiven.Text == "")
            throw new ArgumentException();
        else
        {
            int totaldays =
DateTime.DaysInMonth(Convert.ToInt16(dateTimePicker2.Text),
cbmonth.SelectedIndex);
            if (Convert.ToInt16(tltaken.Text) > totaldays ||
Convert.ToInt16(tlgiven.Text) > totaldays)
                MessageBox.Show("Please enter valid days.\nThe days
must be less than total number of days in month.", "", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            else if (MessageBox.Show("Are You Sure?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                int extral = 0;
                int l = Convert.ToInt16(tlgiven.Text) -
Convert.ToInt16(tltaken.Text);
                if (l > 0)
                    extral = 0;
                else
                    extral = -1 * l;
                c.cmd.CommandText = "update Attendence set
Total_days='"+totaldays+"',Leave_taken='"+tltaken.Text+"',Leave_given='"+tlgiven.Text+"',
Extra_leave='"+extral+"' where Emp_Id='"+cbempid.Text+"' and
Month='"+cbmonth.SelectedItem.ToString()+"' and
Year='"+dateTimePicker2.Text+"''";
                c.cmd.ExecuteNonQuery();
                MessageBox.Show("Attendance Details Updated!", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
                displayall();
                clearall();
            }
        }
    }
}

```

```

        }
    }
}
catch (ArgumentException)
{
    MessageBox.Show("Enter all the details", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}

private void searchemp_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsDigit(e.KeyChar));
}
}

```

Emp Manage User:

```

using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsApplication
{
    public partial class Emp_Manage_User : Form
    {
        Connect c = new Connect();
        public Emp_Manage_User()
        {
            InitializeComponent();
        }

        private void Manage_User_Load(object sender, EventArgs e)
        {
            refreshPage();
        }

        private void refreshPage()
        {
            //refresh function
            textBox1.Text = "";
            errorProvider1.Clear();
            errorProvider2.Clear();
            errorProvider3.Clear();
            errorProvider4.Clear();
            //Combobox1 - unlock
            SqlCommand sc = new SqlCommand("select * from Employee where
Status='Locked'", c.cn);
            SqlDataAdapter adp = new SqlDataAdapter(sc);
            DataSet ds = new DataSet();
            adp.Fill(ds);
            sc.ExecuteNonQuery();
            comboBox1.DataSource = ds.Tables[0];
            comboBox1.DisplayMember = "Emp_Id";
            comboBox1.ValueMember = "Emp_Id";
        }
    }
}

```

```

        comboBox1.Text = "Select";
        //Combobox2 - lock
        SqlCommand sc1 = new SqlCommand("select * from Employee where
Status='Active' or Status='Inactive'", c.cn);
        SqlDataAdapter adp1 = new SqlDataAdapter(sc1);
        DataSet ds1 = new DataSet();
        adp1.Fill(ds1);
        sc1.ExecuteNonQuery();
        comboBox2.DataSource = ds1.Tables[0];
        comboBox2.DisplayMember = "Emp_Id";
        comboBox2.ValueMember = "Emp_Id";
        comboBox2.Text = "Select";
        //Combobox3 - reset
        SqlCommand sc2 = new SqlCommand("select * from Employee where
Status='Active'", c.cn);
        SqlDataAdapter adp2 = new SqlDataAdapter(sc2);
        DataSet ds2 = new DataSet();
        adp2.Fill(ds2);
        sc2.ExecuteNonQuery();
        comboBox3.DataSource = ds2.Tables[0];
        comboBox3.DisplayMember = "Emp_Id";
        comboBox3.ValueMember = "Emp_Id";
        comboBox3.Text = "Select";
    }

    private void toplock_Click(object sender, EventArgs e)
{
    //lock panel
    errorProvider1.Clear();
    errorProvider2.Clear();
    errorProvider3.Clear();
    errorProvider4.Clear();
    panellock.BringToFront();
    toplock.BackColor = Color.FromArgb(176, 0, 32);
    topunlock.BackColor = Color.Gray;
}

private void topunlock_Click(object sender, EventArgs e)
{
    //unlock panel
    errorProvider1.Clear();
    errorProvider2.Clear();
    errorProvider3.Clear();
    errorProvider4.Clear();
    panelunlock.BringToFront();
    topunlock.BackColor = Color.ForestGreen;
    toplock.BackColor = Color.Gray;
}

private void btnunlock_Click(object sender, EventArgs e)
{
    if (comboBox1.Text == "Select")
        errorProvider1.SetError(comboBox1, "Please Select Employee
Id");
    else
    {

```

```

        c.cmd.CommandText = "update Employee set Status='Active' where
Emp_Id='" + comboBox1.Text + "'";
        c.cmd.ExecuteNonQuery();
        MessageBox.Show("Employee unlocked", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        refreshPage();
    }
}

private void btnlock_Click(object sender, EventArgs e)
{
    if (comboBox2.Text == "Select")
        errorProvider2.SetError(comboBox2, "Please Select Employee
Id");
    else
    {
        c.cmd.CommandText = "update Employee set Status='Locked' where
Emp_Id='" + comboBox2.Text + "'";
        c.cmd.ExecuteNonQuery();
        MessageBox.Show("Employee locked", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        refreshPage();
    }
}

private void button1_Click(object sender, EventArgs e)
{
    if (comboBox3.Text == "Select")
        errorProvider3.SetError(comboBox3, "Please Select Employee
Id");
    else if (textBox1.Text == "")
        errorProvider4.SetError(textBox1, "Required!");
    else
    {
        c.cmd.CommandText = "update Login set password='" +
textBox1.Text + "' where Emp_Id='" + comboBox3.Text + "'";
        c.cmd.ExecuteNonQuery();
        MessageBox.Show("Password Changed!", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        refreshPage();
    }
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    errorProvider4.Clear();
}

private void comboBox3_SelectedIndexChanged(object sender, EventArgs e)
{
    errorProvider3.Clear();
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{

```

```

        errorProvider1.Clear();
    }

    private void comboBox2_SelectedIndexChanged(object sender, EventArgs
e)
{
    errorProvider2.Clear();
}

}

```

Emp Salary:

```

using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsApplication
{
    public partial class btn_calculate : Form
    {
        Connect c = new Connect();
        public btn_calculate()
        {
            InitializeComponent();
        }

        private void Emp_Salary_Load(object sender, EventArgs e)
        {
            //page load
            Supplier s = new Supplier();
            s.datagridviewstyle(dataGridView1);
            SqlCommand sc = new SqlCommand("select * from Employee where
Status='Active' ", c.cn);
            SqlDataAdapter adp = new SqlDataAdapter(sc);
            DataSet ds = new DataSet();
            adp.Fill(ds);
            sc.ExecuteNonQuery();
            cbempid.DataSource = ds.Tables[0];
            cbempid.DisplayMember = "Emp_Id";
            cbempid.ValueMember = "Emp_Id";
            cbempid.Text = "Select";
            dateTimePicker2.MaxDate = DateTime.Today;
            displayall();
        }

        private void displayall()
        {
            SqlDataAdapter sda = new SqlDataAdapter("select * from Salary",
c.cn);
            DataSet ds = new DataSet();
            sda.Fill(ds, "dt");
            dataGridView1.DataSource = ds.Tables["dt"];
        }

        private void tbonus_KeyPress(object sender, KeyPressEventArgs e)

```

```

        {
            //accept only decimal numbers
            if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) &&
e.KeyChar != '.')
                e.Handled = true;
            else
                e.Handled = false;
        }

        private void cbempid_Leave(object sender, EventArgs e)
{
    //employeeid leave
    if (!(cbempid.Text == "Select" || cbempid.Text==""))
    {
        SqlCommand cmd = new SqlCommand("Select * from Employee where
Emp_Id=' " + cbempid.Text + "'", c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        adp.Fill(dt);
        if (dt.Rows.Count > 0)
        {
            for (int i = 0; i < dt.Rows.Count; i++)
            {
                tname.Text = dt.Rows[i]["Name"].ToString();
                tbasicpay.Text = dt.Rows[i]["Basic_Pay"].ToString();

            }
        }
        SqlDataAdapter adp1 = new SqlDataAdapter("select * from
Attendance where Emp_Id=' " + cbempid.Text + "'", c.cn);
        DataSet ds = new DataSet();
        adp1.Fill(ds);
        cbmonth.DataSource = ds.Tables[0];
        cbmonth.DisplayMember = "Month";
        cbmonth.ValueMember = "Month";
        cbmonth.Text = "Select";
    }
}

private void clearall()
{
    cbempid.Text = "Select";
    cbmonth.Text = "Select";
    dateTimePicker2.Text = DateTime.Today.ToString();
    ttax.Text = "";
    tname.Text = "";
    tbasicpay.Text = "";
    thra.Text = "";
    tpf.Text = "";
    ttotaldays.Text = "";
    textraleave.Text = "";
    ttotalthay.Text = "";
    tnetpay.Text = "";
    filter.ResetText();
    filter.Text = "Sort By";
}

```

```

        dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
        searchemp.Text = "search...";
        searchemp.ForeColor = Color.Silver;
    }

private void nclear_Click(object sender, EventArgs e)
{
    //clear
    clearall();
    displayall();
}

private void nadd_Click(object sender, EventArgs e)
{
    //save
    try
    {
        if (cbempid.Text == "Select" || cbmonth.Text == "Select")
            throw new ArgumentException();
        else if (ttax.Text == "")
            MessageBox.Show("Please calculate salary first!", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        else
        {
            SqlCommand cmd = new SqlCommand("Select * from Salary
where Emp_Id=''" + cbempid.Text + "' and Month=''" + cbmonth.Text + "' and
Year=''" + datepicker2.Text + "'", c.cn);
            SqlDataAdapter adp = new SqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            adp.Fill(dt);
            if (dt.Rows.Count > 0)
                MessageBox.Show("Salary details exist for this
month!", "", MessageBoxButtons.OK, MessageBoxIcon.Error);
            else if (MessageBox.Show("Are You Sure?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                c.cmd.CommandText = "select Count(*) from Salary";
                int cnt = Convert.ToInt16(c.cmd.ExecuteScalar());
                cnt++;
                String payid = "pay" + cnt;
                c.cmd.CommandText = "insert into Salary (Pay_Id)
values('" + payid + "')";
                c.cmd.ExecuteNonQuery();

                c.cmd.CommandText = "update Salary set Emp_Id=''" +
cbempid.Text + "',Date=''" + DateTime.Now.ToString("MM.dd.yyyy") + "'",Year=''" +
datepicker2.Text + "' ,Month=''" + cbmonth.Text + "' ,Tax=''" + ttax.Text + "' ,
HRA=''" + thra.Text + "' ,PF=''" + tpf.Text + "' ,Total_Pay=''" + ttotalpay.Text +
"' ,Net_Pay=''" + tnetpay.Text + "' where Pay_Id=''" + payid + "'";
                c.cmd.ExecuteNonQuery();
                MessageBox.Show("Salary Details Added Successfully.", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
                new
Emp_Salary_Print(cbempid.Text,cbmonth.Text,datepicker2.Text).Show();
                clearall();
                displayall();
            }
        }
    }
}

```

```

        }
    }
}
catch (ArgumentException)
{
    MessageBox.Show("Enter all the details", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
catch (Exception)
{
    throw;
}
}

private void btnCalculate_Click(object sender, EventArgs e)
{
    //calculate
    try
    {
        if (cbEmpID.Text == "Select" || cbMonth.Text == "Select")
            throw new ArgumentException();
        SqlCommand cmd = new SqlCommand("Select * from Attendance
where Emp_Id=''" + cbEmpID.Text + "' and Month=''" + cbMonth.Text + "' and
Year=''" + dateTimePicker2.Text + "' and Emp_Id=''" + cbEmpID.Text + "'", c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        adp.Fill(dt);
        if (dt.Rows.Count > 0)
        {
            for (int i = 0; i < dt.Rows.Count; i++)
            {
                tTotalDays.Text = dt.Rows[i]["Total_days"].ToString();
                extraLeave.Text = dt.Rows[i]
                ["Extra_leave"].ToString();
            }
            double basic = Convert.ToDouble(tbBasicPay.Text);
            double totalDays = Convert.ToDouble(tTotalDays.Text);
            double extraLeave = Convert.ToDouble(extraLeave.Text);
            basic = (basic / totalDays) * (totalDays - extraLeave);
            double tax = basic * 0.15;
            double HRA = basic * 0.4;
            double PF = basic * 0.12;
            double totalPay = HRA + basic;
            double netPay = totalPay - (PF + tax);
            tTax.Text = Math.Round(tax, 2).ToString();
            tHRA.Text = Math.Round(HRA, 2).ToString();
            tPF.Text = Math.Round(PF, 2).ToString();
            tTotalPay.Text = Math.Round(totalPay, 2).ToString();
            tNetPay.Text = Math.Round(netPay, 2).ToString();
        }
    }
    catch (ArgumentException)
    {
        MessageBox.Show("Enter all the details", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

```

```

}

private void searchemp_Click(object sender, EventArgs e)
{
    //search employee Click
    if (searchemp.Text == "search...")
        searchemp.Text = "";
    searchemp.ForeColor = Color.FromArgb(20, 168, 0);
}

private void searchemp_Leave(object sender, EventArgs e)
{
    //search employee leave
    if (searchemp.Text == "")
        searchemp.Text = "search...";
    searchemp.ForeColor = Color.Silver;
}

private void searchemp_KeyUp(object sender, KeyEventArgs e)
{
    //search employee keyup
    filter.ResetText();
    filter.Text = "Sort By";
    SqlDataAdapter sda = new SqlDataAdapter("select * from Salary
where Emp_Id like '" + searchemp.Text + "%'", c.cn);
    DataSet ds = new DataSet();
    sda.Fill(ds, "dt");
    dataGridView1.DataSource = ds.Tables["dt"];
}

private void filter_SelectedIndexChanged(object sender, EventArgs e)
{
    //filter by
    searchemp.Text = "search...";
    sortfunc(filter.SelectedItem.ToString());
}

private void sortfunc(String value)
{
    //sortfunc function
    SqlDataAdapter adp = new SqlDataAdapter();
    DataSet ds = new DataSet();
    c.cmd = new SqlCommand("select * from Salary", c.cn);
    adp.SelectCommand = c.cmd;
    ds.Clear();
    adp.Fill(ds);
    dataGridView1.Sort(dataGridView1.Columns[value],
ListSortDirection.Ascending);
}

private void searchemp_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsDigit(e.KeyChar));
}

```

```
    }  
}
```

Emp Salary Print:

```
using System.Windows.Forms;  
using System.Data.SqlClient;  
  
namespace WindowsApplication  
{  
    public partial class Emp_Salary_Print : Form  
    {  
        Connect c = new Connect();  
        string empid, month, year;  
        public Emp_Salary_Print(string empno, string m, string y)  
        {  
            InitializeComponent();  
            empid = empno;  
            month = m;  
            year = y;  
        }  
  
        private void Emp_Salary_Print_Load(object sender, EventArgs e)  
        {  
            label52.Text = DateTime.Now.ToString();  
            SqlDataAdapter adp = new SqlDataAdapter("Select * from Employee  
where Emp_Id='"+empid+"'", c.cn);  
            DataTable dt = new DataTable();  
            adp.Fill(dt);  
            if (dt.Rows.Count > 0)  
            {  
                label35.Text = dt.Rows[0]["Emp_Id"].ToString();  
                label34.Text = dt.Rows[0]["Name"].ToString();  
                label36.Text = dt.Rows[0]["Designation"].ToString();  
                DateTime dt1 = Convert.ToDateTime(dt.Rows[0]["Doj"]);  
                label37.Text = dt1.ToShortDateString();  
                label55.Text = dt.Rows[0]["Basic_Pay"].ToString();  
                label39.Text = dt.Rows[0]["Contact_No"].ToString();  
                label42.Text = dt.Rows[0]["Gender"].ToString();  
            }  
            SqlDataAdapter adp1 = new SqlDataAdapter("Select * from Attendance  
where Emp_Id=' " + empid + " ' and Month=' " + month + " ' and Year=' " + year + " '  
and Emp_Id=' " + empid + " '", c.cn);  
            DataTable dt2 = new DataTable();  
            adp1.Fill(dt2);  
            if (dt2.Rows.Count > 0)  
            {  
                label41.Text = dt2.Rows[0]["Total_days"].ToString();  
                label45.Text = dt2.Rows[0]["Extra_leave"].ToString();  
            }  
            SqlDataAdapter adp3 = new SqlDataAdapter("Select * from Salary  
where Emp_Id=' " + empid + " ' and Month=' " + month + " ' and Year=' " + year + " '  
and Emp_Id=' " + empid + " '", c.cn);  
            DataTable dt3 = new DataTable();  
            adp3.Fill(dt3);  
            if (dt3.Rows.Count > 0)
```

```

        {
            double ld = (Convert.ToDouble(label55.Text) /
Convert.ToDouble(label41.Text)) * (Convert.ToDouble(label41.Text) -
Convert.ToDouble(label45.Text));
            label16.Text= Math.Round((Convert.ToDouble(label55.Text)-
ld),2).ToString();
            if (label16.Text == "0")
                label16.Text = "0.0";
            label46.Text = dt3.Rows[0]["HRA"].ToString();
            label47.Text = dt3.Rows[0]["PF"].ToString();
            label49.Text = dt3.Rows[0]["Tax"].ToString();
            label48.Text = (Convert.ToDouble(label46.Text) +
Convert.ToDouble(label55.Text)).ToString();
            label50.Text = (Convert.ToDouble(label47.Text) +
Convert.ToDouble(label49.Text)).ToString();
            label51.Text = dt3.Rows[0]["Net_Pay"].ToString();
            string amt = label51.Text;
            String word =
ConvertNumberToWords(Convert.ToInt32(amt.Split('.')[0])) + ' ' + "RUPEES" + ' '
+ ConvertNumberToWords(Convert.ToInt32(amt.Split('.')[1])) + ' ' + "PAISA
ONLY";
            label26.Text = word;
            string m = dt3.Rows[0]["Month"].ToString();
            string y = dt3.Rows[0]["Year"].ToString();
            label8.Text = m + " - " + y;
            label53.Text = dt3.Rows[0]["Pay_Id"].ToString();
        }
    }

    public static string ConvertNumberToWords(int number)
{
    //amount in words
    if (number == 0)
        return "ZERO";
    if (number < 0)
        return "minus " + ConvertNumberToWords(Math.Abs(number));
    string words = "";
    if ((number / 1000000000) > 0)
    {
        words += ConvertNumberToWords(number / 1000000000) + " Billion
";
        number %= 1000000000;
    }
    if ((number / 10000000) > 0)
    {
        words += ConvertNumberToWords(number / 10000000) + " CRORE ";
        number %= 10000000;
    }
    if ((number / 1000000) > 0)
    {
        words += ConvertNumberToWords(number / 1000000) + " MILLION ";
        number %= 1000000;
    }
    if ((number / 100000) > 0)
    {
        words += ConvertNumberToWords(number / 100000) + " LAKH ";
    }
}

```

```

        number %= 100000;
    }
    if ((number / 1000) > 0)
    {
        words += ConvertNumbertoWords(number / 1000) + " THOUSAND ";
        number %= 1000;
    }
    if ((number / 100) > 0)
    {
        words += ConvertNumbertoWords(number / 100) + " HUNDRED ";
        number %= 100;
    }
    if (number > 0)
    {
        if (words != "")
            words += "AND ";
        string[] unitsMap = new string[20] { "ZERO", "ONE", "TWO",
"THREE", "FOUR", "FIVE", "SIX", "SEVEN", "EIGHT", "NINE", "TEN", "ELEVEN",
"TWELVE", "THIRTEEN", "FOURTEEN", "FIFTEEN", "SIXTEEN", "SEVENTEEN",
"EIGHTEEN", "NINETEEN" };
        string[] tensMap = new string[10] { "ZERO", "TEN", "TWENTY",
"THIRTY", "FORTY", "FIFTY", "SIXTY", "SEVENTY", "EIGHTY", "NINETY" };
        if (number < 20)
            words += unitsMap[number];
        else
        {
            words += tensMap[number / 10];
            if ((number % 10) > 0)
                words += " " + unitsMap[number % 10];
        }
    }
    return words;
}

Bitmap img;
private void capturescreen()
{
    Panel p = new Panel();
    this.Controls.Add(p);
    Graphics g = p.CreateGraphics();
    Size s = this.ClientSize;
    img = new Bitmap(s.Width, s.Height, g);
    Graphics mg = Graphics.FromImage(img);
    Point pt = PointToScreen(p.Location);
    mg.CopyFromScreen(pt.X, pt.Y, 0, 0, s);
}

private void button1_Click(object sender, EventArgs e)
{
    try
    {
        button1.Visible = false;
        capturescreen();
        printDocument1.Print();
    }
    catch (Exception)
    {

```

```

        MessageBox.Show("bill cannot be printed");
    }
}

private void printDocument1_PrintPage(object sender,
System.Drawing.Printing.PrintPageEventArgs e)
{
    e.Graphics.DrawImage(img, 50, 100);
}
}
}

```

Program.cs:

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace WindowsApplication
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            //Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}

```

Purchase Bill:

```

using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsApplication
{
    public partial class Purchase_Bill : Form
    {
        Connect c = new Connect();
        String pbid,itemid,pitemid;
        int qty;
        double totalamt;
        public Purchase_Bill()
        {
            InitializeComponent();
        }

        private void Purchase_Bill_Load(object sender, EventArgs e)
        {

```

```

//page load
Supplier s = new Supplier();
s.datagridviewstyle(dgvpurchasebilld);
s.datagridviewstyle(dgvpurchaseoraderd);
btnnext.Enabled = false;
dateTimePicker1.Value = DateTime.Today;
dropdownpurchaseorer();
dateTimePicker1.MinDate = DateTime.Today;
dateTimePicker1.MaxDate = DateTime.Today.AddMonths(1);
}

private void dropdownpurchaseorer()
{
    SqlCommand sc = new SqlCommand("select distinct
Purchase_Order.PO_No from Purchase_Order inner join Purchase_Order_Detail on
Purchase_Order.PO_No=Purchase_Order_Detail.PO_No where
Purchase_Order.Status='Ordered'", c.cn);
    SqlDataAdapter adp = new SqlDataAdapter(sc);
    DataSet ds = new DataSet();
    adp.Fill(ds);
    sc.ExecuteNonQuery();
    cbpurchaseorder.DataSource = ds.Tables[0];
    cbpurchaseorder.DisplayMember = "PO_No";
    cbpurchaseorder.ValueMember = "PO_No";
    cbpurchaseorder.Text = "Select";
}

private void displayall()
{
    dgvpurchaseoraderd.SelectionMode =
DataGridViewSelectionMode.CellSelect;
    SqlDataAdapter adp1 = new SqlDataAdapter("select
Purchase_Order_Detail.Item_Id,Inventory.Name,Purchase_Order_Detail.Qty,Invento
ry.Price,Inventory.Warranty from Purchase_Order_Detail inner join
Purchase_Order on Purchase_Order.PO_No = Purchase_Order_Detail.PO_No inner
join Inventory on Purchase_Order_Detail.Item_Id = Inventory.Item_Id where
Purchase_Order.PO_No=''' + cbpurchaseorder.Text + ''' ", c.cn);
    DataSet ds1 = new DataSet();
    adp1.Fill(ds1, "dt1");
    dgvpurchaseoraderd.DataSource = ds1.Tables["dt1"];
}

private void displayall1()
{
    dgvpurchasebilld.SelectionMode =
DataGridViewSelectionMode.CellSelect;
    SqlDataAdapter adp1 = new SqlDataAdapter("select
PB_No,Item_Id,Qty,Price,Amount,Warranty from Purchase_Bill_Detail where
PB_No=''' + pbid + ''' ", c.cn);
    DataSet ds2 = new DataSet();
    adp1.Fill(ds2, "dt2");
    dgvpurchasebilld.DataSource = ds2.Tables["dt2"];
}

private void totalamts()
{

```

```

        c.cmd.CommandText = "Select Amount from Purchase_Bill_Detail where
PB_No=''' + pbid + '''";
        if (!(c.cmd.ExecuteScalar() == null))
        {
            c.cmd.CommandText = "Select sum(Amount) from
Purchase_Bill_Detail where PB_No=''' + pbid + '''";
            totalamt = Convert.ToDouble(c.cmd.ExecuteScalar());
            ttamt.Text = totalamt.ToString();
        }
        else
            ttamt.Text = "";
    }

    private void clearall()
    {
        dateTimePicker1.Text = "";
        tqty.Text = "";
        tprice.Text = "";
        twarranty.Text = "";
        dateTimePicker1.Enabled = true;
        btnnext.Enabled = false;
        btnsave.Enabled = true;
        nupdate.Enabled = false;
        ndelete.Enabled = false;
        cbpurchaseorder.Enabled = true;
        cbpurchaseorder.Text = "Select";
        tsupid.Text = "";
        tsupname.Text = "";
        ttamt.Text = "";
    }

    private void btnsave_Click(object sender, EventArgs e)
    {
        if (cbpurchaseorder.Text == "Select" || cbpurchaseorder.Text ==
"")
            MessageBox.Show("please select Purchase Order Number", "",

MessageBoxButtons.OK, MessageBoxIcon.Warning);
        else
        {
            dateTimePicker1.Enabled = false;
            SqlCommand cmd = new SqlCommand("Select * from Purchase_Bill
where Total_Amount is Null", c.cn);
            DataTable dtb = new DataTable();
            SqlDataAdapter adp = new SqlDataAdapter(cmd);
            adp.Fill(dtb);
            foreach (DataRow r in dtb.Rows)
            {
                c.cmd.CommandText = "delete from Purchase_Bill_Detail
where PB_No=''' + r["PB_No"] + '''";
                c.cmd.ExecuteNonQuery();
                c.cmd.CommandText = "delete from Purchase_Bill where
PB_No=''' + r["PB_No"] + '''";
                c.cmd.ExecuteNonQuery();
            }
            c.cmd.CommandText = "select Count(*) from Purchase_Bill";
            int cnt = Convert.ToInt16(c.cmd.ExecuteScalar());
            cnt++;
        }
    }
}

```

```

        pbid = "pb" + cnt;
        String date = dateTimePicker1.Value.ToString("MM-dd-yyyy");
        c.cmd.CommandText = "insert into
Purchase_Bill(PB_No,Date,PO_No) values('" + pbid +
"', '" + date + "','" + cbpurchaseorder.Text + "')";
        c.cmd.ExecuteNonQuery();
        MessageBox.Show("Details Saved!", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        btnnext.Enabled = true;
        btnsave.Enabled = false;
        cbpurchaseorder.Enabled = false;
        displayall();
        c.cmd.CommandText = "select Sup_Id from Purchase_Order where
PO_No=''" + cbpurchaseorder.Text + "''";
        tsupid.Text = c.cmd.ExecuteScalar().ToString();
        c.cmd.CommandText = "select Name from Supplier where Sup_Id='"
+ tsupid.Text + "''";
        tsupname.Text = c.cmd.ExecuteScalar().ToString();
    }

    private void dgvpurchaseoraderd_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if (dgvpurchaseoraderd.Rows[e.RowIndex].Cells[e.ColumnIndex].Value
!= null)
    {
        dgvpurchaseoraderd.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        dgvpurchaseoraderd.CurrentRow.Selected = true;
        itemid =
dgvpurchaseoraderd.Rows[e.RowIndex].Cells["Item_Id"].FormattedValue.ToString()
;
        qty =
Convert.ToInt32(dgvpurchaseoraderd.Rows[e.RowIndex].Cells["Qty"].FormattedValue
.ToString());
        tprice.Text =
dgvpurchaseoraderd.Rows[e.RowIndex].Cells["Price"].FormattedValue.ToString();
        twarranty.Text =
dgvpurchaseoraderd.Rows[e.RowIndex].Cells["Warranty"].FormattedValue.ToString()
);
        tqty.Text = qty.ToString();
    }
}

private void tprice_KeyPress(object sender, KeyPressEventArgs e)
{
    //accept only decimal numbers
    e.Handled = !char.IsControl(e.KeyChar) && !
char.IsDigit(e.KeyChar) && e.KeyChar != '.';
}

private void twarranty_KeyPress(object sender, KeyPressEventArgs e)
{
    //accept only decimal numbers
    e.Handled = !char.IsControl(e.KeyChar) && !
char.IsDigit(e.KeyChar) && e.KeyChar != '.';
}

```

```

        }

        private void tqty_KeyPress(object sender, KeyPressEventArgs e)
        {
            //accept only numbers
            e.Handled = !char.IsDigit(e.KeyChar) && !
            char.IsControl(e.KeyChar);
        }

        private void btnnext_Click(object sender, EventArgs e)
        {
            try
            {
                SqlCommand cmd = new SqlCommand("select
PB_No,Item_Id,Qty,Price,Amount,Warranty from Purchase_Bill_Detail where
PB_No=''" + pbid + "' and Item_Id=''" + itemid + "'", c.cn);
                SqlDataAdapter adp = new SqlDataAdapter(cmd);
                DataTable dt = new DataTable();
                adp.Fill(dt);

                if (!(dgvpurchaseoraderd.SelectionMode ==
DataGridViewSelectionMode.FullRowSelect))
                    MessageBox.Show("please select Item", "",

                MessageBoxButtons.OK, MessageBoxIcon.Warning);
                else if (tqty.Text == "" || tqty.Text == "0")
                    errorProvider1.SetError(tqty, "Required!");
                else if (Convert.ToInt16(tqty.Text) > qty)
                    errorProvider1.SetError(tqty, "Quantity should be equal or
less than specified!");
                else if (tprice.Text == "" || tprice.Text == "0.0" ||
tprice.Text == "0")
                    errorProvider2.SetError(tprice, "Required!");
                else if (twarranty.Text == "")
                    errorProvider3.SetError(twarranty, "Required!");
                else if (dt.Rows.Count > 0)
                    MessageBox.Show("You already added this item.\nPlease
edititem if you want to make changes.", "", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
                else
                {
                    double amt = Convert.ToDouble(tqty.Text) *
                    Convert.ToDouble(tprice.Text);
                    c.cmd.CommandText = "insert into Purchase_Bill_Detail
(PB_No,Item_Id,Qty,Price,Amount,Warranty) values(''" + pbid + "','" + itemid +
"', '" + tqty.Text + "','" + tprice.Text + "','" + amt + "','" + twarranty.Text + "')";
                    c.cmd.ExecuteNonQuery();
                    tqty.Text = "";
                    tprice.Text = "";
                    twarranty.Text = "";
                    displayall1();
                    displayall();
                    totalamts();
                }
            }
            catch (OverflowException)
            {
                errorProvider1.SetError(tqty, "Invalid Quantity!");
            }
        }
    }
}

```

```

        }
        catch (SqlException)
        {
            errorProvider2.SetError(tprice, "Invalid Price!");
        }
    }

    private void btngeneratebil_Click(object sender, EventArgs e)
    {
        c.cmd.CommandText = "select Count(*) from Purchase_Bill_Detail
where PB_No=''' + pbid + '''';
        int rc = Convert.ToInt16(c.cmd.ExecuteScalar());
        c.cmd.CommandText = "select Count(*) from Purchase_Bill where
PB_No=''' + pbid + '''';
        int rc2 = Convert.ToInt16(c.cmd.ExecuteScalar());
        if (rc >= 1 && rc2 == 1)
        {
            if (MessageBox.Show("Are You Sure ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                c.cmd.CommandText = "update Purchase_Bill set
Total_Amount=" + ttamt.Text+ " where PB_No=''' + pbid + '''';
                c.cmd.ExecuteNonQuery();
                c.cmd.CommandText = "update Purchase_Order set
Status='Invoiced' where PO_No=''' + cbpurchaseorder.Text + '''';
                c.cmd.ExecuteNonQuery();
                SqlCommand cmd = new SqlCommand("select * from
Purchase_Bill_Detail where PB_No=''' + pbid + '''", c.cn);
                DataTable dt = new DataTable();
                SqlDataAdapter adp = new SqlDataAdapter(cmd);
                adp.Fill(dt);
                foreach (DataRow r in dt.Rows)
                {
                    c.cmd.CommandText = "Select Qty from Inventory where
Item_Id=''' + r["Item_Id"] + '''';
                    int qt = Convert.ToInt32(c.cmd.ExecuteScalar());
                    c.cmd.CommandText = "update Inventory set Qty=" + qt +
" + " + r["Qty"] + ",Price=" + r["Price"] + ",Warranty=" + r["Warranty"] +
",Status='Active' where Item_Id=''' + r["Item_Id"] + '''";
                    c.cmd.ExecuteNonQuery();
                }
                MessageBox.Show("Bill generated Successfully!", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
                clearall();
                dropdownpurchaseorer();
                pbid = null;
                displayall();
                displayall1();
                ttamt.Text = "";
            }
        }
        else
            MessageBox.Show("You have not stored the details!", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }

    private void btncancel_Click(object sender, EventArgs e)

```

```

        {
            c.cmd.CommandText = "select Count(*) from Purchase_Bill_Detail
where PB_No=''' + pbid + '''';
            int rc1 = Convert.ToInt16(c.cmd.ExecuteScalar());
            c.cmd.CommandText = "select Count(*) from Purchase_Bill where
PB_No=''' + pbid + '''';
            int rc2 = Convert.ToInt16(c.cmd.ExecuteScalar());
            if (rc1 >= 1 && rc2 == 1)
            {
                if (MessageBox.Show("Are You Sure ?", "",

MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    c.cmd.CommandText = "delete from Purchase_Bill_Detail
where PB_No=''' + pbid + '''';
                    c.cmd.ExecuteNonQuery();
                    c.cmd.CommandText = "delete from Purchase_Bill where
PB_No=''' + pbid + '''';
                    c.cmd.ExecuteNonQuery();
                    MessageBox.Show("Bill Generation Cancelled!", "",

MessageBoxButtons.OK, MessageBoxIcon.Information);
                    dropdownpurchaseorer();
                    clearall();
                    pbid = null;
                    displayall();
                    displayall1();
                    ttamt.Text = "";
                }
            }
            else
                MessageBox.Show("You have not stored the details!", "",

MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }

        private void dgvpurchasebilld_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
            if (dgvpurchasebilld.Rows[e.RowIndex].Cells[e.ColumnIndex].Value !=

null)
            {
                nupdate.Enabled = true;
                ndelete.Enabled = true;
                btnnext.Enabled = false;
                dgvpurchasebilld.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
                dgvpurchasebilld.CurrentRow.Selected = true;
                tqty.Text =
dgvpurchasebilld.Rows[e.RowIndex].Cells["Qty"].FormattedValue.ToString();
                tprice.Text =
dgvpurchasebilld.Rows[e.RowIndex].Cells["Price"].FormattedValue.ToString();
                twarranty.Text =
dgvpurchasebilld.Rows[e.RowIndex].Cells["Warranty"].FormattedValue.ToString();
                pitemid =
dgvpurchasebilld.Rows[e.RowIndex].Cells["Item_Id"].FormattedValue.ToString();
            }
        }

        private void nupdate_Click(object sender, EventArgs e)

```

```

{
    try
    {
        if (tqty.Text == "" || tqty.Text == "0")
            errorProvider1.SetError(tqty, "Required!");
        else if (Convert.ToInt16(tqty.Text) > qty)
            errorProvider1.SetError(tqty, "Quantity should be equal or
less than specified!");
        else if (tprice.Text == "" || tprice.Text == "0")
            errorProvider2.SetError(tprice, "Required!");
        else if (twarranty.Text == "")
            errorProvider3.SetError(twarranty, "Required!");
        else
        {
            double amt = Convert.ToDouble(tqty.Text) *
Convert.ToDouble(tprice.Text);
            c.cmd.CommandText = "update Purchase_Bill_Detail set Qty="
+ tqty.Text + ",Price=" + tprice.Text + ",Warranty=" + twarranty.Text +
",Amount=" + amt + " where PB_No='" + pbid + "' and Item_Id='" + pitemid + "'";
            c.cmd.ExecuteNonQuery();
            MessageBox.Show("Item Updated!", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            tqty.Text = "";
            tprice.Text = "";
            twarranty.Text = "";
            displayall();
            displayall1();
            totalamts();
            nupdate.Enabled = false;
            ndelete.Enabled = false;
            btnnext.Enabled = true;
        }
    }
    catch (OverflowException)
    {
        errorProvider1.SetError(tqty, "Invalid Quantity!");
    }
    catch (SqlException)
    {
        errorProvider2.SetError(tprice, "Invalid Price!");
    }
}

private void ndelete_Click(object sender, EventArgs e)
{
    c.cmd.CommandText = "delete from Purchase_Bill_Detail where
PB_No=''" + pbid + "' and Item_Id=''" + pitemid + "'";
    c.cmd.ExecuteNonQuery();
    MessageBox.Show("Item Deleted!", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    totalamts();
    tqty.Text = "";
    tprice.Text = "";
    twarranty.Text = "";
    displayall1();
    displayall();
}

```

```

        totalamts();
        nupdate.Enabled = false;
        ndelete.Enabled = false;
        btnnext.Enabled = true;
    }

    private void tqty_TextChanged(object sender, EventArgs e)
    {
        errorProvider1.Clear();
    }

    private void tprice_TextChanged(object sender, EventArgs e)
    {
        errorProvider2.Clear();
    }

    private void twarranty_TextChanged(object sender, EventArgs e)
    {
        errorProvider3.Clear();
    }
}
}

```

Purchase Order:

```

using System.Windows.Forms;
using System.Data.SqlClient;
using System.Net.Mail;
using System.Net;

namespace WindowsApplication
{
    public partial class Purchase_Order : Form
    {
        Connect c = new Connect();
        String itemid,supid,poid,pitemid,supname;
        int qty;
        public Purchase_Order()
        {
            InitializeComponent();
        }

        private void Purchase_Load(object sender, EventArgs e)
        {
            //page load
            Supplier s = new Supplier();
            s.datagridviewstyle(dgvinventoryd);
            s.datagridviewstyle(dgvsupplierd);
            s.datagridviewstyle(dgvpurchaseorderd);
            dgvinventoryd.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;
            displayall();
            btnnextpur.Enabled = false;
            dateTimePicker1.Value = DateTime.Today;
            dateTimePicker1.MinDate = DateTime.Today;
        }
    }
}

```

```

private void displayall()
{
    dgvsupplierd.SelectionMode = DataGridViewSelectionMode.CellSelect;
    dgvinventoryd.SelectionMode =
DataGridViewSelectionMode.CellSelect;
    SqlDataAdapter sda = new SqlDataAdapter("select
Item_Id,Name,Qty,Material,Color,Category from Inventory ", c.cn);
    DataSet ds = new DataSet();
    sda.Fill(ds, "dt");
    dgvinventoryd.DataSource = ds.Tables["dt"];

    SqlDataAdapter adp = new SqlDataAdapter("select
Sup_Id,Name,Contact_No from Supplier where Status='Active'", c.cn);
    DataSet ds1 = new DataSet();
    adp.Fill(ds1, "dt1");
    dgvsupplierd.DataSource = ds1.Tables["dt1"];
}

private void displayall1()
{
    dgvpurchaseorderd.SelectionMode =
DataGridViewSelectionMode.CellSelect;
    dgvinventoryd.SelectionMode =
DataGridViewSelectionMode.CellSelect;
    SqlDataAdapter sda = new SqlDataAdapter("select
Item_Id,Name,Qty,Material,Color,Category from Inventory ", c.cn);
    DataSet ds = new DataSet();
    sda.Fill(ds, "dt");
    dgvinventoryd.DataSource = ds.Tables["dt"];

    SqlDataAdapter adp1 = new SqlDataAdapter("select
Purchase_Order_Detail.PO_No,Purchase_Order.Sup_Id,Purchase_Order_Detail.Item_I
d,Inventory.Name,Purchase_Order_Detail.Qty from Purchase_Order_Detail inner
join Purchase_Order on Purchase_Order_Detail.PO_No=Purchase_Order.PO_No inner
join Inventory on Purchase_Order_Detail.Item_Id= Inventory.Item_Id where
Purchase_Order.PO_No=''' + poid + "' ", c.cn);
    DataSet ds2 = new DataSet();
    adp1.Fill(ds2, "dt2");
    dgvpurchaseorderd.DataSource = ds2.Tables["dt2"];
}

private void clearall()
{
    dateTimePicker1.Text = "";
    tqty.Text = "";
    vsupname.Text = "search...";
    vsupname.ForeColor = Color.Silver;
    searchitem.Text = "search...";
    searchitem.ForeColor = Color.Silver;
    groupBox1.Enabled = true;
    dateTimePicker1.Enabled = true;
    btnnextpur.Enabled = false;
    btnsave.Enabled = true;
    nupdate.Enabled = false;
    ndelete.Enabled = false;
    groupBox2.Enabled = true;
}

```

```

        }

    private void vsupname_Click(object sender, EventArgs e)
    {
        if (vsupname.Text == "search...")
            vsupname.Text = "";
        vsupname.ForeColor = Color.FromArgb(20, 168, 0);
    }

    private void vsupname_Leave(object sender, EventArgs e)
    {
        if (vsupname.Text == "")
            vsupname.Text = "search...";
        vsupname.ForeColor = Color.Silver;
    }

    private void vsupname_KeyUp(object sender, KeyEventArgs e)
    {
        SqlDataAdapter sda = new SqlDataAdapter("select
Sup_Id,Name,Contact_No from Supplier where Status='Active' and Name like '" +
vsupname.Text + "%' or Status='Active' and Sup_Id like '" + vsupname.Text + "%'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgSupplierd.DataSource = ds.Tables["dt"];
    }

    private void searchitem_Click(object sender, EventArgs e)
    {
        //search item Click
        if (searchitem.Text == "search...")
            searchitem.Text = "";
        searchitem.ForeColor = Color.FromArgb(20, 168, 0);
    }

    private void searchitem_Leave(object sender, EventArgs e)
    {
        //search item leave
        if (searchitem.Text == "")
            searchitem.Text = "search...";
        searchitem.ForeColor = Color.Silver;
    }

    private void searchitem_KeyUp(object sender, KeyEventArgs e)
    {
        //search item keyup
        SqlDataAdapter sda = new SqlDataAdapter("select
Item_Id,Name,Qty,Material,Color,Category from Inventory where Name like '" +
searchitem.Text + "%'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgInventoryd.DataSource = ds.Tables["dt"];
    }

    private void dgInventoryd_CellContentClick(object sender,
DataGridViewCellEventArgs e)

```

```

        {
            if (dgvinventoryd.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
            {
                dgvinventoryd.SelectionMode =
                    DataGridViewSelectionMode.FullRowSelect;
                dgvinventoryd.CurrentRow.Selected = true;
                itemid =
                    dgvinventoryd.Rows[e.RowIndex].Cells["Item_Id"].FormattedValue.ToString();
                qty =
                    Convert.ToInt32(dgvinventoryd.Rows[e.RowIndex].Cells["Qty"].FormattedValue.ToString());
                tqty.Text = qty.ToString();
            }
        }

        private void dgvsupplierd_CellContentClick(object sender,
            DataGridViewCellEventArgs e)
        {
            if (dgvsupplierd.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
            {
                dgvsupplierd.SelectionMode =
                    DataGridViewSelectionMode.FullRowSelect;
                dgvsupplierd.CurrentRow.Selected = true;
                supid =
                    dgvsupplierd.Rows[e.RowIndex].Cells["Sup_Id"].FormattedValue.ToString();
                supname =
                    dgvsupplierd.Rows[e.RowIndex].Cells["Name"].FormattedValue.ToString();
            }
        }

        private void btnsave_Click(object sender, EventArgs e)
        {
            if (!(dgvsupplierd.SelectionMode ==
                    DataGridViewSelectionMode.FullRowSelect))
                MessageBox.Show("please select supplier", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            else
            {
                groupBox1.Enabled = false;
                dateTimePicker1.Enabled = false;
                SqlCommand cmd = new SqlCommand("Select * from Purchase_Order
where Status is Null", c.cn);
                DataTable dt = new DataTable();
                SqlDataAdapter adp = new SqlDataAdapter(cmd);
                adp.Fill(dt);
                foreach (DataRow r in dt.Rows)
                {
                    c.cmd.CommandText = "delete from Purchase_Order_Detail
where PO_No=' " + r["PO_No"] + " '";
                    c.cmd.ExecuteNonQuery();
                    c.cmd.CommandText="delete from Purchase_Order where
PO_No=' "+r["PO_No"]+" '";
                    c.cmd.ExecuteNonQuery();
                }
                c.cmd.CommandText = "select Count(*) from Purchase_Order";
            }
        }
    
```

```

        int cnt = Convert.ToInt16(c.cmd.ExecuteScalar());
        cnt++;
        poid = "po" + cnt;
        String date = dateTimePicker1.Value.ToString("MM-dd-yyyy");
        c.cmd.CommandText = "insert into Purchase_Order
(Po_No,Sup_Id,Date) values('" + poid + "','" + supid + "','" + date + ')';
        c.cmd.ExecuteNonQuery();
        MessageBox.Show("Details Saved!", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        btnnextpur.Enabled = true;
        btnSave.Enabled = false;
    }
}

private void btnnextpur_Click(object sender, EventArgs e)
{
    try
    {
        SqlCommand cmd = new SqlCommand("select Po_No,Item_Id,Qty from
Purchase_Order_Detail where Po_No='" + poid + "' and Item_Id='" + itemid +
"', c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        adp.Fill(dt);

        if (!(dgvinventoryd.SelectionMode ==
DataGridViewSelectionMode.FullRowSelect))
            MessageBox.Show("please select Item", "", MessageBoxButtons.OK,
 MessageBoxIcon.Warning);
        else if (tqty.Text == "" || tqty.Text == "0")
            errorProvider1.SetError(tqty, "Required!");
        else if (dt.Rows.Count > 0)
            MessageBox.Show("You already added this item to place
order.\nPlease edit item if you want to make changes.", "", MessageBoxButtons.OK,
 MessageBoxIcon.Warning);
        else
        {
            c.cmd.CommandText = "insert into Purchase_Order_Detail
(Po_No,Item_Id,Qty) values('" + poid + "','" + itemid + "','" + tqty.Text +
")";
            c.cmd.ExecuteNonQuery();
            tqty.Text = "";
            displayall1();
        }
    }
    catch (SqlException)
    {
        errorProvider1.SetError(tqty, "Invalid Quantity!");
    }
}

private void btncancel_Click(object sender, EventArgs e)
{
    c.cmd.CommandText = "select Count(*) from Purchase_Order_Detail
where Po_No='" + poid + "'";
    int rc1 = Convert.ToInt16(c.cmd.ExecuteScalar());
}

```

```

        c.cmd.CommandText = "select Count(*) from Purchase_Order where
PO_No=''' + poid + '''";
        int rc2 = Convert.ToInt16(c.cmd.ExecuteScalar());
        if (rc1 >= 1 && rc2 == 1)
        {
            if (MessageBox.Show("Are You Sure ?", "",

MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                c.cmd.CommandText = "delete from Purchase_Order_Detail
where PO_No=''' + poid + '''";
                c.cmd.ExecuteNonQuery();
                c.cmd.CommandText = "delete from Purchase_Order where
PO_No=''' + poid + '''";
                c.cmd.ExecuteNonQuery();
                MessageBox.Show("Order Cancelled!", "",

MessageBoxButtons.OK, MessageBoxIcon.Information);
                poid = null;
                clearall();
                displayall();
                displayall1();
            }
        }
        else
            MessageBox.Show("You have not saved any details to cancel
order!", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }

    private void tqty_TextChanged(object sender, EventArgs e)
    {
        errorProvider1.Clear();
    }

    public string gethtml(DataGridView dgv)
    {
        try
        {
            string mb = "<br><br>Supplier Name : " + supname + "<br><br>PO
Id : " + poid + "<br><br>Date : " + DateTime.Now + "<br><br>";
            if (dgv.Rows.Count == 0)
                return mb;
            string hts = "<table border='1' text-align:center;\\">";
            string hte = "</table>";
            string hhrs = "<tr style=\"background-
color:#6FA1D2;color:#ffffff;\\">";
            string hhre = "</tr>";
            string htrs = "<tr style=\"color:#555555;\\">";
            string htre = "</tr>";
            string htds = "<td style=\"border-color:#5c87b2;borer-
style:solid;border-width:thin;padding:20px;\\">";
            string htde = "</td>";
            mb += hts;
            mb += hhrs;
            mb += htds + "Item ID" + htde;
            mb += htds + "Name" + htde;
            mb += htds + "Quantity" + htde;
            mb += hhre;
        }
    }
}

```

```

        for (int i = 0; i <= dgv.Rows.Count-1; i++)
        {
            mb = mb + htrs;
            mb = mb + htde + dgv.Rows[i].Cells[2].Value + htde;
            mb = mb + htde + dgv.Rows[i].Cells[3].Value + htde;
            mb = mb + htde + dgv.Rows[i].Cells[4].Value + htde;
            mb = mb + htde;
        }
        mb = mb + hte;
        mb = mb + "<center><b><h3>Furniture Shop Management System
</h3><h5> Renovate Your Home</h5><b></center>";
        return mb;
    }
    catch (Exception)
    {
        return null;
    }
}

private void btnplaceorder_Click(object sender, EventArgs e)
{
    c.cmd.CommandText = "select Count(*) from Purchase_Order_Detail
where PO_No=''' + poid + '''";
    int rc1 = Convert.ToInt16(c.cmd.ExecuteScalar());
    c.cmd.CommandText = "select Count(*) from Purchase_Order where
PO_No=''' + poid + '''";
    int rc2 = Convert.ToInt16(c.cmd.ExecuteScalar());
    if (rc1 >= 1 && rc2 == 1)
    {
        if (MessageBox.Show("Are You Sure ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            string hs = gethtml(dgvpurchaseorderd);
            try
            {
                c.cmd.CommandText = "select Email from Supplier where
Sup_Id='''+supid+'''";
                String email =
Convert.ToString(c.cmd.ExecuteScalar());
                MailMessage mail = new MailMessage();
                SmtpClient smtpserver = new
SmtpClient("smtp.gmail.com");
                mail.From = new
MailAddress("bhakthiprabhu15@gmail.com");
                mail.To.Add(email);
                mail.Subject = "Purchase Order";
                mail.Body = hs;
                mail.IsBodyHtml = true;
                smtpserver.Port = 587;
                smtpserver.Credentials = new
NetworkCredential("bhakthiprabhu15@gmail.com", "bhakthi15");
                smtpserver.EnableSsl = true;
                btnplaceorder.Cursor = Cursors.WaitCursor;
                smtpserver.Send(mail);
                c.cmd.CommandText = "update Purchase_Order set
Status='Ordered' where PO_No=''' + poid + '''";
                c.cmd.ExecuteNonQuery();
            }
        }
    }
}

```

```

        MessageBox.Show("Order Placed Successfully!", "", 
MessageBoxButtons.OK, MessageBoxIcon.Information);
        poid = null;
        displayall();
        displayall1();
        clearall();
    }
    catch (Exception)
    {
        MessageBox.Show("No internet connection");
    }
}
else
    MessageBox.Show("You have not saved the details to place
order!", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}

private void dgvpurchaseorderd_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if
(dgvpurchaseorderd.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
    {
        nupdate.Enabled = true;
        ndelete.Enabled = true;
        groupBox2.Enabled = false;
        btnnextpur.Enabled = false;
        dgvpurchaseorderd.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        dgvpurchaseorderd.CurrentRow.Selected = true;
        tqty.Text =
dgvpurchaseorderd.Rows[e.RowIndex].Cells["Qty"].FormattedValue.ToString();
        pitemid =
dgvpurchaseorderd.Rows[e.RowIndex].Cells["Item_Id"].FormattedValue.ToString();
    }
}

private void nupdate_Click(object sender, EventArgs e)
{
    try
    {
        if (tqty.Text == "" || tqty.Text == "0")
            errorProvider1.SetError(tqty, "Required!");
        else
        {
            c.cmd.CommandText = "update Purchase_Order_Detail set
Qty='"+tqty.Text+"' where PO_No='"+poid+"' and Item_Id='"+pitemid+"' ";
            c.cmd.ExecuteNonQuery();
            MessageBox.Show("Item Quantity Updated!", "", 
MessageBoxButtons.OK, MessageBoxIcon.Information);
            displayall1();
            nupdate.Enabled = false;
            ndelete.Enabled = false;
            groupBox2.Enabled = true;
            tqty.Text = "";
            btnnextpur.Enabled = true;
        }
    }
}

```

```

        }
    }
    catch (SqlException)
    {
        errorProvider1.SetError(tqty, "Invalid Quantity!");
    }
}

private void ndelete_Click(object sender, EventArgs e)
{
    c.cmd.CommandText = "delete from Purchase_Order_Detail where
PO_No=''" + poid + "' and Item_Id=''" + pitemid + "'";
    c.cmd.ExecuteNonQuery();
    MessageBox.Show("Item Deleted!", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    displayall1();
    nupdate.Enabled = false;
    ndelete.Enabled = false;
    groupBox2.Enabled = true;
    tqty.Text = "";
    btnnextpur.Enabled = true;
}

private void tqty_KeyPress(object sender, KeyPressEventArgs e)
{
    //accept only numbers
    e.Handled = !char.IsDigit(e.KeyChar) && !
char.IsControl(e.KeyChar);
}

private void searchitem_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || char.IsDigit(e.KeyChar));
}

private void vsupname_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || char.IsDigit(e.KeyChar));
}
}

```

Purchase Return:

```

using System.Windows.Forms;
using System.Data.SqlClient;
using System.Net.Mail;
using System.Net;

namespace WindowsApplication
{
    public partial class Purchase_Order : Form
    {
        Connect c = new Connect();

```

```

        String itemid,supid,poid,pitemid,supname;
        int qty;
        public Purchase_Order()
        {
            InitializeComponent();
        }

        private void Purchase_Load(object sender, EventArgs e)
        {
            //page load
            Supplier s = new Supplier();
            s.datagridviewstyle(dgvinventoryd);
            s.datagridviewstyle(dgvsupplierd);
            s.datagridviewstyle(dgvpurchaseorderd);
            dgvinventoryd.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;
            displayall();
            btnnextpur.Enabled = false;
            dateTimePicker1.Value = DateTime.Today;
            dateTimePicker1.MinDate = DateTime.Today;
        }

        private void displayall()
        {
            dgvsupplierd.SelectionMode = DataGridViewSelectionMode.CellSelect;
            dgvinventoryd.SelectionMode =
DataGridViewSelectionMode.CellSelect;
            SqlDataAdapter sda = new SqlDataAdapter("select
Item_Id,Name,Qty,Material,Color,Category from Inventory ", c.cn);
            DataSet ds = new DataSet();
            sda.Fill(ds, "dt");
            dgvinventoryd.DataSource = ds.Tables["dt"];

            SqlDataAdapter adp = new SqlDataAdapter("select
Sup_Id,Name,Contact_No from Supplier where Status='Active'", c.cn);
            DataSet ds1 = new DataSet();
            adp.Fill(ds1, "dt1");
            dgvsupplierd.DataSource = ds1.Tables["dt1"];
        }

        private void displayall1()
        {
            dgvpurchaseorderd.SelectionMode =
DataGridViewSelectionMode.CellSelect;
            dgvinventoryd.SelectionMode =
DataGridViewSelectionMode.CellSelect;
            SqlDataAdapter sda = new SqlDataAdapter("select
Item_Id,Name,Qty,Material,Color,Category from Inventory ", c.cn);
            DataSet ds = new DataSet();
            sda.Fill(ds, "dt");
            dgvinventoryd.DataSource = ds.Tables["dt"];

            SqlDataAdapter adp1 = new SqlDataAdapter("select
Purchase_Order_Detail.PO_No,Purchase_Order.Sup_Id,Purchase_Order_Detail.Item_I
d,Inventory.Name,Purchase_Order_Detail.Qty from Purchase_Order_Detail inner
join Purchase_Order on Purchase_Order_Detail.PO_No=Purchase_Order.PO_No inner

```

```

join Inventory on Purchase_Order_Detail.Item_Id= Inventory.Item_Id where
Purchase_Order.PO_No=''' + poid + ''' , c.cn);
        DataSet ds2 = new DataSet();
        adp1.Fill(ds2, "dt2");
        dgvpurchaseorderd.DataSource = ds2.Tables["dt2"];
    }

private void clearall()
{
    dateTimePicker1.Text = "";
    tqty.Text = "";
    vsupname.Text = "search...";
    vsupname.ForeColor = Color.Silver;
    searchitem.Text = "search...";
    searchitem.ForeColor = Color.Silver;
    groupBox1.Enabled = true;
    dateTimePicker1.Enabled = true;
    btnnextpur.Enabled = false;
    btnsave.Enabled = true;
    nupdate.Enabled = false;
    ndelete.Enabled = false;
    groupBox2.Enabled = true;
}

private void vsupname_Click(object sender, EventArgs e)
{
    if (vsupname.Text == "search...")
        vsupname.Text = "";
    vsupname.ForeColor = Color.FromArgb(20, 168, 0);
}

private void vsupname_Leave(object sender, EventArgs e)
{
    if (vsupname.Text == "")
        vsupname.Text = "search...";
    vsupname.ForeColor = Color.Silver;
}

private void vsupname_KeyUp(object sender, KeyEventArgs e)
{
    SqlDataAdapter sda = new SqlDataAdapter("select
Sup_Id,Name,Contact_No from Supplier where Status='Active' and Name like '" +
vsupname.Text + "%' or Status='Active' and Sup_Id like '" + vsupname.Text +
"%'", c.cn);
    DataSet ds = new DataSet();
    sda.Fill(ds, "dt");
    dgvsupplierd.DataSource = ds.Tables["dt"];
}

private void searchitem_Click(object sender, EventArgs e)
{
    //search item Click
    if (searchitem.Text == "search...")
        searchitem.Text = "";
    searchitem.ForeColor = Color.FromArgb(20, 168, 0);
}

```

```

private void searchitem_Leave(object sender, EventArgs e)
{
    //search item leave
    if (searchitem.Text == "")
        searchitem.Text = "search...";
    searchitem.ForeColor = Color.Silver;
}

private void searchitem_KeyUp(object sender, KeyEventArgs e)
{
    //search item keyup
    SqlDataAdapter sda = new SqlDataAdapter("select
Item_Id,Name,Qty,Material,Color,Category from Inventory where Name like '" +
searchitem.Text + "%'", c.cn);
    DataSet ds = new DataSet();
    sda.Fill(ds, "dt");
    dgvinventoryd.DataSource = ds.Tables["dt"];
}

private void dgvinventoryd_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if (dgvinventoryd.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
    {
        dgvinventoryd.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        dgvinventoryd.CurrentRow.Selected = true;
        itemid =
dgvinventoryd.Rows[e.RowIndex].Cells["Item_Id"].FormattedValue.ToString();
        qty =
Convert.ToInt32(dgvinventoryd.Rows[e.RowIndex].Cells["Qty"].FormattedValue.ToString());
        tqty.Text = qty.ToString();
    }
}

private void dgsupplierd_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if (dgsupplierd.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
    {
        dgsupplierd.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        dgsupplierd.CurrentRow.Selected = true;
        supid =
dgsupplierd.Rows[e.RowIndex].Cells["Sup_Id"].FormattedValue.ToString();
        supname =
dgsupplierd.Rows[e.RowIndex].Cells["Name"].FormattedValue.ToString();
    }
}

private void btnsave_Click(object sender, EventArgs e)
{
}

```

```

        if (!(dgvsupplierd.SelectionMode ==
DataGridViewSelectionMode.FullRowSelect))
            MessageBox.Show("please select supplier", "", 
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        else
        {
            groupBox1.Enabled = false;
            dateTimePicker1.Enabled = false;
            SqlCommand cmd = new SqlCommand("Select * from Purchase_Order
where Status is Null", c.cn);
            DataTable dt = new DataTable();
            SqlDataAdapter adp = new SqlDataAdapter(cmd);
            adp.Fill(dt);
            foreach (DataRow r in dt.Rows)
            {
                c.cmd.CommandText = "delete from Purchase_Order_Detail
where PO_No=" + r["PO_No"] + "";
                c.cmd.ExecuteNonQuery();
                c.cmd.CommandText="delete from Purchase_Order where
PO_No='"+r["PO_No"]+"'";
                c.cmd.ExecuteNonQuery();
            }
            c.cmd.CommandText = "select Count(*) from Purchase_Order";
            int cnt = Convert.ToInt16(c.cmd.ExecuteScalar());
            cnt++;
            poid = "po" + cnt;
            String date = dateTimePicker1.Value.ToString("MM-dd-yyyy");
            c.cmd.CommandText = "insert into Purchase_Order
(PO_No,Sup_Id,Date) values('" + poid + "','" + supid + "','" + date + ' )';
            c.cmd.ExecuteNonQuery();
            MessageBox.Show("Details Saved!", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            btnnextpur.Enabled = true;
            btnsave.Enabled = false;
        }
    }

private void btnnextpur_Click(object sender, EventArgs e)
{
    try
    {
        SqlCommand cmd = new SqlCommand("select PO_No,Item_Id,Qty from
Purchase_Order_Detail where PO_No='" + poid + "' and Item_Id=' " + itemid +
"'", c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        adp.Fill(dt);

        if (!(dgvinventoryd.SelectionMode ==
DataGridViewSelectionMode.FullRowSelect))
            MessageBox.Show("please select Item", "", 
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        else if (tqty.Text == "" || tqty.Text == "0")
            errorProvider1.SetError(tqty, "Required!");
        else if (dt.Rows.Count > 0)
    }
}

```

```

        MessageBox.Show("You already added this item to place
order.\nPlease edit item if you want to make changes.", "",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    else
    {
        c.cmd.CommandText = "insert into Purchase_Order_Detail
(P0_No,Item_Id,Qty) values('" + poid + "','" + itemid + "','" + tqty.Text +
")";
        c.cmd.ExecuteNonQuery();
        tqty.Text = "";
        displayall1();
    }
}
catch (SqlException)
{
    errorProvider1.SetError(tqty, "Invalid Quantity!");
}
}

private void btncancel_Click(object sender, EventArgs e)
{
    c.cmd.CommandText = "select Count(*) from Purchase_Order_Detail
where P0_No=" + poid + "";
    int rc1 = Convert.ToInt16(c.cmd.ExecuteScalar());
    c.cmd.CommandText = "select Count(*) from Purchase_Order where
P0_No=" + poid + "";
    int rc2 = Convert.ToInt16(c.cmd.ExecuteScalar());
    if (rc1 >= 1 && rc2 == 1)
    {
        if (MessageBox.Show("Are You Sure ?", "",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            c.cmd.CommandText = "delete from Purchase_Order_Detail
where P0_No=" + poid + "";
            c.cmd.ExecuteNonQuery();
            c.cmd.CommandText = "delete from Purchase_Order where
P0_No=" + poid + "";
            c.cmd.ExecuteNonQuery();
            MessageBox.Show("Order Cancelled!", "",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            poid = null;
            clearall();
            displayall();
            displayall1();
        }
    }
    else
        MessageBox.Show("You have not saved any details to cancel
order!", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}

private void tqty_TextChanged(object sender, EventArgs e)
{
    errorProvider1.Clear();
}

public string gethtml(DataGridView dgv)

```

```

{
    try
    {
        string mb = "<br><br>Supplier Name : " + supname + "<br><br>PO
Id : " + poid + "<br><br>Date : " + DateTime.Now + "<br><br>";
        if (dgv.Rows.Count == 0)
            return mb;
        string hts = "<table border='1' text-align:center;\">";
        string hte = "</table>";
        string hhrs = "<tr style=\"background-
color:#6FA1D2;color:#ffffff;\">";
        string hhre = "</tr>";
        string htrs = "<tr style=\"color:#555555;\">";
        string htre = "</tr>";
        string htds = "<td style=\"border-color:#5c87b2;borer-
style:solid;border-width:thin;padding:20px;\">";
        string htde = "</td>";
        mb += hts;
        mb += hhrs;
        mb += htds + "Item ID" + htde;
        mb += htds + "Name" + htde;
        mb += htds + "Quantity" + htde;
        mb += hhre;

        for (int i = 0; i <= dgv.Rows.Count-1; i++)
        {
            mb = mb + htrs;
            mb = mb + htds + dgv.Rows[i].Cells[2].Value + htde;
            mb = mb + htds + dgv.Rows[i].Cells[3].Value + htde;
            mb = mb + htds + dgv.Rows[i].Cells[4].Value + htde;
            mb = mb + htre;
        }
        mb = mb + hte;
        mb = mb + "<center><b><h3>Furniture Shop Management System
</h3><h5> Renovate Your Home</h5><b></center>";
        return mb;
    }
    catch (Exception)
    {
        return null;
    }
}

private void btnplaceorder_Click(object sender, EventArgs e)
{
    c.cmd.CommandText = "select Count(*) from Purchase_Order_Detail
where PO_No=''' + poid + '''";
    int rc1 = Convert.ToInt16(c.cmd.ExecuteScalar());
    c.cmd.CommandText = "select Count(*) from Purchase_Order where
PO_No=''' + poid + '''";
    int rc2 = Convert.ToInt16(c.cmd.ExecuteScalar());
    if (rc1 >= 1 && rc2 == 1)
    {
        if (MessageBox.Show("Are You Sure ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            string hs = gethtml(dgvpurchaseorderd);

```

```

        try
        {
            c.cmd.CommandText = "select Email from Supplier where
Sup_Id='"+supid+"'";
            String email =
Convert.ToString(c.cmd.ExecuteScalar());
            MailMessage mail = new MailMessage();
            SmtpClient smtpserver = new
SmtpClient("smtp.gmail.com");
            mail.From = new
MailAddress("bhakthiprabhu15@gmail.com");
            mail.To.Add(email);
            mail.Subject = "Purchase Order";
            mail.Body = hs;
            mail.IsBodyHtml = true;
            smtpserver.Port = 587;
            smtpserver.Credentials = new
NetworkCredential("bhakthiprabhu15@gmail.com", "bhakthi15");
            smtpserver.EnableSsl = true;
            btnplaceorder.Cursor = Cursors.WaitCursor;
            smtpserver.Send(mail);
            c.cmd.CommandText = "update Purchase_Order set
Status='Ordered' where PO_No='"+ poid + "'";
            c.cmd.ExecuteNonQuery();
            MessageBox.Show("Order Placed Successfully!", "",

MessageBoxButtons.OK, MessageBoxIcon.Information);
            poid = null;
            displayall();
            displayall1();
            clearall();
        }
        catch (Exception)
        {
            MessageBox.Show("No internet connection");
        }
    }
}
else
{
    MessageBox.Show("You have not saved the details to place
order!", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}

private void dgvpurchaseorderd_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if
(dgvpurchaseorderd.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
    {
        nupdate.Enabled = true;
        ndelete.Enabled = true;
        groupBox2.Enabled = false;
        btnnextpur.Enabled = false;
        dgvpurchaseorderd.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        dgvpurchaseorderd.CurrentRow.Selected = true;
        tqty.Text =
dgvpurchaseorderd.Rows[e.RowIndex].Cells["Qty"].FormattedValue.ToString();
    }
}

```

```

        pitemid =
dgvpurchaseorderd.Rows[e.RowIndex].Cells["Item_Id"].FormattedValue.ToString();
    }
}

private void nupdate_Click(object sender, EventArgs e)
{
    try
    {
        if (tqty.Text == "" || tqty.Text == "0")
            errorProvider1.SetError(tqty, "Required!");
        else
        {
            c.cmd.CommandText = "update Purchase_Order_Detail set
Qty='"+tqty.Text+"' where PO_No='"+poid+"' and Item_Id='"+pitemid+"' ";
            c.cmd.ExecuteNonQuery();
            MessageBox.Show("Item Quantity Updated!", "",

MessageBoxButtons.OK, MessageBoxIcon.Information);
            displayall1();
            nupdate.Enabled = false;
            ndelete.Enabled = false;
            groupBox2.Enabled = true;
            tqty.Text = "";
            btnnextpur.Enabled = true;
        }
    }
    catch (SqlException)
    {
        errorProvider1.SetError(tqty, "Invalid Quantity!");
    }
}

private void ndelete_Click(object sender, EventArgs e)
{
    c.cmd.CommandText = "delete from Purchase_Order_Detail where
PO_No=''" + poid + "' and Item_Id='" + pitemid + "'";
    c.cmd.ExecuteNonQuery();
    MessageBox.Show("Item Deleted!", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    displayall1();
    nupdate.Enabled = false;
    ndelete.Enabled = false;
    groupBox2.Enabled = true;
    tqty.Text = "";
    btnnextpur.Enabled = true;
}

private void tqty_KeyPress(object sender, KeyPressEventArgs e)
{
    //accept only numbers
    e.Handled = !char.IsDigit(e.KeyChar) && !
char.IsControl(e.KeyChar);
}

private void searchitem_KeyPress(object sender, KeyPressEventArgs e)
{

```

```

        e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || char.IsDigit(e.KeyChar));
    }

    private void vsupname_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || char.IsDigit(e.KeyChar));
    }
}

```

Sales Bill:

```

using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsApplication
{
    public partial class Sale_Bill : Form
    {
        Connect c = new Connect();
        String itemid, cno, sbid, pitemid, username;
        int qty;
        double totalamt, netamt, discount;
        public Sale_Bill(string str)
        {
            username = str;
            InitializeComponent();
        }

        private void Sale_Bill_Load(object sender, EventArgs e)
        {
            Supplier s = new Supplier();
            s.datagridviewstyle(dgvsalebilld);
            s.datagridviewstyle(dgvcustomerd);
            s.datagridviewstyle(dgvinventoryd);
            dgvinventoryd.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;
            dgvsalebilld.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;
            btnnext.Enabled = false;
            dateTimePicker1.Value = DateTime.Today;
            dateTimePicker1.MinDate = DateTime.Today;
            dateTimePicker1.MaxDate = DateTime.Today.AddMonths(1);
            displayall1();
            display();
        }

        private void display()
        {
            dgvcustomerd.SelectionMode = DataGridViewSelectionMode.CellSelect;
            SqlDataAdapter adp = new SqlDataAdapter("select
Name, Contact_No, Address from Customer", c.cn);
            DataSet ds1 = new DataSet();
            adp.Fill(ds1, "dt1");

```

```

        dgvcustomerd.DataSource = ds1.Tables["dt1"];
    }

    private void displayall1()
    {
        dgvinventoryd.SelectionMode =
DataGridViewSelectionMode.CellSelect;
        SqlDataAdapter sda = new SqlDataAdapter("select
Item_Id,Name,Qty,Price,Material,Color,Category,Warranty from Inventory where
Status='Active'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgvinventoryd.DataSource = ds.Tables["dt"];
    }

    private void displayall()
    {
        dgvsalebilld.SelectionMode = DataGridViewSelectionMode.CellSelect;
        dgvinventoryd.SelectionMode =
DataGridViewSelectionMode.CellSelect;
        SqlDataAdapter sda = new SqlDataAdapter("select
Item_Id,Name,Qty,Price,Material,Color,Category,Warranty from Inventory where
Status='Active'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgvinventoryd.DataSource = ds.Tables["dt"];

        SqlDataAdapter adp1 = new SqlDataAdapter("select
Sale_Bill_Detail.SB_No,Sale_Bill_Detail.Item_Id,Sale_Bill_Detail.Qty,Sale_Bill
_Detail.Rate,Sale_Bill_Detail.Price,Sale_Bill_Detail.Tax,Sale_Bill_Detail.Amou
nt,Inventory.Warranty from Sale_Bill_Detail inner join Inventory on
Sale_Bill_Detail.Item_Id=Inventory.Item_Id where SB_No='" + sbid + "' ", 
c.cn);
        DataSet ds2 = new DataSet();
        adp1.Fill(ds2, "dt2");
        dgvsalebilld.DataSource = ds2.Tables["dt2"];
    }

    private void clearall()
    {
        dateTimePicker1.Text = "";
        tqty.Text = "";
        tprice.Text = "";
        twarranty.Text = "";
        dateTimePicker1.Enabled = true;
        btnnext.Enabled = false;
        btnsave.Enabled = true;
        nupdate.Enabled = false;
        ndelete.Enabled = false;
        ttamt.Text = "";
        tnetamt.Text = "";
        tdiscount.Text = "";
        searchcno.Text = "search...";
        searchcno.ForeColor = Color.Silver;
        searchitem.Text = "search...";
        searchitem.ForeColor = Color.Silver;
    }

```

```

}

private void searchitem_Click(object sender, EventArgs e)
{
    //search item Click
    if (searchitem.Text == "search...")
        searchitem.Text = "";
    searchitem.ForeColor = Color.FromArgb(20, 168, 0);
}

private void searchitem_Leave(object sender, EventArgs e)
{
    //search item leave
    if (searchitem.Text == "")
        searchitem.Text = "search...";
    searchitem.ForeColor = Color.Silver;
}

private void searchitem_KeyUp(object sender, KeyEventArgs e)
{
    //search item keyup
    SqlDataAdapter sda = new SqlDataAdapter("select
Item_Id,Name,Qty,Price,Material,Color,Category,Warranty from Inventory where
Status='Active' and Name like '" + searchitem.Text + "%'", c.cn);
    DataSet ds = new DataSet();
    sda.Fill(ds, "dt");
    dgvinventoryd.DataSource = ds.Tables["dt"];
}

private void searchcno_Click(object sender, EventArgs e)
{
    //search contact no click
    if (searchcno.Text == "search...")
        searchcno.Text = "";
    searchcno.ForeColor = Color.FromArgb(20, 168, 0);
}

private void searchcno_Leave(object sender, EventArgs e)
{
    //search contact no leave
    if (searchcno.Text == "")
        searchcno.Text = "search...";
    searchcno.ForeColor = Color.Silver;
}

private void searchcno_KeyUp(object sender, KeyEventArgs e)
{
    //search contact no keyup
    SqlDataAdapter sda = new SqlDataAdapter("select * from Customer
where Contact_No like '" + searchcno.Text + "%'", c.cn);
    DataSet ds = new DataSet();
    sda.Fill(ds, "dt");
    dgvcustomerd.DataSource = ds.Tables["dt"];
}

private void searchcno_KeyPress(object sender, KeyPressEventArgs e)
{
}

```

```

        //accept only numbers
        e.Handled = !char.IsDigit(e.KeyChar) && !
char.IsControl(e.KeyChar);
    }

    private void dgvinventoryd_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if (dgvinventoryd.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
    {
        dgvinventoryd.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        dgvinventoryd.CurrentRow.Selected = true;
        itemid =
dgvinventoryd.Rows[e.RowIndex].Cells["Item_Id"].FormattedValue.ToString();
        qty =
Convert.ToInt32(dgvinventoryd.Rows[e.RowIndex].Cells["Qty"].FormattedValue.ToString());
        twarranty.Text =
dgvinventoryd.Rows[e.RowIndex].Cells["Warranty"].FormattedValue.ToString();
    }
}

private void dgvcustomerd_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if (dgvcustomerd.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
    {
        dgvcustomerd.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        dgvcustomerd.CurrentRow.Selected = true;
        cno =
dgvcustomerd.Rows[e.RowIndex].Cells["Contact_No"].FormattedValue.ToString();
    }
}

private void totalamts()
{
    c.cmd.CommandText = "Select Amount from Sale_Bill_Detail where
SB_No=''' + sbid + '''";
    if (!(c.cmd.ExecuteScalar() == null))
    {
        c.cmd.CommandText = "Select sum(Amount) from Sale_Bill_Detail
where SB_No=''' + sbid + '''";
        totalamt = Convert.ToDouble(c.cmd.ExecuteScalar());
        ttamt.Text = Math.Round(totalamt,2).ToString();
        if (totalamt >= 10000.0)
            discount = totalamt * 0.15;
        else if (totalamt >= 5000.0 )
            discount = totalamt * 0.1;
        else if (totalamt >= 1000.0 && totalamt<5000.0)
            discount = totalamt * 0.05;
        else
            discount = totalamt * 0.0;
        tdiscount.Text =Math.Round(discount,2).ToString();
    }
}

```

```

        netamt = totalamt - discount;
        tnetamt.Text = Math.Round(netamt, 2).ToString();
    }
    else
    {
        ttamt.Text = "";
        tnetamt.Text = "";
        tdiscount.Text = "";
    }
}

private void btnsave_Click(object sender, EventArgs e)
{
    if (!(dgvcustomerd.SelectionMode ==
DataGridViewSelectionMode.FullRowSelect))
        MessageBox.Show("please select Customer", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    else
    {
        groupBox1.Enabled = false;
        dateTimePicker1.Enabled = false;
        SqlCommand cmd = new SqlCommand("Select * from Sale_Bill where
Total_Amount is Null", c.cn);
        DataTable dt = new DataTable();
        SqlDataAdapter adp = new SqlDataAdapter(cmd);
        adp.Fill(dt);
        foreach (DataRow r in dt.Rows)
        {
            c.cmd.CommandText = "delete from Sale_Bill_Detail where
SB_No=" + r["SB_No"] + "'";
            c.cmd.ExecuteNonQuery();
            c.cmd.CommandText = "delete from Sale_Bill where SB_No=" +
+ r["SB_No"] + "'";
            c.cmd.ExecuteNonQuery();
        }
        c.cmd.CommandText = "select Count(*) from Sale_Bill";
        int cnt = Convert.ToInt16(c.cmd.ExecuteScalar());
        cnt++;
        sbid = "sb" + cnt;
        String date = dateTimePicker1.Value.ToString("MM-dd-yyyy");
        c.cmd.CommandText = "insert into
Sale_Bill(SB_No,Date,Contact_No) values('" + sbid + "','" + date + "','" + cno
+ "')";
        c.cmd.ExecuteNonQuery();
        MessageBox.Show("Details Saved!", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        btnnext.Enabled = true;
        btnsave.Enabled = false;
    }
}

private void tprice_KeyPress(object sender, KeyPressEventArgs e)
{
    //accept only decimal numbers
    e.Handled = !char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)
&& e.KeyChar != '.';
}

```

```

private void twarranty_KeyPress(object sender, KeyPressEventArgs e)
{
    //accept only decimal numbers
    e.Handled = !char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)
&& e.KeyChar != '.';
}

private void tqty_KeyPress(object sender, KeyPressEventArgs e)
{
    //accept only numbers
    e.Handled = !char.IsDigit(e.KeyChar) && !
char.IsControl(e.KeyChar);
}

private void btnnext_Click(object sender, EventArgs e)
{
    SqlCommand cmd = new SqlCommand("select
SB_No,Item_Id,Qty,Rate,Price,Tax,Amount from Sale_Bill_Detail where SB_No='"
+ sbid + "' and Item_Id='"
+ itemid + "'", c.cn);
    SqlDataAdapter adp = new SqlDataAdapter(cmd);
    DataTable dt = new DataTable();
    adp.Fill(dt);

    try
    {
        if (!(dgvinventoryd.SelectionMode ==
DataGridViewSelectionMode.FullRowSelect))
            MessageBox.Show("please select Item", "", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        else if (tqty.Text == "" || tqty.Text == "0")
            errorProvider1.SetError(tqty, "Required!");
        else if (Convert.ToInt64(tqty.Text) > qty)
            errorProvider1.SetError(tqty, "Quantity should be equal or
less than specified!");
        else if (tprice.Text == "" || tprice.Text == "0.0" ||
tprice.Text == "0")
            errorProvider2.SetError(tprice, "Required!");
        else if (dt.Rows.Count > 0)
            MessageBox.Show("You already added this item.\nPlease edit
item if you want to make changes.", "", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        else
        {
            double price = Convert.ToDouble(tqty.Text) *
Convert.ToDouble(tprice.Text);
            double tax = price * 0.18;
            double amt = price + tax;
            c.cmd.CommandText = "insert into Sale_Bill_Detail
(SB_No,Item_Id,Qty,Rate,Price,Tax,Amount) values('"
+ sbid + "','" + itemid +
"', '" + tqty.Text + "," + tprice.Text + "," + price + "," + tax +
"," + amt + ")";
            c.cmd.ExecuteNonQuery();
            tqty.Text = "";
            tprice.Text = "";
            twarranty.Text = "";
            displayall1();
        }
    }
}

```

```

        displayall();
        totalamts();
    }
}
catch (OverflowException)
{
    errorProvider1.SetError(tqty, "Invalid Quantity!");
}
catch (SqlException)
{
    errorProvider2.SetError(tprice, "Invalid Price!");
}
}

private void tqty_TextChanged(object sender, EventArgs e)
{
    errorProvider1.Clear();
}

private void tprice_TextChanged(object sender, EventArgs e)
{
    errorProvider2.Clear();
}

private void btngeneratebil_Click(object sender, EventArgs e)
{
    c.cmd.CommandText = "select Count(*) from Sale_Bill_Detail where
SB_No=''' + sbid + '''";
    int rc = Convert.ToInt16(c.cmd.ExecuteScalar());
    c.cmd.CommandText = "select Count(*) from Sale_Bill where SB_No=''
+ sbid + '''";
    int rc2 = Convert.ToInt16(c.cmd.ExecuteScalar());
    if (rc >= 1 && rc2 == 1)
    {
        if (MessageBox.Show("Are You Sure ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            c.cmd.CommandText = "update Sale_Bill set Total_Amount=" +
ttamt.Text + ",Discount='"+tdiscount.Text+"',Net_Amount='"+tnetamt.Text+"'
where SB_No=''' + sbid + '''";
            c.cmd.ExecuteNonQuery();

            SqlCommand cmd = new SqlCommand("select * from
Sale_Bill_Detail where SB_No=''' + sbid + '''", c.cn);
            DataTable dt = new DataTable();
            SqlDataAdapter adp = new SqlDataAdapter(cmd);
            adp.Fill(dt);
            foreach (DataRow r in dt.Rows)
            {
                c.cmd.CommandText = "Select Qty from Inventory where
Item_Id=''' + r["Item_Id"] + '''";
                int qt = Convert.ToInt32(c.cmd.ExecuteScalar());
                qt = qt - Convert.ToInt32(r["Qty"]);
                if (qt <= 0)
                {
                    c.cmd.CommandText = "update Inventory set Status
='Inactive',Qty=0 where Item_Id=''' + r["Item_Id"] + '''";
                }
            }
        }
    }
}

```

```

                c.cmd.ExecuteNonQuery();
            }
            else
            {
                c.cmd.CommandText = "update Inventory set Qty =" +
qt + " where Item_Id=''' + r["Item_Id"] + '''";
                c.cmd.ExecuteNonQuery();
            }
        }

        MessageBox.Show("Bill generated Successfully!", "",

MessageBoxButtons.OK, MessageBoxIcon.Information);
        new Sale_Bill_Print(sbid,username).Show();
        clearall();
        sbid = null;
        displayall();
        displayall1();
        display();
        ttamt.Text = "";
        tprice.Text = "";
        twarranty.Text = "";
        groupBox1.Enabled = true;
    }
}
else
    MessageBox.Show("You have not stored the details!", "",

MessageBoxButtons.OK, MessageBoxIcon.Warning);
}

private void btncancel_Click(object sender, EventArgs e)
{
    c.cmd.CommandText = "select Count(*) from Sale_Bill_Detail where
SB_No=''' + sbid + '''";
    int rc = Convert.ToInt16(c.cmd.ExecuteScalar());
    c.cmd.CommandText = "select Count(*) from Sale_Bill where SB_No='"
+ sbid + '''";
    int rc2 = Convert.ToInt16(c.cmd.ExecuteScalar());
    if (rc >= 1 && rc2 == 1)
    {
        if (MessageBox.Show("Are You Sure?", "",

MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            c.cmd.CommandText = "delete from Sale_Bill_Detail where
SB_No=''' + sbid + '''";
            c.cmd.ExecuteNonQuery();
            c.cmd.CommandText = "delete from Sale_Bill where SB_No='"
+ sbid + '''";
            c.cmd.ExecuteNonQuery();
            MessageBox.Show("Bill Generation Cancelled!", "",

MessageBoxButtons.OK, MessageBoxIcon.Information);
            clearall();
            sbid = null;
            displayall();
            displayall1();
            display();
            ttamt.Text = "";
            tprice.Text = ";

```

```

        twarranty.Text = "";
        groupBox1.Enabled = true;
    }
}
else
    MessageBox.Show("You have not stored the details!", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}

private void dgvsalebilld_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if (dgvsalebilld.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
    {
        nupdate.Enabled = true;
        ndelete.Enabled = true;
        btnnext.Enabled = false;
        dgvsalebilld.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        dgvsalebilld.CurrentRow.Selected = true;
        tqty.Text =
dgvsalebilld.Rows[e.RowIndex].Cells["Qty"].FormattedValue.ToString();
        tprice.Text =
dgvsalebilld.Rows[e.RowIndex].Cells["Rate"].FormattedValue.ToString();
        twarranty.Text =
dgvsalebilld.Rows[e.RowIndex].Cells["Warranty"].FormattedValue.ToString();
        pitemid =
dgvsalebilld.Rows[e.RowIndex].Cells["Item_Id"].FormattedValue.ToString();
    }
}

private void nupdate_Click(object sender, EventArgs e)
{
    try
    {
        if (tqty.Text == "" || tqty.Text == "0")
            errorProvider1.SetError(tqty, "Required!");
        else if (Convert.ToInt16(tqty.Text) > qty)
            errorProvider1.SetError(tqty, "Quantity should be equal or
less than specified!");
        else if (tprice.Text == "" || tprice.Text == "0")
            errorProvider2.SetError(tprice, "Required!");
        else
        {
            double price = Convert.ToDouble(tqty.Text) *
Convert.ToDouble(tprice.Text);
            double tax = price * 0.18;
            double amt = price + tax;
            c.cmd.CommandText = "update Sale_Bill_Detail set Qty=" +
tqty.Text + ",Rate=" + tprice.Text + ",Price=" + price + ",Tax=" + tax +
",Amount=" + amt + " where SB_No='" + sbid + "' and Item_Id='" + pitemid + "' ";
            c.cmd.ExecuteNonQuery();
            MessageBox.Show("Item Updated!", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            tqty.Text = "";
    }
}

```

```

        tprice.Text = "";
        twarranty.Text = "";
        displayall();
        displayall1();
        totalamts();
        nupdate.Enabled = false;
        ndelete.Enabled = false;
        btnnext.Enabled = true;
    }
}
catch (OverflowException)
{
    errorProvider1.SetError(tqty, "Invalid Quantity!");
}
catch (SqlException)
{
    errorProvider2.SetError(tprice, "Invalid Price!");
}
}

private void ndelete_Click(object sender, EventArgs e)
{
    c.cmd.CommandText = "delete from Sale_Bill_Detail where SB_No='"
    sbid+ "' and Item_Id='"
    + pitemid + "'";
    c.cmd.ExecuteNonQuery();
    MessageBox.Show("Item Deleted!", "", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
    totalamts();
    tqty.Text = "";
    tprice.Text = "";
    twarranty.Text = "";
    displayall1();
    displayall();
    totalamts();
    nupdate.Enabled = false;
    ndelete.Enabled = false;
    btnnext.Enabled = true;
}

private void searchitem_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
    (char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || char.IsDigit(e.KeyChar));
}

}
}

```

Sales Bill Print:

```

using System.Windows.Forms;
using System.Data.SqlClient;
using System.Drawing.Printing;

namespace WindowsApplication
{

```

```

public partial class Sale_Bill_Print : Form
{
    string sbno,username;
    Connect c = new Connect();
    public Sale_Bill_Print(String sbid,string str)
    {
        InitializeComponent();
        username = str;
        sbno = sbid;
    }

    private void Sale_Bill_Print_Load(object sender, EventArgs e)
    {
        label27.Text = username;
        c.cmd.CommandText = "select Customer.Name from Customer inner join
Sale_Bill on Customer.Contact_No = Sale_Bill.Contact_No where
Sale_Bill.SB_No=''' + sbno + '''";
        label9.Text = c.cmd.ExecuteScalar().ToString();
        label11.Text = sbno.ToString();
        c.cmd.CommandText = "select Contact_No from Sale_Bill where
Sale_Bill.SB_No=''' + sbno + '''";
        label10.Text = c.cmd.ExecuteScalar().ToString();
        label12.Text = DateTime.Now.ToString();
        c.cmd.CommandText = "select Total_Amount from Sale_Bill where
Sale_Bill.SB_No=''' + sbno + '''";
        label15.Text = c.cmd.ExecuteScalar().ToString();
        c.cmd.CommandText = "select Discount from Sale_Bill where
Sale_Bill.SB_No=''' + sbno + '''";
        label16.Text = c.cmd.ExecuteScalar().ToString();
        c.cmd.CommandText = "select Net_Amount from Sale_Bill where
Sale_Bill.SB_No=''' + sbno + '''";
        label18.Text = (c.cmd.ExecuteScalar()).ToString();
        string amt = label18.Text;
        label27.Text = username;
        String word = ConvertNumbertoWords(Convert.ToInt32(amt.Split('.')[0])) + ' ' + "RUPEES" + ' ' +
ConvertNumbertoWords(Convert.ToInt32(amt.Split('.')[1])) + ' ' + "PAISE ONLY";
        label26.Text = word;
        dataGridView1.EnableHeadersVisualStyles = false;
        dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill;
        dataGridView1.RowTemplate.Height = 50;
        dataGridView1.ColumnHeadersHeight = 50;
        dataGridView1.BackgroundColor = Color.White;
        dataGridView1.MultiSelect = false;
        dataGridView1.AllowUserToAddRows = false;
        dataGridView1.AllowUserToDeleteRows = false;
        dataGridView1.AllowUserToResizeColumns = false;
        dataGridView1.AllowUserToResizeRows = false;
        dataGridView1.ReadOnly = true;
        dataGridView1.BorderStyle = BorderStyle.None;
        dataGridView1.ShowEditingIcon = false;
        dataGridView1.DefaultCellStyle.WrapMode =
DataGridViewTriState.True;
        dataGridView1.DefaultCellStyle.ForeColor = Color.Black;
        dataGridView1.DefaultCellStyle.BackColor = Color.White;
        dataGridView1.DefaultCellStyle.SelectionBackColor = Color.White;
    }
}

```

```

        dataGridView1.DefaultCellStyle.SelectionForeColor = Color.Black;
        dataGridView1.DefaultCellStyle.Font = new Font("Arial", 10,
FontStyle.Regular);
        dataGridView1.DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
        dataGridView1.CellBorderStyle =
DataGridViewCellBorderStyle.Single;
        dataGridView1.ShowCellToolTips = false;
        dataGridView1.ShowCellErrors = false;
        dataGridView1.ShowEditingIcon = false;
        dataGridView1.ShowRowErrors = false;
        dataGridView1.RowHeadersVisible = false;
        dataGridView1.RowHeadersWidthSizeMode =
DataGridViewRowHeadersWidthSizeMode.DisableResizing;
        dataGridView1.RowHeadersBorderStyle =
DataGridViewHeaderBorderStyle.Single;
        dataGridView1.RowHeadersDefaultCellStyle.BackColor = Color.White;
        dataGridView1.RowHeadersDefaultCellStyle.Font = new Font("Arial",
10.25f, FontStyle.Bold);
        dataGridView1.RowHeadersDefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
        dataGridView1.AlternatingRowsDefaultCellStyle.BackColor =
Color.White;
        dataGridView1.ColumnHeadersDefaultCellStyle.WrapMode =
DataGridViewTriState.True;
        dataGridView1.ColumnHeadersDefaultCellStyle.Font = new
Font("Arial", 10.25f, FontStyle.Bold);
        dataGridView1.ColumnHeadersDefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
        dataGridView1.ColumnHeadersDefaultCellStyle.BackColor =
Color.White;
        dataGridView1.ColumnHeadersDefaultCellStyle.ForeColor =
Color.Black;
        dataGridView1.ColumnHeadersBorderStyle =
DataGridViewHeaderBorderStyle.Single;
        dataGridView1.ColumnHeadersHeightSizeMode =
DataGridViewColumnHeadersHeightSizeMode.DisableResizing;

        SqlDataAdapter adp1 = new SqlDataAdapter("select
Inventory.Name ,Inventory.Warranty ,Sale_Bill_Detail.Qty,Sale_Bill_Detail.Rate
,Sale_Bill_Detail.Price,Sale_Bill_Detail.Tax,Sale_Bill_Detail.Amount from
Sale_Bill_Detail inner join Inventory on
Sale_Bill_Detail.Item_Id=Inventory.Item_Id where SB_No=' " + sbno + " ' ,
c.cn);
        DataSet ds2 = new DataSet();
        adp1.Fill(ds2, "dt2");
        dataGridView1.DataSource = ds2.Tables["dt2"];
        dataGridView1.Columns["Name"].HeaderText = "Detail of Product";
        dataGridView1.Columns["Qty"].HeaderText = "Quantity";
        dataGridView1.Columns["Warranty"].HeaderText = "Warranty in
Years";
    }

    Bitmap img;
    private void capturescreen()
    {
        Panel p = new Panel();

```

```

        this.Controls.Add(p);
        Graphics g = p.CreateGraphics();
        Size s = this.ClientSize;
        img = new Bitmap(s.Width, s.Height, g);
        Graphics mg = Graphics.FromImage(img);
        Point pt = PointToScreen(p.Location);
        mg.CopyFromScreen(pt.X, pt.Y, 0, 0, s);
    }

    private void button1_Click(object sender, EventArgs e)
    {
        //print
        try
        {
            button1.Visible = false;
            capturescreen();
            printDocument1.Print();
        }
        catch (Exception)
        {
            MessageBox.Show("bill cannot be printed");
        }
    }

    private void printDocument1_PrintPage(object sender,
System.Drawing.Printing.PrintPageEventArgs e)
    {
        e.Graphics.DrawImage(img, 50, 100);
    }

    public static string ConvertNumbertowords(int number)
    {
        //amount in words
        if (number == 0)
            return "ZERO";
        if (number < 0)
            return "minus " + ConvertNumbertowords(Math.Abs(number));
        string words = "";
        if ((number / 1000000000) > 0)
        {
            words += ConvertNumbertowords(number / 1000000000) + " Billion ";
            number %= 1000000000;
        }
        if ((number / 10000000) > 0)
        {
            words += ConvertNumbertowords(number / 10000000) + " CRORE ";
            number %= 10000000;
        }
        if ((number / 1000000) > 0)
        {
            words += ConvertNumbertowords(number / 1000000) + " MILLION ";
            number %= 1000000;
        }
        if ((number / 100000) > 0)
        {
            words += ConvertNumbertowords(number / 100000) + " LAKH ";
        }
    }
}

```

```

        number %= 1000000;
    }
    if ((number / 1000) > 0)
    {
        words += ConvertNumberToWords(number / 1000) + " THOUSAND ";
        number %= 1000;
    }
    if ((number / 100) > 0)
    {
        words += ConvertNumberToWords(number / 100) + " HUNDRED ";
        number %= 100;
    }
    if (number > 0)
    {
        if (words != "")
            words += "AND ";
        string[] unitsMap = new string[20] { "ZERO", "ONE", "TWO",
"THREE", "FOUR", "FIVE", "SIX", "SEVEN", "EIGHT", "NINE", "TEN", "ELEVEN",
"TWELVE", "THIRTEEN", "FOURTEEN", "FIFTEEN", "SIXTEEN", "SEVENTEEN",
"EIGHTEEN", "NINETEEN" };
        string[] tensMap = new string[10] { "ZERO", "TEN", "TWENTY",
"THIRTY", "FORTY", "FIFTY", "SIXTY", "SEVENTY", "EIGHTY", "NINETY" };
        if (number < 20)
            words += unitsMap[number];
        else
        {
            words += tensMap[number / 10];
            if ((number % 10) > 0)
                words += " " + unitsMap[number % 10];
        }
    }
    return words;
}
}

```

Sales Order:

```
using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsApplication
{
    public partial class Sale_Order : Form
    {
        Connect c = new Connect();
        string cno, itemid, soid;
        public Sale_Order()
        {
            InitializeComponent();
        }

        private void Sale_Order_Load(object sender, EventArgs e)
        {
            Supplier s = new Supplier();
            s.datagridviewstyle(dgvinventoryd);
        }
    }
}
```

```

        s.datagridviewstyle(dgvcustomerd);
        s.datagridviewstyle(dgvsaleorder);
        dgvinventoryd.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;
        displayall();
    }

    private void displayall()
    {
        dgvcustomerd.SelectionMode = DataGridViewSelectionMode.CellSelect;
        dgvsaleorder.SelectionMode = DataGridViewSelectionMode.CellSelect;
        dgvinventoryd.SelectionMode =
DataGridViewSelectionMode.CellSelect;
        SqlDataAdapter sda = new SqlDataAdapter("select
Item_Id,Name,Qty,Material,Color,Category from Inventory where
Status='Inactive'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgvinventoryd.DataSource = ds.Tables["dt"];

        SqlDataAdapter adp = new SqlDataAdapter("select
Name,Contact_No,Address from Customer", c.cn);
        DataSet ds1 = new DataSet();
        adp.Fill(ds1, "dt1");
        dgvcustomerd.DataSource = ds1.Tables["dt1"];

        SqlDataAdapter adp2 = new SqlDataAdapter("select
Sale_Order.SO_No,Sale_Order.Date,Sale_Order.Contact_No,Sale_Order.Item_Id,Inve
ntory.Name from Sale_Order inner join Inventory on
Sale_Order.Item_Id=Inventory.Item_Id", c.cn);
        DataSet ds2 = new DataSet();
        adp2.Fill(ds2, "dt2");
        dgvsaleorder.DataSource = ds2.Tables["dt2"];
    }

    private void clearall()
    {
        searchcno.Text = "search...";
        searchcno.ForeColor = Color.Silver;
        searchitem.Text = "search...";
        searchitem.ForeColor = Color.Silver;
        tsearch.Text = "search...";
        tsearch.ForeColor = Color.Silver;
    }

    private void searchitem_Click(object sender, EventArgs e)
    {
        //search item Click
        if (searchitem.Text == "search...")
            searchitem.Text = "";
        searchitem.ForeColor = Color.FromArgb(20, 168, 0);
    }

    private void searchitem_Leave(object sender, EventArgs e)
    {
        //search item leave
        if (searchitem.Text == "")

```

```

        searchitem.Text = "search...";
        searchitem.ForeColor = Color.Silver;
    }

    private void searchitem_KeyUp(object sender, KeyEventArgs e)
    {
        //search item keyup
        SqlDataAdapter sda = new SqlDataAdapter("select
Item_Id,Name,Qty,Material,Color,Category from Inventory where
Status='Inactive' and Name like '" + searchitem.Text + "%'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgvinventoryd.DataSource = ds.Tables["dt"];
    }

    private void searchcno_Click(object sender, EventArgs e)
    {
        //search contact no click
        if (searchcno.Text == "search...")
            searchcno.Text = "";
        searchcno.ForeColor = Color.FromArgb(20, 168, 0);
    }

    private void searchcno_Leave(object sender, EventArgs e)
    {
        //search contact no leave
        if (searchcno.Text == "")
            searchcno.Text = "search...";
        searchcno.ForeColor = Color.Silver;
    }

    private void searchcno_KeyUp(object sender, KeyEventArgs e)
    {
        //search contact no keyup
        SqlDataAdapter sda = new SqlDataAdapter("select * from Customer
where Contact_No like '" + searchcno.Text + "%'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgvcustomerd.DataSource = ds.Tables["dt"];
    }

    private void tsearch_KeyUp(object sender, KeyEventArgs e)
    {
        SqlDataAdapter sda = new SqlDataAdapter("select
Sale_Order.SO_No,Sale_Order.Date,Sale_Order.Contact_No,Sale_Order.Item_Id,Inve
ntory.Name from Sale_Order inner join Inventory on
Sale_Order.Item_Id=Inventory.Item_Id where Sale_Order.Contact_No like '" +
tsearch.Text + "' or Inventory.Name like '" + tsearch.Text + "%'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgsaleorder.DataSource = ds.Tables["dt"];
    }

    private void tsearch_Leave(object sender, EventArgs e)
    {
        if (tsearch.Text == "")
            tsearch.Text = "search...";
    }

```

```

        tsearch.ForeColor = Color.Silver;
    }

    private void tsearch_Click(object sender, EventArgs e)
    {
        if (tsearch.Text == "search...")
            tsearch.Text = "";
        tsearch.ForeColor = Color.FromArgb(20, 168, 0);
    }

    private void searchcno_KeyPress(object sender, KeyPressEventArgs e)
    {
        //accept only numbers
        e.Handled = !char.IsDigit(e.KeyChar) && !
char.IsControl(e.KeyChar);
    }

    private void dgvinventoryd_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {
        if (dgvinventoryd.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
        {
            dgvinventoryd.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
            dgvinventoryd.CurrentRow.Selected = true;
            itemid =
dgvinventoryd.Rows[e.RowIndex].Cells["Item_Id"].FormattedValue.ToString();
        }
    }

    private void dgvcustomerd_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {
        if (dgvcustomerd.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
        {
            dgvcustomerd.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
            dgvcustomerd.CurrentRow.Selected = true;
            cno =
dgvcustomerd.Rows[e.RowIndex].Cells["Contact_No"].FormattedValue.ToString();
        }
    }

    private void btnplaceorder_Click(object sender, EventArgs e)
    {
        if (!(dgvinventoryd.SelectionMode ==
DataGridViewSelectionMode.FullRowSelect))
            MessageBox.Show("please select Item", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        else if (!(dgvcustomerd.SelectionMode ==
DataGridViewSelectionMode.FullRowSelect))
            MessageBox.Show("please select Customer", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        else
    }

```

```

        if (MessageBox.Show("Are You Sure ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        c.cmd.CommandText = "select Count(*) from Sale_Order";
        int cnt = Convert.ToInt16(c.cmd.ExecuteScalar());
        cnt++;
        soid = "so" + cnt;
        String date = DateTime.Now.ToString("MM-dd-yyyy");
        c.cmd.CommandText = "insert into Sale_Order
(SO_No, Date, Item_Id, Contact_No) values('" + soid + "','" + date + "','" +
itemid + "','" + cno + "')";
        c.cmd.ExecuteNonQuery();
        MessageBox.Show("Order Placed Successfully!", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
        displayall();
        clearall();
    }
}

private void btncancel_Click(object sender, EventArgs e)
{
    displayall();
    clearall();
}

private void searchitem_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || char.IsDigit(e.KeyChar));
}

private void tsearch_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || char.IsDigit(e.KeyChar));
}
}

```

Sans Reset:

```

using System.Net.Mail;
using System.Net;

namespace WindowsApplication
{
    public partial class Sans_Reset : Form
    {
        public Sans_Reset()
        {
            InitializeComponent();
        }

        private void Sans_Reset_Load(object sender, EventArgs e)
        {

```

```

        this.Owner.Enabled = false;
        this.Owner.Hide();
        this.WindowState = FormWindowState.Normal;
        Connect c = new Connect();
        c.cmd.CommandText = "select email from Admin_Create";
        String email = Convert.ToString(c.cmd.ExecuteScalar());
        c.cmd.CommandText = "select sans from Admin_Create";
        String sans = Convert.ToString(c.cmd.ExecuteScalar());
        c.cmd.CommandText = "select sqns from Admin_Create";
        String sqns = Convert.ToString(c.cmd.ExecuteScalar());
        try
        {
            MailMessage mail = new MailMessage();
            SmtpClient smtpserver = new SmtpClient("smtp.gmail.com");
            mail.From = new MailAddress("bhakthiprabhu15@gmail.com");
            mail.To.Add(email);
            mail.Subject = "Forgot your security answer?";
            mail.Body = "The security answer for the security question\n"
+ sqns + " is\n" + sans;
            smtpserver.Port = 587;
            smtpserver.Credentials = new
NetworkCredential("bhakthiprabhu15@gmail.com", "bhakthi15");
            smtpserver.EnableSsl = true;
            smtpserver.Send(mail);
            pictureBox1.Visible = true;
            pictureBox3.Visible = false;
            label2.Text = "We sent an email to " + email + " with a
security answer. Don't forget to check your spam folder.";
        }
        catch (Exception)
        {
            pictureBox1.Visible = false;
            pictureBox3.Visible = true;
            label1.Text = "Connection Error!";
            label2.Text = "No data connection.Consider turning on mobile
data or turning on Wi-Fi";
        }
    }

    private void close_Click(object sender, EventArgs e)
    {
        //close
        this.Owner.Enabled = true;
        this.Owner.Show();
        this.WindowState = FormWindowState.Normal;
        this.Close();
    }
}

```

Service:

```

using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsApplication

```

```

{
    public partial class Service : Form
    {
        Connect c = new Connect();
        string srno;
        public Service()
        {
            InitializeComponent();
        }

        private void Service_Load(object sender, EventArgs e)
        {
            Supplier s = new Supplier();
            s.datagridviewstyle(dataGridView1);
            dateTimePicker1.Value = DateTime.Today;
            dateTimePicker1.MinDate = DateTime.Today;
            dateTimePicker1.MaxDate = DateTime.Today.AddMonths(1);
            displayall();
            dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;

            SqlCommand sc = new SqlCommand("select * from Sale_Bill where
Net_Amount is not null", c.cn);
            SqlDataAdapter adp = new SqlDataAdapter(sc);
            DataSet ds = new DataSet();
            adp.Fill(ds);
            sc.ExecuteNonQuery();
            cb1.DataSource = ds.Tables[0];
            cb1.DisplayMember = "SB_No";
            cb1.ValueMember = "SB_No";
            cb1.Text = "Select";
        }

        private void cb1_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (!(cb1.Text == "" || cb1.Text == "Select"))
            {
                SqlCommand sc = new SqlCommand("select * from Sale_Bill_Detail
where SB_No='"
+ cb1.SelectedValue.ToString() + "'", c.cn);
                SqlDataAdapter adp = new SqlDataAdapter(sc);
                DataSet ds = new DataSet();
                adp.Fill(ds);
                sc.ExecuteNonQuery();
                cb2.DataSource = ds.Tables[0];
                cb2.DisplayMember = "Item_Id";
                cb2.ValueMember = "Item_Id";
                cb2.Text = "Select";
            }
        }

        private void comboBox1_SelectedIndexChanged(object sender, EventArgs
e)
        {
            if (comboBox1.Text == "Warranty")
            {
                ttamt.Text = "0.0";
                ttax.Text = "0.0";

```

```

        ttamt.Text = "0.0";
        ttamt.ReadOnly = true;
    }
    else
    {
        ttamt.Text = "";
        ttax.Text = "";
        tgtotal.Text = "";
        ttamt.ReadOnly = false;
    }
}

private void clearall()
{
    ttamt.Text = "";
    ttax.Text = "";
    tgtotal.Text = "";
    ttamt.ReadOnly = false;
    cb2.Text = "Select";
    cb1.Text = "Select";
    comboBox1.Text = "Select";
    dateTimePicker1.Text = "";
    tremark.Text = "";
    nupdate.Enabled = false;
    ndelete.Enabled = false;
    btngeneratebil.Enabled = true;
}

private void displayall()
{
    dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
    SqlDataAdapter adp1 = new SqlDataAdapter("select * from Service",
c.cn);
    DataSet ds2 = new DataSet();
    adp1.Fill(ds2, "dt2");
    dataGridView1.DataSource = ds2.Tables["dt2"];
}

private void btncancel_Click(object sender, EventArgs e)
{
    clearall();
    displayall();
}

private void ttamt_KeyUp(object sender, KeyEventArgs e)
{
    if (ttamt.Text != "")
    {
        ttax.Text = (Convert.ToDouble(ttamt.Text) * 0.18).ToString();
        tgtotal.Text = (Convert.ToDouble(ttamt.Text) +
Convert.ToDouble(ttax.Text)).ToString();
    }
    else
    {
        ttax.Text = "";
        tgtotal.Text = "";
    }
}

```

```

        }

    }

    private void tsearch_KeyUp(object sender, KeyEventArgs e)
    {
        SqlDataAdapter sda = new SqlDataAdapter("select * from Service
where SR_No like '" + tsearch.Text + "%'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dataGridView1.DataSource = ds.Tables["dt"];
    }

    private void tsearch_Leave(object sender, EventArgs e)
    {
        if (tsearch.Text == "")
            tsearch.Text = "search...";
        tsearch.ForeColor = Color.Silver;
    }

    private void tsearch_Click(object sender, EventArgs e)
    {
        if (tsearch.Text == "search...")
            tsearch.Text = "";
        tsearch.ForeColor = Color.FromArgb(20, 168, 0);
    }

    private void btngeneratebil_Click(object sender, EventArgs e)
    {
        try
        {
            if (cb1.Text == "" || cb1.Text == "Select" || cb2.Text == ""
|| cb2.Text == "Select" || comboBox1.Text == "" || comboBox1.Text == "Select"
|| ttamt.Text == "" || tremark.Text == "")
                throw new ArgumentException();
            else
            {
                if (MessageBox.Show("Are You Sure ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    c.cmd.CommandText = "select Count(*) from Service";
                    int cnt = Convert.ToInt16(c.cmd.ExecuteScalar());
                    cnt++;
                    srno = "sr" + cnt;
                    String date = dateTimePicker1.Value.ToString("MM-dd-yyyy");
                    c.cmd.CommandText = "insert into Service
(SR_No, Date, SB_No, Service_Type, Total_Amount, Tax, Grand_Total, Remark, Item_Id)
values('" + srno + "','" + date + "','" + cb1.Text + "','" + comboBox1.Text +
','" + ttamt.Text + "','" + ttax.Text + "','" + tgttotal.Text + "','" +
tremark.Text + "','" + cb2.Text + "')";
                    c.cmd.ExecuteNonQuery();
                    MessageBox.Show("Bill generated Successfully!", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    displayall();
                    clearall();
                }
            }
        }
    }
}

```

```

        }
        catch (ArgumentException)
        {
            MessageBox.Show("Enter all the details", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
        catch (SqlException)
        {
            errorProvider1.SetError(ttamt, "Invalid Amount!");
        }
    }

    private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if (dataGridView1.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
    {
        nupdate.Enabled = true;
        ndelete.Enabled = true;
        btngeneratebil.Enabled = false;
        dataGridView1.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        dataGridView1.CurrentRow.Selected = true;
        srno =
dataGridView1.Rows[e.RowIndex].Cells["SR_No"].FormattedValue.ToString();
        dateTimePicker1.Text =
dataGridView1.Rows[e.RowIndex].Cells["Date"].FormattedValue.ToString();
        cb1.Text =
dataGridView1.Rows[e.RowIndex].Cells["SB_No"].FormattedValue.ToString();
        cb2.Text =
dataGridView1.Rows[e.RowIndex].Cells["Item_Id"].FormattedValue.ToString();
        comboBox1.Text =
dataGridView1.Rows[e.RowIndex].Cells["Service_Type"].FormattedValue.ToString();
        ttamt.Text =
dataGridView1.Rows[e.RowIndex].Cells["Total_Amount"].FormattedValue.ToString();
        ttax.Text =
dataGridView1.Rows[e.RowIndex].Cells["Tax"].FormattedValue.ToString();
        tgtotal.Text =
dataGridView1.Rows[e.RowIndex].Cells["Grand_Total"].FormattedValue.ToString();
        tremark.Text =
=dataGridView1.Rows[e.RowIndex].Cells["Remark"].FormattedValue.ToString();
    }
}

private void nupdate_Click(object sender, EventArgs e)
{
    try
    {
        if (cb1.Text == "" || cb1.Text == "Select" || cb2.Text == ""
|| cb2.Text == "Select" || comboBox1.Text == "" || comboBox1.Text == "Select"
|| ttamt.Text == "" || tremark.Text == "")
            throw new ArgumentException();
    }
}

```

```

        else
        {
            if (MessageBox.Show("Are You Sure ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                String date = dateTimePicker1.Value.ToString("MM-dd-yyyy");
                c.cmd.CommandText = "update Service set SR_No='"
                srno + "',Date='"
                + date + "',SB_No='"
                + cb1.Text + "',Service_Type='"
                + comboBox1.Text + "',Total_Amount='"
                + ttamt.Text + "',Tax='"
                + ttax.Text +
                "',Grand_Total='"
                + ttotal.Text + "',Remark='"
                + tremark.Text +
                "',Item_Id='"
                + cb2.Text + "' where SR_No='"
                + srno + "'";
                c.cmd.ExecuteNonQuery();
                MessageBox.Show("Bill updated Successfully!", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
                displayall();
                clearall();
            }
        }
    }
    catch (ArgumentException)
    {
        MessageBox.Show("Enter all the details", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void ndelete_Click(object sender, EventArgs e)
{
    try
    {
        if (cb1.Text == "" || cb1.Text == "Select" || cb2.Text == ""
        || cb2.Text == "Select" || comboBox1.Text == "" || comboBox1.Text == "Select"
        || ttamt.Text == "" || tremark.Text == "")
            throw new ArgumentException();
        else
        {
            if (MessageBox.Show("Are You Sure ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                c.cmd.CommandText = "delete from Service where
SR_No='"
                + srno + "'";
                c.cmd.ExecuteNonQuery();
                MessageBox.Show("Bill deleted Successfully!", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
                displayall();
                clearall();
            }
        }
    }
    catch (ArgumentException)
    {
        MessageBox.Show("Enter all the details", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

```

```

        private void ttamt_KeyPress(object sender, KeyPressEventArgs e)
        {
            e.Handled = !(char.IsDigit(e.KeyChar) || e.KeyChar == '.' || 
e.KeyChar == (char)Keys.Back);
        }

        private void tremark_KeyPress(object sender, KeyPressEventArgs e)
        {
            e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar == 
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || char.IsDigit(e.KeyChar));
        }

        private void tsearch_KeyPress(object sender, KeyPressEventArgs e)
        {
            e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar == 
(char)Keys.Back || char.IsDigit(e.KeyChar));
        }

        private void ttamt_TextChanged(object sender, EventArgs e)
        {
            errorProvider1.Clear();
        }
    }
}

```

Supplier:

```

using System.Text.RegularExpressions;
using System.Net.Mail;
using System.Data.SqlClient;

namespace WindowsApplication
{
    public partial class Supplier : Form
    {
        Connect c = new Connect();
        String sid;
        public Supplier()
        {
            InitializeComponent();
        }

        private void displayall()
        {
            //display function
            if ((cbactive.Checked == true && cbinactive.Checked == true) || 
(cbactive.Checked == false && cbinactive.Checked == false))
            {
                SqlDataAdapter sda = new SqlDataAdapter("select * from 
Supplier", c.cn);
                DataSet ds = new DataSet();
                sda.Fill(ds, "dt");
                dataGridView1.DataSource = ds.Tables["dt"];
            }
            else if (cbactive.Checked == true && cbinactive.Checked == false)
            {

```

```

        SqlDataAdapter sda = new SqlDataAdapter("select * from
Supplier where Status='Active'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dataGridView1.DataSource = ds.Tables["dt"];
    }
    else if (cbactive.Checked == false && cbinactive.Checked == true)
    {
        SqlDataAdapter sda = new SqlDataAdapter("select * from
Supplier where Status='Inactive'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dataGridView1.DataSource = ds.Tables["dt"];
    }
}

private void clearall()
{
    //clear function
    supname.Text = "";
    address.Text = "";
    email.Text = "";
    cno.Text = "";
    vsupname.Text = "";
    filter.ResetText();
    filter.Text = "Sort By";
    msg.Text = "";
    nupdate.Enabled = false;
    ndelete.Enabled = false;
    nadd.Enabled = true;
    dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
    vsupname.Text = "search...";
    vsupname.ForeColor = Color.Silver;
}

public void datagridviewstyle(DataGridView dgv)
{
    //datagridview
    dgv.EnableHeadersVisualStyles = false;
    dgv.SelectionMode = DataGridViewSelectionMode.CellSelect;
    dgv.BackgroundColor = Color.FromArgb(30, 35, 38);
    dgv.MultiSelect = false;
    dgv.AllowUserToAddRows = false;
    dgv.AllowUserToDeleteRows = false;
    dgv.AllowUserToResizeColumns = false;
    dgv.AllowUserToResizeRows = false;
    dgv.ReadOnly = true;
    dgv.BorderStyle = BorderStyle.None;
    dgv.ShowEditingIcon = false;
    dgv.DefaultCellStyle.ForeColor = Color.Black;
    dgv.DefaultCellStyle.BackColor = Color.Gainsboro;
    dgv.DefaultCellStyle.SelectionBackColor = Color.SeaGreen;
    dgv.DefaultCellStyle.SelectionForeColor = Color.WhiteSmoke;
    dgv.DefaultCellStyle.Font = new Font("Arial", 10, FontStyle.Bold);
    dgv.CellBorderStyle = DataGridViewCellBorderStyle.Single;
    dgv.RowHeadersVisible = false;
}

```

```

        dgv.RowHeadersWidthSizeMode =
DataGridViewRowHeadersWidthSizeMode.DisableResizing;
        dgv.AlternatingRowsDefaultCellStyle.BackColor =
Color.FromArgb(235, 236, 240);
        dgv.ColumnHeadersDefaultCellStyle.Font = new Font("Arial", 10.25f,
FontStyle.Bold);
        dgv.ColumnHeadersDefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
        dgv.ColumnHeadersDefaultCellStyle.BackColor = Color.FromArgb(45,
66, 91);
        dgv.ColumnHeadersDefaultCellStyle.ForeColor = Color.White;
        dgv.ColumnHeadersBorderStyle =
DataGridViewHeaderBorderStyle.Single;
        dgv.ColumnHeadersDefaultCellStyle.HeightSizeMode =
DataGridViewColumnHeadersHeightSizeMode.DisableResizing;
    }

    private void Supplier_Load(object sender, EventArgs e)
{
    //page load
    nupdate.Enabled = false;
    ndelete.Enabled = false;
    displayall();
    datagridviewstyle(dataGridView1);
}

private void nadd_Click(object sender, EventArgs e)
{
    //register
    try
    {
        Regex cnum = new Regex("^[6-9][0-9]{9}$");
        if (supname.Text == "" || address.Text == "" ||
Convert.ToString(cno.Text) == "" || email.Text == "")
            throw new ArgumentException();
        else
        {
            new MailAddress(email.Text);
            if (!(cnum.IsMatch(cno.Text)))
                errorProvider1.SetError(cno, "Enter valid mobile
number");
            else
            {
                SqlCommand cmd = new SqlCommand("Select * from
Supplier where Name='" + supname.Text + "' and Contact_No='" + cno.Text + "'"
and Email=''" + email.Text + "'", c.cn);
                SqlDataAdapter adp = new SqlDataAdapter(cmd);
                DataTable dt = new DataTable();
                adp.Fill(dt);
                if (dt.Rows.Count > 0)
                    MessageBox.Show("Record Exists!\n Please Edit
Supplier if you want to make changes!", "", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                else if (MessageBox.Show("Are You Sure ?", "", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    String supid;

```

```

        c.cmd.CommandText = "select Count(*) from
Supplier";
        int cnt = Convert.ToInt16(c.cmd.ExecuteScalar());
        cnt++;
        supid = "sup" + cnt;
        c.cmd.CommandText = "insert into Supplier (Sup_Id)
values('" + supid + "')";
        c.cmd.ExecuteNonQuery();
        c.cmd.CommandText = "update Supplier set Name='" +
supname.Text + "',Address='" + address.Text + "',Email='" + email.Text +
"',Contact_No= '" + cno.Text + "',Status='Active' where Sup_Id='" + supid + 
"';
        c.cmd.ExecuteNonQuery();
        MessageBox.Show("Supplier Added Successfully!",
"", MessageBoxButtons.OK, MessageBoxIcon.Information);
        displayall();
        clearall();
    }
}
}
catch (FormatException)
{
    errorProvider2.SetError(email, "Enter valid format email");
}
catch (ArgumentException)
{
    MessageBox.Show("Enter all the details", "",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}

private void email_TextChanged(object sender, EventArgs e)
{
    errorProvider2.Clear();
}

private void cno_TextChanged(object sender, EventArgs e)
{
    errorProvider1.Clear();
}

private void nclear_Click(object sender, EventArgs e)
{
    //clear
    clearall();
    cbinactive.Checked = false;
    cbactive.Checked = false;
    displayall();
}

private void vsupname_Click(object sender, EventArgs e)
{
    if (vsupname.Text == "search...")
        vsupname.Text = "";
    vsupname.ForeColor = Color.FromArgb(20, 168, 0);
}

```

```

private void vsupname_Leave(object sender, EventArgs e)
{
    if (vsupname.Text == "")
        vsupname.Text = "search...";
    vsupname.ForeColor = Color.Silver;
}

private void vsupname_KeyUp(object sender, KeyEventArgs e)
{
    filter.ResetText();
    filter.Text = "Sort By";
    cbinactive.Checked = false;
    cbactive.Checked = false;
    SqlDataAdapter sda = new SqlDataAdapter("select * from Supplier
where Name like '" + vsupname.Text + "%' or Sup_Id like '" + vsupname.Text +
"%"', c.cn);
    DataSet ds = new DataSet();
    sda.Fill(ds, "dt");
    dataGridView1.DataSource = ds.Tables["dt"];
}

private void filter_SelectedIndexChanged(object sender, EventArgs e)
{
    //filter by
    vsupname.Text = "search...";
    cbactive.Checked = false;
    cbinactive.Checked = false;
    sortfunc(filter.SelectedItem.ToString());
}

private void sortfunc(String value)
{
    //sortfunc function
    SqlDataAdapter adp = new SqlDataAdapter();
    DataSet ds = new DataSet();
    c.cmd = new SqlCommand("select * from Supplier", c.cn);
    adp.SelectCommand = c.cmd;
    ds.Clear();
    adp.Fill(ds);

    dataGridView1.Sort(dataGridView1.Columns[value], ListSortDirection.Ascending);
}

private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    msg.Visible = false;
    if (dataGridView1.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
    {
        nupdate.Enabled = true;
        ndelete.Enabled = true;
        nadd.Enabled = false;
        dataGridView1.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
}

```

```

        dataGridView1.CurrentRow.Selected = true;
        sid =
dataGridView1.Rows[e.RowIndex].Cells["Sup_Id"].FormattedValue.ToString();
        supname.Text =
dataGridView1.Rows[e.RowIndex].Cells["Name"].FormattedValue.ToString();
        email.Text =
dataGridView1.Rows[e.RowIndex].Cells["Email"].FormattedValue.ToString();
        cno.Text =
dataGridView1.Rows[e.RowIndex].Cells["Contact_No"].FormattedValue.ToString();
        address.Text =
dataGridView1.Rows[e.RowIndex].Cells["Address"].FormattedValue.ToString();
    }

private void nupdate_Click(object sender, EventArgs e)
{
    //update
    try
    {
        Regex rx = new Regex("^[6-9][0-9]{9}$");
        if (supname.Text == "" || address.Text == "" || Convert.ToString(cno.Text) == "" || email.Text == "")
            throw new ArgumentException();
        else if (!(rx.IsMatch(cno.Text)))
            errorProvider1.SetError(cno, "Enter valid mobile number");
        else
        {
            new MailAddress(email.Text);
            if (MessageBox.Show("Are You Sure want to Update ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                c.cmd.CommandText = "update Supplier set Name='" +
supname.Text + "',Address='" + address.Text + "',Email='" + email.Text +
"',Contact_No= '" + cno.Text + "' where Sup_Id=''" + sid + "'";
                c.cmd.ExecuteNonQuery();
                MessageBox.Show("Supplier Updated Successfully!", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
                dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
                dataGridView1.Rows[0].Cells[0].Selected = true;
                displayall();
                clearall();
            }
        }
    catch (FormatException)
    {
        errorProvider2.SetError(email, "Enter valid format email");
    }
    catch (ArgumentException)
    {
        MessageBox.Show("Enter all the details", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void ndelete_Click(object sender, EventArgs e)

```

```

{
    //delete
    c.cmd.CommandText = "select Status from Supplier where Sup_Id='"
    sid + "'";
    if (Convert.ToString(c.cmd.ExecuteScalar()) == "Inactive")
    {
        msg.Visible = true;
        msg.Text = "Supplier already Inactivated!";
    }
    else
    {
        if (MessageBox.Show("Are You Sure want to Inactivate " +
    supname.Text + " ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
    DialogResult.Yes)
        {
            c.cmd.CommandText = "update Supplier set Status='Inactive'
where Sup_Id='"
    sid + "'";
            c.cmd.ExecuteNonQuery();
            MessageBox.Show("Supplier Inactivated!", "", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
            dataGridView1.SelectionMode =
DataGridViewSelectionMode.CellSelect;
            dataGridView1.Rows[0].Cells[0].Selected = true;
            displayall();
            clearall();
        }
    }
}

private void cbactive_CheckedChanged(object sender, EventArgs e)
{
    //checkbox active
    vsupname.Text = "Search...";
    filter.ResetText();
    filter.Text = "Sort By";
    displayall();
}

private void cbinactive_CheckedChanged(object sender, EventArgs e)
{
    //checkbox inactive
    vsupname.Text = "Search...";
    filter.ResetText();
    filter.Text = "Sort By";
    displayall();
}

private void supname_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || e.KeyChar == '.');
}

private void address_KeyPress(object sender, KeyPressEventArgs e)
{
}

```

```

        e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || char.IsDigit(e.KeyChar) || 
e.KeyChar == ',' || e.KeyChar == '-' || e.KeyChar == '&'amp; || e.KeyChar == '.');
    }

    private void vsupname_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Back || char.IsWhiteSpace(e.KeyChar) || char.IsDigit(e.KeyChar));
    }
}

```

Display:

```

using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsApplication
{
    public partial class Display : Form
    {
        Connect c = new Connect();
        public Display()
        {
            InitializeComponent();
        }

        private void Display_Load(object sender, EventArgs e)
        {
            panelpo.BringToFront();
            po.BackColor = Color.FromArgb(24, 123, 205);
            pb.BackColor = Color.Gray;
            pr.BackColor = Color.Gray;
            sb.BackColor = Color.Gray;

            Supplier s = new Supplier();
            s.datagridviewstyle(dgvpo);
            s.datagridviewstyle(dgvpod);
            SqlCommand sc = new SqlCommand("select * from Purchase_Order",
c.cn);
            SqlDataAdapter adp = new SqlDataAdapter(sc);
            DataSet ds = new DataSet();
            adp.Fill(ds);
            sc.ExecuteNonQuery();
            cbpurchaseorder.DataSource = ds.Tables[0];
            cbpurchaseorder.DisplayMember = "PO_No";
            cbpurchaseorder.ValueMember = "PO_No";
            cbpurchaseorder.Text = "Select";
        }

        private void po_Click(object sender, EventArgs e)
        {
            panelpo.BringToFront();
            panelpo.Show();
        }
    }
}

```

```

        po.BackColor = Color.FromArgb(24, 123, 205);
        pb.BackColor = Color.Gray;
        pr.BackColor = Color.Gray;
        sb.BackColor = Color.Gray;

        Supplier s = new Supplier();
        s.datagridviewstyle(dgvo);
        s.datagridviewstyle(dgvpod);
        SqlCommand sc = new SqlCommand("select * from Purchase_Order",
c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(sc);
        DataSet ds = new DataSet();
        adp.Fill(ds);
        sc.ExecuteNonQuery();
        cbpurchaseorder.DataSource = ds.Tables[0];
        cbpurchaseorder.DisplayMember = "PO_No";
        cbpurchaseorder.ValueMember = "PO_No";
        cbpurchaseorder.Text = "Select";
    }

    private void cbpurchaseorder_SelectedIndexChanged(object sender,
EventArgs e)
{
    if (!(cbpurchaseorder.Text == "" || cbpurchaseorder.Text ==
"Select"))
    {
        dgvo.SelectionMode = DataGridViewSelectionMode.CellSelect;
        SqlCommand sc = new SqlCommand("select * from Purchase_Order
where PO_No=' " + cbpurchaseorder.SelectedValue.ToString() + "'", c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(sc);
        DataSet ds = new DataSet();
        adp.Fill(ds, "dt");
        dgvo.DataSource = ds.Tables["dt"];

        dgvpod.SelectionMode = DataGridViewSelectionMode.CellSelect;
        SqlCommand sc1 = new SqlCommand("select
Purchase_Order_Detail.PO_No,Purchase_Order_Detail.Item_Id,Inventory.Name,Purch
ase_Order_Detail.Qty from Purchase_Order_Detail inner join Inventory on
Purchase_Order_Detail.Item_Id = Inventory.Item_Id where
Purchase_Order_Detail.PO_No=' " + cbpurchaseorder.SelectedValue.ToString() +
"'", c.cn);
        SqlDataAdapter adp1 = new SqlDataAdapter(sc1);
        DataSet ds1 = new DataSet();
        adp1.Fill(ds1, "dt1");
        dgvpod.DataSource = ds1.Tables["dt1"];
    }
}

private void searchitem_KeyUp(object sender, KeyEventArgs e)
{
    if (cbpurchaseorder.Text == "Select" || cbpurchaseorder.Text ==
"")
    {
        SqlDataAdapter sda = new SqlDataAdapter("select
Purchase_Order_Detail.PO_No,Purchase_Order_Detail.Item_Id,Inventory.Name,Purch
ase_Order_Detail.Qty from Purchase_Order_Detail inner join Inventory on

```

```

Purchase_Order_Detail.Item_Id = Inventory.Item_Id where Inventory.Name like ''
+ searchitem.Text + "%'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgvpod.DataSource = ds.Tables["dt"];
    }
    else
    {
        SqlDataAdapter sda = new SqlDataAdapter("select
Purchase_Order_Detail.PO_No,Purchase_Order_Detail.Item_Id,Inventory.Name,Purch
ase_Order_Detail.Qty from Purchase_Order_Detail inner join Inventory on
Purchase_Order_Detail.Item_Id = Inventory.Item_Id where Inventory.Name like ''
+ searchitem.Text + '%' and Purchase_Order_Detail.PO_No='"
+ cbpurchaseorder.SelectedValue.ToString() + "'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgvpod.DataSource = ds.Tables["dt"];
    }
}

private void searchitem_Leave(object sender, EventArgs e)
{
    if (searchitem.Text == "")
        searchitem.Text = "search...";
    searchitem.ForeColor = Color.Silver;
}

private void searchitem_Click(object sender, EventArgs e)
{
    if (searchitem.Text == "search...")
        searchitem.Text = "";
    searchitem.ForeColor = Color.FromArgb(20, 168, 0);
}

private void cbpurchaseorder_TextChanged(object sender, EventArgs e)
{
    searchitem.Text = "search...";
    dgvo.SelectionMode = DataGridViewSelectionMode.CellSelect;
    SqlCommand sc = new SqlCommand("select * from Purchase_Order",
c.cn);
    SqlDataAdapter adp = new SqlDataAdapter(sc);
    DataSet ds = new DataSet();
    adp.Fill(ds, "dt");
    dgvo.DataSource = ds.Tables["dt"];

    dgvpod.SelectionMode = DataGridViewSelectionMode.CellSelect;
    SqlCommand sc1 = new SqlCommand("select
Purchase_Order_Detail.PO_No,Purchase_Order_Detail.Item_Id,Inventory.Name,Purch
ase_Order_Detail.Qty from Purchase_Order_Detail inner join Inventory on
Purchase_Order_Detail.Item_Id = Inventory.Item_Id", c.cn);
    SqlDataAdapter adp1 = new SqlDataAdapter(sc1);
    DataSet ds1 = new DataSet();
    adp1.Fill(ds1, "dt1");
    dgvpod.DataSource = ds1.Tables["dt1"];
}

private void pb_Click(object sender, EventArgs e)

```

```

    {
        panelpb.BringToFront();
        po.BackColor = Color.Gray;
        pb.BackColor = Color.FromArgb(24, 123, 205);
        pr.BackColor = Color.Gray;
        sb.BackColor = Color.Gray;

        Supplier s = new Supplier();
        s.datagridviewstyle(dgvpb);
        s.datagridviewstyle(dgvpbd);
        SqlCommand sc = new SqlCommand("select * from Purchase_Bill",
c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(sc);
        DataSet ds = new DataSet();
        adp.Fill(ds);
        sc.ExecuteNonQuery();
        comboBox1.DataSource = ds.Tables[0];
        comboBox1.DisplayMember = "PB_No";
        comboBox1.ValueMember = "PB_No";
        comboBox1.Text = "Select";
    }

    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (!(comboBox1.Text == "" || comboBox1.Text == "Select"))
        {
            dgvpb.SelectionMode = DataGridViewSelectionMode.CellSelect;
            SqlCommand sc = new SqlCommand("select * from Purchase_Bill
where PB_No='"
+ comboBox1.SelectedValue.ToString() + "'", c.cn);
            SqlDataAdapter adp = new SqlDataAdapter(sc);
            DataSet ds = new DataSet();
            adp.Fill(ds, "dt");
            dgvpb.DataSource = ds.Tables["dt"];

            dgvpbd.SelectionMode = DataGridViewSelectionMode.CellSelect;
            SqlCommand sc1 = new SqlCommand("select
Purchase_Bill_Detail.PB_No,Purchase_Bill_Detail.Item_Id,Inventory.Name,Purchas
e_Bill_Detail.Qty,Purchase_Bill_Detail.Price,Purchase_Bill_Detail.Amount,Purch
ase_Bill_Detail.Warranty from Purchase_Bill_Detail inner join Inventory on
Purchase_Bill_Detail.Item_Id = Inventory.Item_Id where
Purchase_Bill_Detail.PB_No='"
+ comboBox1.SelectedValue.ToString() + "'", c.cn);
            SqlDataAdapter adp1 = new SqlDataAdapter(sc1);
            DataSet ds1 = new DataSet();
            adp1.Fill(ds1, "dt1");
            dgvpbd.DataSource = ds1.Tables["dt1"];
        }
    }

    private void comboBox1_TextChanged(object sender, EventArgs e)
    {
        tsearch.Text="search...";
        dgvpb.SelectionMode = DataGridViewSelectionMode.CellSelect;
        SqlCommand sc = new SqlCommand("select * from Purchase_Bill",
c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(sc);

```

```

        DataSet ds = new DataSet();
        adp.Fill(ds, "dt");
        dgvpb.DataSource = ds.Tables["dt"];

        dgvpbd.SelectionMode = DataGridViewSelectionMode.CellSelect;
        SqlCommand sc1 = new SqlCommand("select
Purchase_Bill_Detail.PB_No,Purchase_Bill_Detail.Item_Id,Inventory.Name,Purchas
e_Bill_Detail.Qty,Purchase_Bill_Detail.Price,Purchase_Bill_Detail.Amount,Purch
ase_Bill_Detail.Warranty from Purchase_Bill_Detail inner join Inventory on
Purchase_Bill_Detail.Item_Id = Inventory.Item_Id", c.cn);
        SqlDataAdapter adp1 = new SqlDataAdapter(sc1);
        DataSet ds1 = new DataSet();
        adp1.Fill(ds1, "dt1");
        dgvpbd.DataSource = ds1.Tables["dt1"];
    }

    private void tsearch_Leave(object sender, EventArgs e)
{
    if (tsearch.Text == "")
        tsearch.Text = "search...";
    tsearch.ForeColor = Color.Silver;
}

private void tsearch_KeyUp(object sender, KeyEventArgs e)
{
    if (comboBox1.Text == "Select" || comboBox1.Text == "")
    {
        SqlDataAdapter sda = new SqlDataAdapter("select
Purchase_Bill_Detail.PB_No,Purchase_Bill_Detail.Item_Id,Inventory.Name,Purchas
e_Bill_Detail.Qty,Purchase_Bill_Detail.Price,Purchase_Bill_Detail.Amount,Purch
ase_Bill_Detail.Warranty from Purchase_Bill_Detail inner join Inventory on
Purchase_Bill_Detail.Item_Id = Inventory.Item_Id where Inventory.Name like '" +
tsearch.Text + "%'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgvpbd.DataSource = ds.Tables["dt"];
    }
    else
    {
        SqlDataAdapter sda = new SqlDataAdapter("select
Purchase_Bill_Detail.PB_No,Purchase_Bill_Detail.Item_Id,Inventory.Name,Purchas
e_Bill_Detail.Qty,Purchase_Bill_Detail.Price,Purchase_Bill_Detail.Amount,Purch
ase_Bill_Detail.Warranty from Purchase_Bill_Detail inner join Inventory on
Purchase_Bill_Detail.Item_Id = Inventory.Item_Id where Inventory.Name like '" +
tsearch.Text + "%' and
Purchase_Bill_Detail.PB_No='"+comboBox1.SelectedValue.ToString()+"'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgvpbd.DataSource = ds.Tables["dt"];
    }
}

private void tsearch_Click(object sender, EventArgs e)
{
    if (tsearch.Text == "search...")
        tsearch.Text = "";
    tsearch.ForeColor = Color.FromArgb(20, 168, 0);
}

```

```

        }

    private void pr_Click(object sender, EventArgs e)
    {
        panelpr.BringToFront();
        po.BackColor = Color.Gray;
        pb.BackColor = Color.Gray;
        pr.BackColor = Color.FromArgb(24, 123, 205);
        sb.BackColor = Color.Gray;

        Supplier s = new Supplier();
        s.datagridviewstyle(dgvpr);
        s.datagridviewstyle(dgvpred);
        SqlCommand sc = new SqlCommand("select * from Purchase_Return",
c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(sc);
        DataSet ds = new DataSet();
        adp.Fill(ds);
        sc.ExecuteNonQuery();
        comboBox2.DataSource = ds.Tables[0];
        comboBox2.DisplayMember = "PR_No";
        comboBox2.ValueMember = "PR_No";
        comboBox2.Text = "Select";
    }

    private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (!(comboBox2.Text == "" || comboBox2.Text == "Select"))
        {
            dgvpr.SelectionMode = DataGridViewSelectionMode.CellSelect;
            SqlCommand sc = new SqlCommand("select * from Purchase_Return
where PR_No='"
+ comboBox2.SelectedValue.ToString() + "'", c.cn);
            SqlDataAdapter adp = new SqlDataAdapter(sc);
            DataSet ds = new DataSet();
            adp.Fill(ds, "dt");
            dgvpr.DataSource = ds.Tables["dt"];

            dgvpred.SelectionMode = DataGridViewSelectionMode.CellSelect;
            SqlCommand sc1 = new SqlCommand("select
Purchase_Return_Detail.PR_No, Purchase_Return_Detail.Item_Id, Inventory.Name, Pur
chase_Return_Detail.Qty from Purchase_Return_Detail inner join Inventory on
Purchase_Return_Detail.Item_Id = Inventory.Item_Id where
Purchase_Return_Detail.PR_No='"
+ comboBox2.SelectedValue.ToString() + "'", c.cn);
            SqlDataAdapter adp1 = new SqlDataAdapter(sc1);
            DataSet ds1 = new DataSet();
            adp1.Fill(ds1, "dt1");
            dgvpred.DataSource = ds1.Tables["dt1"];
        }
    }

    private void comboBox2_TextChanged(object sender, EventArgs e)
    {
        textBox1.Text = "search...";
        dgvpr.SelectionMode = DataGridViewSelectionMode.CellSelect;
    }
}

```

```

        SqlCommand sc = new SqlCommand("select * from Purchase_Return",
c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(sc);
        DataSet ds = new DataSet();
        adp.Fill(ds, "dt");
        dgvpr.DataSource = ds.Tables["dt"];

        dgvprd.SelectionMode = DataGridViewSelectionMode.CellSelect;
        SqlCommand sc1 = new SqlCommand("select
Purchase_Return_Detail.PR_No,Purchase_Return_Detail.Item_Id,Inventory.Name,Pur
chase_Return_Detail.Qty from Purchase_Return_Detail inner join Inventory on
Purchase_Return_Detail.Item_Id = Inventory.Item_Id", c.cn);
        SqlDataAdapter adp1 = new SqlDataAdapter(sc1);
        DataSet ds1 = new DataSet();
        adp1.Fill(ds1, "dt1");
        dgvprd.DataSource = ds1.Tables["dt1"];
    }

    private void textBox1_KeyUp(object sender, KeyEventArgs e)
{
    if (comboBox2.Text == "Select" || comboBox2.Text == "")
    {
        SqlDataAdapter sda = new SqlDataAdapter("select
Purchase_Return_Detail.PR_No,Purchase_Return_Detail.Item_Id,Inventory.Name,Pur
chase_Return_Detail.Qty from Purchase_Return_Detail inner join Inventory on
Purchase_Return_Detail.Item_Id = Inventory.Item_Id where Inventory.Name like
'" + textBox1.Text + "%'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgvprd.DataSource = ds.Tables["dt"];
    }
    else
    {
        SqlDataAdapter sda = new SqlDataAdapter("select
Purchase_Return_Detail.PR_No,Purchase_Return_Detail.Item_Id,Inventory.Name,Pur
chase_Return_Detail.Qty from Purchase_Return_Detail inner join Inventory on
Purchase_Return_Detail.Item_Id = Inventory.Item_Id where Inventory.Name like
'" + textBox1.Text + "%' and
Purchase_Return_Detail.PR_No='"+comboBox2.SelectedValue.ToString()+"'", c.cn);
        DataSet ds = new DataSet();
        sda.Fill(ds, "dt");
        dgvprd.DataSource = ds.Tables["dt"];
    }
}

private void textBox1_Leave(object sender, EventArgs e)
{
    if (textBox1.Text == "")
        textBox1.Text = "search...";
    textBox1.ForeColor = Color.Silver;
}

private void textBox1_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "search...")
        textBox1.Text = "";
    textBox1.ForeColor = Color.FromArgb(20, 168, 0);
}

```

```

        }

    private void sb_Click(object sender, EventArgs e)
    {
        panelsb.BringToFront();
        po.BackColor = Color.Gray;
        pb.BackColor = Color.Gray;
        pr.BackColor = Color.Gray;
        sb.BackColor = Color.FromArgb(24, 123, 205);

        Supplier s = new Supplier();
        s.datagridviewstyle(dgvsb);
        s.datagridviewstyle(dgvsbd);
        SqlCommand sc = new SqlCommand("select * from Sale_Bill", c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(sc);
        DataSet ds = new DataSet();
        adp.Fill(ds);
        sc.ExecuteNonQuery();
        comboBox3.DataSource = ds.Tables[0];
        comboBox3.DisplayMember = "SB_No";
        comboBox3.ValueMember = "SB_No";
        comboBox3.Text = "Select";
    }

    private void comboBox3_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (!(comboBox3.Text == "" || comboBox3.Text == "Select"))
        {
            dgvsb.SelectionMode = DataGridViewSelectionMode.CellSelect;
            SqlCommand sc = new SqlCommand("select * from Sale_Bill where
SB_No=' " + comboBox3.SelectedValue.ToString() + "'", c.cn);
            SqlDataAdapter adp = new SqlDataAdapter(sc);
            DataSet ds = new DataSet();
            adp.Fill(ds, "dt");
            dgvsb.DataSource = ds.Tables["dt"];

            dgvsbd.SelectionMode = DataGridViewSelectionMode.CellSelect;
            SqlCommand sc1 = new SqlCommand("select
Sale_Bill_Detail.SB_No,Sale_Bill_Detail.Item_Id,Inventory.Name,Sale_Bill_Detail.Qty,Sale_Bill_Detail.Rate,Sale_Bill_Detail.Price,Sale_Bill_Detail.Tax,Sale_Bill_Detail.Amount from Sale_Bill_Detail inner join Inventory on
Sale_Bill_Detail.Item_Id=Inventory.Item_Id where Sale_Bill_Detail.SB_No=' " +
comboBox3.SelectedValue.ToString() + "'", c.cn);
            SqlDataAdapter adp1 = new SqlDataAdapter(sc1);
            DataSet ds1 = new DataSet();
            adp1.Fill(ds1, "dt1");
            dgvsbd.DataSource = ds1.Tables["dt1"];
        }
    }

    private void comboBox3_TextChanged(object sender, EventArgs e)
    {
        textBox2.Text = "search...";
        dgvsb.SelectionMode = DataGridViewSelectionMode.CellSelect;
        SqlCommand sc = new SqlCommand("select * from Sale_Bill", c.cn);
        SqlDataAdapter adp = new SqlDataAdapter(sc);
    }
}

```

```

        DataSet ds = new DataSet();
        adp.Fill(ds, "dt");
        dgvsb.DataSource = ds.Tables["dt"];

        dgvsbd.SelectionMode = DataGridViewSelectionMode.CellSelect;
        SqlCommand sc1 = new SqlCommand("select
Sale_Bill_Detail.SB_No,Sale_Bill_Detail.Item_Id,Inventory.Name,Sale_Bill_Detail.Qty,Sale_Bill_Detail.Rate,Sale_Bill_Detail.Price,Sale_Bill_Detail.Tax,Sale_Bill_Detail.Amount from Sale_Bill_Detail inner join Inventory on
Sale_Bill_Detail.Item_Id=Inventory.Item_Id", c.cn);
        SqlDataAdapter adp1 = new SqlDataAdapter(sc1);
        DataSet ds1 = new DataSet();
        adp1.Fill(ds1, "dt1");
        dgvsbd.DataSource = ds1.Tables["dt1"];
    }

    private void textBox2_KeyUp(object sender, KeyEventArgs e)
    {
        if (comboBox3.Text == "Select" || comboBox3.Text == "")
        {
            SqlDataAdapter sda = new SqlDataAdapter("select
Sale_Bill_Detail.SB_No,Sale_Bill_Detail.Item_Id,Inventory.Name,Sale_Bill_Detail.Qty,Sale_Bill_Detail.Rate,Sale_Bill_Detail.Price,Sale_Bill_Detail.Tax,Sale_Bill_Detail.Amount from Sale_Bill_Detail inner join Inventory on
Sale_Bill_Detail.Item_Id=Inventory.Item_Id where Inventory.Name like '" +
textBox2.Text + "%'", c.cn);
            DataSet ds = new DataSet();
            sda.Fill(ds, "dt");
            dgvsbd.DataSource = ds.Tables["dt"];
        }
        else
        {
            SqlDataAdapter sda = new SqlDataAdapter("select
Sale_Bill_Detail.SB_No,Sale_Bill_Detail.Item_Id,Inventory.Name,Sale_Bill_Detail.Qty,Sale_Bill_Detail.Rate,Sale_Bill_Detail.Price,Sale_Bill_Detail.Tax,Sale_Bill_Detail.Amount from Sale_Bill_Detail inner join Inventory on
Sale_Bill_Detail.Item_Id=Inventory.Item_Id where Inventory.Name like '" +
textBox2.Text + "%' and
Sale_Bill_Detail.SB_No='"+comboBox3.SelectedValue.ToString()+"'", c.cn);
            DataSet ds = new DataSet();
            sda.Fill(ds, "dt");
            dgvsbd.DataSource = ds.Tables["dt"];
        }
    }

    private void textBox2_Leave(object sender, EventArgs e)
    {
        if (textBox2.Text == "")
            textBox2.Text = "search...";
        textBox2.ForeColor = Color.Silver;
    }

    private void textBox2_Click(object sender, EventArgs e)
    {
        if (textBox2.Text == "search...")
            textBox2.Text = "";
        textBox2.ForeColor = Color.FromArgb(20, 168, 0);
    }

```

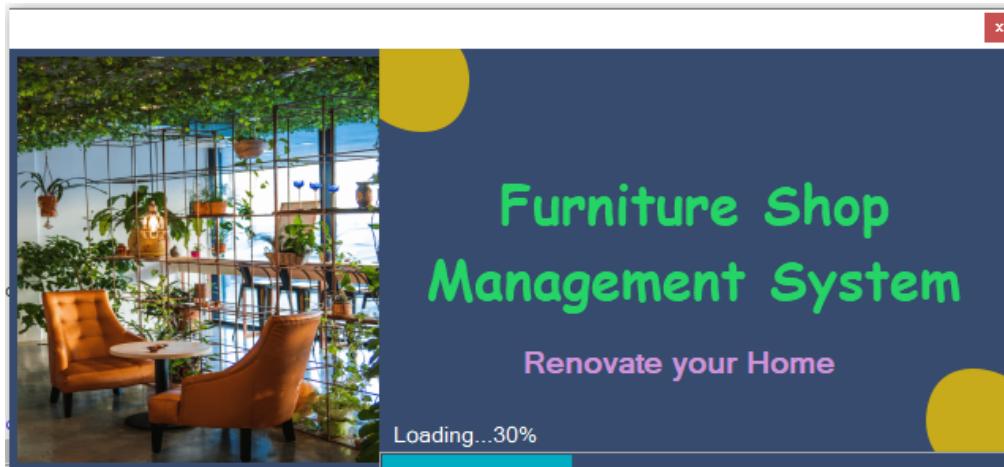
```

        }
    }
}

```

User Manual:

Start-up-Screen:



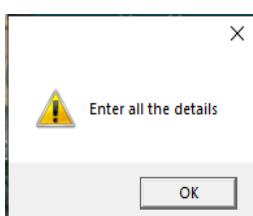
Creating Admin:

First Name	<input type="text"/>
Last Name	<input type="text"/>
Email	<input type="text"/>
User Name	admin
Password	<input type="password"/>
Retype Password	<input type="password"/>
Security Question	Select
Answer	<input type="text"/>
<input type="button" value="Clear"/> <input type="button" value="Sign Up"/>	

First Name	abc
Last Name	efg
Email	abc
User Name	admin
Password	aaaaaaaa
Retype Password	*****
Security Question	Select
Answer	<input type="text"/>
<input type="button" value="Clear"/> <input type="button" value="Sign Up"/>	

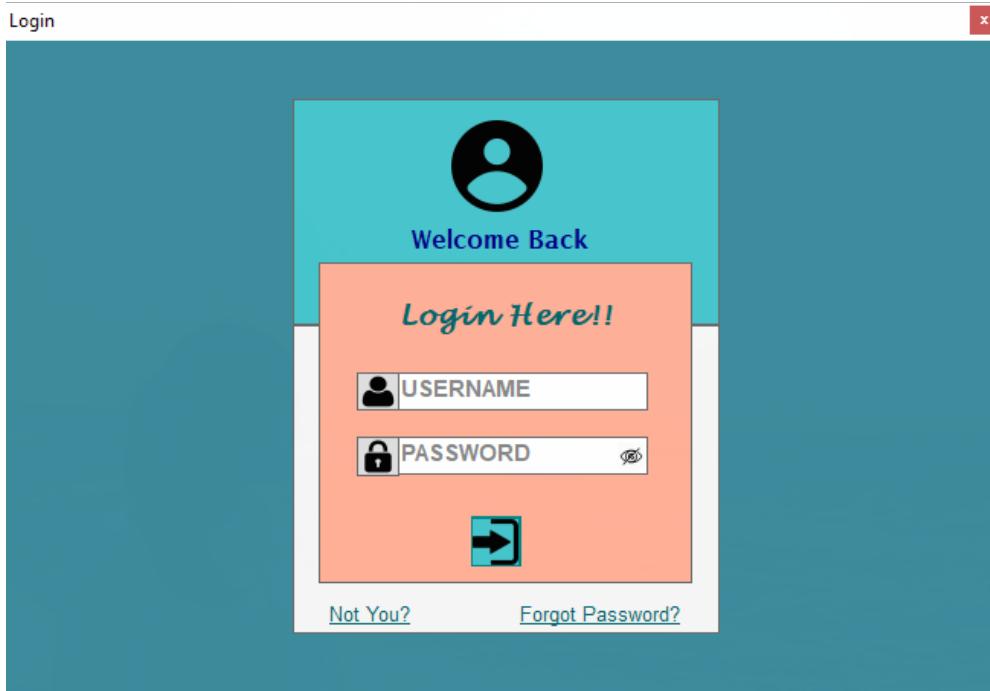
It has 3 validations:

- Enter all the details
- Invalid format of Email
- Password is matched or not matched



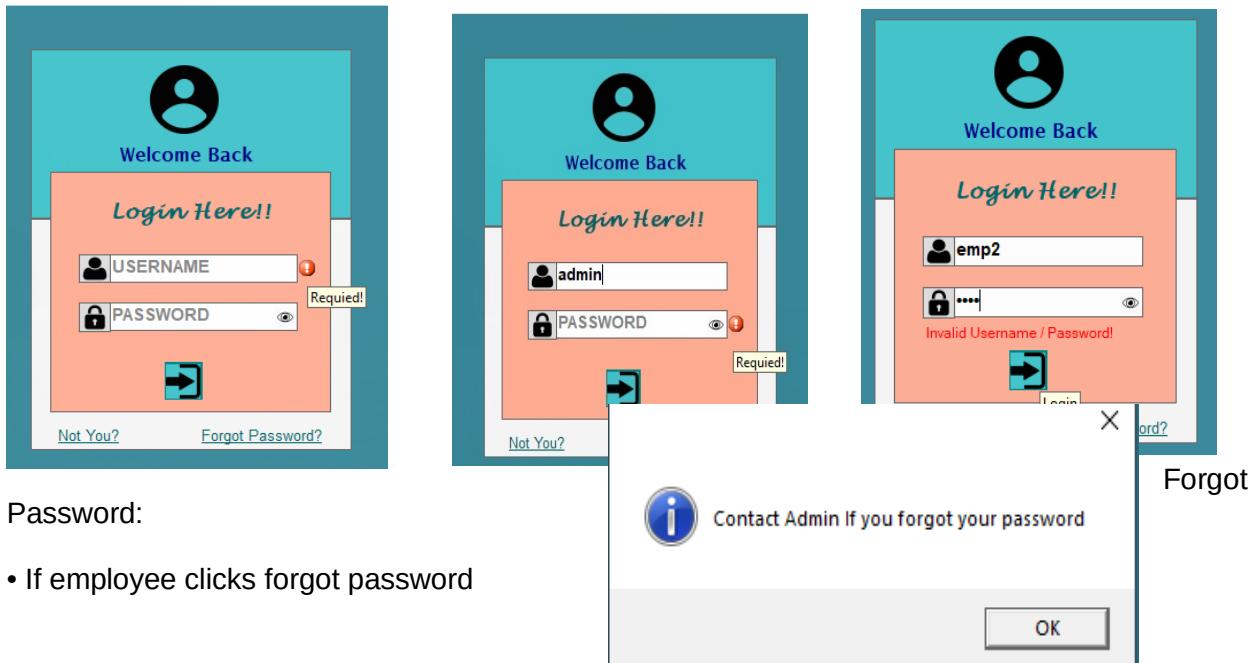
First Name	abc
Last Name	efg
Email	abc
User Name	admin
Password	aaaaaaaa
Retype Password	*****
Security Question	In what country were you born
Answer	India
<input type="button" value="Clear"/> <input type="button" value="Sign Up"/>	

Login:



Validation to be followed when entering values to controls in login form:

When we enter correct password and username the controls moves to the next page.
If not entered properly validation Controls light up.



Password:

- If employee clicks forgot password

- For admin

Forgot Password

Forgot Password?

Enter the answer to the security question you set during sign up

What is your work address ?

Confirm

[Forgot your security answer?](#)

Reset Password

Forgot Password?

Enter the answer to the security question you set during sign up

What is your work address ?

Confirm

[Forgot your security answer?](#)

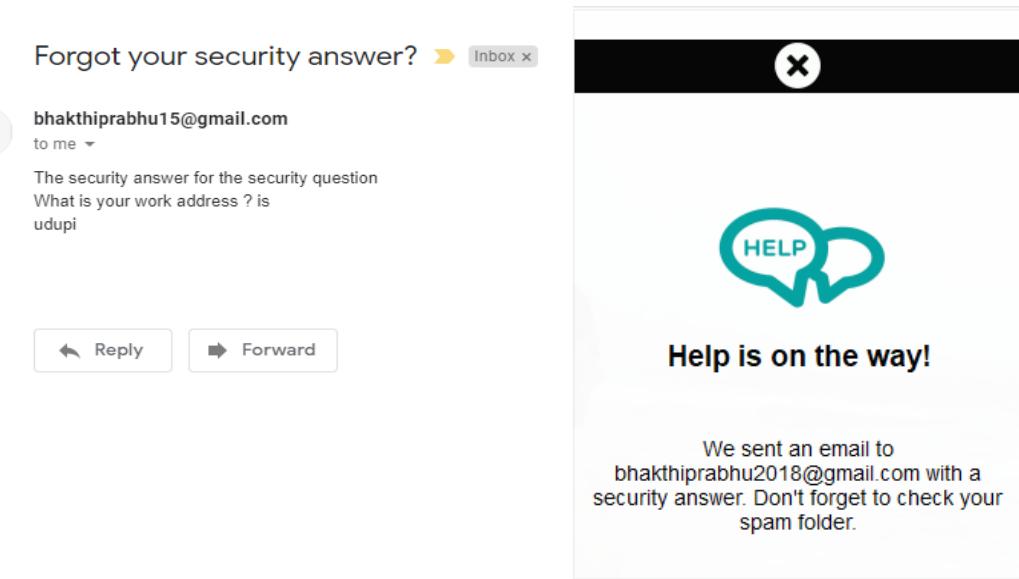
Reset Password

Enter your new password

New Password :

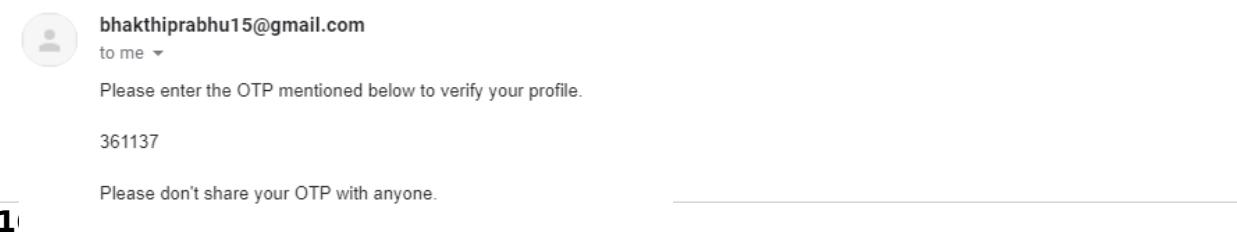
Confirm

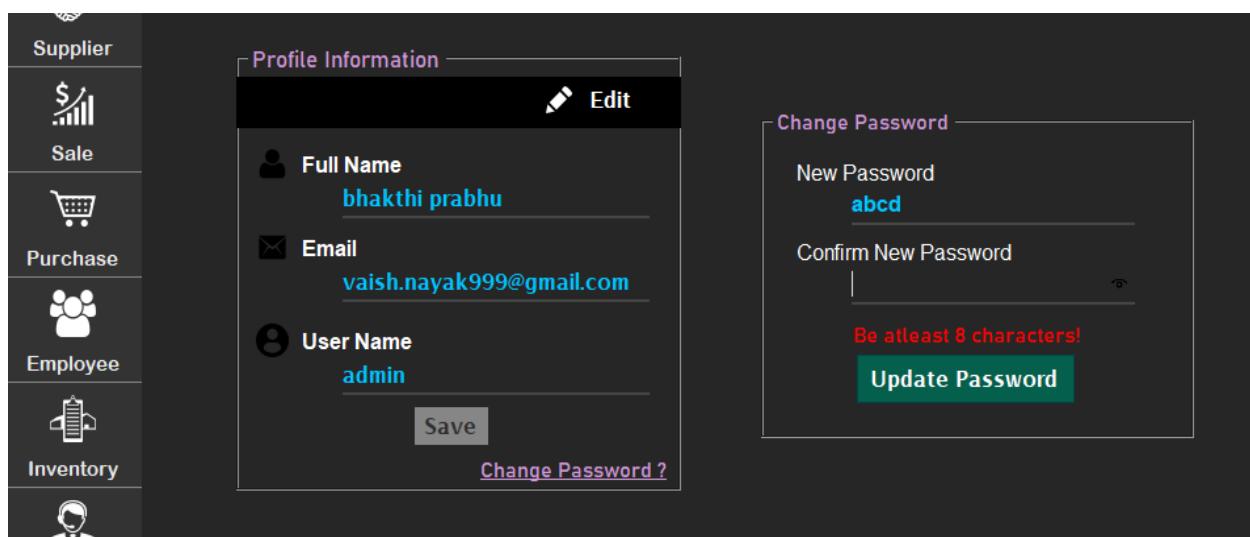
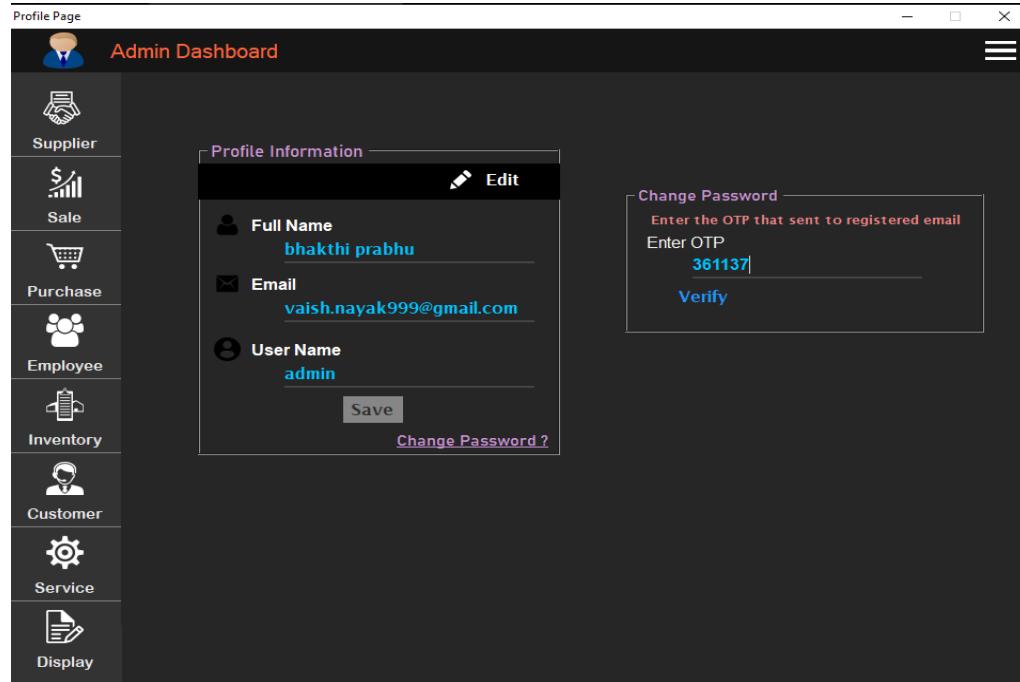
- If the admin forgot the security question then otp is sent to the mailed of the admin, then the password is reset._



When the admin wants to change password manually, an otp is sent for verification to the mail-id of the admin.

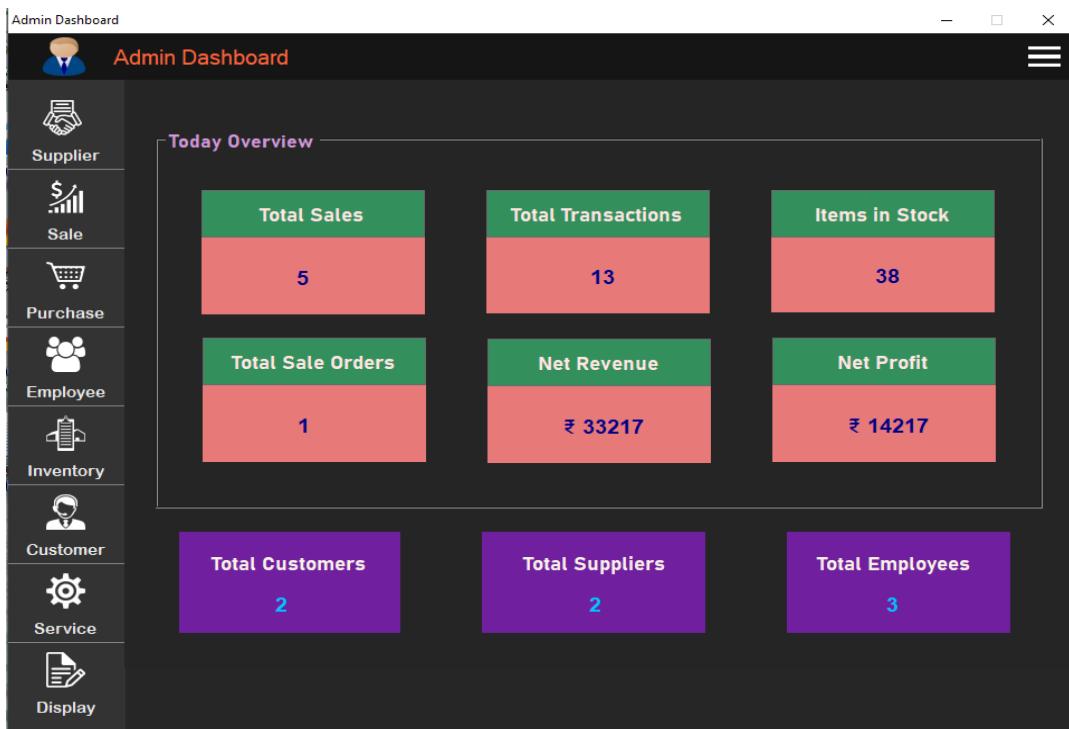
Verify Profile - OTP ➤ **Inbox**



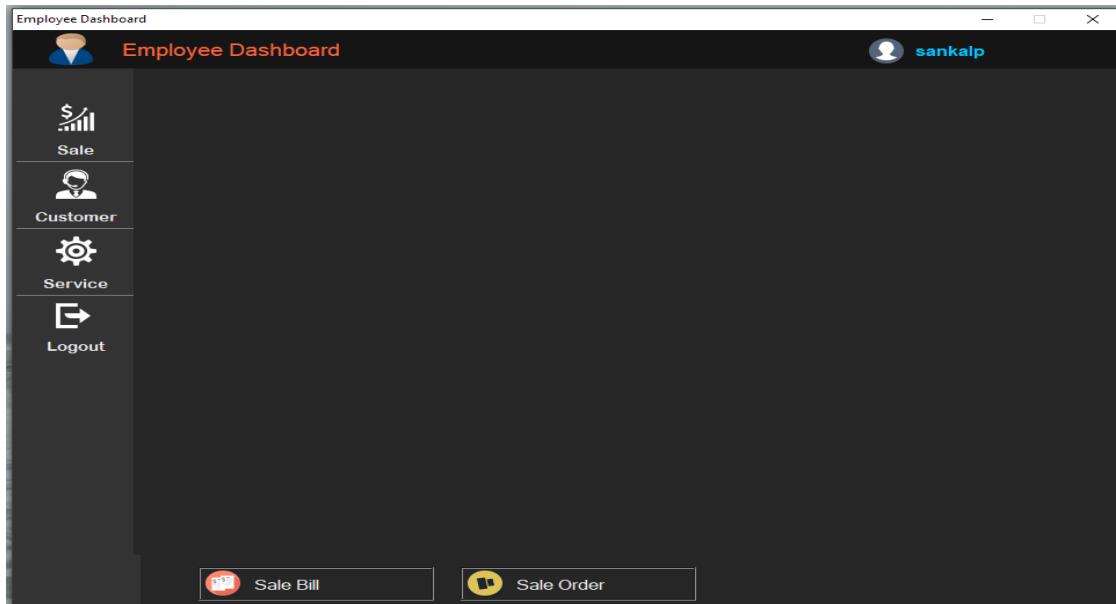


Dashboard:

- Admin Dashboard: (After logging in successfully)



- Employee Dashboard: (After logging in successfully)



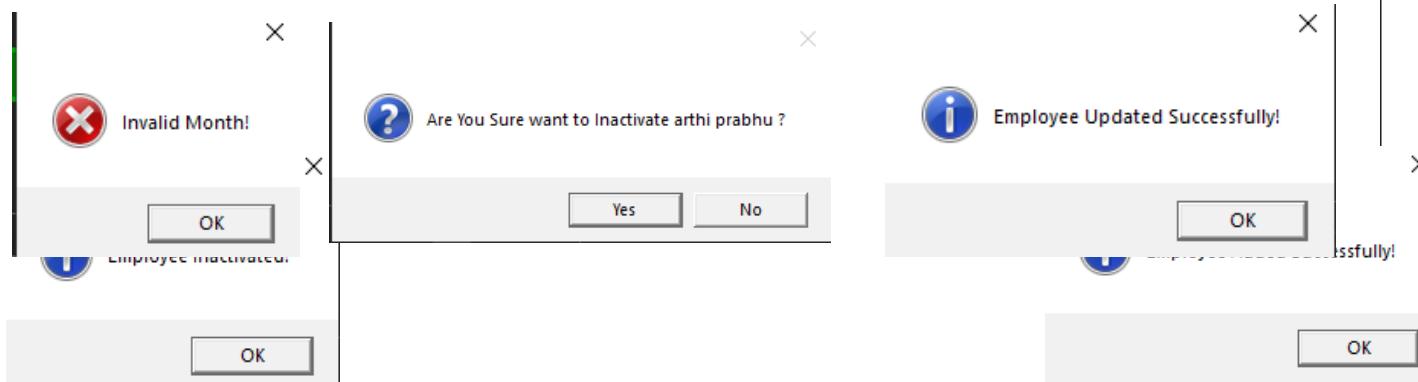
Create Employee (Admin rights):

Employee Page - Add New

Admin Dashboard

	Full Name _____	Gender <input type="radio"/> Female <input type="radio"/> Male <input type="radio"/> Others																																								
	Date of Birth _____	Contact No. _____																																								
	Age _____	Email ID _____																																								
	Address _____	Joined Date <input type="date" value="06 Oct 2021"/>																																								
	Designation _____	Basic Pay _____																																								
	Buttons: Register Update Delete Reset																																									
	Employee Name / ID <input type="text" value="search..."/> Filter Sort By																																									
	<table border="1"> <thead> <tr> <th>Emp_Id</th> <th>Name</th> <th>Dob</th> <th>Gender</th> <th>Age</th> <th>Contact_No</th> <th>Address</th> <th>Email</th> </tr> </thead> <tbody> <tr> <td>emp1</td> <td>sankalp</td> <td>06-04-2000</td> <td>Male</td> <td>21</td> <td>9988776655</td> <td>udipi</td> <td>vaish.na</td> </tr> <tr> <td>emp2</td> <td>sri</td> <td>06-04-2000</td> <td>Male</td> <td>21</td> <td>9445678944</td> <td>barkur</td> <td>vaish.na</td> </tr> <tr> <td>emp3</td> <td>varsha</td> <td>08-09-1998</td> <td>Female</td> <td>23</td> <td>9866454243</td> <td>malpe</td> <td>tyuiog@gr</td> </tr> <tr> <td>emp4</td> <td>qwerty</td> <td>09-09-1999</td> <td>Male</td> <td>22</td> <td>9887867574</td> <td>asdfg</td> <td>asdfg@g</td> </tr> </tbody> </table> <p>*** Click Above Cell Content to Edit or Delete Employee Details ***</p>		Emp_Id	Name	Dob	Gender	Age	Contact_No	Address	Email	emp1	sankalp	06-04-2000	Male	21	9988776655	udipi	vaish.na	emp2	sri	06-04-2000	Male	21	9445678944	barkur	vaish.na	emp3	varsha	08-09-1998	Female	23	9866454243	malpe	tyuiog@gr	emp4	qwerty	09-09-1999	Male	22	9887867574	asdfg	asdfg@g
Emp_Id	Name	Dob	Gender	Age	Contact_No	Address	Email																																			
emp1	sankalp	06-04-2000	Male	21	9988776655	udipi	vaish.na																																			
emp2	sri	06-04-2000	Male	21	9445678944	barkur	vaish.na																																			
emp3	varsha	08-09-1998	Female	23	9866454243	malpe	tyuiog@gr																																			
emp4	qwerty	09-09-1999	Male	22	9887867574	asdfg	asdfg@g																																			

Validations to be followed when entering values to controls in Employee option:



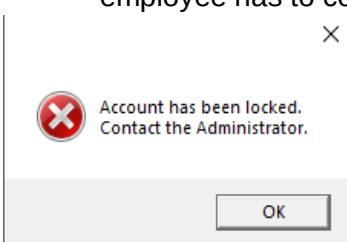
Admin Dashboard

	Full Name <input type="text" value="arthi prabhu"/>	Gender <input type="radio"/> Female <input type="radio"/> Male <input type="radio"/> Others
	Date of Birth <input type="text" value="08-04-1995"/>	Contact No. <input type="text" value="2235467876"/> * Add
	Age <input type="text" value="26"/>	Email ID <input type="text" value="arthi"/> * Enter valid mobile number

Admin Dashboard

	Full Name <input type="text" value="arthi prabhu"/>	Gender <input type="radio"/> Female <input type="radio"/> Male <input type="radio"/> Others
	Date of Birth <input type="text" value="86-58-9776"/> * Enter valid birth date in format dd/mm/yyyy	Contact No. <input type="text" value="9589800990"/>
	Age <input type="text" value="21"/>	Email ID <input type="text" value="aaa@gmail.com"/>
	Address <input type="text" value="asdas"/>	Joined Date <input type="date" value="06 Oct 2024"/>
	Buttons: Add New Manage User	

- If the employee types wrong password thrice while login, the account will be locked. The employee has to contact admin to unlock his profile.



Validations to be followed when entering values to Controls in Admin Dashboard.

- When we click lock or unlock and reset Password without selecting the Employee ID.

The screenshot shows the Admin Dashboard interface. On the left, there's a sidebar with icons for Supplier, Sale, Purchase, Employee, and Inventory. The main area has two sections: "Employee Password Reset" and "Lock/Unlock". In the "Lock/Unlock" section, there's a dropdown menu labeled "Select Employee ID to Lock Employee" with an error message "Please Select Employee Id". Below it are "Lock" and "Reset" buttons. To the right, there's another section titled "Employee Password Reset" with a dropdown menu labeled "Select Employee ID to Reset the Password" and a "Reset" button. On the right side of the screen, there are two validation messages: "Employee unlocked" with an "OK" button and "Password Changed!" with an "OK" button.

Admin can block the employee profile manually too.

This screenshot shows the Admin Dashboard with the "Employee" section selected in the sidebar. In the main area, the "Lock/Unlock" section is active. The dropdown menu "Select Employee ID to Lock Employee" now has "emp1" selected. The "Lock" button is highlighted in green, indicating it's active. To the right, the "Employee Password Reset" section is visible with its own dropdown menu. On the right side, there are validation messages: "Employee unlocked" with an "OK" button and "Password Changed!" with an "OK" button.

Attendance details can be only viewed by admin only:

Admin Dashboard

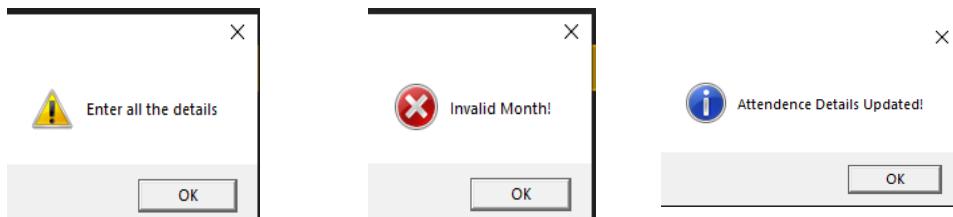
The Admin Dashboard interface includes a sidebar with icons for Supplier, Sale, Purchase, Employee, Inventory, Customer, Service, and Display. The main area shows employee attendance details with fields for Employee ID, Month, Year, Full Name, Leave Taken, and Leave Given. Buttons for Update, Save, and Reset are present. A search bar for Employee ID and a sort dropdown are also available. A table displays attendance data for employees emp6 and emp7 across September and August 2021.

Emp_Id	Month	Year	Total_days	Leave_taken	Leave_given	Extra_leave
emp6	September	2021	30	2	2	0
emp6	August	2021	31	6	4	2

*** Click Above Cell Content to Edit Attendance Details ***

Validations to be followed when entering values to Controls in Attendance details are:

- When all details are not entered
- When we choose wrong month from the drop down list.



- Customer can be registered through both employee and admin logins.

Customer Page

Admin Dashboard

Supplier Sale Purchase Employee Inventory Customer Service Display

Full Name _____ **Add**

Address _____ **Update**

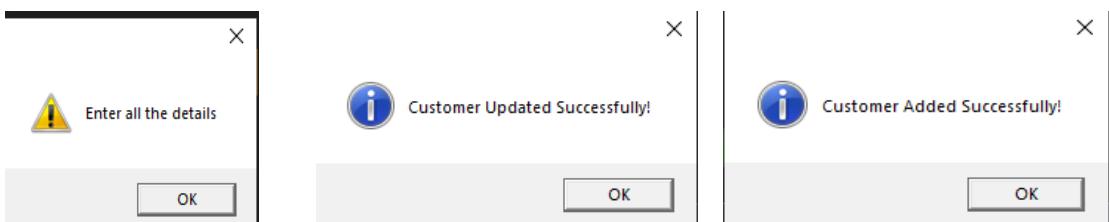
Contact No. _____

Email ID _____ **Reset**

Contact No. **Sort By**

Name	Contact_No	Address	Email
alfiya	9944455543	yermall	qwerty@gmail.com
anvitha	9977554433	karavali	jklp@gmail.com

Validations to be followed when entering values to Controls in customer registration are:



- When all details are not entered
- When proper format of email is not entered.
- When contact number is not written in proper format.

Supplier Sale Purchase

Full Name **pooja** **Add**

Address **udipi**

Contact No. **9338538499** **Update**

Email ID **vaish** **Reset**

Enter valid format email

Customer Search (contact number is used to search) in the table:

Name	Contact_No	Address	Email
pooja	9338538499	udipi	vaish.nayak999@gmail.c...

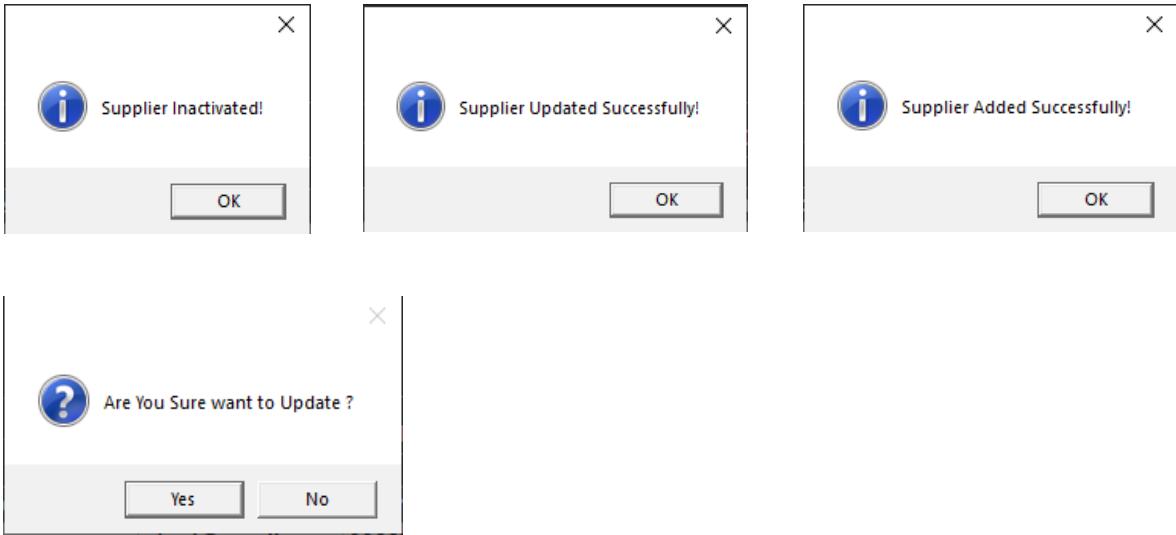
Customer is sorted by address and name:

Name	Contact_No	Address	Email
anvitha	9977554433	karavali	jklp@gmail.com
pooja	9338538499	udipi	vaish.nayak999@gmail.c...
alfiya	9944455543	yermall	qwerty@gmail.com

Admin Dashboard

Sup_Id	Name	Address	Email	Contact_No	Status
sup1	varshini	malpe	abcd@gmail.com	9988772233	Inactive
sup2	sushmitha	indralli	qwerty@gmail....	9933445566	Inactive
sup3	tanvi	pangala	asdfg@gmail.c...	9933445567	Inactive
sup4	vaish	udipi	vaish.nayak999...	9986775635	Active
sup5	bhakthi	udipi	vaish.nayak999...	8088308112	Active

Validations to be followed when entering values to Controls in the supplier page:



Validations are:

- When all details are not entered.
- When proper format of email is not entered.
- When contact number is not written in proper format.

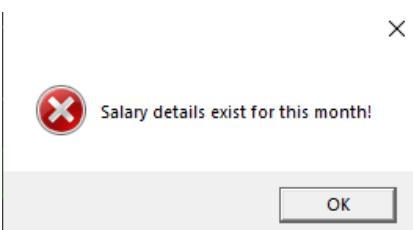
The screenshot shows the Admin Dashboard interface. On the left, there's a sidebar with icons for Supplier, Sale, and Address. The main area has tabs for 'Admin Dashboard' and 'Supplier'. Under the 'Supplier' tab, there are fields for 'Full Name' (pooja), 'Email ID' (aish.nayak), and 'Contact No.' (2343425349). The 'Email ID' field has a tooltip 'Enter valid format email'. The 'Contact No.' field has a tooltip 'Enter valid mobile number'. A green 'Add' button is next to the name field, and a blue 'Update' button is next to the email field. Below these, there are 'Delete' and 'Reset' buttons.

Employee Salary page:

Admin Dashboard

Pay_Id	Emp_Id	Date	Year	Month	Tax	HRA	PF
pay1	emp6	07-10-2021	2021	September	1800.00	4800.00	1440.00
pay2	emp6	07-10-2021	2021	August	1683.87	4490.32	1347.10

Message displayed when payslip is already created for employee:



When salary is successfully generated:



Payslip print:

Furniture Shop Management System Renovate Your Home			
<u>Pay Slip for August - 2021</u>			
P A Y S		08-10-2021 21:55:32	
Employee Name	sankalp	Contact No.	9988776655
Employee ID	emp1	Gender	Male
Designation	engineer	Extra Leave	2
Date Of Join	16-07-2021	Total Working Days	31
Earnings		Deductions	
Basic Pay	20000.00	Provident Fund	2245.16
H.R.A	7483.87	Total Tax	2806.45
Total Earnings	27483.87	Total Deductions	5051.61
Extra Leave Salary Deduction		1290.32	
Net Salary		21141.94	
Amount In Words : TWENTY ONE THOUSAND ONE HUNDRED AND FORTY ONE RUPEES NINETY FOUR PAISA ONLY			
THIS IS COMPUTER GENERATED PRINTOUT WHICH REQUIRES NO SIGNATURE.			Print Bill

Sales Order:

Sale Page - Order

Admin Dashboard

Select Item Here :

Item_Id	Name	Qty	Material	Color	Category
item4	comptable	0	fiber	black	Console table

Select Customer Here :

Name	Contact_No	Address
pooja	9338538499	udupi
alfiya	9944455543	yermall
anvitha	9977554433	karavali

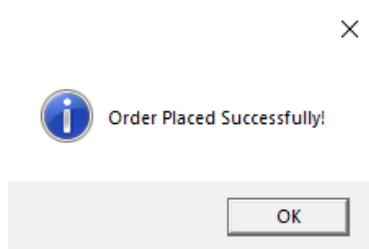
Place Order **Cancel**

Contact No. / Item Name

SO_No	Date	Contact_No	Item_Id	Name
so1	18-09-2021	9977554433	item3	chair

Sale Bill **Sale Order**

When details are successfully added in sales order page:



Sales bill can be controlled by both employee and the admin.

Sale Page - Bill

Admin Dashboard

Supplier	Select Item Here :					Select Customer Here :		
Sale	Item Name	search...				Contact No.	search...	
Purchase	Item_Id	Name	Qty	Price	Material	Color		
Employee	item1	table	12	3000.00	wood	brown		
Inventory	item2	bed	6	20000.00	foam	pink		
Customer	item3	chair	10	500.00	fiber	blue		
Service	item4	comptable	7	2500.00	fiber	black		
Display								

Date Save

Quantity

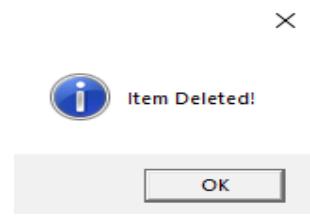
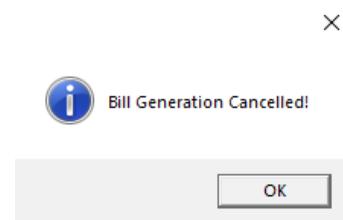
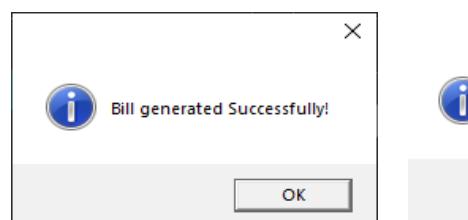
Rate

Warranty
In years

Add Total Amount

Discount Net Amount Generate Bill Cancel

Sale Bill Sale Order



Validations are to be followed when entering values to Controls in the Sales page:

- When quantity is not specified and the given quantity is greater than the stock. And When rate is not specified.

Purchase

Employee	Select Item Here :					Select Customer Here :		
Inventory	Item Name	search...				Contact No.	search...	
Customer	Item_Id	Name	Qty	Price	Material	Color		
	item1	table	7					
	item2	bed	0					
	item3	chair	1.0					
	item4	comptable						

Date Save

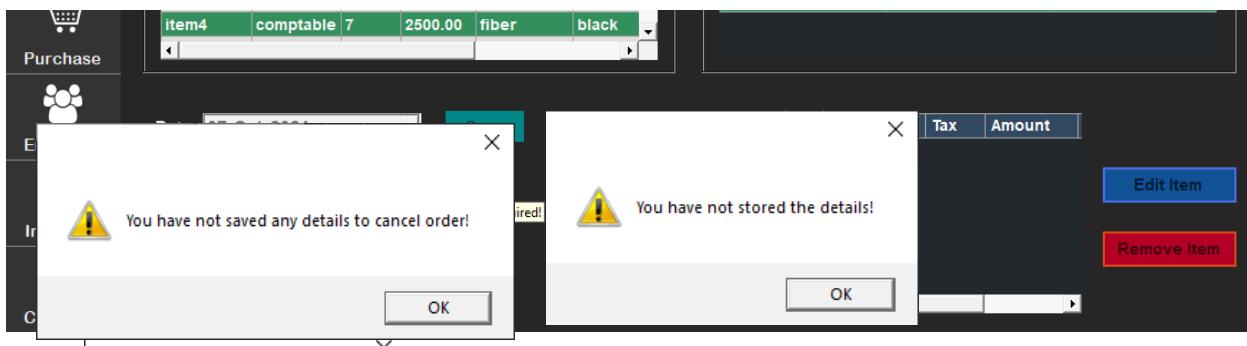
Quantity

Rate Required!

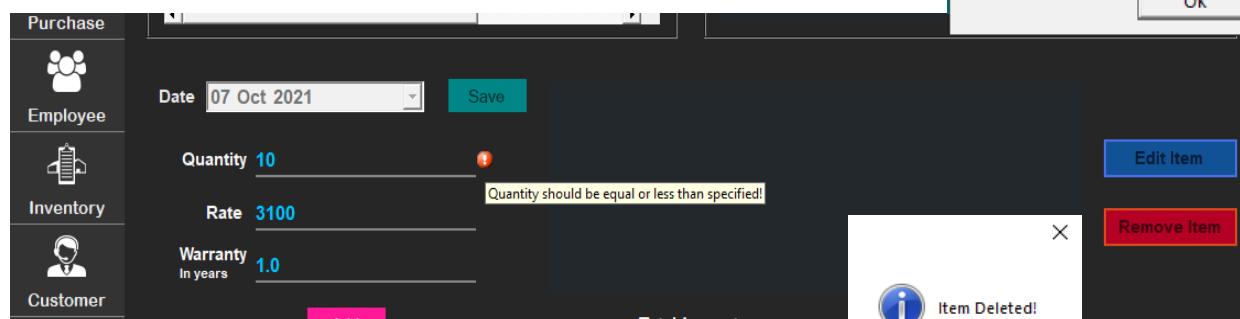
Warranty
In years

Add Total Amount

Edit Item Remove Item



please select Customer



Sales Invoice:

Furniture Shop Management System
Renovate Your Home

Sale Invoice						
Invoice No. : SB8	Invoice Given By : admin					
Invoice To : anvitha	Invoiced Date : 07-10-2021 13:03:33					
Contact No. : 9977554433						
Detail of Product	Warranty in Years	Quantity	Rate	Price	Tax	Amount
chair	1.0	1	750.00	750.00	135.00	885.00

THANK YOU !! VISIT AGAIN
Goods Once Sold Will Not Be Exchanged or Taken Back.

Total Amount : 885.00
Discount : 0.00
Grand Total : 885.00

Amount In Words : EIGHT HUNDRED AND EIGHTY FIVE RUPEES ZERO PAISE ONLY

THIS IS COMPUTER GENERATED PRINTOUT WHICH REQUIRES NO SIGNATURE.

Purchase Order:

Purchase Page - Order

Admin Dashboard

Item_Id	Name	Qty	Material	Color	Category
item1	table	12	wood	brown	Coffee table
item2	bed	6	foam	pink	Bed
item3	chair	10	fiber	blue	Chair
item4	comptable	7	fiber	black	Console
item5	lamp	9	cotton	blue	Nightstand

Sup_Id	Name	Contact_No
sup4	vaish	9986775635
sup5	bhakthi	8088308112

Date: 07 Oct 2021

Quantity:

When Invalid Quantity is given:

PO_No	Sup_Id	Item_Id	Name	Qty
po18	sup5	item5	lamp	10000000

Date: 08 Oct 2021

Quantity: 1000000000000000

When the Purchase order is placed the supplier gets a mail to his mail-id regarding the items orders.

Purchase Order ➔ Inbox



bhakthiprabhu15@gmail.com
to me ▾

Supplier Name : bhakthi

PO Id : po17

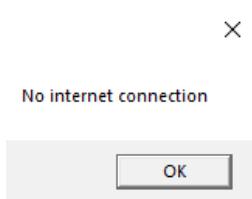
Date : 07-10-2021 12:44:16

Item ID	Name	Quantity
item5	lamp	1

Furniture Shop Management System

Renovate Your Home

If the mail is not sent:



Purchase Bill:

Purchase Page - Bill

Admin Dashboard

Purchase Order No. Date Save

Supplier Id _____

Supplier Name _____

Quantity _____

Price _____

Warranty In years _____

Add Edit Item Remove Item

Total Amount _____

Save Cancel

Purchase Order Purchase Bill Purchase Return

Purchase Page - Bill

Admin Dashboard

Purchase Order No. Date Save

Supplier Id _____

Supplier Name _____

Quantity _____

Price _____

Warranty In years _____

Add Edit Item Remove Item

Total Amount _____

Save Cancel

Purchase Order Purchase Bill Purchase Return

Purchase Page - Bill

Admin Dashboard

Purchase Order No. Date Save

Supplier Id _____
 Supplier Name _____

Quantity _____
 Price _____
 Warranty In years _____

Add

Total Amount _____

Supplier

Sale

Purchase

Employee

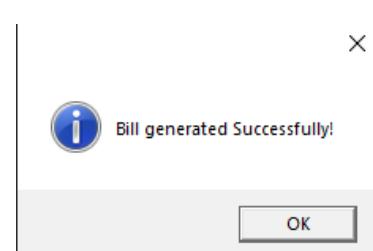
Inventory

Customer

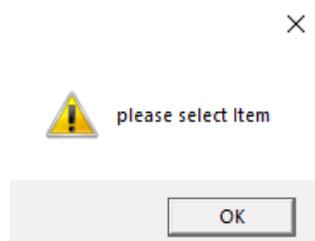
Service

Display

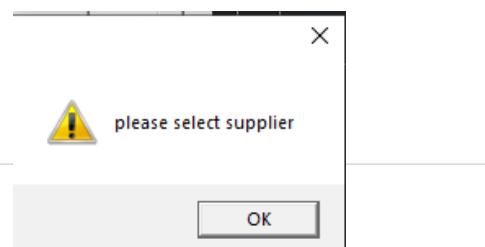
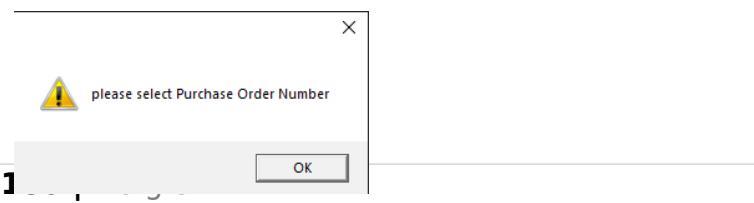
 Item Deleted!



If we click add without selecting item:



If we click save without selecting order number: If we click save without selecting supplier:



Purchase Return:

Purchase Page - Return

Admin Dashboard

- Supplier**
- Sale**
- Purchase**
- Employee**
- Inventory**
- Customer**
- Service**
- Display**

Purchase Bill No. Date Save

Supplier Id _____ Supplier Name _____

Quantity _____



Purchase Returned Successfully!



Purchase return Cancelled!

When the item is selected 2 times:



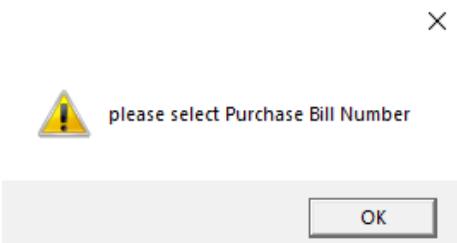
You already added this item.
Please edititem if you want to make changes.



please select item

If we select add without selecting item:

If we click save without selecting Purchase Bill Number:



Inventory page is handled by Admin only.

Inventory Page

Admin Dashboard

Supplier Sale Purchase Employee Inventory Customer Service Display

Item Name _____ Date 06 Oct 2021
Quantity 0 Category Select
Price 0.0 Color _____
Warranty 0.0 Material _____

Add Update Delete Reset

Item Name search... Filter By Active Inactive SortBy

Item_Id	Name	Qty	Material	Color	Price	Date	Warranty	Category	Status
item1	table	12	wood	brown	3000.00	16-09-2021	2.0	Coffee table	Active
item2	bed	6	foam	pink	20000.00	16-09-2021	5.0	Bed	Active
item3	chair	10	fiber	blue	500.00	17-09-2021	1.0	Chair	Active
item4	comptable	7	fiber	black	2500.00	18-09-2021	1.0	Console table	Active
item5	lamp	9	cotton	blue	400.00	20-09-2021	2.0	Nightstand	Active

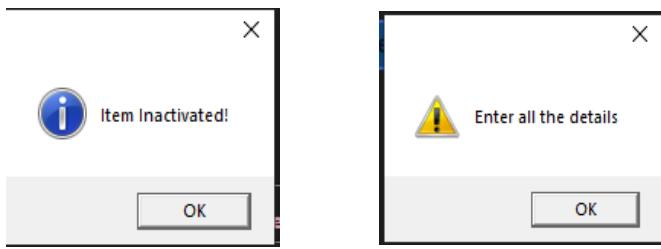
*** Click Above Cell Content to Edit or Delete Inventory Details ***

Inventory Updated Successfully!

Are You Sure want to Inactivate sofa ?

Item Added Successfully!

OK Yes No OK



Searching the items from the list:

Employee	Inventory	Customer	Service																				
	<p>Item Name <input type="text" value="ta"/> <input type="button" value="Search"/></p> <p>Filter By <input type="checkbox"/> Active <input type="checkbox"/> Inactive</p> <p>Sort By <input type="button" value="▼"/></p> <table border="1"><thead><tr><th>Item_Id</th><th>Name</th><th>Qty</th><th>Material</th><th>Color</th><th>Price</th><th>Date</th><th>Warranty</th><th>Category</th><th>Status</th></tr></thead><tbody><tr><td>item1</td><td>table</td><td>12</td><td>wood</td><td>brown</td><td>3000.00</td><td>16-09-2021</td><td>2.0</td><td>Coffee table</td><td>Active</td></tr></tbody></table>	Item_Id	Name	Qty	Material	Color	Price	Date	Warranty	Category	Status	item1	table	12	wood	brown	3000.00	16-09-2021	2.0	Coffee table	Active		
Item_Id	Name	Qty	Material	Color	Price	Date	Warranty	Category	Status														
item1	table	12	wood	brown	3000.00	16-09-2021	2.0	Coffee table	Active														

Service page is operated by both employee and the admin:

Service Page

Admin Dashboard


Supplier

Sale

Purchase

Employee

Inventory

Customer

Service

Display

Sale Bill No.

Service Type

Item Id

Date

Total Amount

Tax

Remark

Grand Total

Update
Save
Reset
Delete

Service No.

Search

SR_No	Date	SB_No	Item_Id	Service_Type	Total_Amount	Tax	Grand_Total	Remark
sr1	18-09-2021	sb3	item1	Non-Warranty	1000.00	180.00	1180.00	polishing
sr2	18-09-2021	sb3	item4	Warranty	0.00	0.00	0.00	oil change
sr3	20-09-2021	sb4	item1	Non-Warranty	1000.00	180.00	1180.00	polishing
sr4	20-09-2021	sb2	item4	Warranty	0.00	0.00	0.00	oil change

<
>

*** Click Above Cell Content to Edit or Delete Service Details ***

Service Without warranty:

Service Page

Admin Dashboard


Supplier

Sale

Purchase

Employee

Inventory

Customer

Service

Display

Sale Bill No.

Service Type

Item Id

Date

Total Amount

Tax

Remark

Grand Total

Update
Save
Reset
Delete

Service No.

Search

SR_No	Date	SB_No	Item_Id	Service_Type	Total_Amount	Tax	Grand_Total	Remark
sr1	18-09-2021	sb3	item1	Non-Warranty	1000.00	180.00	1180.00	polishing
sr2	18-09-2021	sb3	item4	Warranty	0.00	0.00	0.00	oil change
sr3	20-09-2021	sb4	item1	Non-Warranty	1000.00	180.00	1180.00	polishing
sr4	20-09-2021	sb2	item4	Warranty	0.00	0.00	0.00	oil change

<
>

*** Click Above Cell Content to Edit or Delete Service Details ***

Display option is only available for the admin.

The screenshot shows two instances of the Admin Dashboard interface. The left sidebar lists various modules: Supplier, Sale, Purchase, Employee, Inventory, Customer, Service, and Display. The 'Purchase Orders' module is currently selected, indicated by a blue background.

The main area displays a table for Purchase Orders. In the first instance (top), a dropdown menu labeled 'Purchase Order No.' contains the option 'Select'. In the second instance (bottom), the dropdown shows 'po9' and the table below it displays the following data:

PO_No	Date	Sup_Id	Status
po9	29-09-2021	sup5	Ordered

Below the table, there is a search input field labeled 'Item Name' containing 'ch' and a small green search icon. A partial table is visible, showing:

PO_No	Item_Id	Name	Qty
po9	item3	chair	10

Search option in display page.

TESTING AND IMPLEMENTATION:

TESTING PHASE

Testing is the process of detecting errors. Testing is a major party control used in software development. During testing, the program to be tested is executed with a set of test cases evaluated to determine if the program is performing as expected. Thus the goal of testing is to uncover the errors requirements, design and coding errors in the program.

Every time the program is executed to be tested. Therefore we have to execute the program before it gets to the donor with the specific intent of finding and removing all errors.

Psychology of testing:

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of phase is to detect errors that may be present in the program. Hence one should not start testing with the intent to show that a program doesn't work. Testing is the process of executing program with the intent of finding errors.

Testing Objectives:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time, starting formally, we can say, testing is a process of executing a program with the intent of finding errors.

A successful test is one that uncovers an as yet undiscovered error. A good test case is one that has a high probability of finding error, if it exists.

The tests are inadequate to detect possibly present errors. The software more or less confirms to the quality and reliable standards

System Testing: Software is only one element of a large computer based system. Ultimately, software is incorporated with other system element and a series of system integration and validation tests are conducted.

The various types of testing on the system are:

- Unit testing
- Integrated testing
- Validation testing
- Output testing
- User Acceptance testing

Unit Testing:

Unit testing focuses verification effort on the smallest unit of software that is the module. Using detailed design and the process specifications testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit testing before integrated testing begins.

In this project each service can be thought of a module. There are three basic modules. Giving different set of inputs has tested each module. When developing the modules works without any error. The inputs are validated when accepting from user. In this application developer tests the programs up as system. Software units in a system are the modules and routines that are assembled and integrated, to form a specific function. Unit testing is first done on modules, independent of one another to locate errors. This enables to detect errors, through these errors resulting from the interaction between modules are initially avoided.

Integrated Testing:

After the unit testing, we have to perform integration testing. The goal here is to see modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design hence the emphasis on testing modules interactions.

Validation testing:

At the culmination of the integration testing, the software was completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software validation began. Here we test the system in a manner that can be reasonably expected by the customer, the system was tested against system requirement specification.

Output testing:

After performing validation test the next phase is output test of the system, since no system could be useful if it does not produce the desired output in desired format. By considering the format of the report/output is generated or displayed and is tested. Here output format is generated or displayed and is a screen and other is on printed form.

User Acceptance Testing:

Acceptance test is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behavior of the system; the internal logic of program is not emphasized.

Test cases should be selected so that the largest number of attributes is an important part of the software developed. It is the process of finding errors and missing operations

and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

TEST CASE

Unit testing:

Test case: 1

Objectives: To change password (admin).

Test data:

Valid: Otp is generated to the admin's mail-id, then otp is verified and allows to write the new password and the same has to be written in the confirm password.

Invalid: If new password and old password doesn't match.

Output:

Valid: Allows changing the password.

Invalid: New password is not updated.

Result:

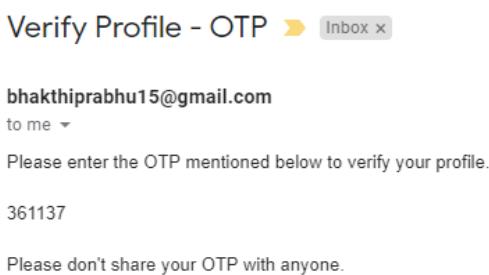
Valid: The password has been changed.

Invalid: The record is not updated to the database.

Conclusion: Both the valid and invalid results are tested.

Output matches with the desired result.

Hence Test case is successful



A screenshot of the Admin Dashboard interface. On the left, there is a sidebar with various menu items: Supplier, Sale, Purchase, Employee, Purchase, Inventory, Customer, Service, and Display. The main area shows "Profile Information" with fields for Full Name (bhakthi prabhu), Email (vaish.nayak999@gmail.com), and User Name (admin). An "Edit" button is visible above these fields. To the right, a "Change Password" dialog box is open, containing fields for "New Password" (abcd) and "Confirm New Password" (*****). Below these fields are "Update" and "Cancel" buttons. A success message "Password Changed" with an info icon is displayed at the bottom right. At the very bottom left, the page number "188" is visible.

Test case: 2

Objectives: Test for adding Employee Information

Test data:

Valid: All required fields are entered and in their correct format.

Invalid: Some fields are left blank or written incorrectly.

Output:

Valid: Allows records to be added to the database.

Invalid: The user is prompt with an error message.

Result:

Valid: Records will be saved in the database.

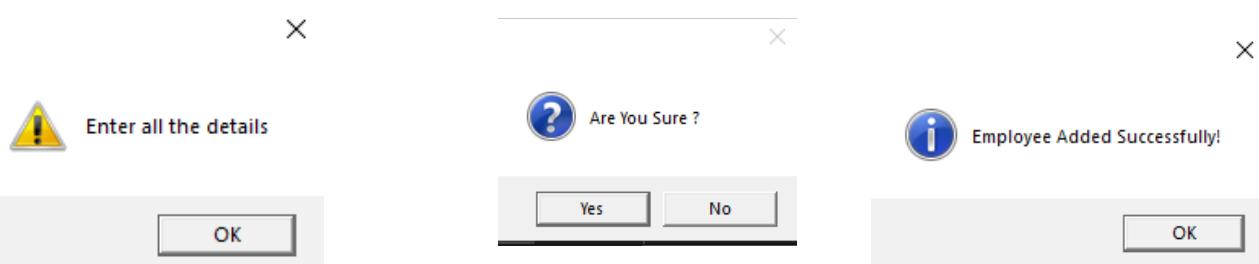
Invalid: The record is not updated to the database.

Conclusion: Both the valid and invalid results are tested and the record will be saved to the database.

The screenshot shows the Admin Dashboard interface. On the left, there is a sidebar with icons for Supplier, Sale, Purchase, Employee, Inventory, Customer, Service, and Display. The main area has input fields for Full Name (ram), Date of Birth (08-01-1995), Age (26), Address (kaup), Designation (manager), Gender (Male), Contact No. (9934573623), Email ID (ram@gmail.com), Joined Date (09 Oct 2021), and Basic Pay (10000). Below these fields are buttons for Register (green), Update (blue), Delete (red), and Reset (yellow). To the right of these buttons are three more buttons: Add New (blue), Manage User (red), and Salary (green). At the bottom, there is a table titled "Employee Name / ID" with columns: Emp_Id, Name, Dob, Gender, Age, Contact_No, Address, and Email. The table contains data for seven employees (emp1 to emp7). A search bar and filter options (Active, Inactive) are also present.

Emp_Id	Name	Dob	Gender	Age	Contact_No	Address	Email
emp1	sankalp	06-04-2000	Male	21	9988776655	udupi	vaish.na
emp2	sri	06-04-2000	Male	21	9445678944	barkur	vaish.na
emp3	varsha	08-09-1998	Female	23	9866454243	malpe	tyulo@gr
emp4	qwerty	09-09-1999	Male	22	9887867574	asdgf	asdgf@g
emp5	arthi prabh	08-04-1995	Female	26	8235467876	udupi	vaish.na
emp6	pooja prab...	03-05-1995	Female	26	9534534523	udupi	vaish.na
emp7	poorn	03-06-1990	Female	31	9335465636	udupi	vaish.na

*** Click Above Cell Content to Edit or Delete Employee Details ***



Test case: 3

Objectives: Test for adding customer information

Test Data:

Valid: All required fields are entered and in their correct format.

Invalid: Some fields are left blank or written incorrectly.

Output:

Valid: Allows records to be added to the database.

Invalid: The user is prompt with an error message.

Result:

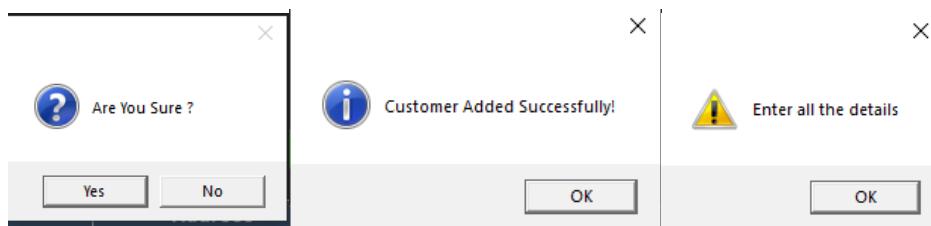
Valid: Records will be saved in the database.

Invalid: The record is not updated to the database.

Conclusion: Both the valid and invalid results are tested and the record will be saved to the database.

The screenshot shows the Admin Dashboard interface. On the left, there is a sidebar with icons and labels for Supplier, Sale, Purchase, Employee, Inventory, Customer, Service, and Display. The main area has four input fields: 'Full Name' (abcd), 'Address' (efgh), 'Contact No.' (9956756424), and 'Email ID' (qwerty@gmail.com). Below these are 'Add' (green button), 'Update' (blue button), and 'Reset' (orange button) buttons. A search bar labeled 'Contact No.' with placeholder 'search...' and a 'Sort By' dropdown are also present. At the bottom, a table displays customer data with columns: Name, Contact_No, Address, and Email. The table rows are: pooja (9338538499, udupi, vaish.nayak999@gmail.com...), alfiya (9944455543, yermall, qwerty@gmail.com), and anvitha (9977554433, karavali, jkip@gmail.com). A note at the bottom says: *** Click Above Cell Content to Edit Customer Details ***.

Name	Contact_No	Address	Email
pooja	9338538499	udipi	vaish.nayak999@gmail.c...
alfiya	9944455543	yermall	qwerty@gmail.com
anvitha	9977554433	karavali	jkip@gmail.com



Test case: 4

Objectives: Test for adding supplier information

Test Data:

Valid: All required fields are entered and in their correct format.

Invalid: Some fields are left blank or written incorrectly.

Output:

Valid: Allows records to be added to the database.

Invalid: The user is prompt with an error message.

Result:

Valid: Records will be saved.

Invalid: The record will not be saved.

Conclusion: Both the valid and invalid results are tested and the record will be saved to the database.

Admin Dashboard

Supplier

Sale

Purchase

Employee

Inventory

Customer

Service

Display

Full Name

Email ID

Contact No.

Address

Add

Update

Delete

Reset

Supplier Name / ID Filter By Active Inactive Sort By

Sup_Id	Name	Address	Email	Contact_No	Status
sup1	varshini	malpe	abcd@gmail.com	9988772233	Inactive
sup2	sushmitha	indralli	qwert@gmail...	9933445566	Inactive
sup3	tanvi	pangala	asdg@gmail.c...	9933445567	Inactive
sup4	vaish	udupi	bhaktiprabhu...	9986775635	Active
sup5	bhakthi	udupi	vaish.nayak999...	8088308112	Active
sup6	aarthi	udupi	vaish.nayak999...	9887654323	Inactive

*** Click Above Cell Content to Edit or Delete Supplier Details ***

Are You Sure ?

Supplier Added Successfully!

OK

Test case: 5

Objectives: Test for adding Item information.

Test Data:

Valid: All required fields are entered and in their correct format.

Invalid: Some fields are left blank or written incorrectly.

Output:

Valid: Allows records to be added to the database.

Invalid: The user is prompt with an error message.

X



Enter all the details

OK

Result:

Valid: Records will be saved.

Invalid: The record will not be saved.

Conclusion: Both the valid and invalid results are tested and the record will be saved to the database.

Admin Dashboard

Supplier

Sale

Purchase

Employee

Inventory

Customer

Service

Display

Item_Id	Name	Qty	Material	Color	Price	Date	Warranty	Category	Status
Item1	table	12	wood	brown	3000.00	16-09-2021	2.0	Coffee table	Active
item2	bed	6	foam	pink	20000.00	16-09-2021	5.0	Bed	Active
item3	chair	9	fiber	blue	500.00	17-09-2021	1.0	Chair	Active
item4	comptable	0	fiber	black	2500.00	18-09-2021	1.0	Console table	Inactive
item5	lamp	9	cotton	blue	400.00	20-09-2021	2.0	Nightstand	Active
item6	sofa	5	foam	peach	200.00	06-10-2021	0.0	Sectional sofa	Active
item7	desk	0	wood	brown	5000.00	08-10-2021	3.0	Desk	Inactive

Item Name: rocking chair Date: 09 Oct 2021

Quantity: 2 Category: Chair

Price: 5500.00 Color: black

Warranty: 2.0 Material: wood

Add Update Delete Reset

Item Name: search... Filter By: Active Inactive Sort By:

*** Click Above Cell Content to Edit or Delete Inventory Details ***

Are You Sure ?

Yes No

Item Added Successfully!

Enter all the details

OK

Test case: 6

Objectives: Test for adding Salary information

Test Data:

Valid: All required fields are entered and in their correct format.

Invalid: Some fields are left blank or written incorrectly.

Output:

Valid: Allows records to be added to the database.

Invalid: The user is prompt with an error message.

Result:

Valid: Records will be saved.

Invalid: The records will not be saved.

Conclusion: Both the valid and invalid results are tested and the record will be saved to the database.

X



Enter all the details

OK

Admin Dashboard

Supplier

Sale

Purchase

Employee

Inventory

Customer

Service

Display

Employee Id: emp6

Full Name: pooja prabhu

Year: 2021

Basic Pay: 12000.00

Month: September

Total Days: 30

Tax: 1800

Extra Leave: 0

H.R.A: 4800

Total Pay: 16800

Provident Fund: 1440

Net Pay: 13560

Add New

Manage User

Attendance

Salary

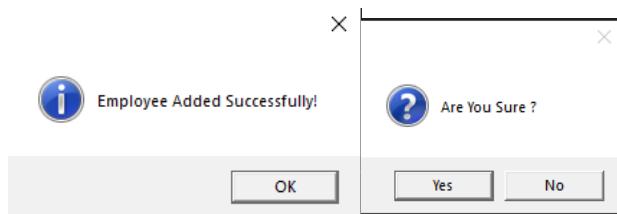
Generate

Calculate

Reset

Employee ID: search... Sort By

Pay_Id	Emp_Id	Date	Year	Month	Tax	HRA	PF
pay1	emp6	07-10-2021	2021	September	1800.00	4800.00	1440.00
pay2	emp6	07-10-2021	2021	August	1683.87	4490.32	1347.10
pay3	emp1	08-10-2021	2021	August	2806.45	7483.87	2245.16



System Testing:

Test case: 7

Objectives: To test the user name and password entry.

Test data:

Valid: Valid user name and password to be entered in the program.

Invalid: Use of invalid user name and password in the program

Output:

Valid: Entering into the software.

Invalid: Displaying an error message.

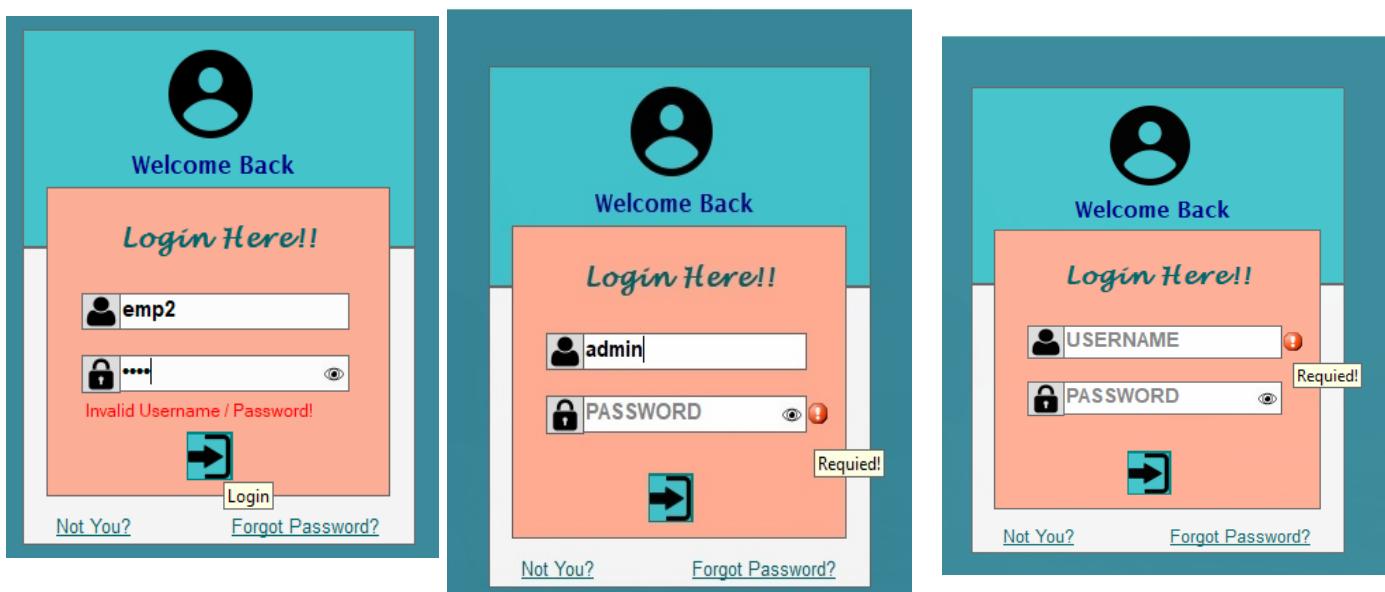
Result:

Valid: The user is allowed to enter the software.

Invalid: Display of error message which restrict the user to enter the software

Conclusion: Both the valid and invalid results are tested.

Output matches with the required result, hence the test is successful.



Test case: 8

Objectives: Test for adding purchase order.

Test data:

Valid: All required fields are entered in correct format.

Invalid: Some fields are blank or incorrect.

Output:

Valid: Allows records to be added to the database and a mail is generated to be sent to the supplier's mail-id.

Invalid: The user is prompt with an error message.

Result:

Valid: The records will be saved.

Invalid: Records will not be saved.

Conclusion: Both the valid and invalid results are tested and the record will be saved to the database.

Admin Dashboard

Select Item Here :

Item_Id	Name	Qty	Material	Color	Category
item1	table	12	wood	brown	Coffee Table
item2	bed	6	foam	pink	Bed
item3	chair	9	fiber	blue	Chair
item4	comptable	0	fiber	black	Console
item5	lamp	9	cotton	blue	Nightstand

Select Supplier Here :

Sup_Id	Name	Contact_No
sup4	vaish	9986775635
sup5	bhakthi	8088308112

Date: 09 Oct 2021

PO_No: po18

Sup_Id: sup5

Item_Id: item5

Name: lamp

Qty: 9

Save Edit Item Add Remove Item Place Order Cancel Order

Details Saved!

OK

Order Placed Successfully!

OK

Purchase Order

Inbox

bhakthiprabhu15@gmail.com
to me

Supplier Name : bhakthi
PO Id : po17
Date : 07-10-2021 12:44:16

Item ID	Name	Quantity
item5	lamp	1

Test case: 9

195 | Page

Objectives: Test case for inserting attendance information.

Test data:

Valid: Attendance details are saved.

Invalid: Attendance details are not saved.

Output:

Valid: Details will be inserted.

Invalid: Prompts an error message.

Result:

Valid: Record will be inserted.

Invalid: Record will not be inserted.

Conclusion: Both the valid and invalid results are tested and the record will be saved to the database.

Admin Dashboard

Supplier Sale Purchase Employee Inventory Customer Service Display

Employee Id: emp4 Full Name: qwerty

Month: May Year: 2021

Leave Taken: 8 Leave Given: 5

Update Save Reset

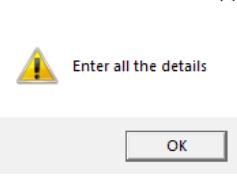
Employee ID: search... Sort By:

Emp_Id	Month	Year	Total_days	Leave_taken	Leave_given	Extra_leave
emp6	September	2021	30	2	2	0
emp6	August	2021	31	6	4	2
emp1	August	2021	31	5	3	2

*** Click Above Cell Content to Edit Attendance Details ***

Enter all the details

OK



Employee Page - Attendance

Admin Dashboard

Supplier Sale Purchase Employee Inventory Customer Service Display

Employee Id: emp4 Full Name: qwerty

Month: January Year: 2019

Leave Taken: 3 Leave Given: 7

Update

Employee ID: search... Sort By:

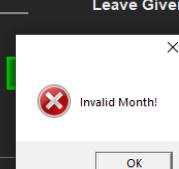
Emp_Id	Month	Year	Total_days	Leave_taken	Leave_given	Extra_leave
emp2	April	2021	30	7	2	5
emp3	April	2021	30	5	4	1
emp3	March	2021	31	3	4	0
emp2	September	2021	30	2	2	0

Invalid Month!

OK

*** Click Above Cell Content to Edit Attendance Details ***

196 |



Test case: 10

Objectives: Test for modifying employee information.

Test data:

Valid: All required fields are entered in correct format.

Invalid: Some fields are blank or incorrect.

Output:

Valid: Allows records to be added to the database.

Invalid: The user is prompt with an error message.

Result:

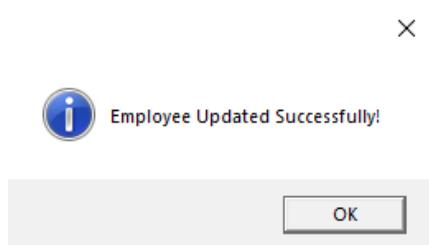
Valid: The records will be updated.

Invalid: Records will not be updated.

Conclusion: Both the valid and invalid results are tested and the record will be saved to the database.

The screenshot shows an 'Admin Dashboard' interface. A modal window is open, asking 'Are You Sure want to Update ?'. It contains two buttons: 'Yes' and 'No'. Below the modal, the main dashboard shows employee details for an employee named 'qwerty'. The form fields include: Full Name (qwerty), Gender (radio buttons for Female, Male, Others), Date of Birth (09-09-1999), Contact No. (9887867574), Age (22), Email ID (asdfg@gmail.com), Address (asdfg), Joined Date (10 Aug 2021), Designation (vvvvv), and Basic Pay (10000.00). On the right side of the dashboard, there are buttons for 'Add New', 'Manage User', 'Attendance', and 'Salary'. Below the form, there is a table of employee data with columns: Emp_Id, Name, Dob, Gender, Age, Contact_No, Address, and Email. The table lists employees from emp1 to emp7. A note at the bottom says: '*** Click Above Cell Content to Edit or Delete Employee Details ***'.

Emp_Id	Name	Dob	Gender	Age	Contact_No	Address	Email
emp1	sankalp	06-04-2000	Male	21	988776655	udupi	vaish.nay
emp2	sri	06-04-2000	Male	21	9445678944	barkur	vaish.nay
emp3	varsha	08-09-1998	Female	23	9866454243	malpe	tyuio@gr
emp4	qwerty	09-09-1999	Male	22	9887867574	asdfg	asdfg@g
emp5	arthi prabhu	08-04-1995	Female	26	8235467876	udupi	vaish.nay
emp6	pooja prab...	03-05-1995	Female	26	9534534523	udupi	vaish.nay
emp7	poorn	03-06-1990	Female	31	9335465636	udupi	vaish.nay



Test case: 11

Objectives: Test for modifying customer information.

Test data:

Valid: All required fields are entered in correct format.

Invalid: Some fields are blank or incorrect.

Output:

Valid: Allows records to be added to the database.

Invalid: The user is prompt with an error message.

Result:

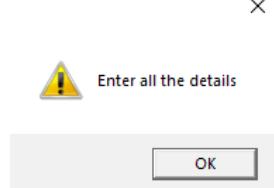
Valid: The records will be updated.

Invalid: Records will not be updated.

Conclusion: Both the valid and invalid results are tested and the record will be saved to the database.

A screenshot of the Admin Dashboard. On the left is a sidebar with icons for Supplier, Sale, Purchase, Employee, Inventory, Customer, Service, and Display. The main area shows a form for updating a customer record. The form fields are: Full Name (anvitha), Address (karavali), Contact No. (9977554433), and Email ID (jklp@gmail.com). Below the form is a table with columns: Name, Contact_No, Address, and Email. The table contains four rows: pooja, udupi, vaish.nayak999@gmail.c...; alfiya, yermall, qwerty@gmail.com; anvitha, 9977554433, jklp@gmail.com. A search bar and a sort by dropdown are also present. A note at the bottom says "*** Click Above Cell Content to Edit Customer Details ***".





Test case: 12

Objectives: Test case to delete supplier information.

Test data:

Valid: Supplier information is deleted from all foreign key tables.

Invalid: Supplier information is not deleted from foreign key tables.

Output:

Valid: Record will be deleted.

Invalid: Prompts an error message.

Result:

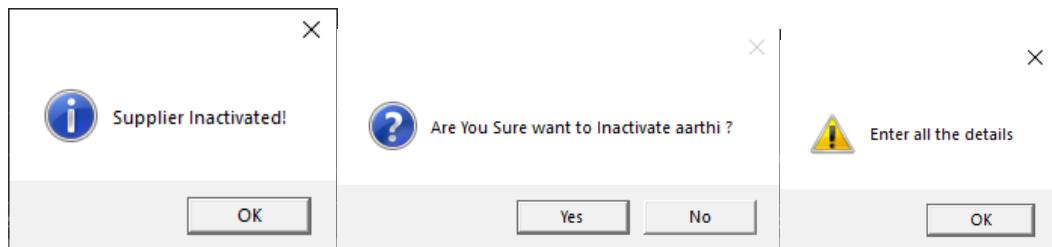
Valid: Supplier information will be deleted.

Invalid: Supplier information will not be deleted.

Conclusion: Both the valid and invalid results are tested and the record will be saved to the database.

Sup_Id	Name	Address	Email	Contact_No	Status
sup1	varshini	malpe	abcd@gmail.com	9988772233	Inactive
sup2	sushmitha	indrali	qwerty@gmail....	9933445566	Inactive
sup3	tanvii	pangala	asdfg@gmail.c...	9933445567	Inactive
sup4	vaish	udipi	bhakthiprabhu...	9986775635	Active
sup5	bhakthi	udipi	vaish.nayak999...	8088308112	Active
sup6	aarthi	udipi	vaish.nayak999...	9887654323	Inactive

Sup_Id	Name	Address	Email	Contact_No	Status
sup1	varshini	malpe	abcd@gmail.com	9988772233	Inactive
sup2	sushmitha	indrali	qwerty@gmail....	9933445566	Inactive
sup3	tanvii	pangala	asdfg@gmail.c...	9933445567	Inactive
sup4	vaish	udipi	bhakthiprabhu...	9986775635	Active
sup5	bhakthi	udipi	vaish.nayak999...	8088308112	Active
sup6	aarthi	udipi	vaish.nayak999...	9887654323	Inactive



Test case: 13

Objectives: Test for modifying supplier information.

Test data:

Valid: All required fields are entered in correct format.

Invalid: Some fields are blank or incorrect.

Output:

Valid: Allows records to be added to the database.

Invalid: The user is prompt with an error message.

Result:

Valid: The records will be updated.

Invalid: Records will not be updated.

Conclusion: Both the valid and invalid results are tested and the record will be saved to the database.

Admin Dashboard

- User
- Supplier
- Sale
- Purchase
- Employee
- Inventory
- Customer
- Service
- Display

Full Name

Email ID

Contact No.

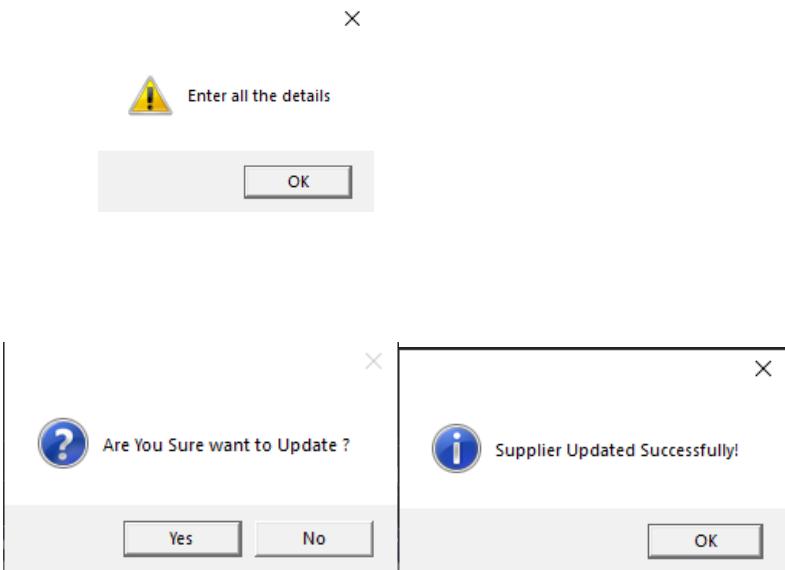
Address

Add
Update
Delete
Reset

Supplier Name / ID
Filter By Active Inactive
Sort By

Sup_Id	Name	Address	Email	Contact_No	Status
sup1	varshini	malpe	abcd@gmail.com	9988772233	Inactive
sup2	sushmitha	indralli	qwerty@gmail.com	9933445566	Inactive
sup3	tanvii	pangala	asdfg@gmail.c...	9933445567	Inactive
sup4	vaish	udipi	bhakthiprabhu...	9988775635	Active
sup5	bhakthi	udipi	vaish.nayak999...	8088308112	Active
sup6	aarthi	udipi	vaish.nayak999...	9887654323	Inactive

*** Click Above Cell Content to Edit or Delete Supplier Details ***



Integration Testing:

Test case: 14

Objectives: Test for the relationship between purchase bill and stock.

Test data:

Valid: When all information for the purchase bill is entered in correct format.

Invalid: Some fields are blank or incorrect.

Output:

Valid: Purchase bill details are saved in the database and hence the updating is done automatically in the stock.

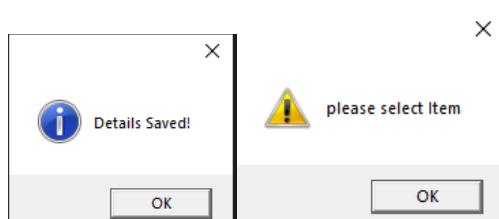
Invalid: The user is prompt with an error message.

Result:

Valid: Relationship between the two forms works successfully.

Invalid: Relationship fails.

Conclusion: Both the valid and invalid results are tested. The output match with the required output hence the test is successful.



Admin Dashboard

Purchase Order No. po17 Date 09 Oct 2021 Save

Item_Id	Name	Qty	Price	Warranty
item5	lamp	1	400.00	2.0

Supplier Id sup5
Supplier Name bhakthi

Quantity _____
Price _____
Warranty In years _____

PB_No	Item_Id	Qty	Price	Amount	Warranty
pb11	item5	1	400.00	400.00	2.0

Edit Item Remove Item

Total Amount 400 Save Cancel

Purchase Order **Purchase Bill** **Purchase Return**

Are You Sure ? Bill generated Successfully!

Yes No OK

Item_Id	Name	Qty	Material	Color	Price	Date	Warranty	Category	Status
item1	table	12	wood	brown	3000.00	16-09-2021	2.0	Coffee table	Active
item2	bed	6	foam	pink	20000.00	16-09-2021	5.0	Bed	Active
item3	chair	9	fiber	blue	500.00	17-09-2021	1.0	Chair	Active
item4	comptable	0	fiber	black	2500.00	18-09-2021	1.0	Console table	Inactive
item5	lamp	10	cotton	blue	400.00	20-09-2021	2.0	Nightstand	Active
item6	sofa	5	foam	peach	200.00	06-10-2021	0.0	Sectional sofa	Active
item7	desk	0	wood	brown	5000.00	08-10-2021	3.0	Desk	Inactive

Test case: 15

Objectives: Test for the relationship between sales bill and stock.

Test data:

Valid: When all information for the sales bill is entered in correct format.

Invalid: Some fields are blank or incorrect.

Output:

Valid: Sales bill details are saved in the database and hence the updating is done automatically in the stock.

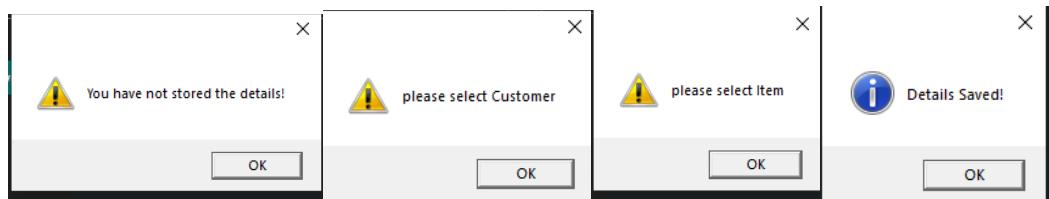
Invalid: The user is prompt with an error message.

Result:

Valid: Relationship between the two forms works successfully.

Invalid: Relationship fails.

Conclusion: Both the valid and invalid results are tested. The output match with the required output hence the test is successful.



Admin Dashboard

Select Item Here :

Item_Id	Name	Qty	Price	Material	Color
item1	table	12	3000.00	wood	brown
item2	bed	6	20000.00	foam	pink
item3	chair	9	500.00	fiber	blue
item5	lamp	10	400.00	cotton	blue

Select Customer Here :

Name	Contact_No	Address
pooja	9338538499	udupi
alfiya	9944455543	yermall
anvitha	9977554433	karavali

Date : 09 Oct 2021 SB_No : sb9 Item_Id : item5 Qty : 1 Rate : 600.00 Price : 600.00 Tax : 108.00 Amount : 708.00

Quantity :
Rate :
Warranty : In years

Add Total Amount : 708

Discount : 0 Net Amount : 708

Supplier
Sale
Purchase
Employee
Inventory
Customer
Service
Display



Furniture Shop Management System									
Renovate Your Home									
<u>Sale Invoice</u>									
Invoice No. : SB9	Invoice Given By: admin								
Invoice To: arnitha	Invoiced Date: 09-10-2021 03:29:56								
Contact No.: 9977554433									
Detail of Product	Warranty in Years	Quantity	Rate	Price	Tax	Amount			
lamp	2.0	1	600.00	600.00	108.00	708.00			
THANK YOU !! VISIT AGAIN Goods Once Sold Will Not Be Exchanged or Taken Back.				Total Amount:	708.00				
				Discount:	0.00				
				Grand Total:	708.00				
Amount In Words : SEVEN HUNDRED AND EIGHT RUPEES ZERO PAISE ONLY									
THIS IS COMPUTER GENERATED PRINTOUT WHICH REQUIRES NO SIGNATURE. Print Bill									
Item_Id	Name	Qty	Material	Color	Price	Date	Warranty	Category	Status
item1	table	12	wood	brown	3000.00	16-09-2021	2.0	Coffee table	Active
item2	bed	6	foam	pink	20000.00	16-09-2021	5.0	Bed	Active
item3	chair	9	fiber	blue	500.00	17-09-2021	1.0	Chair	Active
item4	comptable	0	fiber	black	2500.00	18-09-2021	1.0	Console table	Inactive
item5	lamp	9	cotton	blue	400.00	20-09-2021	2.0	Nightstand	Active
item6	sofa	5	foam	peach	200.00	06-10-2021	0.0	Sectional sofa	Active
item7	desk	0	wood	brown	5000.00	08-10-2021	3.0	Desk	Inactive

Test case: 16

Objectives: Test for the relationship between sales order and stock.

Test data:

Valid: When all information for the sales order is entered in correct format.

Invalid: Some fields are blank or incorrect.

Output:

Valid: Sales order details are saved in the database and hence the updating is done automatically in the stock.

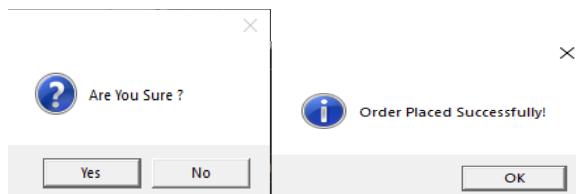
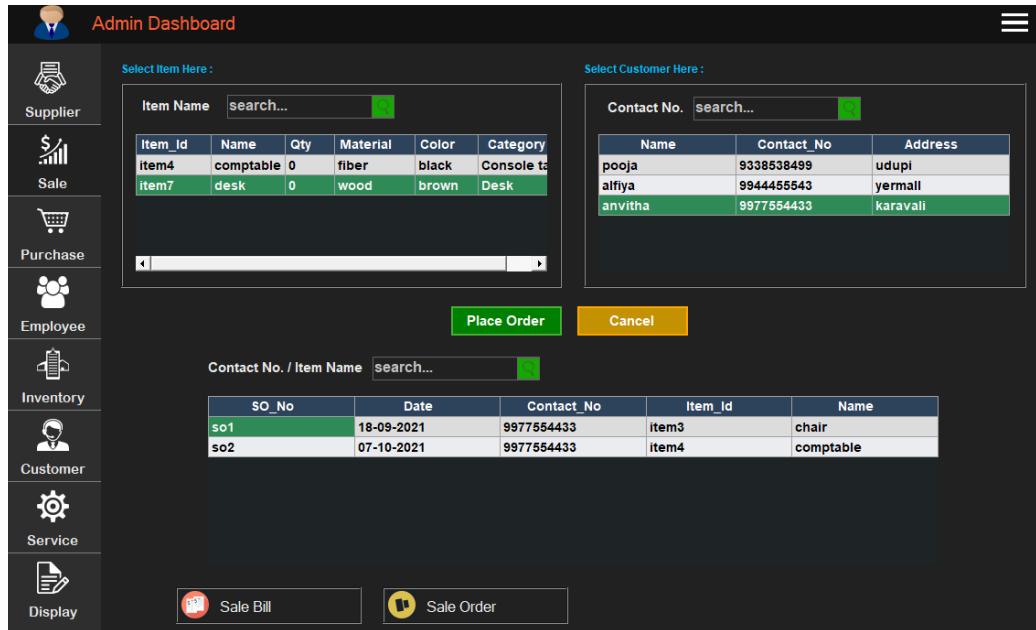
Invalid: The user is prompt with an error message.

Result:

Valid: Relationship between the two forms works successfully.

Invalid: Relationship fails.

Conclusion: Both the valid and invalid results are tested. The output match with the required output hence the test is successful.



Test case: 17

Objectives: Test for the relationship between purchase return and stock.

Test data:

Valid: When all information for the purchase return is entered in correct format.

Invalid: Some fields are blank or incorrect.

Output:

Valid: Purchase return details are saved in the database and hence the updating is done automatically in the stock.

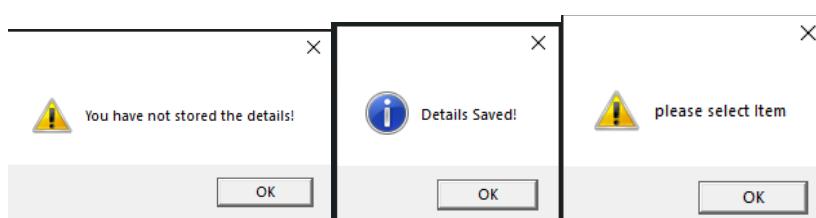
Invalid: The user is prompt with an error message.

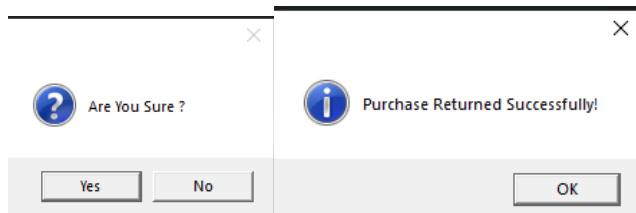
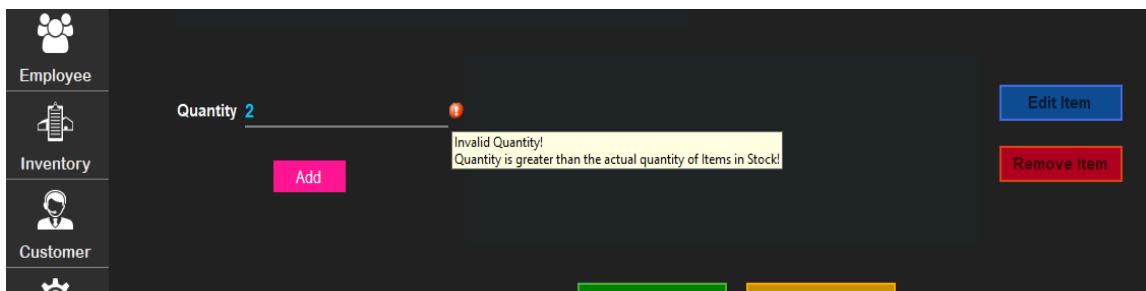
Result:

Valid: Relationship between the two forms works successfully.

Invalid: Relationship fails.

Conclusion: Both the valid and invalid results are tested. The output match with the required output hence the test is successful.





Item_Id	Name	Qty	Material	Color	Category
item1	table	12	wood	brown	Coffee Table
item2	bed	6	foam	pink	Bed
item3	chair	9	fiber	blue	Chair
item4	comptable	0	fiber	black	Console
item5	lamp	8	cotton	blue	Nightstand

Test case: 18

Objectives: Test for the relationship between purchase order and stock.

Test data:

Valid: When all information for the service is entered in correct format.

Invalid: Some fields are blank or incorrect.

Output:

Valid: Purchase order details are saved in the database and hence the updating is done automatically in the stock.

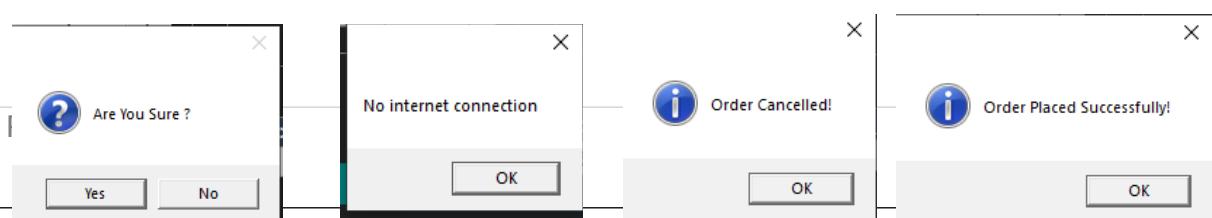
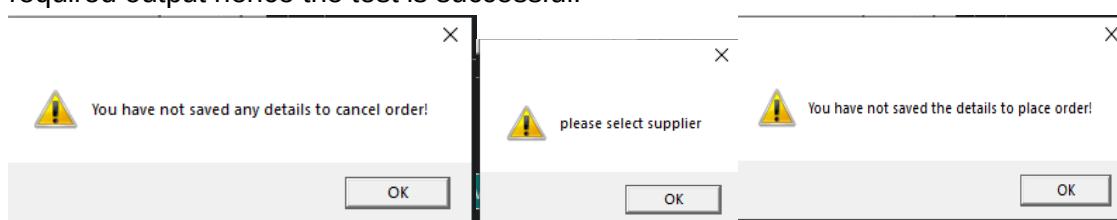
Invalid: The user is prompt with an error message.

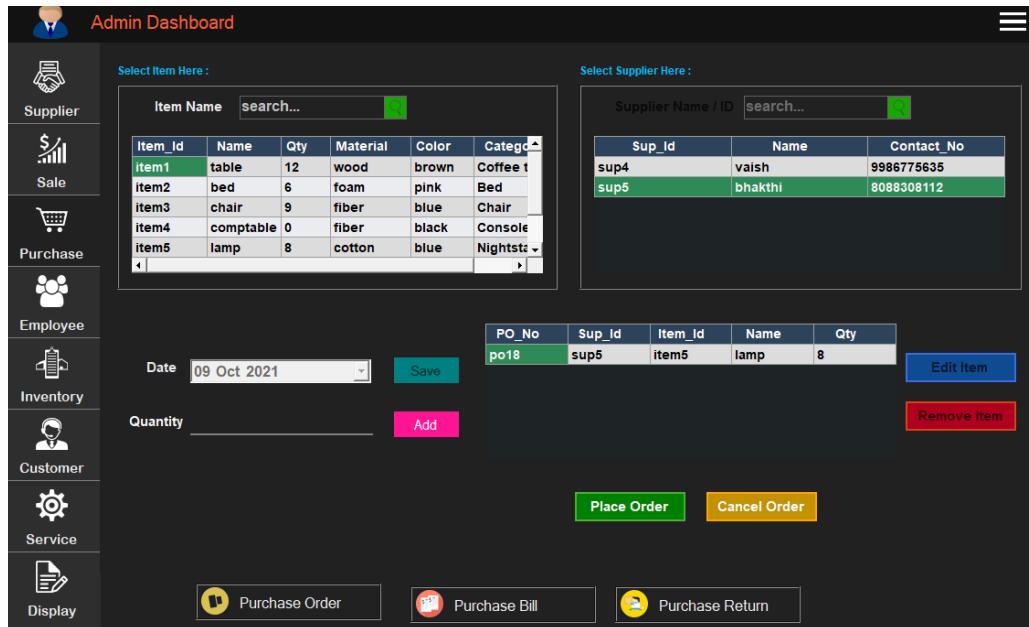
Result:

Valid: Relationship between the two forms works successfully.

Invalid: Relationship fails.

Conclusion: Both the valid and invalid results are tested. The output match with the required output hence the test is successful.





Limitations and Future Scope:

- **Limitations:**
 - Biometric options are not available to maintain employee attendance.
 - There is no card payment option (ATM) in this project.
 - SQL Server has to be installed on the system in which this software needs to be used.
 - Barcode system is not available.
- **Future Scope and enhancement:**
 - The software can be updated to allow online transactions.
 - The software can be enhanced to allow a biometric option to track employees' attendance.

DEFINITIONS, ACRONYMS AND ABBREVIATION

We use connection string for connecting to SQL server. Connection defines three data sources, initial catalogue and integrated security.

- o Data source: This defines the source from which the data needs to be extracted.
- o Initial catalogue: It is the name of the database required.

- o Integrated security: This asks if to use the application the login name given in the beginning is enough or before starting the application if new password is required.
 - DFD: Data Flow Diagram
 - CFD: Control Flow Diagram
 - SQL: Structured Query Language
 - Visual C#

BIBLIOGRAPHY

- **An Integrated approach to Software Engineering** By: Panchal Jalote
- www.stackoverflow.com