COMPUTER SHOP SYSTEM

A

PROJECT REPORT

Submitted in partial fulfilment
of the requirement for
the award of Degree

BACHELOR OF COMPUTER APPLICATIONS

Submitted to

Submitted by

Gaurav Mahawar
BCA DS 3rd Sem

Submitted to

School of computer application

CAREER POINT UNIVERSITY,ALANIYA,KOTA
Session (22-23)

DECLARATION BY CANDIDATE

I am Gaurav Mahawar student of **CAREER POINT UNIVERSITY,ALANIYA,KOTA**, hereby declare that the work presented in this project is outcome of my own work, is bonfire, correct to the best of my knowledge and this work has been carried out taking care of IT Ethics. The work presented does not infringe any patented work and has not been submitted to any University for the award of any degree.

B.C.A (D.S) 3rd semester

ACKNOWLEDGEMENT

First of all, I would like to express my heartfelt gratitude to School of Computer application, Career Point University, Kota, Rajasthan for giving me this precious opportunity to follow a world recognized degree program.

My heartfelt gratitude goes to, Dr. Garima Tyagi ma'am (HOD of School Of Computer Application), Mr. Arshad Hussain, Dr. Abid Hussain sir, Dr. Amit Sharma sir, Ms. Shalini Chawla ma'am, Mr. Praveen Goyal of school of Computer application at Career Point University for their valuable suggestions and directly or indirectly for completing this project.

Finally, I would like to express my special thanks to All my Classmates for wisdom Computer Technologies and staff members for giving me the great support and for providing their valuable time to help me for successful completing This project.

Thanks goes to all those who helped, whether through their comments, feedback, edits or suggestion.

Table of Contents

Chapter 1

INTRODUCTION

Computer Shop Management System is a large database system which can be used for managing computer shop system .In Computer Shop Management System, it store records of suppliers, hardware, sale, purchase and customer's records are maintained and manipulated. There is now some investigations are involved to find out or to correctly done the work.

It can assist the staff to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.

The aim of our project is to develop a system that is meant to partially computerize the work performed in the Computer Shop System like generating customer bills, record of stock and dealer, store record of the customer and item that we ordered.

The overall project report is divided into some parts. These parts are further divided into their subparts.

We use Microsoft Visual Basic 4.0 as front end and MS-Access 2016 as back end for developing our project. Visual Basic is primarily a visual design environment.

a.) Objective :-

This project deals with the management of the Computer Shop Management System. Shop is a place where the work is to sale the hardware, prepare bill, maintain purchasing and stock in shop and keeps their records, and prepare reports. The aim is to automate its existing manual system by the help of computerized equipment and fulfilling their requirements.

- To assist the staff in capturing the effort spent on their respective working areas.
- To keep and manipulate suppliers information.
- To keep and manipulate hardware details.
- To provide facility to sale at shop and prepare customer bill.
- To make enquiry of a particular sale

To maintain the purchase and stock at shop.

b.) Scope :-

Our project aims at Business process automation, i.e. we have tried to computerize various processes of Computer Shop Management System. In the sector of computer shop we have computerizes their and stock maintenance. Scope of any software depends upon the following things:

1. It satisfy the user requirement
2. Be easy to understand by the user and operator
3. Be easy to operate
4. Have a good user interface
5. Be expandable
6. Delivered on schedule within the budget.

We have tried to make such type of software, which satisfy the above given requirement.

Chapter 2

SYSTEM ANALYSIS

a.) Feasibility :-

Feasibility study is the phase in which the analyst checks that the candidate system is feasible for the organization or not. This entails identification, description & evaluation of the system. Feasibility study is done to select the best system that meets the performance requirement.

Functional requirement of the user with a Computer Shop Management System are that is should be able to maintain information of sales, dealer, stock and ordered item and also generate the bill of customer purchasing. And easily check the report of customer, stock and ordered items and can search their information.

This system is also covers the non-functional requirement such as fast to save information, retrieve information and update information. It is secure, less chance of errors and user friendly interface.

If the feasibility study is to serve as a decision document, it must answer key questions.

1. Is there a new and better way to do the job that will benefit the user?
2. What are the costs and savings of the alternatives?
3. What is recommended?

The most successful system projects are not necessarily the biggest or most visible in the business but rather those truly meet user's expectations.

Feasibility considerations :-

Three key considerations are involved in the feasibility study. They are as follows:-

Economic Feasibility :

Economic analysis is the most frequently used method for evaluating the effectiveness of the computerized system. We analyse the computerized system is feasible as than the manual system because it saves the money, time and manpower. It also feasible according to cost benefits analysis.

Technical Feasibility :

Technical feasibility centers around the technology used. It means the candidate system is technically feasible i.e. it don't have any technical fault and work properly in the given environment. Our system is technically feasible; it is providing us required output.

Operational Feasibility:

No doubt the proposed system is fully GUI based that is very user friendly. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with new system. As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing all the complex activities itself.

b.) Requirement specification :-

1. **Developer minimum requirement :-**

Hardwar

Processor	Quad core processor or higher
Hard disk space	256 GB
RAM	4 GB

Software

Operating System	Windows 7(or higher) or Linux or Mac OS
Database	MS-Access 2007 or above
Tool	MS Visual Studio 2012 or above
Language	Visual Basic.NET

2. **User specification :-**

Hardware

Processor	Dual core processor or higher
Hard disk	128 GB
RAM	2 GB

Software

Operating System	Windows 7(or higher) or Linux or Mac OS
Database	MS-Access 2007 or higher
Tool	Visual Studio 2012 or higher

Chapter 3

SYSTEM DESIGN

a.) Significance features of Language Used :-

- **Front-end Environment (.NET Framework) :**

The Internet revolution of the late 1990s represented a dramatic shift in the way individuals and organizations communicate with each other. Traditional applications, such as word processors and accounting packages, are modelled as stand-alone applications: they offer users the capability to perform tasks using data stored on the system the application resides and executes on. Most new software, in contrast, is modelled based on a distributed computing model where applications collaborate to provide services and expose functionality to each other.

- **Benefits of the .NET Framework :**

The .NET Framework offers a number of benefits to developers:

- A consistent programming models
- Direct support for security
- Simplified development efforts
- Easy application deployment and maintenance

The .NET Class Library is a key component of the .NET Framework — it is sometimes referred to as the Base Class Library (BCL). The .NET Class Library contains hundreds of classes you can use for tasks such as the following:-

- Working with data from multiple data sources
- Debugging your code and working with event logs
- Working with data streams and files
- Managing the run-time environment
- Developing Web services, components, and standard Windows applications
- Working with application security

The functionality that the .NET Class Library provides is available to all .NET languages, resulting in a consistent object model regardless of the programming language developer's use.

- **About VB.net Language :**

Visual Basic was developed forms the BASIC programming language. In 1970's Microsoft got its start by developing ROM based interpreted Basic for the early microprocessor-based computer. In 1982, Microsoft QuickIasic revolutionized Basic and legitimized as a serious development language for MS-DOS environment. Later on, Microsoft Corporation created the enhanced version of BASIC called Visual Basic for Windows

- **Features of Visual Basic:**

1. Visual basic provides a GUI which and therefore screen looks very attractive.
2. Work on client / server computing model.
3. Object Oriented programming approach.
4. Visual basic provides several tool bars, which make working quick and easy.
5. It is front end and DBMS as a back end so it uses all the features of RDBMS like referential integrity foreign key etc.
6. It uses a micro help line, which visual basic uses to display starters to display information through the session.

- **Extensibility Model:**

The new model allows to programmatically extending the development environment and control projects, events, code, and Visual elements Declare.

b.) Description of Modules :-

This project includes the following modules for the development of the project. There are as follows :-

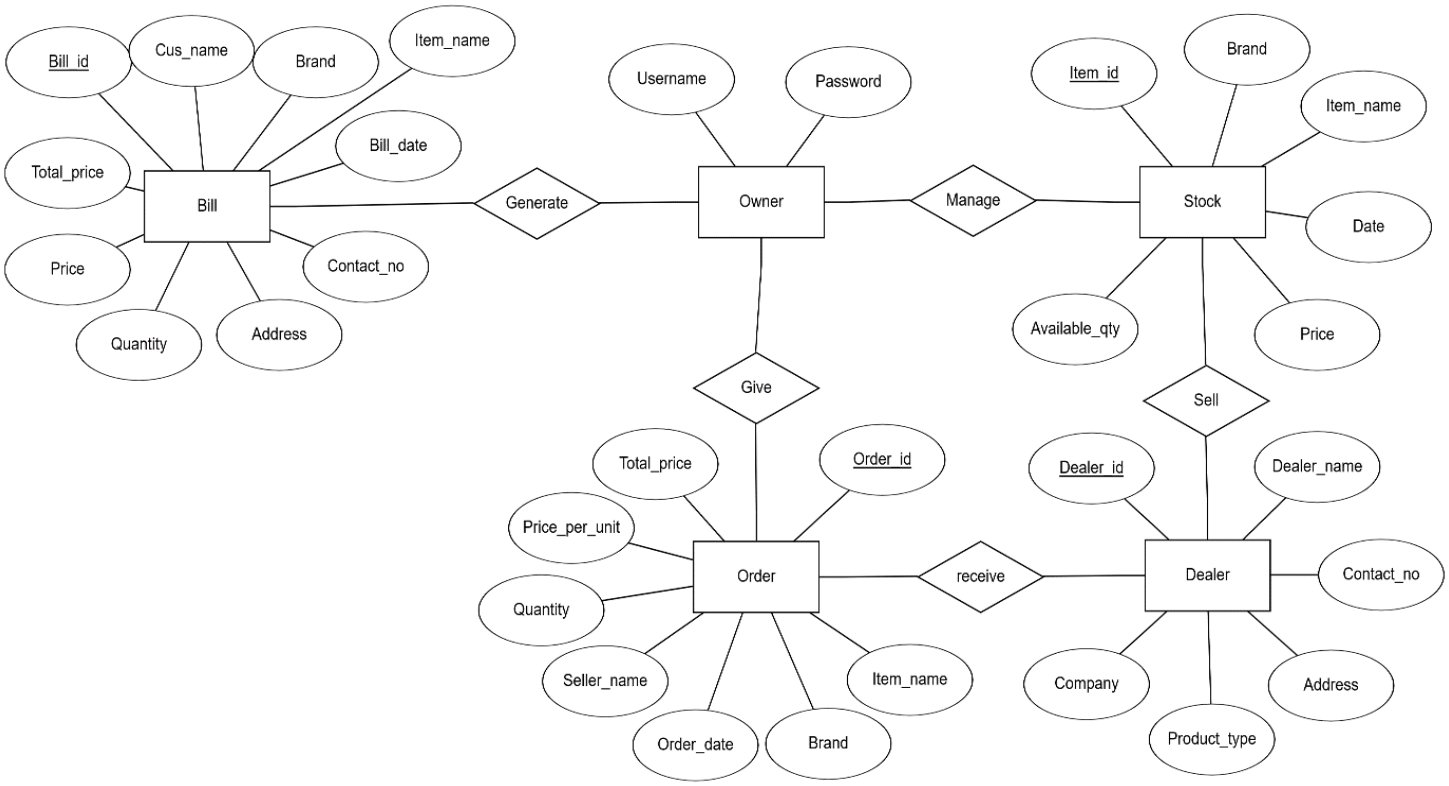
1. **Login Module :-** This second form our project this asks user to enter username and password then it check both on the database to access user to system else show a error message.
2. **Master Module :-** This module have four submodules, there are as follows :-

- **Customer Module :** In this module user can keep the record of customer information like customer details, purchasing etc.
- **Dealer Module :** In this module user can keep the record of dealers and user can also add, update and delete dealer and their information.

- **Order Module :** In this module user can saves the details of items that they order and perform tasks like add, delete order.
- **Stock Module :** In this module user can keep the record of available stock in shop and add items, delete items and update items information.

3. **Search Module :-** In this module user can search the information like customer, stock, dealer and ordered items information by their unique identification(primary key).
4. **Report Module :-** This module is used to check the report of customer, stock and ordered items.
5. **Bill generate Module :-** In this module user can generate the bill of customer purchasing.

c.) ER-Diagram :-



d.) Database :

Field	Datatype	Description
Username	Character	
Password	Character	

Login Table

Field	Datatype	Description
Dealer_id	Number	Primary key
Dealer_name	Character	
Address	Character	
Contact_no	Number	
Product_type	Character	
Company_name	Character	

Dealer Table

Field	Datatype	Description
Item_id	Number	Primary key
Brand	Character	
Item_name	Character	
Date	Date/time	
Price	Number	
Available_qty	Number	

Stock Table

Field	Datatype	Description
Bill_id	Number	Primary Key
Customer_name	Character	
Brand	Character	
Item_name	Character	
Bill_date	Date/time	
Address	Character	
Contact_no	Number	
Quantity	Number	
Price	Number	
Total_price	Number	

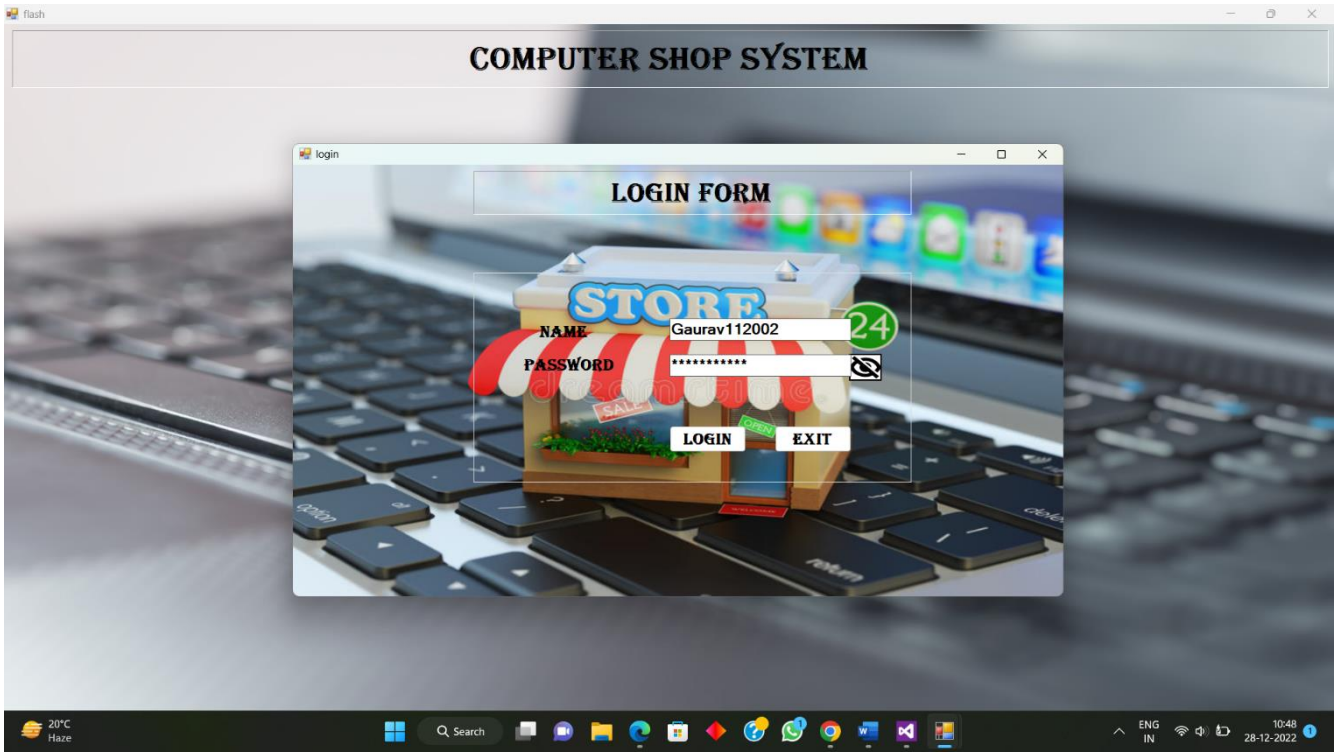
Bill Table

Field	Datatype	Description
Order_id	Number	Primary key
Brand	Character	
Item_name	Character	
Order_date	Date/time	
Seller	Character	
Quantity	Number	
Price_per_unit	Number	
Total_price	Number	

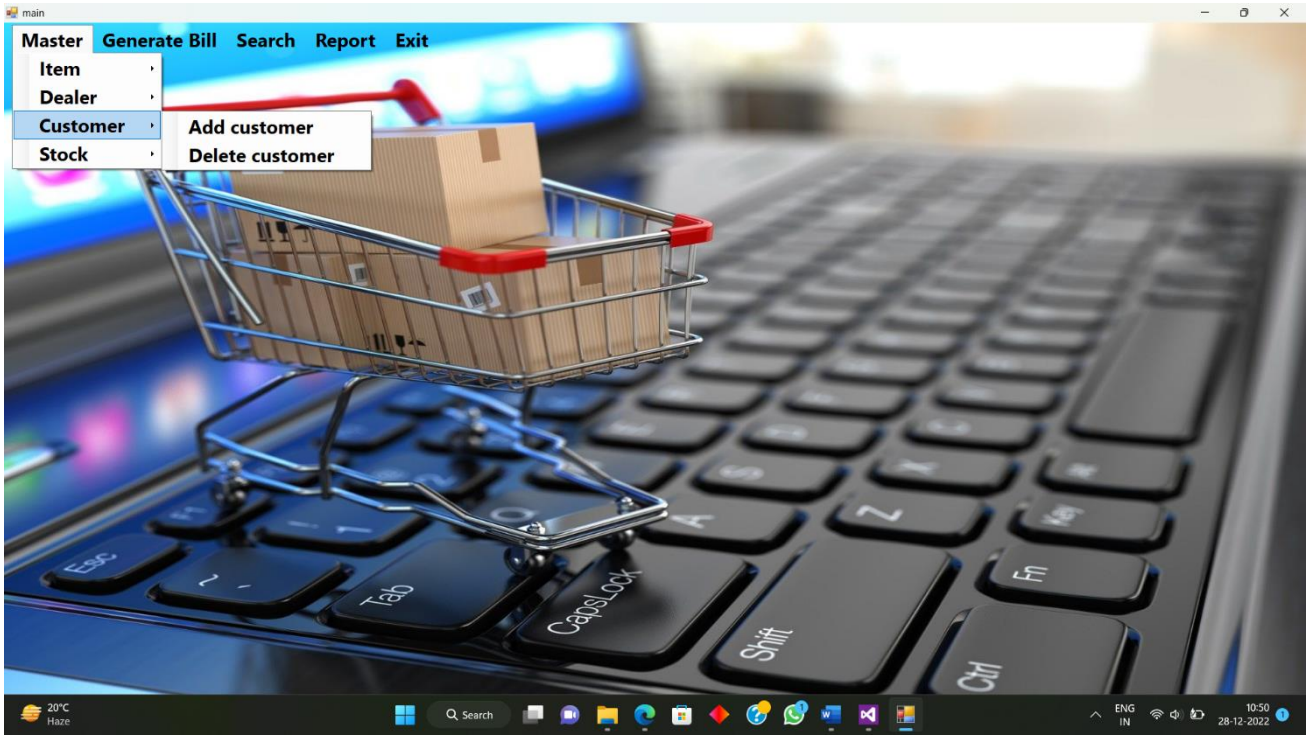
Order Table

e.) Input/output design :

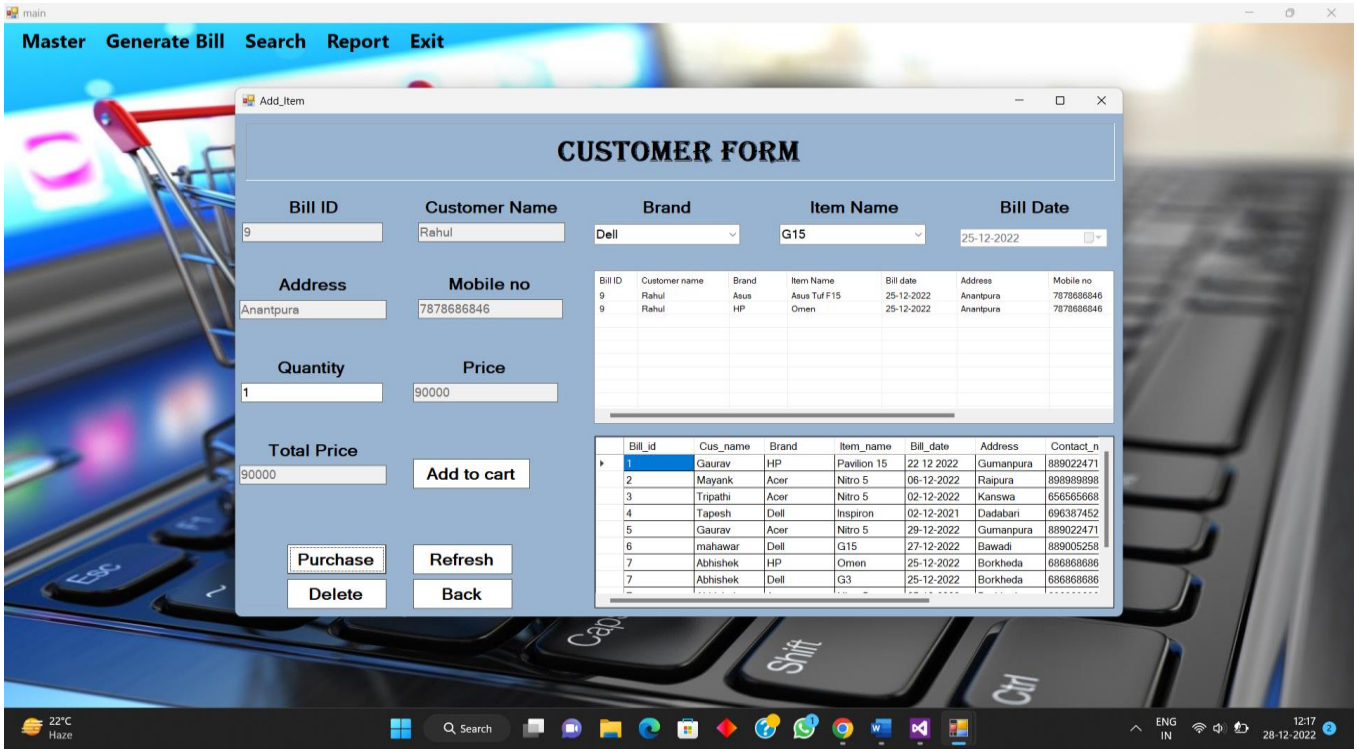
I. Login Form



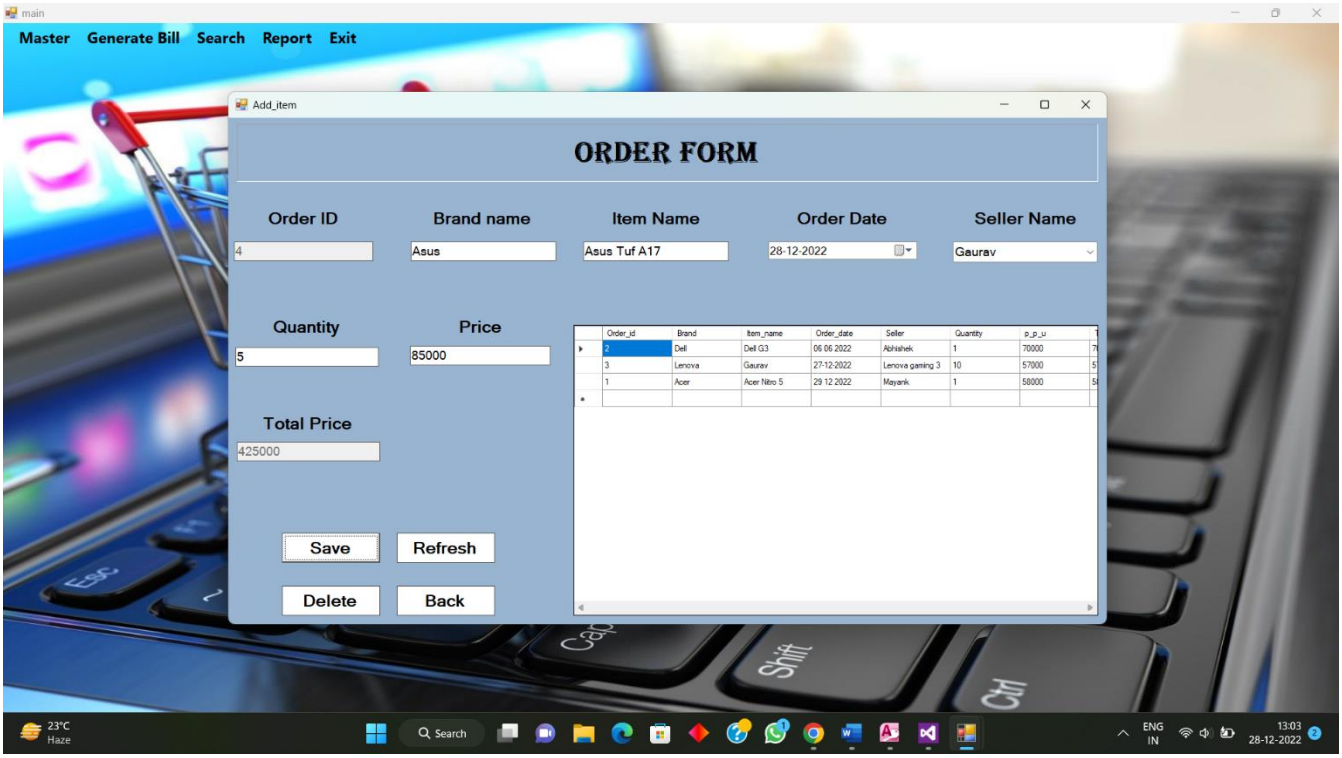
II. Main menu Form



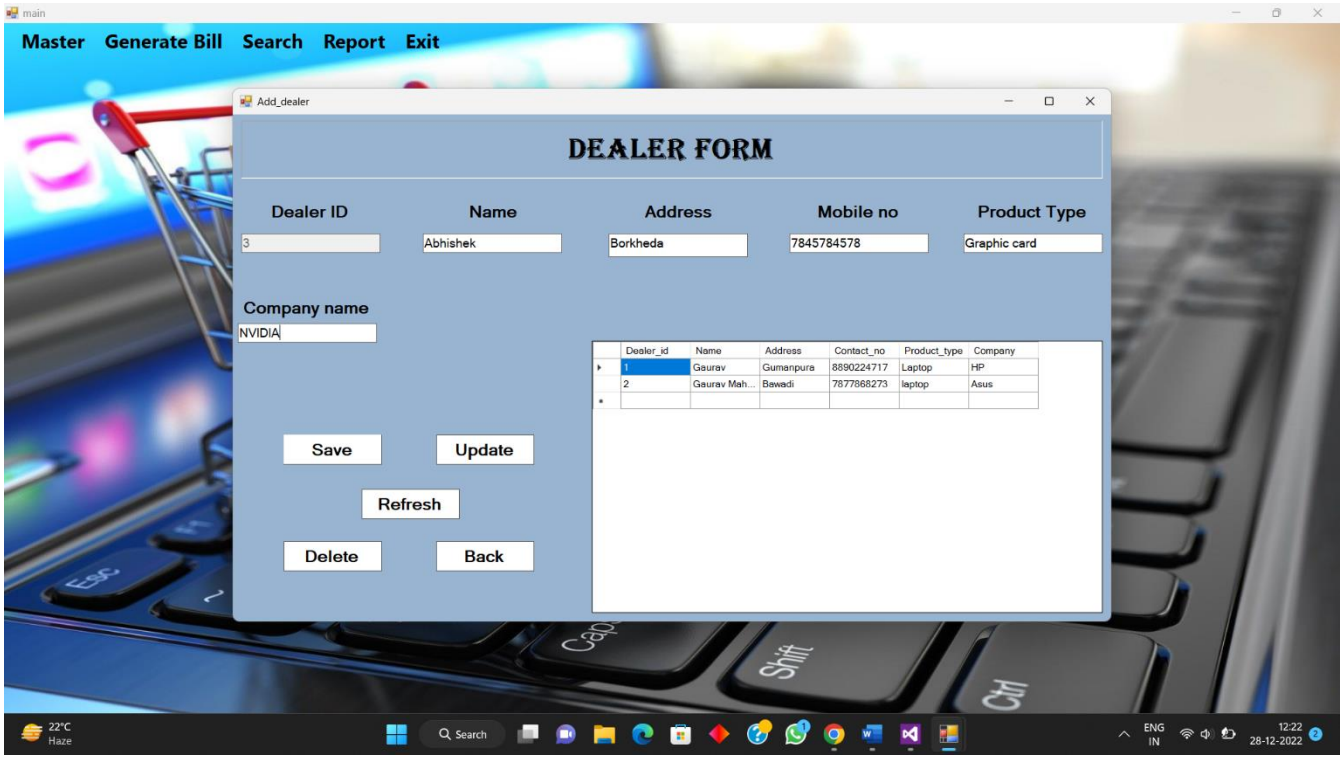
III. Sale Form



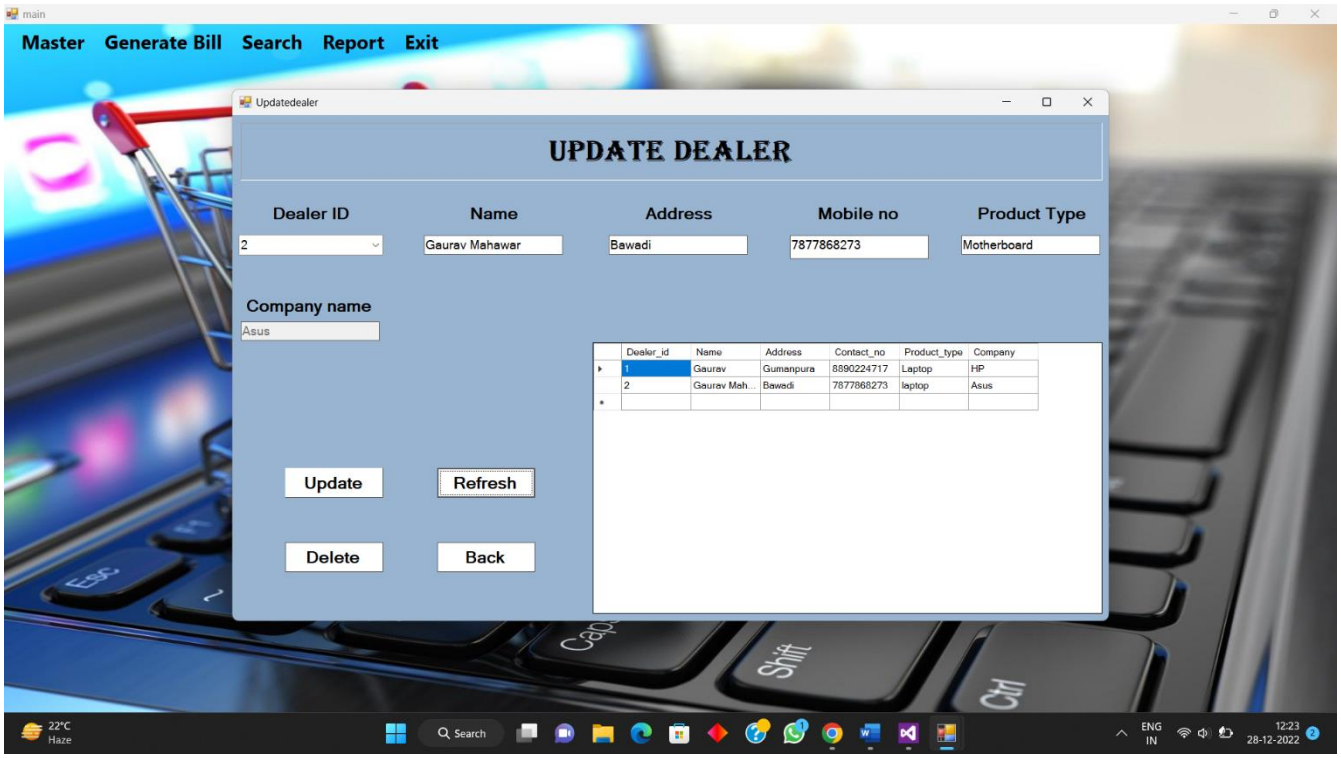
IV. Order Item Form



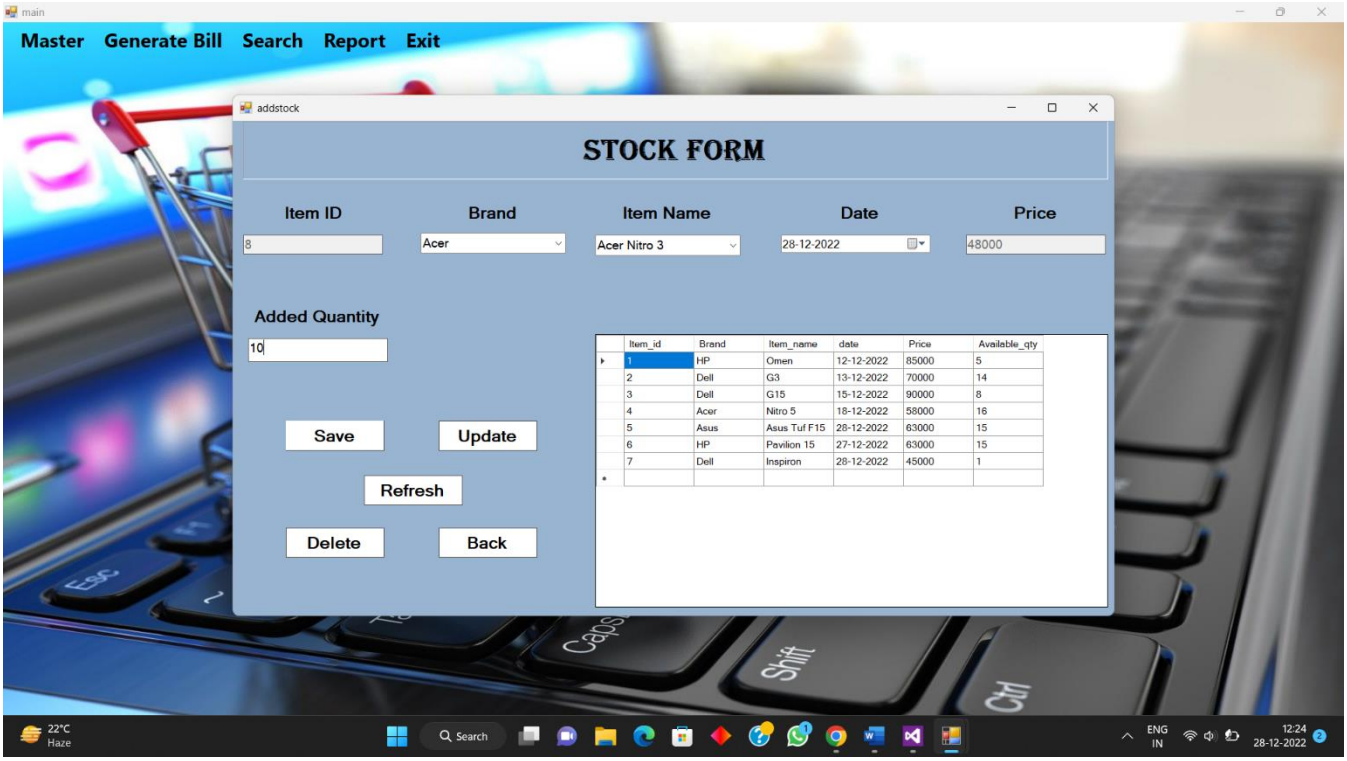
V. Add Dealer Form



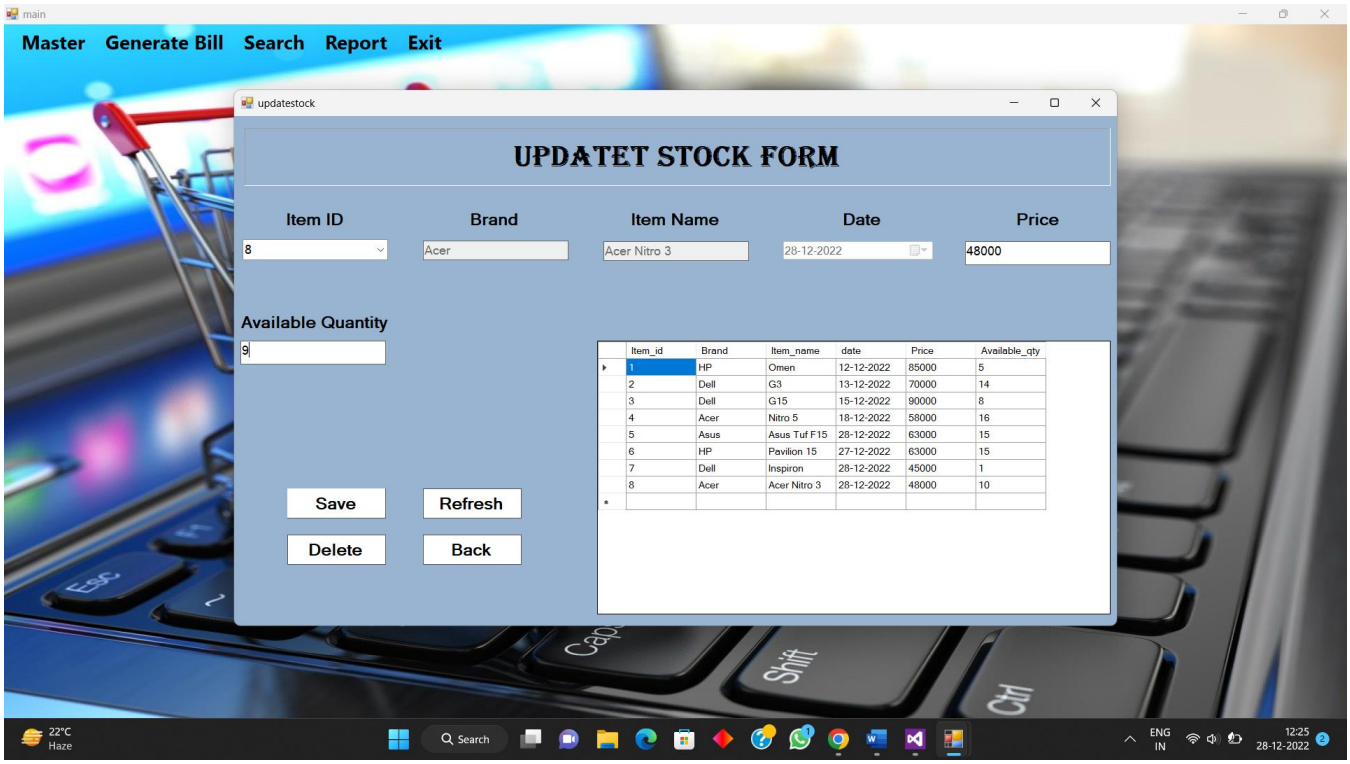
VI. Update Dealer Form



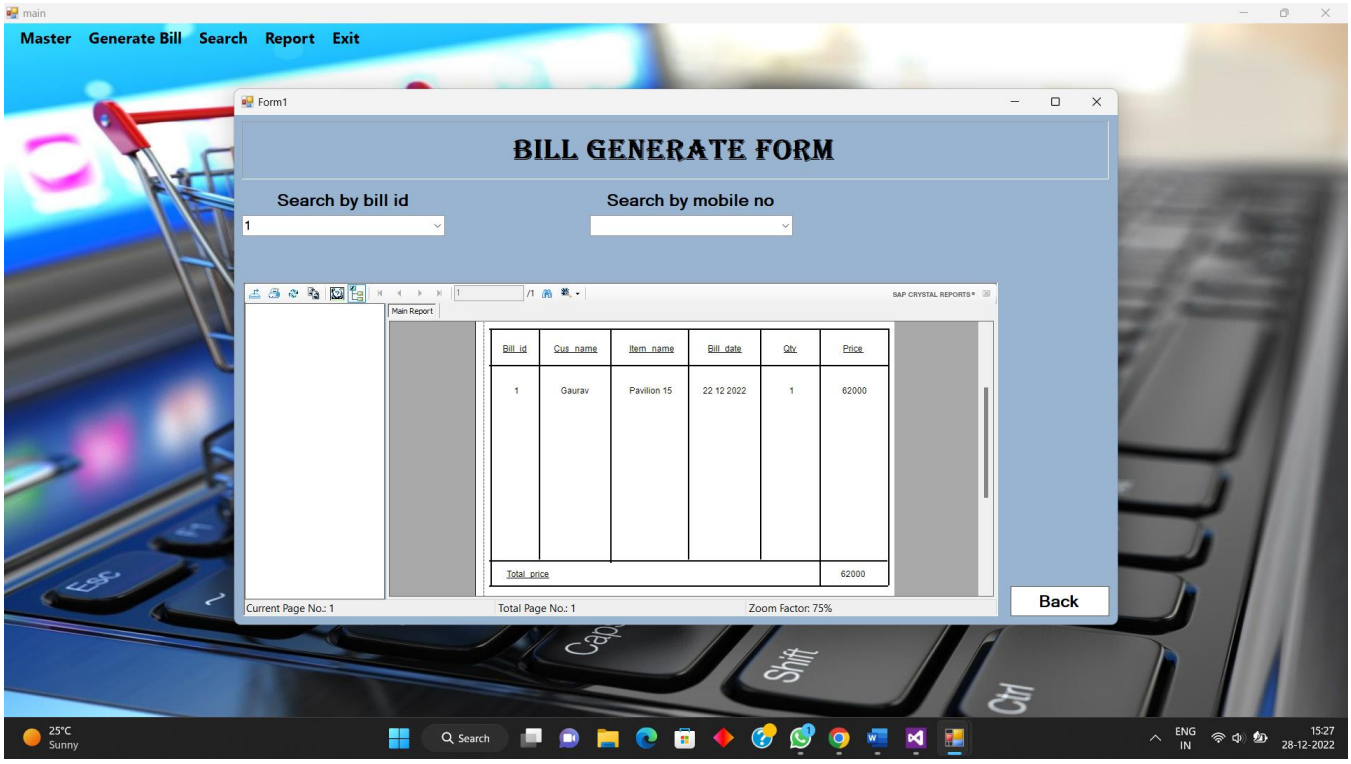
VII. Add Stock Form



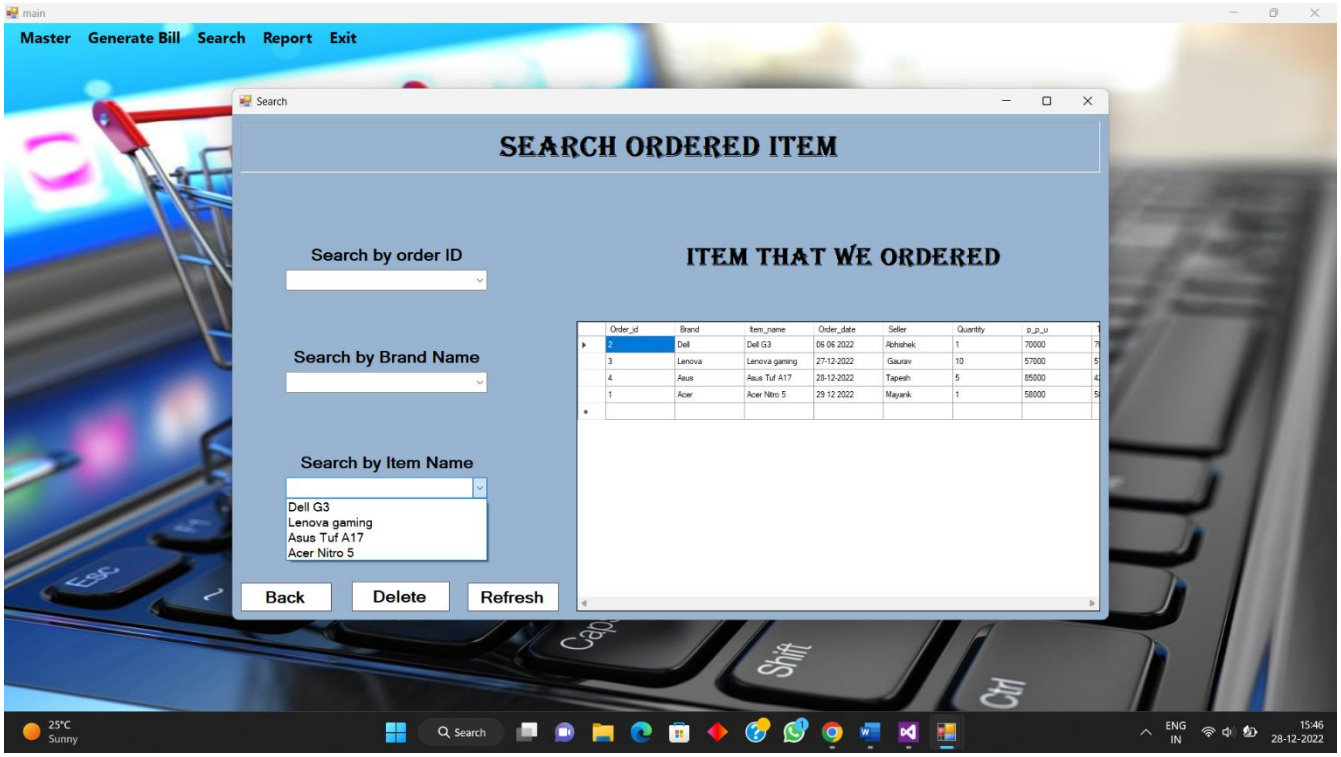
VIII. Update Stock Form



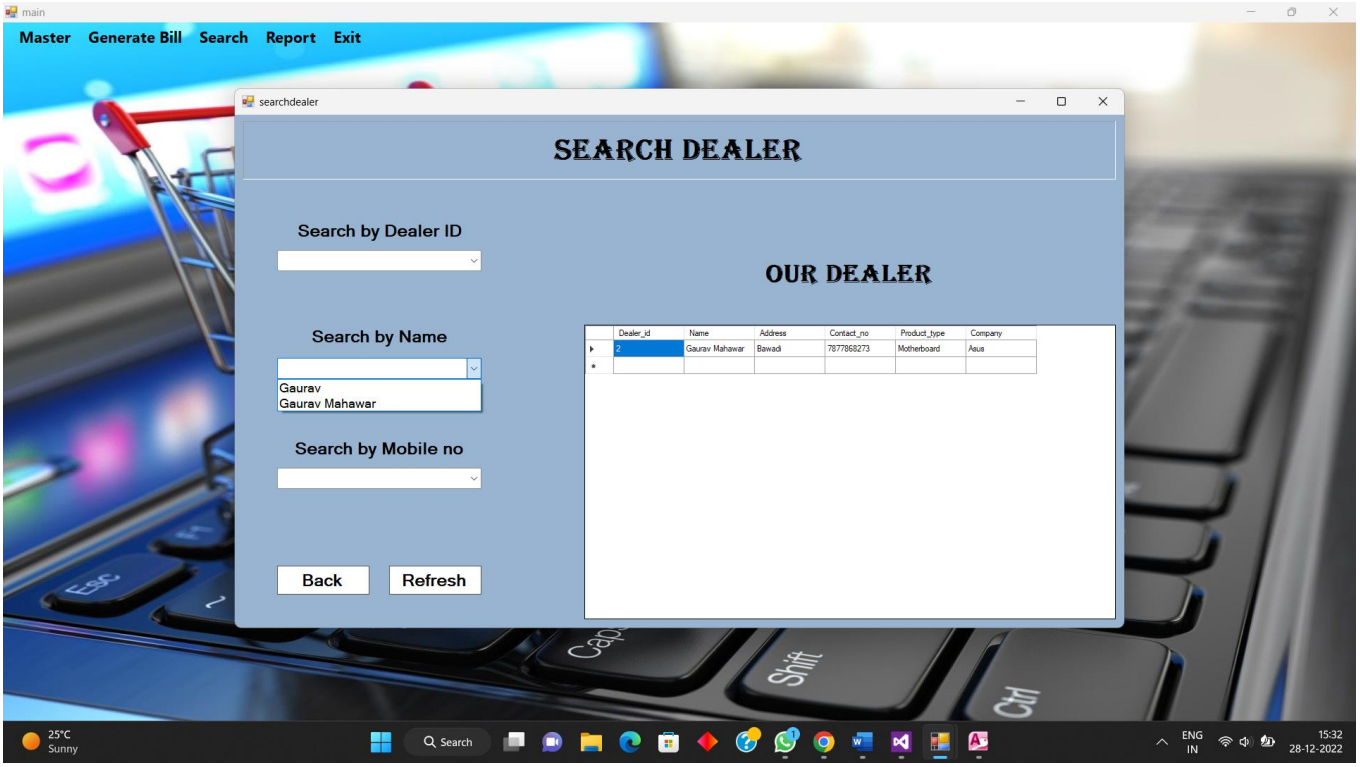
IX. Generate Bill Form



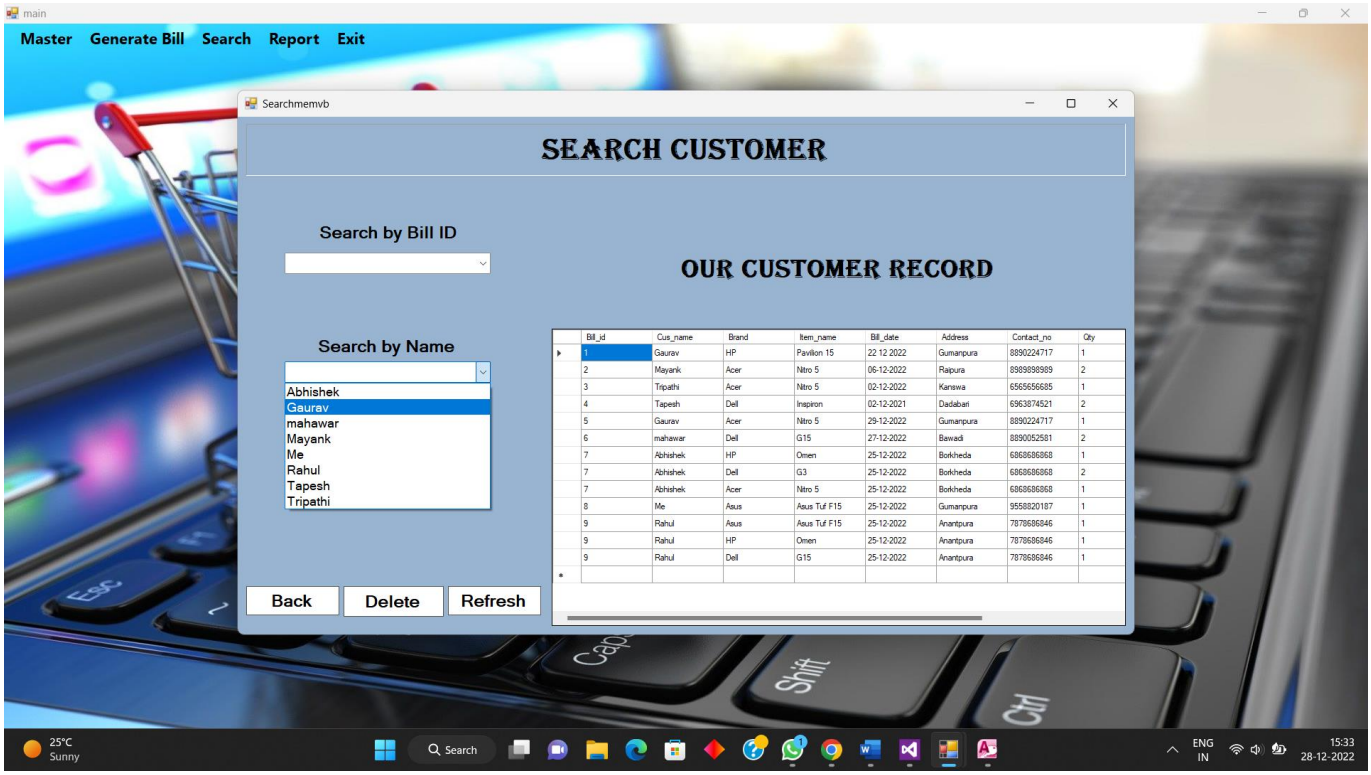
X. Search Ordered Item Form(with delete order)



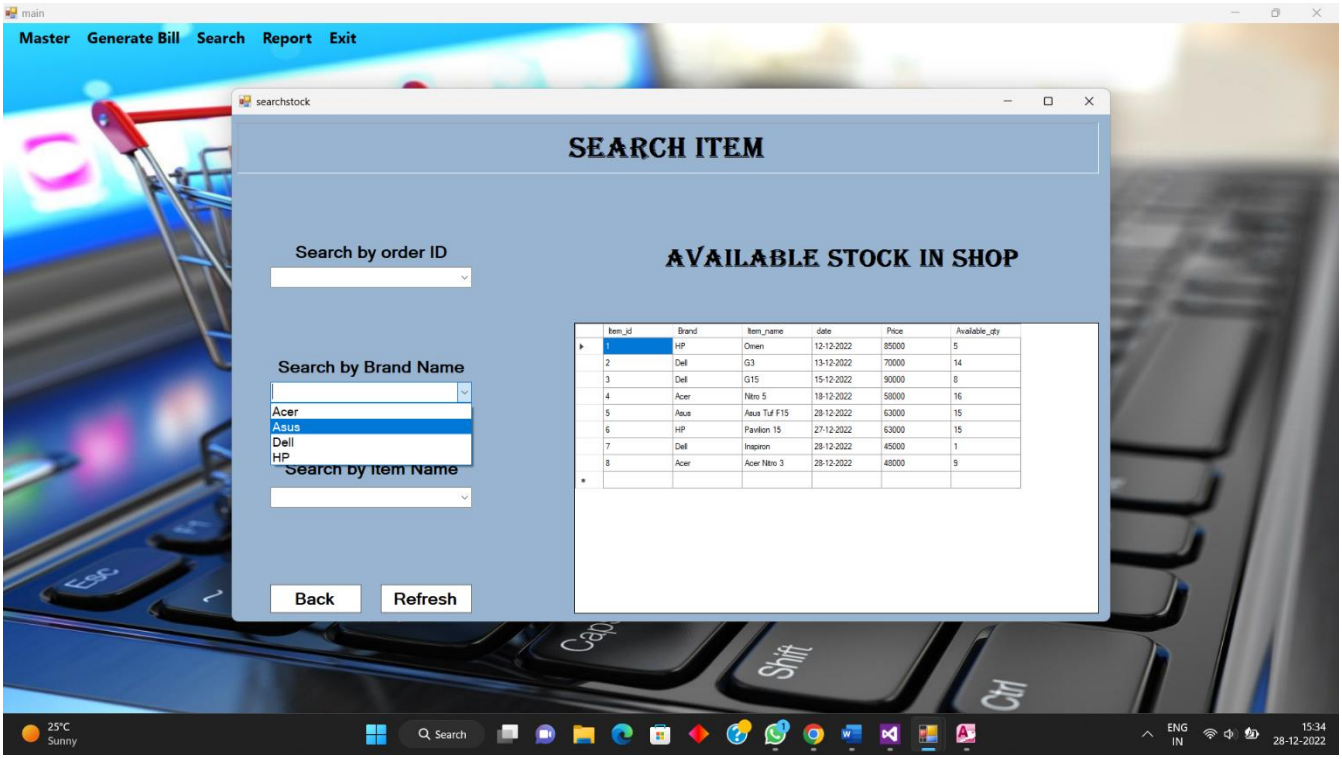
XI. Search Dealer Form



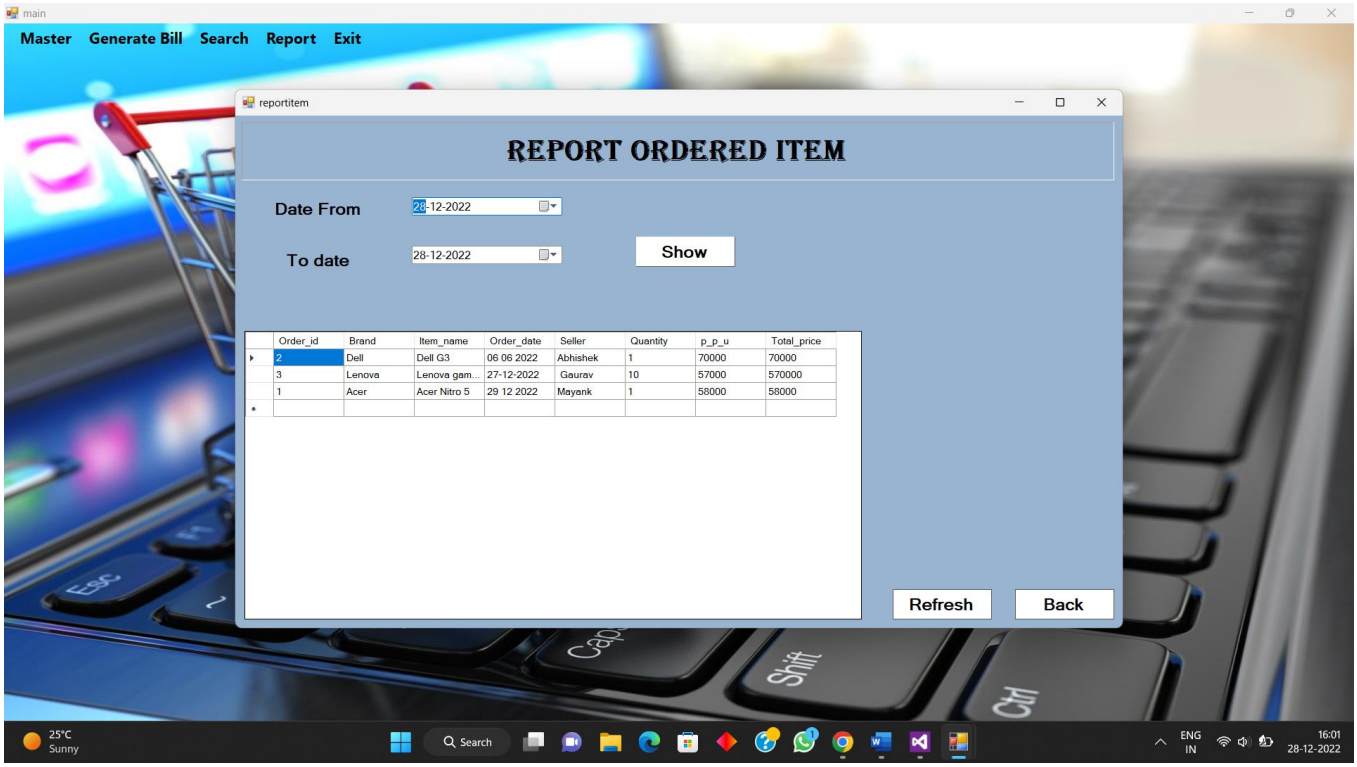
XII. Search Customer Form(with delete customer)



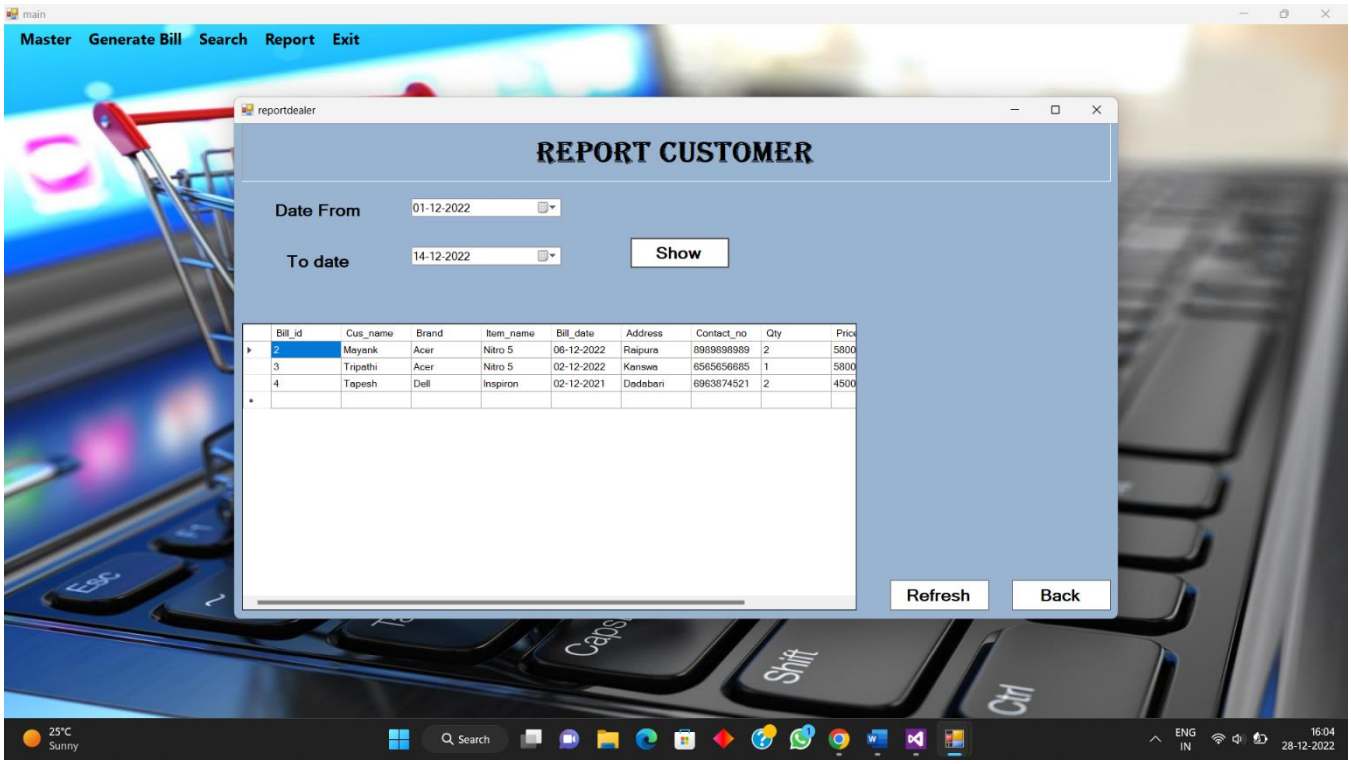
XIII. Search Stock Form



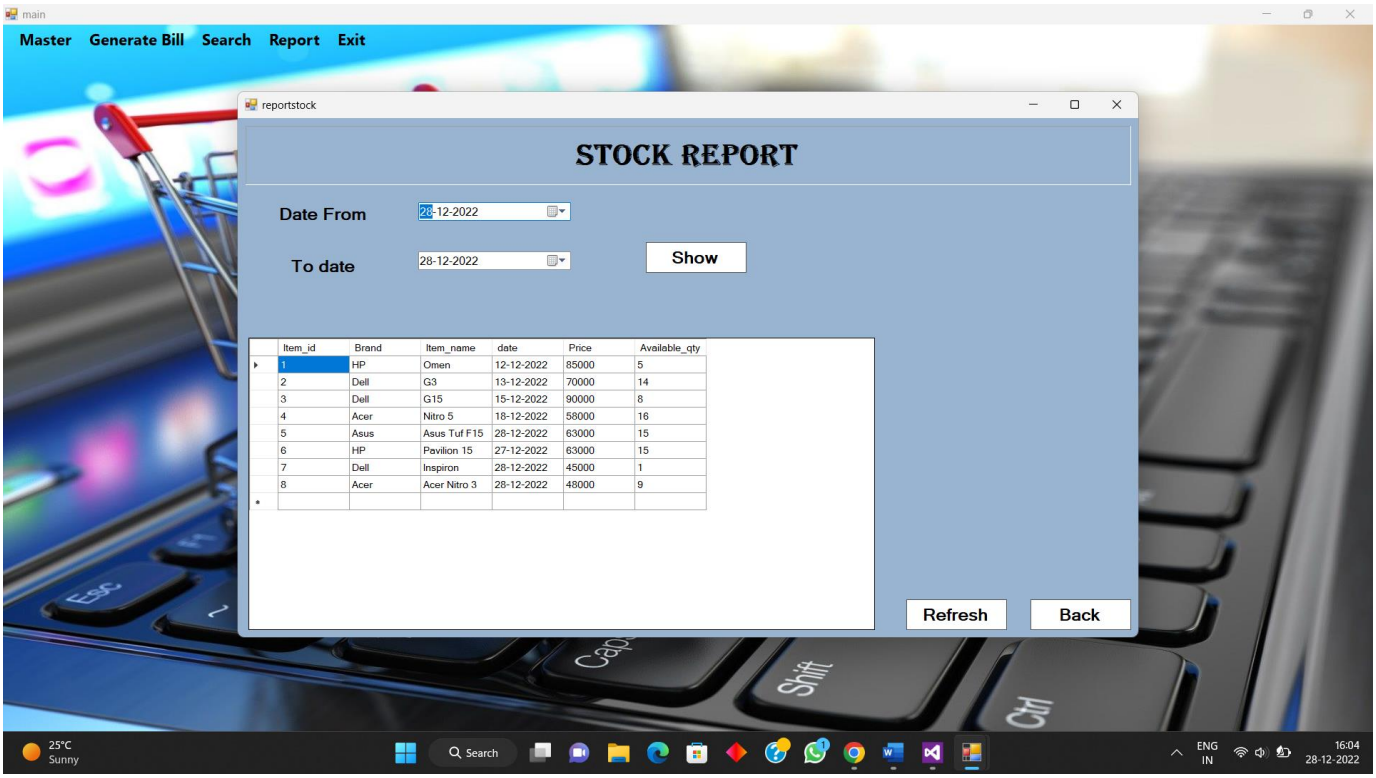
XIV. Report of Order item Form



XV. Report of Customer Form



XVI. Report of Stock Form



Chapter 4

CODING

Class coding :-

```
Imports System.Data
Imports System.Data.OleDb
Imports System.Text.RegularExpressions

Public Class Class1

    Dim e As KeyPressEventArgs

    Public Function sele(ByVal s As String) As DataTable
        Dim adp As New OleDbDataAdapter(s, "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\hp\OneDrive\Documents\Visual Studio
2012\Projects\Minor\Minor\Minor.accdb")
        Dim dt As New DataTable()
        adp.Fill(dt)
        sele = dt
    End Function

    Public Function ins(ByVal s As String)
        Dim adp As New OleDbDataAdapter(s, "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\hp\OneDrive\Documents\Visual Studio nnb
77782012\Projects\Minor\Minor\Minor.accdb")
        Dim dt As New DataTable()
        adp.Fill(dt)
        ins = dt
    End Function

    Function valmob(ByVal m As String) As Boolean
        Return m.Length = 10 And Regex.IsMatch(m, "^[1-9]")
    End Function

    Function valnum(ByVal no As String) As Boolean
        Return Regex.IsMatch(no, "^[1-9]")
    End Function
End Class
```

Login form coding :-

```
Imports System.Data
Imports System.Data.OleDb

Public Class login

    Dim obj As New Class1
    Dim dt As DataTable
    Dim i As Integer, f As Integer = 0

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        For i = 0 To dt.Rows.Count - 1 Step 1
            If (TextBox1.Text = dt.Rows(i)(0) And TextBox2.Text = dt.Rows(i)(1)) Then
                f = 1
            Else
                f = 0
            End If
        Next

        If (f = 0) Then
            MessageBox.Show("Invalid username and password")
        Else
            Me.Hide()
            flash.Hide()
            main.Show()
        End If
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Application.Exit()
    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        TextBox2.PasswordChar = ""
        Button3.Visible = True
        Button4.Visible = False
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        TextBox2.PasswordChar = "*"
        Button4.Visible = True
```

```
        Button3.Visible = False
    End Sub

    Private Sub login_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select * from Login")
    End Sub
End Class
```

Main menu form coding :-

```
Public Class main
    Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ExitToolStripMenuItem.Click
        Application.Exit()
    End Sub

    Private Sub AddToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
AddToolStripMenuItem.Click
        order_item.Show()
    End Sub

    Private Sub CustomerToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles CustomerToolStripMenuItem.Click
        Searchcus.Show()
    End Sub

    Private Sub DealerToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles DealerToolStripMenuItem.Click
        searchdealer.Show()
    End Sub

    Private Sub StockToolStripMenuItem1_Click(sender As Object, e As EventArgs) Handles
StockToolStripMenuItem1.Click
        Searchitem.Show()
    End Sub

    Private Sub AddDealerToolStripMenuItem1_Click(sender As Object, e As EventArgs)
Handles AddDealerToolStripMenuItem1.Click
        Add_dealer.Show()
    End Sub
```

```
    Private Sub AddSalesToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles AddSalesToolStripMenuItem.Click
        Add_cus.Show()
    End Sub

    Private Sub GenerateBillToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles GenerateBillToolStripMenuItem.Click
        Form1.Show()
    End Sub

    Private Sub UpdateToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
UpdateToolStripMenuItem.Click
        Searchitem.Show()
    End Sub

    Private Sub UpdateDealerToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles UpdateDealerToolStripMenuItem.Click
        Updatedealer.Show()
    End Sub

    Private Sub OrderToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
OrderToolStripMenuItem.Click
        reportitem.Show()
    End Sub

    Private Sub CustomerToolStripMenuItem1_Click(sender As Object, e As EventArgs)
Handles CustomerToolStripMenuItem1.Click
        reportcus.Show()
    End Sub

    Private Sub UpdateSalesToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles UpdateSalesToolStripMenuItem.Click
        Searchcus.Show()
    End Sub

    Private Sub AddStockToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles AddStockToolStripMenuItem.Click
        addstock.Show()
    End Sub

    Private Sub ToolStripMenuItem2_Click(sender As Object, e As EventArgs) Handles
ToolStripMenuItem2.Click
```

```
        searchstock.Show()
    End Sub

    Private Sub UpdateStockToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles UpdateStockToolStripMenuItem.Click
        updatestock.Show()
    End Sub

    Private Sub StockToolStripMenuItem2_Click(sender As Object, e As EventArgs) Handles
StockToolStripMenuItem2.Click
        reportstock.Show()
    End Sub

    Private Sub main_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    End Sub
End Class
```

Order Item form coding :-

```
Imports System.Data
Imports System.Data.OleDb
Imports System.Text.RegularExpressions

    Public Class order_item

        Dim obj As New Class1
        Dim dt As New DataTable()
        Dim dt1 As New DataTable()
        Dim i As Integer
        Dim f As Boolean
        Dim a As Integer = 0

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click

            If String.IsNullOrEmpty(TextBox1.Text.Trim) Then
                ErrorProvider1.SetError(TextBox1, "Please fill epmty slot")

            ElseIf String.IsNullOrEmpty(TextBox3.Text.Trim) Then
                ErrorProvider1.SetError(TextBox3, "Please fill empty slot")
```

```
            ElseIf String.IsNullOrEmpty(TextBox4.Text.Trim) Then
                ErrorProvider1.SetError(TextBox4, "Please fill empty slot")

            ElseIf String.IsNullOrEmpty(TextBox6.Text.Trim) Then
                ErrorProvider1.SetError(TextBox6, "Please fill empty slot")

        Else
            dt = obj.sele("select * from gaurav")
        Dim flag As Integer = 0
            For i = 0 To dt.Rows.Count - 1
                If (TextBox1.Text = dt.Rows(i)(0)) Then
                    flag = 0
                    MessageBox.Show("Item id is same to correct id press 'Alt+p'")
                    Return
                Else
                    flag = 1
                End If
            Next

            If (flag = 0) Then
                MessageBox.Show("Item id is same to correct id press 'Alt+p'")
            Else
                obj.ins("insert into gaurav values (" + TextBox1.Text + ",'" +
TextBox7.Text + "','" + ComboBox3.SelectedItem + "','" + DateTimePicker1.Text + "','" +
TextBox3.Text + "'," + TextBox4.Text + "," + TextBox5.Text + "," + TextBox6.Text + ")")
                MessageBox.Show("Record saved successfully")
                dt = obj.sele("select * from gaurav")
                DataGridView1.DataSource = dt
                a = dt.Rows.Count - 1
                TextBox1.Text = dt.Rows(a)(0) + 1
                ComboBox3.Text = ""
                TextBox3.Text = ""
                TextBox4.Text = ""
                TextBox5.Text = ""
                TextBox6.Text = ""
                TextBox7.Text = ""
                DateTimePicker1.Text = ""
            End If
            End If
        End Sub
```

```
        Private Sub TextBox5_Leave(sender As Object, e As EventArgs) Handles
TextBox5.Leave
            f = obj.valnum(TextBox5.Text)
            If (f = False) Then
                MessageBox.Show("Cannot enter character")
                TextBox5.Text = ""
            ElseIf TextBox4.Text = "" Or TextBox5.Text = "" Then
                MessageBox.Show("Please fill Quantity or price")
            Else
                TextBox6.Text        =        (Convert.ToInt64(TextBox4.Text)        *
Convert.ToInt64(TextBox5.Text)).ToString
            End If
        End Sub

        Private Sub Add_Item_Load(sender As Object, e As EventArgs) Handles MyBase.Load
            dt = obj.sele("select * from gaurav")
            DataGridView1.DataSource = dt

            a = dt.Rows.Count - 1
            TextBox1.Text = dt.Rows(a)(0) + 1

            dt = obj.sele("select distinct item_name from gaurav")

            dt1 = obj.sele("select * from Dealer")
            For i = 0 To dt1.Rows.Count - 1 Step 1
                ComboBox3.Items.Add(dt1.Rows(i)(1))
            Next
            TextBox1.Enabled = False
        End Sub

        Private Sub TextBox2_KeyPress(sender As Object, e As KeyPressEventArgs)
            If Not Char.IsLetter(e.KeyChar) And Not e.KeyChar = Chr(Keys.Delete) And
Not e.KeyChar = Chr(Keys.Back) And Not e.KeyChar = Chr(Keys.Space) Then
                e.Handled = True
                MessageBox.Show("Wrong input")
            End If
        End Sub

        Private Sub TextBox4_Leave(sender As Object, e As EventArgs) Handles
TextBox4.Leave
            f = obj.valnum(TextBox4.Text)
            If (f = False) Then
```

```
                MessageBox.Show("Wrong input in price")
                TextBox4.Text = ""
            End If
        End Sub

        Private Sub Button5_Click(sender As Object, e As EventArgs) Handles
Button5.Click
            Me.Hide()
        End Sub

        Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
            dt = obj.sele("select * from gaurav")
            DataGridView1.DataSource = dt
            a = dt.Rows.Count - 1
            TextBox1.Text = dt.Rows(a)(0) + 1
            ComboBox3.Text = ""
            TextBox3.Text = ""
            TextBox4.Text = ""
            TextBox5.Text = ""
            TextBox6.Text = ""
            DateTimePicker1.Text = ""
            TextBox1.Enabled = False
        End Sub

        Private Sub Button4_Click(sender As Object, e As EventArgs) Handles
Button4.Click
            Searchitem.Show()
        End Sub
        Dim c As Integer = 0
        Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
            c = c + 1
            TextBox1.Enabled = True
        End Sub
    End Class
```

Add Dealer form coding :-

```
Imports System.Data
Imports System.Data.OleDb
```

```
Public Class Add_dealer

    Dim obj As New Class1
    Dim dt As New DataTable
    Dim i As Integer
    Dim c As Integer = 0
    Dim f As Boolean

    Private Sub Add_dealer_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select * from Dealer")
        DataGridView1.DataSource = dt

        c = dt.Rows.Count - 1
        TextBox1.Text = dt.Rows(c)(0) + 1
    End Sub

    Private Sub TextBox2_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox2.KeyPress
        If Not Char.IsLetter(e.KeyChar) And Not e.KeyChar = Chr(Keys.Delete) And Not
e.KeyChar = Chr(Keys.Back) And Not e.KeyChar = Chr(Keys.Space) Then
            e.Handled = True
            MessageBox.Show("Wrong input")
        End If
    End Sub

    Private Sub TextBox5_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox5.KeyPress
        If Not Char.IsLetter(e.KeyChar) And Not e.KeyChar = Chr(Keys.Delete) And Not
e.KeyChar = Chr(Keys.Back) And Not e.KeyChar = Chr(Keys.Space) Then
            e.Handled = True
            MessageBox.Show("Wrong input")
        End If
    End Sub

    Private Sub TextBox4_Leave(sender As Object, e As EventArgs) Handles TextBox4.Leave
        f = obj.valmob(TextBox4.Text)
        If (f = False) Then
            MessageBox.Show("Wrong mobile no.")
            TextBox4.Text = ""
        End If
    End Sub
```

```
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        If (TextBox1.Text <> "" And TextBox2.Text <> "" And TextBox3.Text <> "" And
TextBox4.Text <> "" And TextBox5.Text <> "" And TextBox6.Text <> "") Then
            dt = obj.sele("insert into Dealer values (" + TextBox1.Text + ",'" +
TextBox2.Text + "','" + TextBox3.Text + "'," + TextBox4.Text + ",'" + TextBox5.Text +
"','" + TextBox6.Text + "')")
            MessageBox.Show("Record saved")
            dt = obj.sele("select * from Dealer")
            DataGridView1.DataSource = dt
            c = dt.Rows.Count - 1
            TextBox1.Text = dt.Rows(c)(0) + 1
            TextBox2.Text = ""
            TextBox3.Text = ""
            TextBox4.Text = ""
            TextBox5.Text = ""
            TextBox6.Text = ""
        Else
            MessageBox.Show("Please fill all information")
        End If
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        dt = obj.sele("select * from Dealer")
        DataGridView1.DataSource = dt
        c = dt.Rows.Count - 1
        TextBox1.Text = dt.Rows(c)(0) + 1
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        Me.Hide()
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Updatedealer.Show()
    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
```

```
        Updatedealer.Show()
    End Sub
End Class
```

Update dealer form coding :-

```
Public Class Updatedealer

    Dim obj As New Class1
    Dim dt As New DataTable()
    Dim i As Integer
    Dim f As Boolean

    Private Sub Updatedealer_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select * from Dealer")
        DataGridView1.DataSource = dt
        com()
    End Sub

    Public Function com()
        dt = obj.sele("select * from Dealer")
        ComboBox1.Items.Clear()
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox1.Items.Add(dt.Rows(i)(0))
        Next
    End Function

    Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox1.SelectedIndexChanged
        dt = obj.sele("select * from Dealer where Dealer_id = " + ComboBox1.Text + " ")
        TextBox1.Text = dt.Rows(0)(1)
        TextBox2.Text = dt.Rows(0)(2)
        TextBox3.Text = dt.Rows(0)(3)
        TextBox4.Text = dt.Rows(0)(4)
        TextBox5.Text = dt.Rows(0)(5)
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        If (TextBox1.Text <> "" And TextBox2.Text <> "" And TextBox3.Text <> "" And
TextBox4.Text <> "" And TextBox5.Text <> "" And ComboBox1.Text <> "") Then
```

```
            dt = obj.ins("Update Dealer set Name = '" + TextBox1.Text + "', Address =
'" + TextBox2.Text + "', Contact_no = " + TextBox3.Text + ",Product_type = '" +
TextBox4.Text + "' where Dealer_id = " + ComboBox1.Text + " ")
            dt = obj.sele("select * from Dealer")
            MessageBox.Show("Record updated successfully")
            DataGridView1.DataSource = dt
            TextBox1.Text = ""
            TextBox2.Text = ""
            TextBox4.Text = ""
            TextBox5.Text = ""
            TextBox3.Text = ""
            ComboBox1.ResetText()
            com()
        Else
            MessageBox.Show("Please fill all information")
        End If
    End Sub

    Private Sub TextBox4_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox4.KeyPress
        If Not Char.IsLetter(e.KeyChar) And Not e.KeyChar = Chr(Keys.Delete) And Not
e.KeyChar = Chr(Keys.Back) And Not e.KeyChar = Chr(Keys.Space) Then
            e.Handled = True
            MessageBox.Show("Wrong input")
        End If
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        dt = obj.sele("select * from Dealer")
        DataGridView1.DataSource = dt
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox4.Text = ""
        TextBox3.Text = ""
        TextBox5.Text = ""
        ComboBox1.ResetText()
        com()
    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        If (ComboBox1.Text = "") Then
            MessageBox.Show("Select Id first")
```

```
        Else
            dt = obj.ins("delete from Dealer where Dealer_id = " + ComboBox1.Text + "
")
            MessageBox.Show("Record Deleted successfully")
            dt = obj.sele("select * from Dealer")
            DataGridView1.DataSource = dt
            ComboBox1.ResetText()
            com()
        End If
    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        Me.Hide()
    End Sub

    Private Sub TextBox1_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox1.KeyPress
        If Not Char.IsLetter(e.KeyChar) And Not e.KeyChar = Chr(Keys.Delete) And Not
e.KeyChar = Chr(Keys.Back) And Not e.KeyChar = Chr(Keys.Space) Then
            e.Handled = True
            MessageBox.Show("Wrong input")
        End If
    End Sub

    Private Sub TextBox5_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox5.KeyPress
        If Not Char.IsLetter(e.KeyChar) And Not e.KeyChar = Chr(Keys.Delete) And Not
e.KeyChar = Chr(Keys.Back) And Not e.KeyChar = Chr(Keys.Space) Then
            e.Handled = True
            MessageBox.Show("Wrong input")
        End If
    End Sub

    Private Sub TextBox3_Leave(sender As Object, e As EventArgs) Handles TextBox3.Leave
        f = obj.valmob(TextBox3.Text)
        If (f = False) Then
            MessageBox.Show("Wrong mobile no.")
            TextBox3.Text = ""
        End If
    End Sub

End Class
```

Customer form(sale form) coding : -

```
Imports System.Data
Imports System.Data.OleDb

Public Class Add_cus

    Dim obj As New Class1
    Dim dt As New DataTable
    Dim i As Integer
    Dim c As Integer = 0
    Dim dt1 As New DataTable
    Dim f As Boolean

    Private Sub Add_cus_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select * from Stock")

        dt = obj.sele("select distinct Brand from Stock")
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox1.Items.Add(dt.Rows(i)(0))
        Next

        dt1 = obj.sele("select * from Bill")
        DataGridView1.DataSource = dt1
        c = dt1.Rows.Count - 1
        TextBox1.Text = dt1.Rows(c)(0) + 1

    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox1.SelectedIndexChanged
        dt = obj.sele("select * from Stock where Brand = '" + ComboBox1.Text + "' ")
        ComboBox2.Items.Clear()
        ComboBox2.Text = ""
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox2.Items.Add(dt.Rows(i)(2))
        Next
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        For Each item As ListViewItem In ListView1.Items
```

```
            Dim conn As New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\hp\OneDrive\Documents\Visual                                    Studio
2012\Projects\Minor\Minor\Minor.accdb")
            conn.Open()
            Dim     cmd     As     New     OleDbCommand("insert     into     Bill     values
(@id,@cus_nm,@br,@it_nm,@date,@add,@mob,@qty,@pr,@tot)", conn)

            cmd.Parameters.AddWithValue("@id", item.SubItems(0).Text)
            cmd.Parameters.AddWithValue("@cus_nm", item.SubItems(1).Text)
            cmd.Parameters.AddWithValue("@br", item.SubItems(2).Text)
            cmd.Parameters.AddWithValue("@it_nm", item.SubItems(3).Text)
            cmd.Parameters.AddWithValue("@date", item.SubItems(4).Text)
            cmd.Parameters.AddWithValue("@add", item.SubItems(5).Text)
            cmd.Parameters.AddWithValue("@mob", item.SubItems(6).Text)
            cmd.Parameters.AddWithValue("@qty", item.SubItems(7).Text)
            cmd.Parameters.AddWithValue("@pr", item.SubItems(8).Text)
            cmd.Parameters.AddWithValue("@tot", item.SubItems(9).Text)
            cmd.ExecuteNonQuery()
            conn.Close()
            ListView1.Items.Clear()
        Next
        MsgBox("All item saved succesfully")

        dt1 = obj.sele("select * from Bill")
        DataGridView1.DataSource = dt1
        c = dt1.Rows.Count - 1
        TextBox1.Text = dt1.Rows(c)(0) + 1
        TextBox2.Enabled = True
        TextBox3.Enabled = True
        TextBox4.Enabled = True
        TextBox5.Enabled = True
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
        ComboBox1.Text = ""
        ComboBox2.Text = ""
        ComboBox2.Items.Clear()
        DateTimePicker1.Text = ""
    End Sub
```

```
    Private Sub ComboBox2_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox2.SelectedIndexChanged
        dt = obj.sele("select * from Stock where Item_name = '" + ComboBox2.SelectedItem
+ "' ")
        For i = 0 To dt.Rows.Count - 1 Step 1
            TextBox6.Text = dt.Rows(i)(4)
        Next
    End Sub

    Private Sub TextBox5_Leave(sender As Object, e As EventArgs) Handles TextBox5.Leave
        f = obj.valnum(TextBox5.Text)
        If (f = False) Then
            MessageBox.Show("Please fill quantity")
        Else
            If (TextBox6.Text = "") Then
                MessageBox.Show("Please select item")
            ElseIf (TextBox5.Text <> "") Then
                TextBox7.Text = Val(TextBox5.Text * TextBox6.Text).ToString
            Else
                MessageBox.Show("Please fill quantity")
            End If
        End If
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        dt1 = obj.sele("select * from Bill")
        DataGridView1.DataSource = dt1
        c = dt1.Rows.Count - 1
        TextBox1.Text = dt1.Rows(c)(0) + 1
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
        ComboBox1.Text = ""
        ComboBox2.Text = ""
        ComboBox2.Items.Clear()
        DateTimePicker1.Text = ""
    End Sub
```

```
    Private Sub TextBox2_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox2.KeyPress
        If Not Char.IsLetter(e.KeyChar) And Not e.KeyChar = Chr(Keys.Delete) And Not
e.KeyChar = Chr(Keys.Back) And Not e.KeyChar = Chr(Keys.Space) Then
            e.Handled = True
            MessageBox.Show("Wrong name")
        End If
    End Sub

    Private Sub TextBox4_Leave(sender As Object, e As EventArgs) Handles TextBox4.Leave
        f = obj.valmob(TextBox4.Text)
        If (f = False) Then
            MessageBox.Show("Wrong mobile no.")
            TextBox4.Text = ""
        End If
    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        Me.Hide()
    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        Searchcus.Show()
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        If (TextBox1.Text <> "" And TextBox2.Text <> "" And TextBox3.Text <> "" And
TextBox4.Text <> "" And TextBox5.Text <> "" And TextBox6.Text <> "" And TextBox7.Text
<> "" And ComboBox1.Text <> "" And ComboBox2.Text <> "") Then
            Dim item As New ListViewItem(TextBox1.Text)
            item.SubItems.Add(TextBox2.Text)
            item.SubItems.Add(ComboBox1.SelectedItem)
            item.SubItems.Add(ComboBox2.SelectedItem)
            item.SubItems.Add(DateTimePicker1.Value.ToShortDateString)
            item.SubItems.Add(TextBox3.Text)
            item.SubItems.Add(TextBox4.Text)
            item.SubItems.Add(TextBox5.Text)
            item.SubItems.Add(TextBox6.Text)
            item.SubItems.Add(TextBox7.Text)

            ListView1.Items.Add(item)
            TextBox2.Enabled = False
```

```
            TextBox4.Enabled = False
            TextBox3.Enabled = False
            ComboBox2.ResetText()
            TextBox5.Text = ""
            TextBox6.Text = ""
            TextBox7.Text = ""
            Button1.Enabled = True
        Else
            MessageBox.Show("Please fill all information")
        End If
    End Sub

End Class
```

Add stock form coding :-

```
Public Class addstock

    Dim obj As New Class1
    Dim dt As New DataTable
    Dim i As Integer
    Dim c As Integer = 0
    Dim dt1 As New DataTable
    Dim f As Boolean

    Private Sub addstock_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select * from Stock")
        DataGridView1.DataSource = dt
        dt1 = obj.sele("select distinct Brand from gaurav")
        ComboBox1.ResetText()
        ComboBox1.Items.Clear()
        For i = 0 To dt1.Rows.Count - 1 Step 1
            ComboBox1.Items.Add(dt1.Rows(i)(0))
        Next
        c = dt.Rows.Count - 1
        If (c <= 0) Then
            TextBox1.Text = 1
        Else
            TextBox1.Text = dt.Rows(c)(0) + 1
        End If
    End Sub
```

```
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        If (TextBox1.Text <> "" And TextBox2.Text <> "" And TextBox3.Text <> "" And
ComboBox1.Text <> "" And ComboBox2.Text <> "") Then
            dt = obj.ins("insert into Stock values (" + TextBox1.Text + ",'" +
ComboBox1.SelectedItem + "','" + ComboBox2.SelectedItem + "','" +
DateTimePicker1.Value.ToShortDateString + "'," + TextBox2.Text + "," + TextBox3.Text +
")")
            Dim d As Integer
            dt1 = obj.sele("select * from gaurav where Item_name = '" +
ComboBox2.SelectedItem.ToString + "' ")
            For i = 0 To dt1.Rows.Count - 1 Step 1
                d = dt1.Rows(i)(5)
            Next
            d = d - TextBox3.Text
            If (d > 0) Then
                dt1 = obj.ins("Update gaurav set Quantity = " + d.ToString + " where
Item_name = '" + ComboBox2.SelectedItem.ToString + "' ")
            Else
                dt1 = obj.ins("delete from gaurav where Item_name = '" +
ComboBox2.SelectedItem.ToString + "' ")
            End If
            dt = obj.sele("select * from Stock")
            DataGridView1.DataSource = dt
            MessageBox.Show("Record saved successfully")
            c = dt.Rows.Count - 1
            TextBox1.Text = dt.Rows(c)(0) + 1
            TextBox2.Text = ""
            TextBox3.Text = ""
            DateTimePicker1.Text = ""
            ComboBox1.Text = ""
            ComboBox2.Text = ""
        Else
            MessageBox.Show("Please Fill all information")
        End If
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox1.SelectedIndexChanged
        dt1 = obj.sele("select * from gaurav where Brand = '" + ComboBox1.SelectedItem
+ "' ")
        ComboBox2.Items.Clear()
        ComboBox2.Text = ""
```

```
        For i = 0 To dt1.Rows.Count - 1 Step 1
            ComboBox2.Items.Add(dt1.Rows(i)(2))
        Next
    End Sub

    Private Sub ComboBox2_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox2.SelectedIndexChanged
        dt1 = obj.sele("select * from gaurav where Item_name = '" +
ComboBox2.SelectedItem + "' ")
        For i = 0 To dt1.Rows.Count - 1 Step 1
            TextBox2.Text = dt1.Rows(i)(6)
            TextBox3.Text = dt1.Rows(i)(5)
        Next
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        dt1 = obj.sele("select distinct Brand from gaurav")
        ComboBox1.Items.Clear()
        ComboBox1.ResetText()
        For i = 0 To dt1.Rows.Count - 1 Step 1
            ComboBox1.Items.Add(dt1.Rows(i)(0))
        Next
        dt = obj.sele("select *  from Stock")
        DataGridView1.DataSource = dt
        c = dt.Rows.Count - 1
        TextBox1.Text = dt.Rows(c)(0) + 1
        TextBox2.Text = ""
        DateTimePicker1.Text = ""
        TextBox3.Text = ""
        ComboBox2.ResetText()
        ComboBox2.Items.Clear()
    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        Me.Hide()
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        updatestock.Show()
    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
```

```
        updatestock.Show()
    End Sub

    Private Sub TextBox2_Leave(sender As Object, e As EventArgs) Handles TextBox2.Leave
        f = obj.valnum(TextBox2.Text)
        If (f = False) Then
            MessageBox.Show("Enter valid number")
            TextBox2.Text = ""
        End If
    End Sub

    Private Sub TextBox3_Leave(sender As Object, e As EventArgs) Handles TextBox3.Leave
        f = obj.valnum(TextBox3.Text)
        If (f = False) Then
            MessageBox.Show("Enter valid number")
            TextBox3.Text = ""
        End If
    End Sub
End Class
```

Update stock form coding :-

```
Public Class updatestock

    Dim obj As New Class1
    Dim dt As New DataTable
    Dim i As Integer
    Dim c As Integer = 0
    Dim dt1 As New DataTable
    Dim f As Boolean

    Private Sub updatestock_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select * from Stock")
        DataGridView1.DataSource = dt
        ComboBox1.Items.Clear()
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox1.Items.Add(dt.Rows(i)(0))
        Next
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        dt = obj.sele("select * from Stock")
```

```
        ComboBox1.Items.Clear()
        DataGridView1.DataSource = dt
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox1.Items.Add(dt.Rows(i)(0))
        Next
        TextBox1.Text = ""
        TextBox2.Text = ""
        DateTimePicker1.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        ComboBox1.ResetText()
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        If (ComboBox1.Text <> "" And TextBox1.Text <> "" And TextBox2.Text <> "" And
TextBox3.Text <> "" And TextBox4.Text <> "") Then
            dt = obj.ins("Update Stock set Price = " + TextBox3.Text + ", Available_qty
= " + TextBox4.Text + " where Item_id = " + ComboBox1.SelectedItem.ToString + " ")
            dt = obj.sele("select * from Stock")
            DataGridView1.DataSource = dt
            ComboBox1.Items.Clear()
            For i = 0 To dt.Rows.Count - 1 Step 1
                ComboBox1.Items.Add(dt.Rows(i)(0))
            Next
            TextBox1.Text = ""
            TextBox2.Text = ""
            DateTimePicker1.Text = ""
            TextBox3.Text = ""
            TextBox4.Text = ""
            ComboBox1.ResetText()
        Else
            MessageBox.Show("Pleas fill all information")
        End If
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox1.SelectedIndexChanged
        dt = obj.sele("select * from Stock where Item_id = " +
ComboBox1.SelectedItem.ToString + " ")
        For i = 0 To dt.Rows.Count - 1 Step 1
            TextBox1.Text = dt.Rows(i)(1)
            TextBox2.Text = dt.Rows(i)(2)
```

```
            TextBox3.Text = dt.Rows(i)(4)
            DateTimePicker1.Value = dt.Rows(i)(3)
            TextBox4.Text = dt.Rows(i)(5)
        Next
    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        Me.Hide()
    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        If (ComboBox1.Text = "") Then
            MessageBox.Show("Select Id first")
        Else
            dt    =    obj.sele("delete    from    Stock    where    Item_id    =    "    +
ComboBox1.SelectedItem.ToString + " ")
            dt = obj.sele("select * from Stock")
            ComboBox1.Items.Clear()
            DataGridView1.DataSource = dt
            For i = 0 To dt.Rows.Count - 1 Step 1
                ComboBox1.Items.Add(dt.Rows(i)(0))
            Next
            TextBox1.Text = ""
            TextBox2.Text = ""
            DateTimePicker1.Text = ""
            TextBox3.Text = ""
            TextBox4.Text = ""
            ComboBox1.ResetText()
        End If
    End Sub

    Private Sub TextBox3_Leave(sender As Object, e As EventArgs) Handles TextBox3.Leave
        f = obj.valnum(TextBox3.Text)
        If (f = False) Then
            MessageBox.Show("Invalid price")
            TextBox3.Text = ""
        End If
    End Sub

    Private Sub TextBox4_Leave(sender As Object, e As EventArgs) Handles TextBox4.Leave
        f = obj.valnum(TextBox4.Text)
        If (f = False) Then
```

```
            MessageBox.Show("Invalid price")
            TextBox4.Text = ""
        End If
    End Sub

    Private Sub TextBox1_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox1.KeyPress
        If Not Char.IsLetter(e.KeyChar) And Not e.KeyChar = Chr(Keys.Delete) And Not
e.KeyChar = Chr(Keys.Back) And Not e.KeyChar = Chr(Keys.Space) Then
            e.Handled = True
            MessageBox.Show("Name sahi daliye")
        End If
    End Sub

    Private Sub TextBox2_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox2.KeyPress
        If Not Char.IsLetter(e.KeyChar) And Not e.KeyChar = Chr(Keys.Delete) And Not
e.KeyChar = Chr(Keys.Back) And Not e.KeyChar = Chr(Keys.Space) Then
            e.Handled = True
            MessageBox.Show("Name sahi daliye")
        End If
    End Sub
End Class
```

Generate bill form coding :-

```
Public Class Form1

    Dim dt As New DataTable
    Dim obj As New Class1
    Dim cr As New CrystalReport1
    Dim i As Integer

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select distinct Bill_id from Bill")
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox1.Items.Add(dt.Rows(i)(0))
        Next

        dt = obj.sele("select distinct Contact_no from Bill")
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox2.Items.Add(dt.Rows(i)(0))
```

```
        Next
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox1.SelectedIndexChanged
        dt = obj.sele("select * from Bill where Bill_id = " + ComboBox1.Text + " ")
        cr.SetDataSource(dt)
        CrystalReportViewer1.ReportSource = cr
        ComboBox2.ResetText()
    End Sub

    Private Sub ComboBox2_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox2.SelectedIndexChanged
        dt = obj.sele("select * from Bill where Contact_no = " + ComboBox2.Text + " ")
        cr.SetDataSource(dt)
        CrystalReportViewer1.ReportSource = cr
        ComboBox1.ResetText()
    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        Me.Hide()
    End Sub
End Class
```

Search ordered item and delete item form coding :-

```
Public Class Searchitem

    Dim obj As New Class1, dt As New DataTable(), i As Integer

    Private Sub Search_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select * from gaurav")
        DataGridView1.DataSource = dt
            For i = 0 To dt.Rows.Count - 1 Step 1
                ComboBox1.Items.Add(dt.Rows(i)(0))
                ComboBox3.Items.Add(dt.Rows(i)(2))
            Next
        dt = obj.sele("select distinct Brand from gaurav")
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox2.Items.Add(dt.Rows(i)(0))
        Next
    End Sub
```

```
    Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox1.SelectedIndexChanged
        dt = obj.sele("select * from gaurav where Order_id = " + ComboBox1.Text + " ")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub ComboBox2_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox2.SelectedIndexChanged
        ComboBox1.ResetText()
        dt = obj.sele("select * from gaurav where Brand = '" + ComboBox2.Text + "' ")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub ComboBox2_Leave(sender As Object, e As EventArgs) Handles
ComboBox2.Leave
        ComboBox2.ResetText()
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Hide()
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        ComboBox1.Items.Clear()
        ComboBox1.Text = ""
        ComboBox2.Items.Clear()
        ComboBox3.Items.Clear()
        dt = obj.sele("select * from gaurav")
        DataGridView1.DataSource = dt
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox1.Items.Add(dt.Rows(i)(0))
            ComboBox3.Items.Add(dt.Rows(i)(2))
        Next

        dt = obj.sele("select distinct Brand from gaurav")
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox2.Items.Add(dt.Rows(i)(0))
        Next
    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
```

```
        If ComboBox1.Text = "" Then
            MessageBox.Show("Select Id")
        Else
            dt = obj.sele("delete from gaurav where Order_id = " +
ComboBox1.SelectedItem.ToString + " ")
            MessageBox.Show("Record Deleted successfully")
            dt = obj.sele("select  * from gaurav")
            DataGridView1.DataSource = dt
            ComboBox1.Items.Clear()
            ComboBox2.Items.Clear()
            For i = 0 To dt.Rows.Count - 1 Step 1
                ComboBox1.Items.Add(dt.Rows(i)(0))
                ComboBox3.Items.Add(dt.Rows(i)(2))
            Next
            ComboBox1.ResetText()
            dt = obj.sele("select distinct Brand from gaurav")
            For i = 0 To dt.Rows.Count - 1 Step 1
                ComboBox2.Items.Add(dt.Rows(i)(0))
            Next
        End If
    End Sub

    Private Sub ComboBox3_Leave(sender As Object, e As EventArgs) Handles
ComboBox3.Leave
        ComboBox3.ResetText()
        ComboBox1.ResetText()
    End Sub

    Private Sub ComboBox3_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox3.SelectedIndexChanged
        dt = obj.sele("select * from gaurav where Item_name = '" +
ComboBox3.SelectedItem.ToString + "' ")
        DataGridView1.DataSource = dt
    End Sub
End Class
```

Search dealer form coding :-

```
Imports System.Data
Imports System.Data.OleDb
Public Class searchdealer
```

```
    Dim obj As New Class1
    Dim dt As New DataTable
    Dim i As Integer

    Private Sub searchdealer_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select * from Dealer")
        DataGridView1.DataSource = dt
            For i = 0 To dt.Rows.Count - 1 Step 1
                ComboBox1.Items.Add(dt.Rows(i)(0))
                ComboBox2.Items.Add(dt.Rows(i)(1))
                ComboBox3.Items.Add(dt.Rows(i)(3))
            Next
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Hide()
    End Sub

    Private Sub ComboBox1_Leave(sender As Object, e As EventArgs) Handles
ComboBox1.Leave
        ComboBox1.ResetText()
    End Sub

    Private Sub ComboBox2_Leave(sender As Object, e As EventArgs) Handles
ComboBox2.Leave
        ComboBox2.ResetText()
    End Sub

    Private Sub ComboBox3_Leave(sender As Object, e As EventArgs) Handles
ComboBox3.Leave
        ComboBox3.ResetText()
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        ComboBox1.Items.Clear()
        ComboBox2.Items.Clear()
        ComboBox3.Items.Clear()
        dt = obj.sele("select * from Dealer")
        DataGridView1.DataSource = dt
            For i = 0 To dt.Rows.Count - 1 Step 1
                ComboBox1.Items.Add(dt.Rows(i)(0))
                ComboBox2.Items.Add(dt.Rows(i)(1))
```

```
                ComboBox3.Items.Add(dt.Rows(i)(3))
            Next
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox1.SelectedIndexChanged
        dt = obj.sele("select * from Dealer where Dealer_id = " + ComboBox1.Text + " ")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub ComboBox2_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox2.SelectedIndexChanged
        dt = obj.sele("select * from Dealer where Name = '" + ComboBox2.Text + "' ")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub ComboBox3_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox3.SelectedIndexChanged
        dt = obj.sele("select * from Dealer where Contact_no = " + ComboBox3.Text + "
")
        DataGridView1.DataSource = dt
    End Sub
End Class
```

Search customer and delete customer form coding :-

```
Public Class Searchcus

    Dim obj As New Class1
    dim dt As New DataTable
    Dim dt1 As New DataTable
    Dim i As Integer

    Private Sub Searchcus_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select * from Bill")
        DataGridView1.DataSource = dt
            dt = obj.sele("select distinct Bill_id from Bill")
            For i = 0 To dt.Rows.Count - 1 Step 1
                ComboBox1.Items.Add(dt.Rows(i)(0))
            Next

        dt = obj.sele("select distinct Cus_name from Bill")
```

```
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox2.Items.Add(dt.Rows(i)(0))
        Next

        dt = obj.sele("select distinct Contact_no from Bill")
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox3.Items.Add(dt.Rows(i)(0))
        Next

    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox1.SelectedIndexChanged
        dt = obj.sele("select * from Bill where BIll_id = " + ComboBox1.Text + " ")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub ComboBox2_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox2.SelectedIndexChanged
        ComboBox1.ResetText()
        dt = obj.sele("select * from Bill where Cus_name = '" + ComboBox2.Text + "' ")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub ComboBox3_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox3.SelectedIndexChanged
        ComboBox1.ResetText()
        dt = obj.sele("select * from Bill where Contact_no = " + ComboBox3.Text + " ")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub ComboBox2_Leave(sender As Object, e As EventArgs) Handles
ComboBox2.Leave
        ComboBox2.ResetText()
    End Sub

    Private Sub ComboBox3_Leave(sender As Object, e As EventArgs) Handles
ComboBox3.Leave
        ComboBox3.ResetText()
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
```

```
        Me.Hide()
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        ComboBox1.Items.Clear()
        ComboBox1.ResetText()
        ComboBox2.Items.Clear()
        ComboBox3.Items.Clear()
        dt = obj.sele("select * from Bill")
        DataGridView1.DataSource = dt
        dt = obj.sele("select distinct Bill_id from Bill")
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox1.Items.Add(dt.Rows(i)(0))
        Next

        dt = obj.sele("select distinct Cus_name from Bill")
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox2.Items.Add(dt.Rows(i)(0))
        Next

        dt = obj.sele("select distinct Contact_no from Bill")
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox3.Items.Add(dt.Rows(i)(0))
        Next
    End Sub

    Public Function com()
        dt = obj.sele("select distinct Bill_id from Bill")
        ComboBox1.Items.Clear()
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox1.Items.Add(dt.Rows(i)(0))
        Next
    End Function

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        If (ComboBox1.Text = "") Then
            MessageBox.Show("Select Id first")
        Else
            dt = obj.sele("delete from Bill where Bill_id = " + ComboBox1.Text.ToString
+ " ")
            MessageBox.Show("Record deleted successfully")
            dt = obj.sele("select * from Bill")
```

```
            DataGridView1.DataSource = dt
            ComboBox1.ResetText()
            com()
        End If
    End Sub
End Class
```

Search stock form coding :-

```
Public Class searchstock

    Dim obj As New Class1
    Dim dt As New DataTable
    Dim i As Integer
    Dim c As Integer = 0
    Dim dt1 As New DataTable
    Dim f As Boolean

    Private Sub Button4_Click(sender As Object, e As EventArgs)

    End Sub

    Private Sub searchstock_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select * from Stock")
        DataGridView1.DataSource = dt
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox1.Items.Add(dt.Rows(i)(0))
            ComboBox3.Items.Add(dt.Rows(i)(2))
        Next

        dt = obj.sele("select distinct Brand from Stock")
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox2.Items.Add(dt.Rows(i)(0))
        Next

    End Sub

    Private Sub ComboBox1_Leave(sender As Object, e As EventArgs) Handles
ComboBox1.Leave
        ComboBox1.ResetText()
    End Sub
```

```
    Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox1.SelectedIndexChanged
        dt    =    obj.sele("select    *    from    Stock    where    Item_id    =    "    +
ComboBox1.SelectedItem.ToString + " ")
        DataGridView1.DataSource = dt
    End Sub

    Private    Sub    ComboBox2_Leave(sender    As    Object,    e    As    EventArgs)    Handles
ComboBox2.Leave
        ComboBox2.ResetText()
    End Sub

    Private Sub ComboBox2_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox2.SelectedIndexChanged
        dt    =    obj.sele("select    *    from    Stock    where    Brand    =    '"    +
ComboBox2.SelectedItem.ToString + "' ")
        DataGridView1.DataSource = dt
    End Sub

    Private    Sub    ComboBox3_Leave(sender    As    Object,    e    As    EventArgs)    Handles
ComboBox3.Leave
        ComboBox3.ResetText()
    End Sub

    Private Sub ComboBox3_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox3.SelectedIndexChanged
        dt    =    obj.sele("select    *    from    Stock    where    Item_name    =    '"    +
ComboBox3.SelectedItem.ToString + "' ")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        ComboBox1.Items.Clear()
        ComboBox2.Items.Clear()
        ComboBox3.Items.Clear()
        dt = obj.sele("select * from Stock")
        DataGridView1.DataSource = dt
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox1.Items.Add(dt.Rows(i)(0))
            ComboBox3.Items.Add(dt.Rows(i)(2))
        Next
```

```
        dt = obj.sele("select distinct Brand from Stock")
        For i = 0 To dt.Rows.Count - 1 Step 1
            ComboBox2.Items.Add(dt.Rows(i)(0))
        Next
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Hide()
    End Sub
End Class
```

Order item report :-

```
Public Class reportitem

    Dim obj As New Class1
    Dim dt As New DataTable
    Dim i As Integer

    Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
        dt = obj.sele("select * from gaurav where Order_date between '" +
DateTimePicker1.Value.ToShortDateString + "' and '" +
DateTimePicker2.Value.ToShortDateString + "' ")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub reportitem_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select *from gaurav")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        Me.Hide()
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        DateTimePicker1.Text = ""
        DateTimePicker2.Text = ""
        dt = obj.sele("select *from gaurav")
        DataGridView1.DataSource = dt
    End Sub
End Class
```

Customer report :-

```
Public Class reportcus

    Dim obj As New Class1
    Dim dt As New DataTable
    Dim i As Integer

    Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
        dt = obj.sele("select * from Bill where Bill_date between '" +
DateTimePicker1.Value.ToShortDateString + "' and '" +
DateTimePicker2.Value.ToShortDateString + "' ")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub reportdealer_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select * from Bill")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        Me.Hide()
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        dt = obj.sele("select * from Bill")
        DataGridView1.DataSource = dt
        DateTimePicker1.Text = ""
        DateTimePicker2.Text = ""
    End Sub
End Class
```

Stock report :-

```
Public Class reportstock

    Dim obj As New Class1
    Dim dt As New DataTable
    Dim i As Integer
```

```
    Private Sub reportstock_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        dt = obj.sele("select * from Stock")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        dt = obj.sele("select * from Stock")
        DataGridView1.DataSource = dt
    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        Me.Hide()
    End Sub

    Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
        dt = obj.sele("select * from Stock where date between '" +
DateTimePicker1.Value.ToShortDateString + "' and '" +
DateTimePicker2.Value.ToShortDateString + "' ")
        DataGridView1.DataSource = dt
    End Sub
End Class
```

Chapter 5

TESTING & IMPLEMENTATION

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of the software. The results of testing are used later on during maintenance also.

Testing is vital to the success of the system. System testing makes a logical assumption that if the parts of the system are correct, the goal will be successfully achieved. In adequate testing or non-testing leads to errors that may not appear until months or even years later.

This creates two problems:

1.) The time lag between the cause and the appearance of the problem.

2.) The time interval effect of the system errors on files and the records on the system.

A small error can conceivably explode into a much larger problem. Effective testing early in the process translates directly into long term cost savings from a reduced number of errors.

Another reason for system testing is its utility as a user-oriented vehicle before implementation. The best program is worthless if it does not meet the user requirements. Unfortunately, the user's demands are often compromised by efforts to facilitate program or design efficiency in terms of processing time or design efficiency.

Thus, in this phase we went to test the code we wrote. We needed to know if the code compiled with the design or not? Whether the code gave the desired outputs on given inputs? Whether it was ready to be installed on the user's computer or some more modifications were needed?

Testing objectives :

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

- Testing is a process of executing a program with the intent of finding an error.
- 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.

Phases of Testing

1.) White box testing :- In this technique the close examination of the logical parts through the software are tested by cases that exercises pieces sets of condition or loops . All logical parts of the software checked once. When the box testing tests are all independent part within a module a logical decisions on their true and the false side are exercised, all loops and bounds within their operation bounds were exercised and internal data structure to ensure their validity were exercised once.

2.) **Black Box Testing :-** This method enables the software engineer to device sets of input techniques that fully exercise all functional requirement for a program . black box testing tests the input, the output and the external data. It checks whether the input data is correct and whether we are getting the desired output.

3.) **Unit Testing :-** Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the 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 test before the start of the integration testing begins.

4.) **Integration Testing :-** After unit testing, we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions.

5.) **System Testing :-** Here the entire software system is tested. The reference document for this process is the requirements document, and the goal is to see if software meets its requirements.

6.) **Acceptance Testing :-** Acceptance Testing is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behaviour of the system; the internal logic of program is not emphasized. Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of software development. 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.

Chapter 6

LIMITATION & FUTURE SCOPE

Limitation :

- Used by single user at a time
- We cannot send report to others
- It is not manage information of staff members
- There is no information about payment method
- The transaction is done offline
- There is no information about item is delivered or not

Future scope :

- It can be manage large inventory by integrating it with a better data management technique, it can be enabled to handle an large inventory.
- Multiple user can work on this system at a time.
- Extend for staff member and store their information and also have their payment(salary) information.
- Online transaction can be available in future and payment using net banking.
- Make system more reliable and secure.
- Add data recovery in condition of system failure.
- Add more features to make work system better.

Chapter 7

CONCLUSION

We tried to develop a system that can be helpful for the owner of the computer shop. There are some bugs in the system they are solve by testing the system.

At the end of the project we found that this project is provide accurate result in the working area such as add customer, stock, dealer and order, update them and search them. Because the documentation of the project.

We have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them.

In the last we would like to thanks all the persons involved in the development of the system directly or indirectly. We hope that the project will serve its purpose for which it is develop there by underlining success of process.