

**INTRODUCTION TO SYSTEM**

**INTRODUCTION**

**Minimart :**

The project is on Minimart Management System**.** Minimart is the place where customers come to purchase their daily using products and pay for that. So there is a need to calculate how many products are sold and to generate the bill for the customer.

In our project we have 3 users. First is the data entry operator who will enter the products in database. Second one is the administrator who will decide the taxes and commissions on the products and can see the report of any product. Third one is the bill calculating operator who will calculate the bill and print

**OBJECTIVE OF SYSTEM**

“To make software fast in processing, with good user interface so that user can change it and it should be used for a long time without error and maintenance.”

**SCOPE OF SYSTEM**

Our project has a big scope to do.

We can :

1. Calculate the bill.

2. Give the bill to the customer.

3. Store how many products are sold.

4. Store products and their prices and with other information.

5. Set the rates of taxes and commission on the products.

6. Can see the report of the product in a fix period of time.

7. Change the Graphical User Interface of the system.

We can’t :

1. Calculate the expanses on the product.

**INVESTIGATION PHASE**

**EXISTING OF SYSTEM**

Many Minimarts use this type of billing system for a decade. It is also improved many times according to requirements of sellers and customers. It does the same work that is calculating the bill, gives it to the customer and maintain proper database. They are accurate in calculation and printing, they also generate records.

A new concept is also added in the billing system is that they also maintain relationships with the customers who purchase more products from the store regularly. System also concerns their requirements and gives them more commission. It also shows the particular product repots which items are required and .

**PROPOSED OF SYSTEM**

The Existing system there is a need to develop a new system. The new system should concern the requirements of the customer and the sellers. It has the following qualities:

1. Reduction in processing cost.

2. Error reduction.

3. Automatic posting.

4. Improve reporting.

5. Automatic production of the documents and Reports.

6. Faster response time.

7. Ability to meet user requirements.

8. Flexibility.

9. Reduced dependency.

10.Improves resource uses.

11.Reduction in use of the paper.

12.Reduction in Man Power.

Proposed system has these qualities including the qualities of the existing System.

****

**REQUIREMENT ANALYSIS**

**HARDWARE AND SOFTEWARE REQUIREMENT**

System requirement divided into two categories .Hardware and Software requirement are as follows :

* **Hardware Reqirement :**

Hard Disk : 80 GB or More

RAM : 1 GB RAM

* **Software Reqirement :**

Operating System : Windows XP/Windows7/Windows8.

Language Used : VB.NET.

Database : Microsoft SQL Server.

Reporting Tool : Crystal Report (SAP).

****

**SYSTEM ANALYSIS & DESIGN** **FEASIBILITY STUDY**

*“***Feasibility Study***”* is a test of the system according to its workability, impact of the organization, ability to meet user needs and effective use of the resources. We can test our system by different type of the feasibilities. There are 5 types of the feasibilities which are discussed here. These are as follows:

* **Technical Feasibility:**

A study of resources availability that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not. This system can be made in any language that support good user interface and easy database handling.

Technical needs may include:

**Front-End Selection:**

Front-End means a language that is used

for user interface designing and coding. Front-End should have following qualities:

* It must have a graphical user interface that assist employees that are not from some IT background.
* Scalability and Extensibility
* Robustness
* According to the organization requirements and culture.
* Must provide excellent reporting features with good printing support.
* Platform independent.
* Easy to deploy and maintain.
* Event driven programming.
* Front-End must support some popular Back-End like MS Access, SQL Server and Oracle.

According to the above stated features we selected VB.Net as Front –End for developing our project. VB.Net is used in Microsoft Visual Studio 2010.

**Back-End Selection:**

Back-End means a language that is used for database management. Back-End should have following qualities:

* Multiple user support.
* Provide inherent feature for security.
* Efficient data retrieval and maintenance.
* Stored procedures.
* Popularity.
* Operating System compatible.
* Easy to install.
* Various drivers must be available.
* Efficient data handling.
* Easy to implement with Front-End.
* **Economical Feasibility:**

In this we consider following costs:

1. The cost to conduct a full system investigation.

2. The cost of hardware and software for class of application being considered.

3. The benefit in the form of the reduced cost.

Our system has a lot of features at a minimum cost so it is feasible to implement and it will be very much beneficial to the sellers in the reduced cost. It’s software and hardware cost is also low then the existing system.

* **Operational Feasibility:**

In this feasibility we consider following points:

1. What changes will be brought with the system.

2. What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time? In the new system we made some major changes for the staff members so that they have to be trained to use the newly added facilities. These major changes are possible and give a new era in the Minimart in production and sales management.

**SYSTEM DESIGN**

**Zero Level Diagram :**

**Admin**

**Supplier**

**Customer**

**Request Items Payment**

**Issue**

**Purchase**

**Issue Item + Bill Order**

**Payment**

**Issue Items + Bill**

**Maintain Monitor Stock**

**And user Gets Various**

**Reports**

**Entity Relationship Daigram**

has

Issue

Updatee

Issuee

make

Supplier Bills

To

Supplier

Mart

To

Payment

Product

Customer Bill

Maintainance

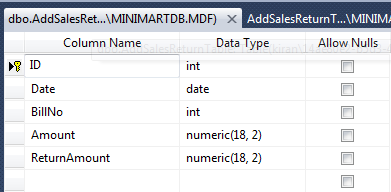
Purchase Order

**

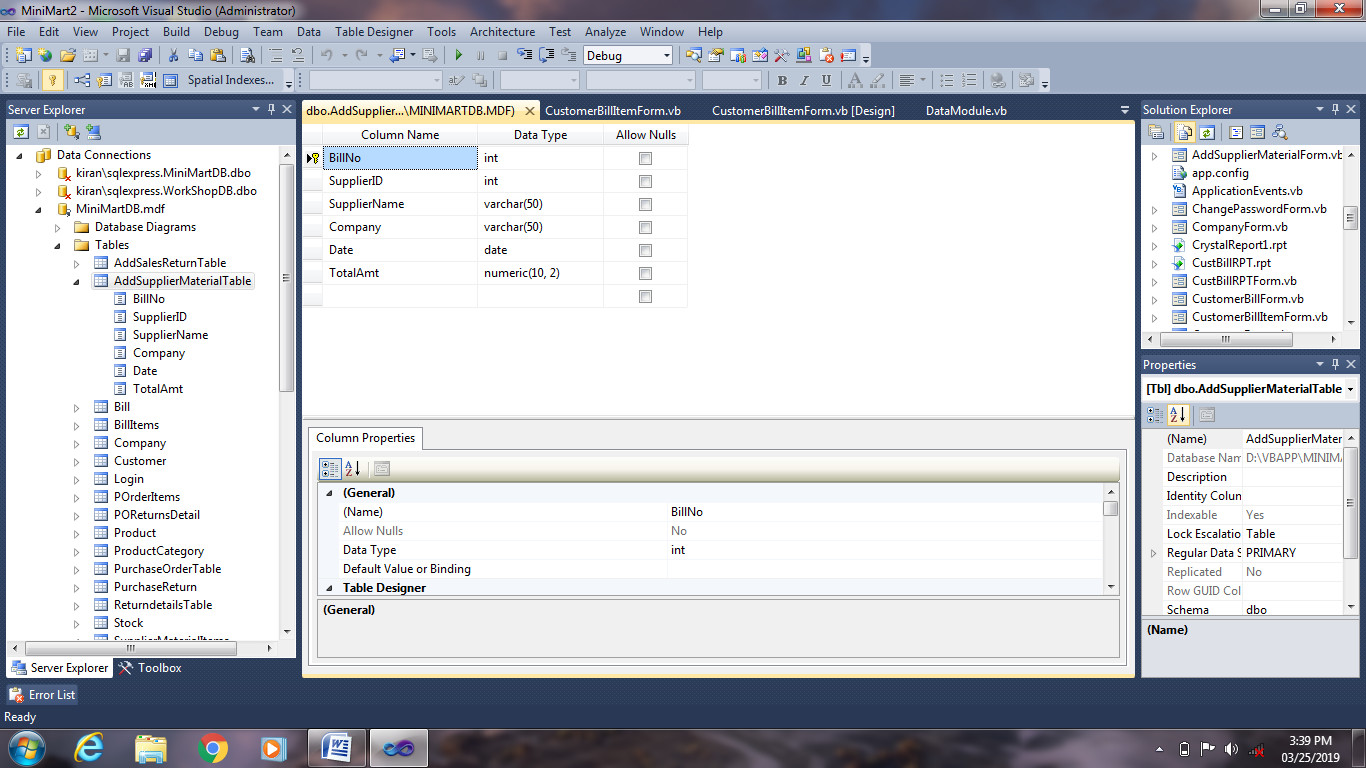
**USER INTERFACE SCREENS**

**DATABASE STRUCTURE**

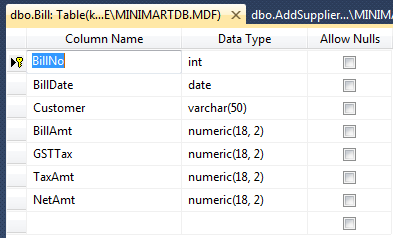
**Sales Return Table :**



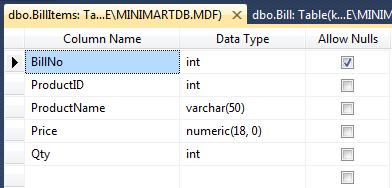
**Supplier Material Table :**



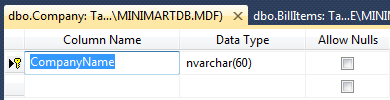
**Bill Table :**



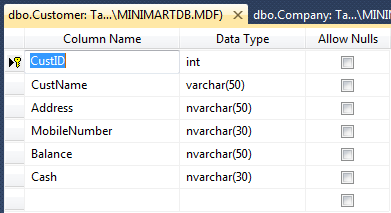
**Bill Items Table :**



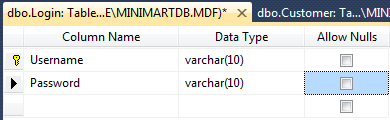
**Company Table :**



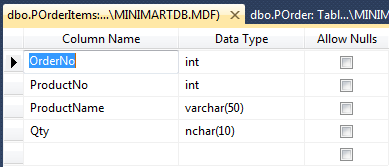
**Customer Table :**



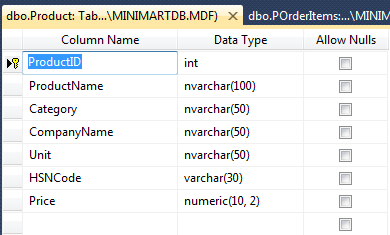
**Login Table :**



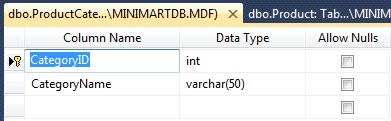
**Purchase Order Items :**



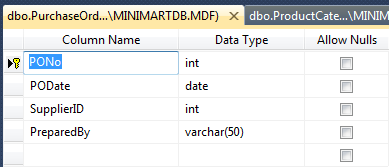
**Product Table :**



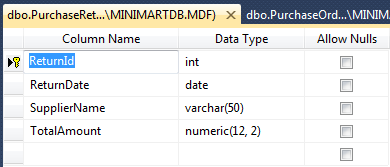
**Product Category :**



**Purchase Order Table :**



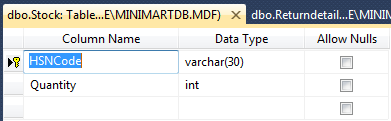
**Purchase Returns Table**

****

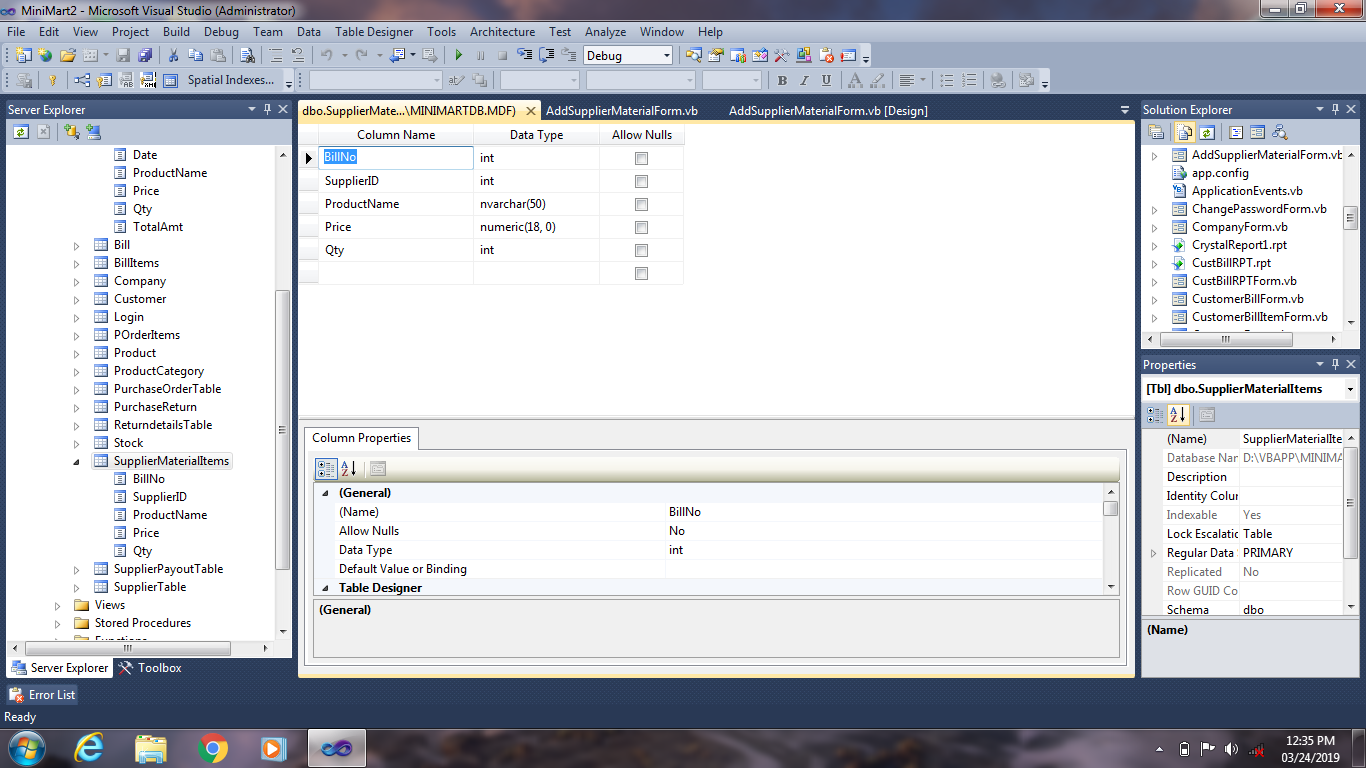
**Return Details Table :**

****

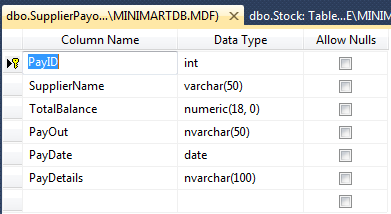
**Stock Table :**

****

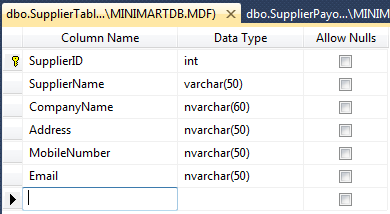
**Supplier Material Items :**

****

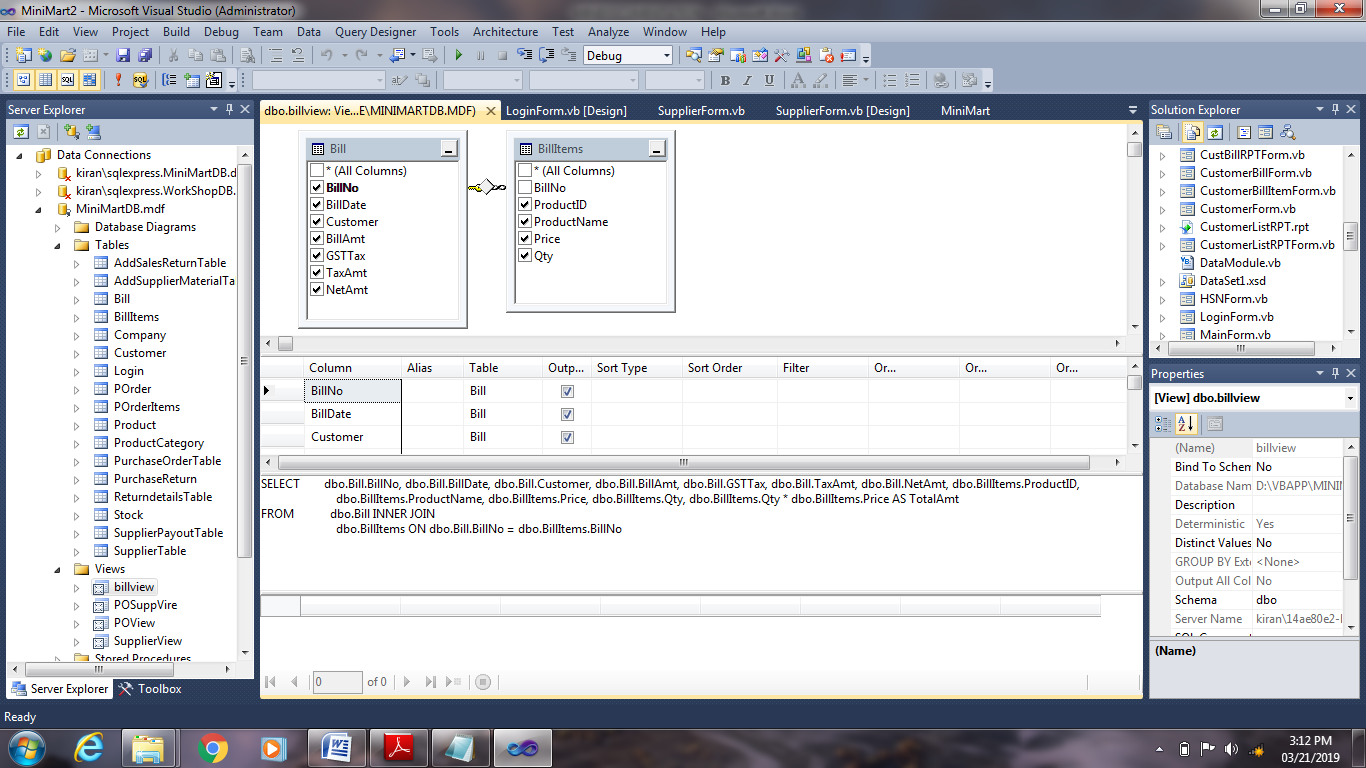
**Supplier Payment Table :**

****

**Supplier Information Table :**

****

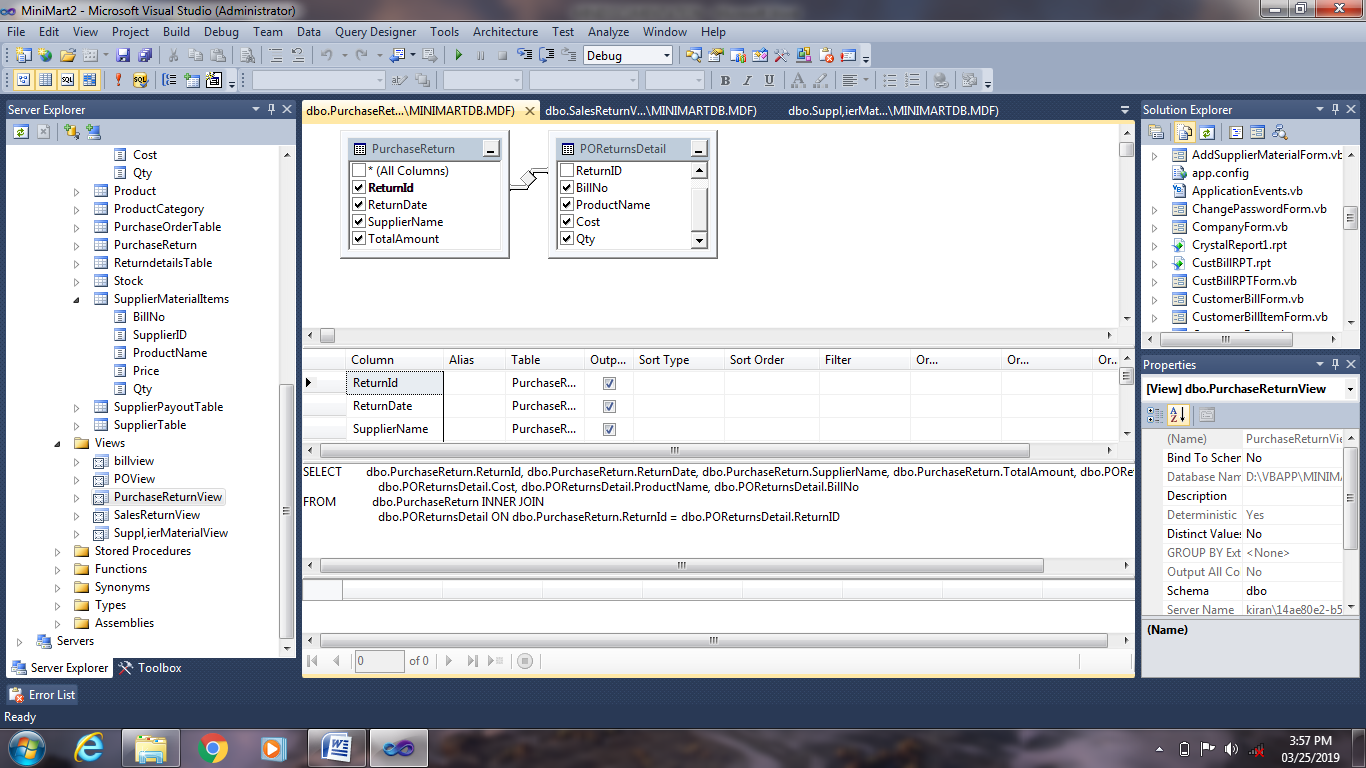
**Bill View :**

****

**Purchase order view :**

****

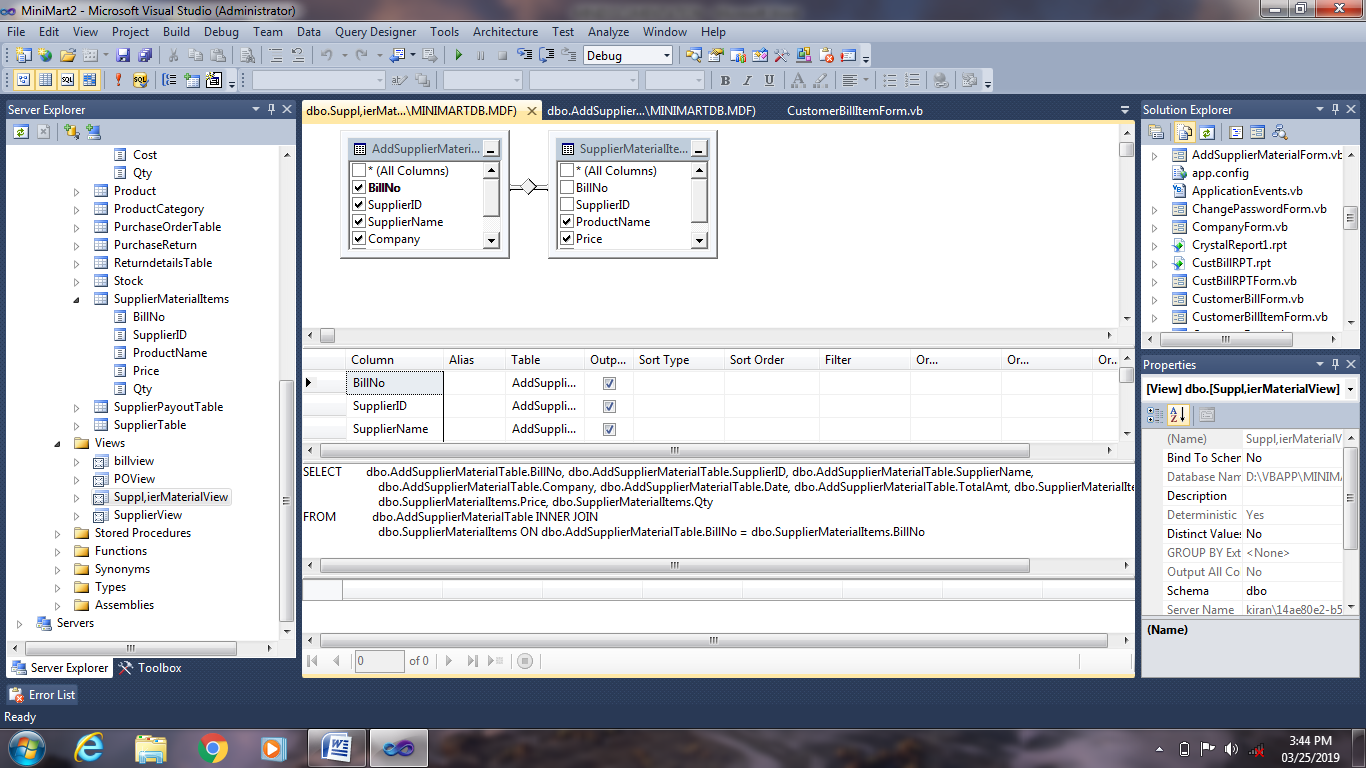
**Purchase Return View:**

****

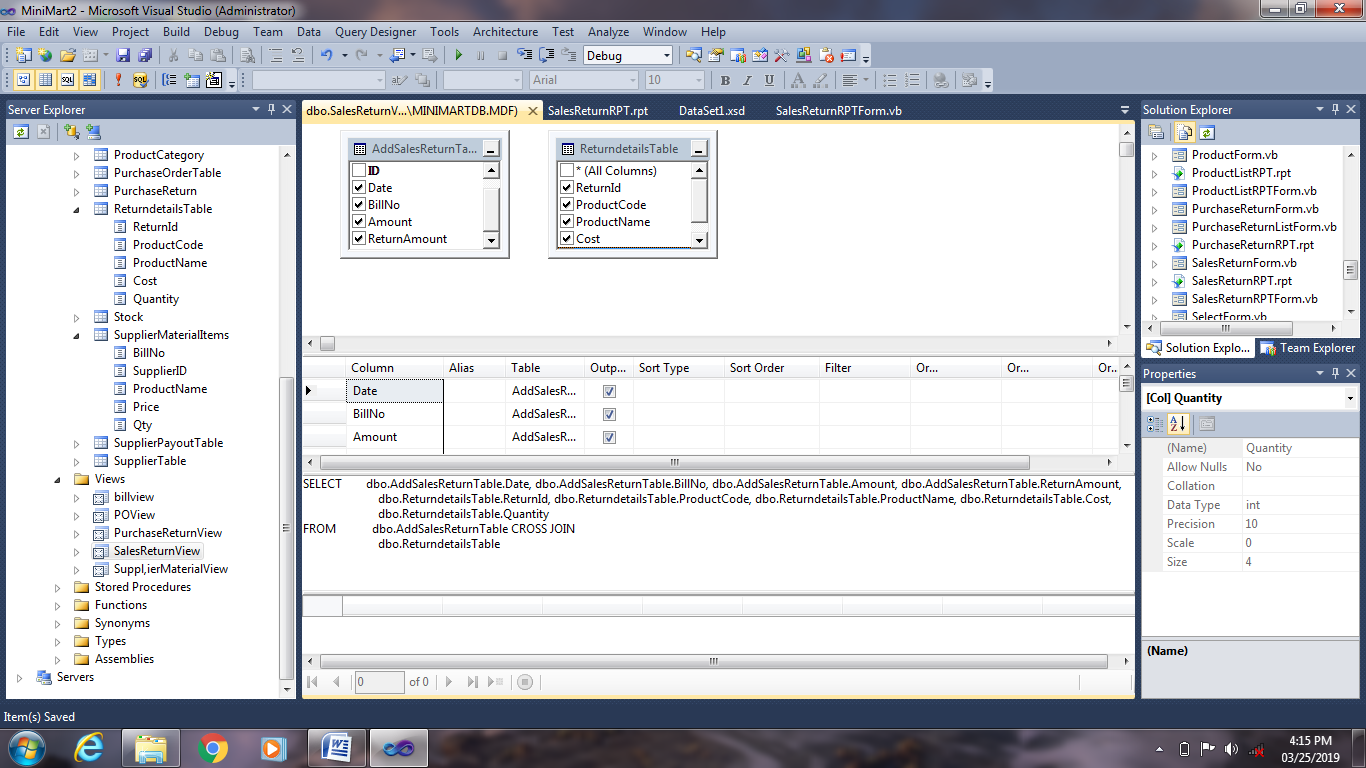
**Supplier view :**

****

**Supplier Material View:**

****

**Sales Return View:**

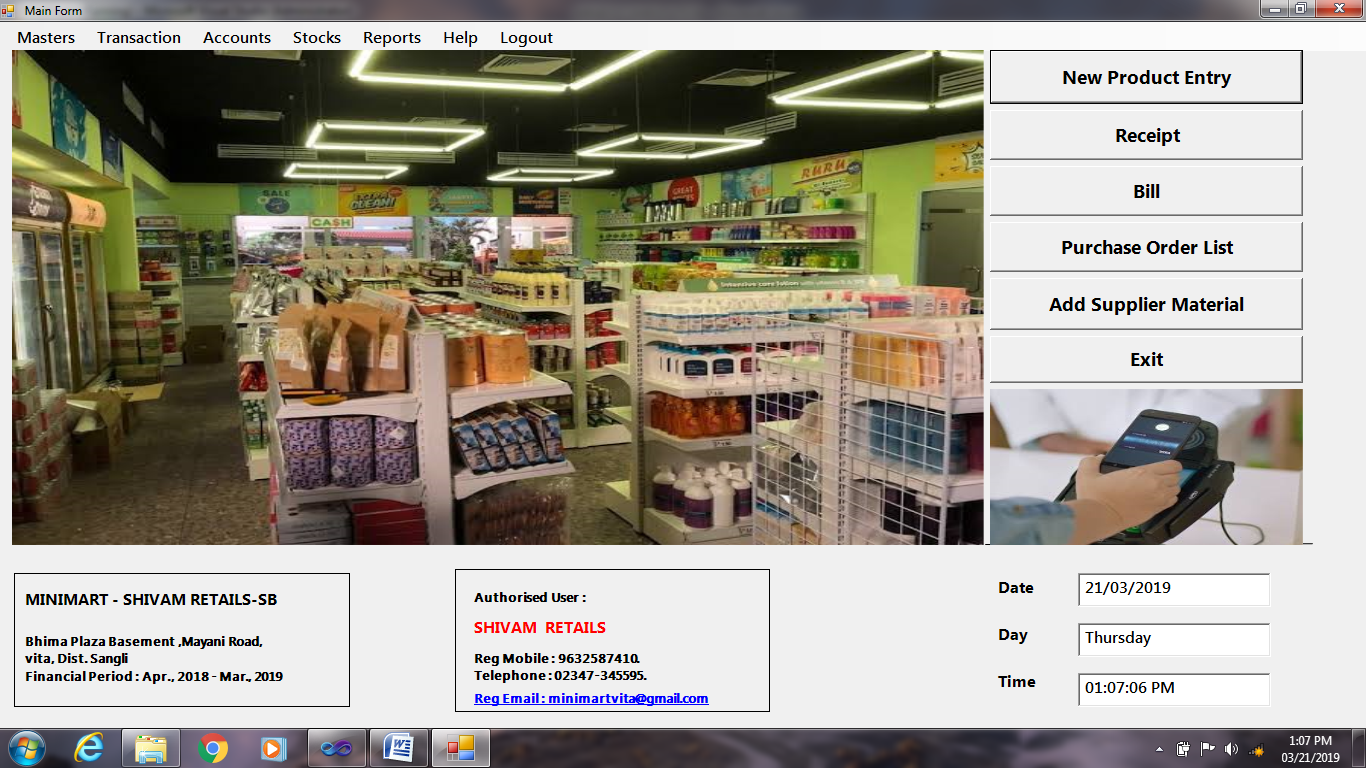
****

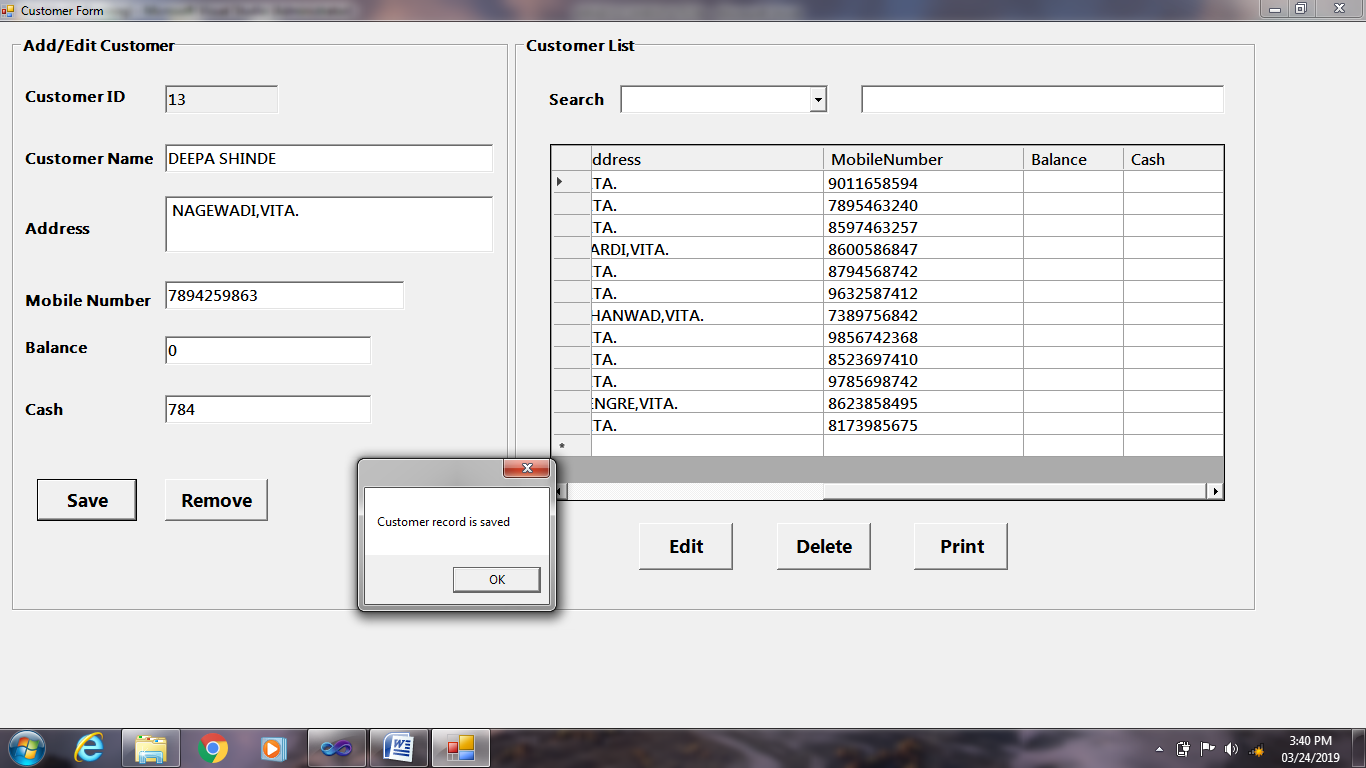
**FORM DESIGN**

**Login Form :**

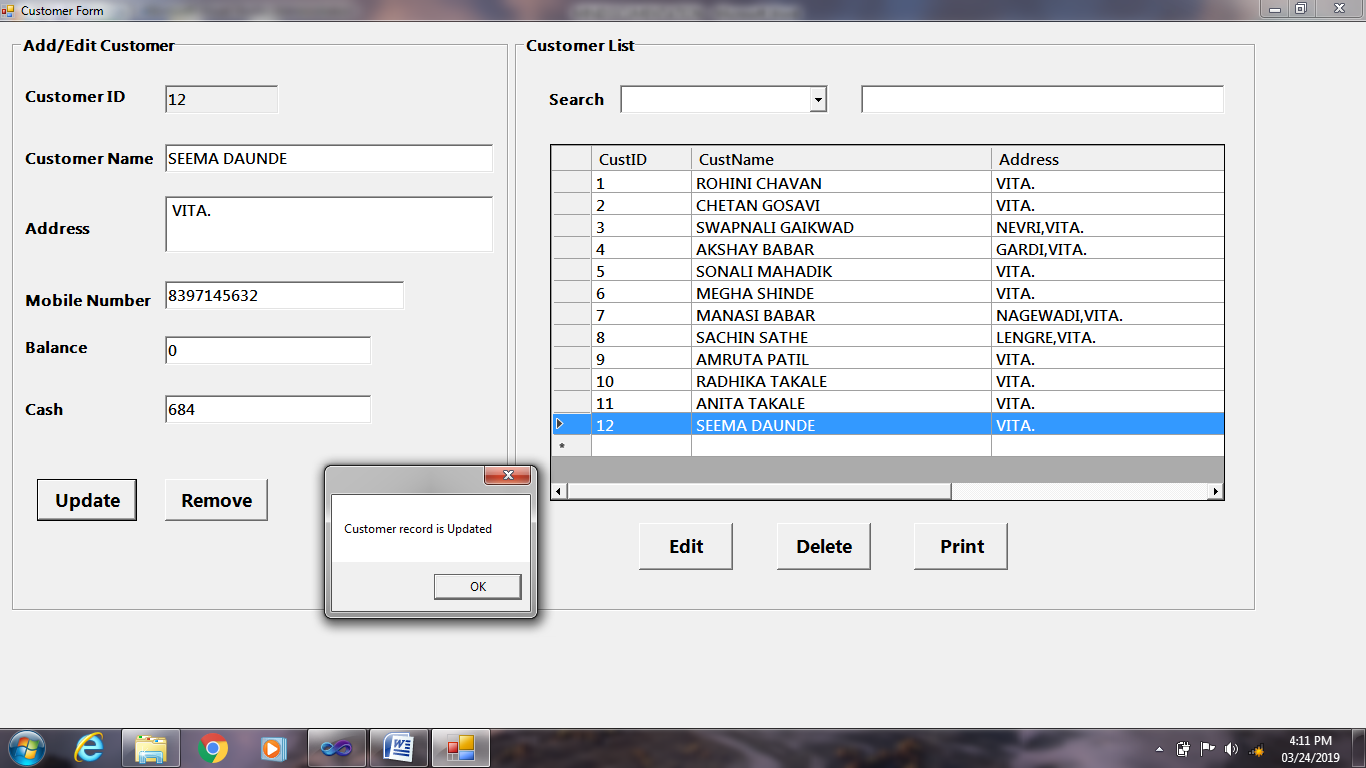
****

**Main Form :**

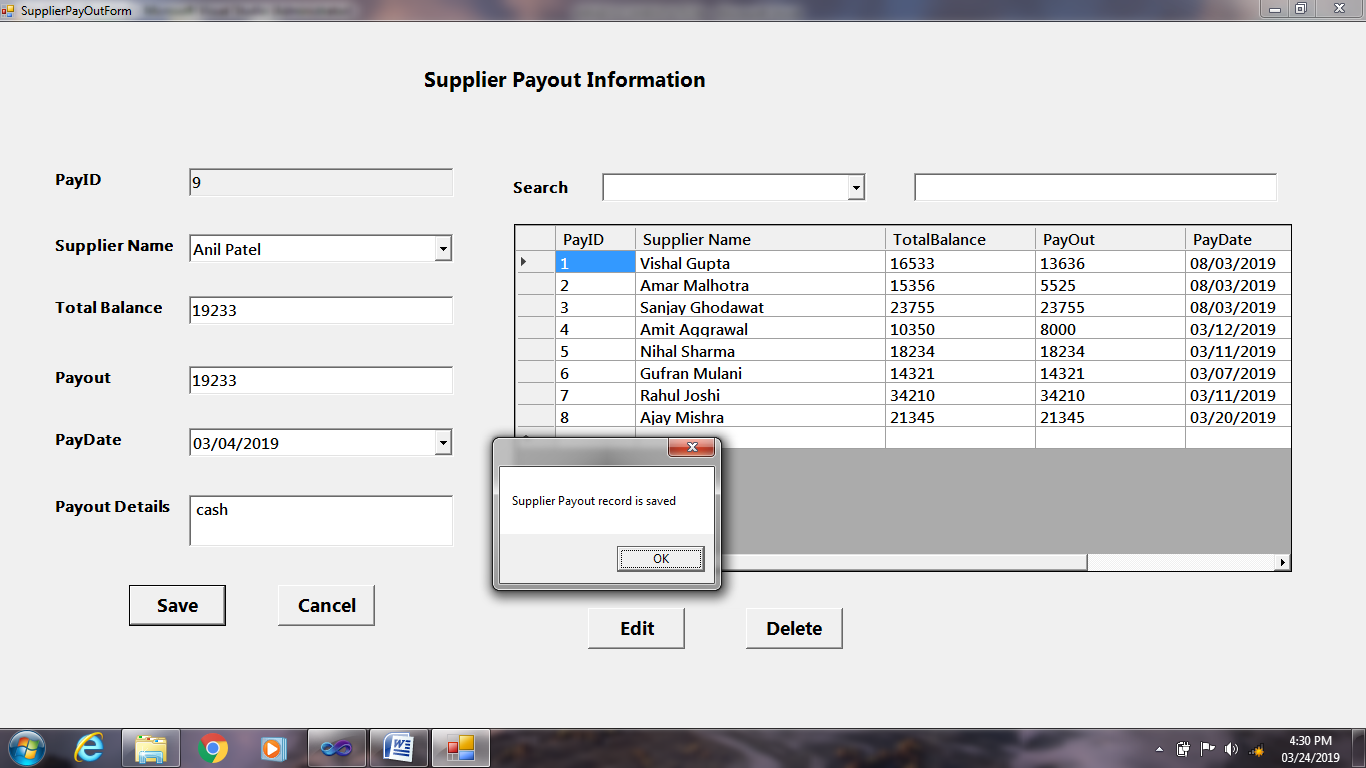
**Supplier Information Form:**

****

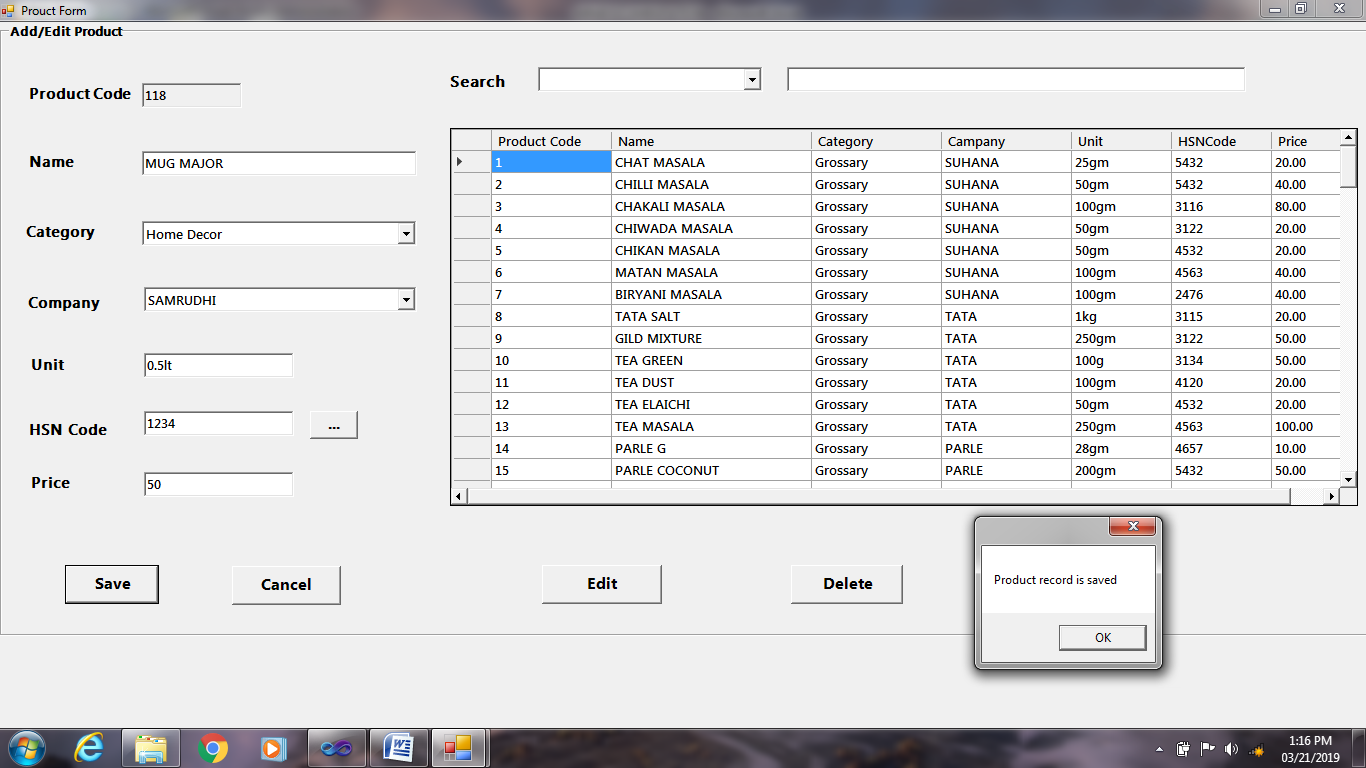
**Customer record:**

****

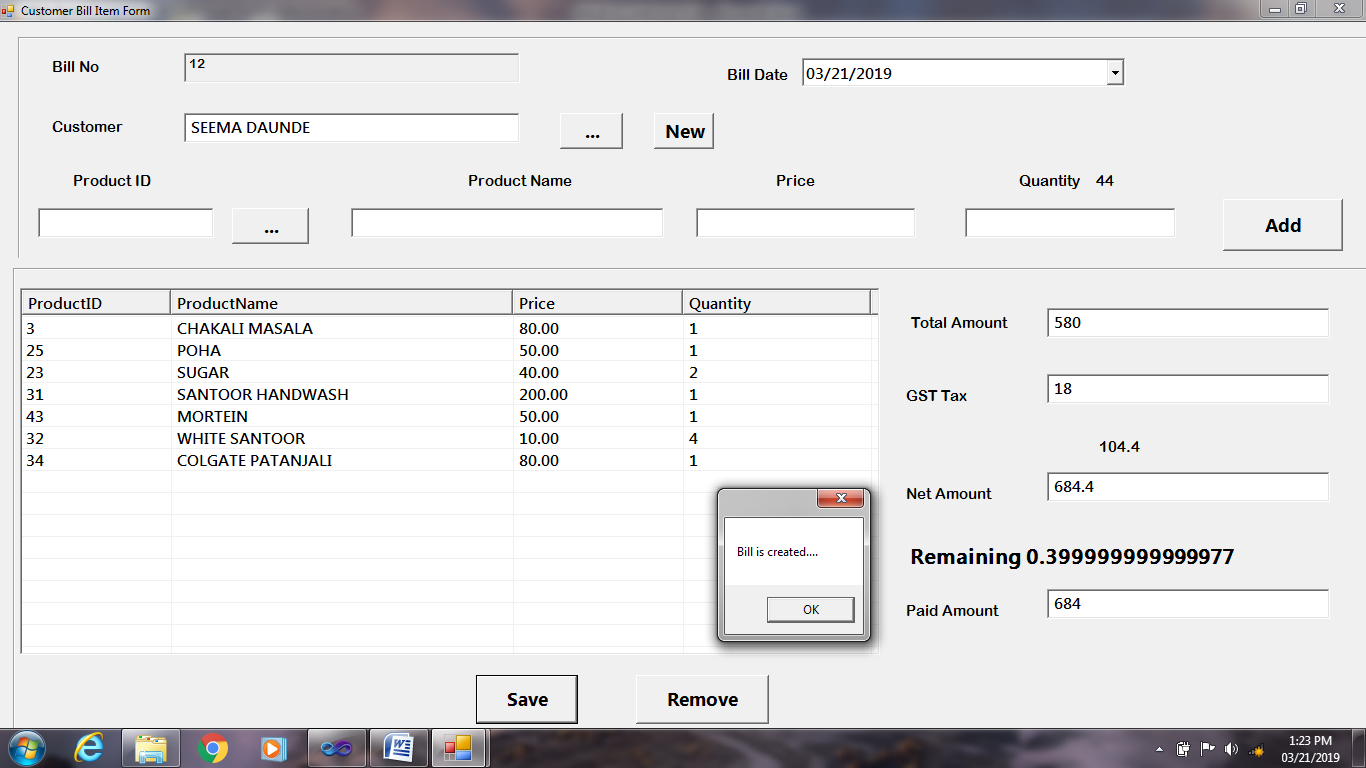
**Supplier Payment Information:**

****

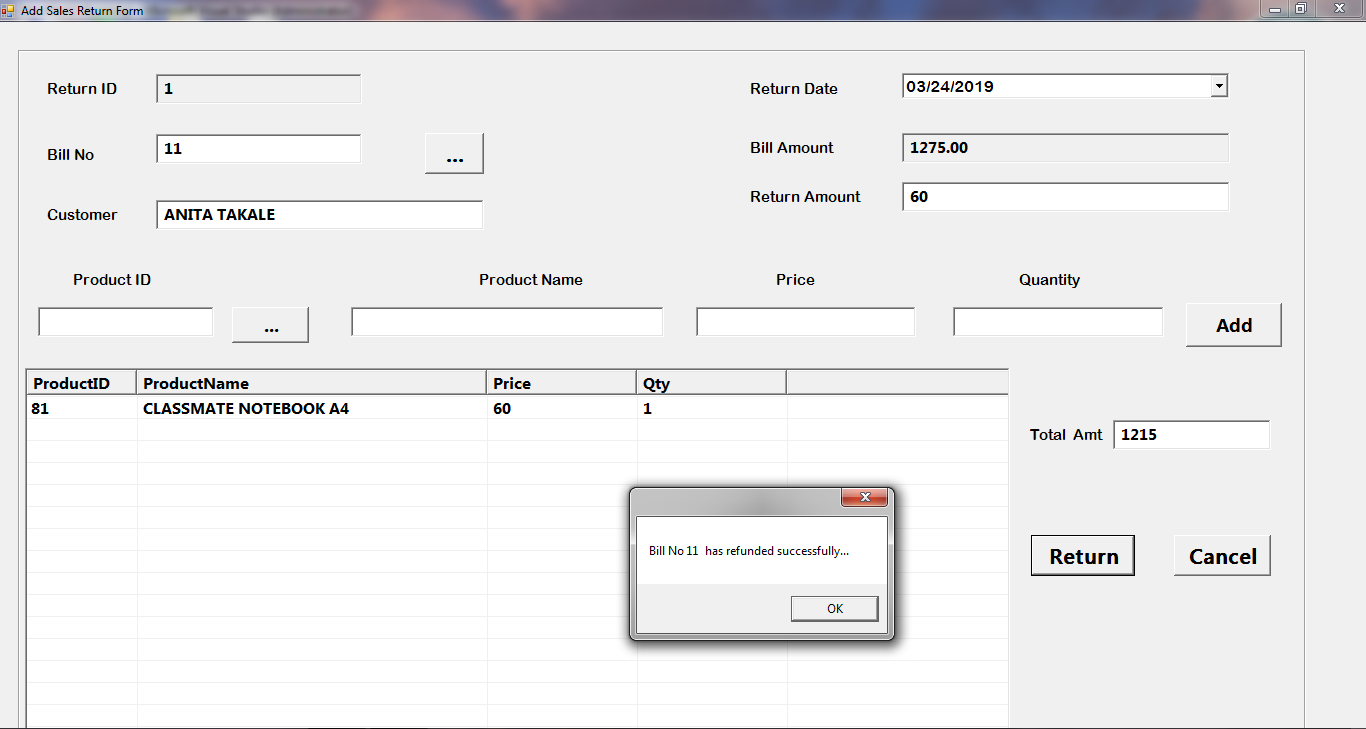
**Product Form :**

****

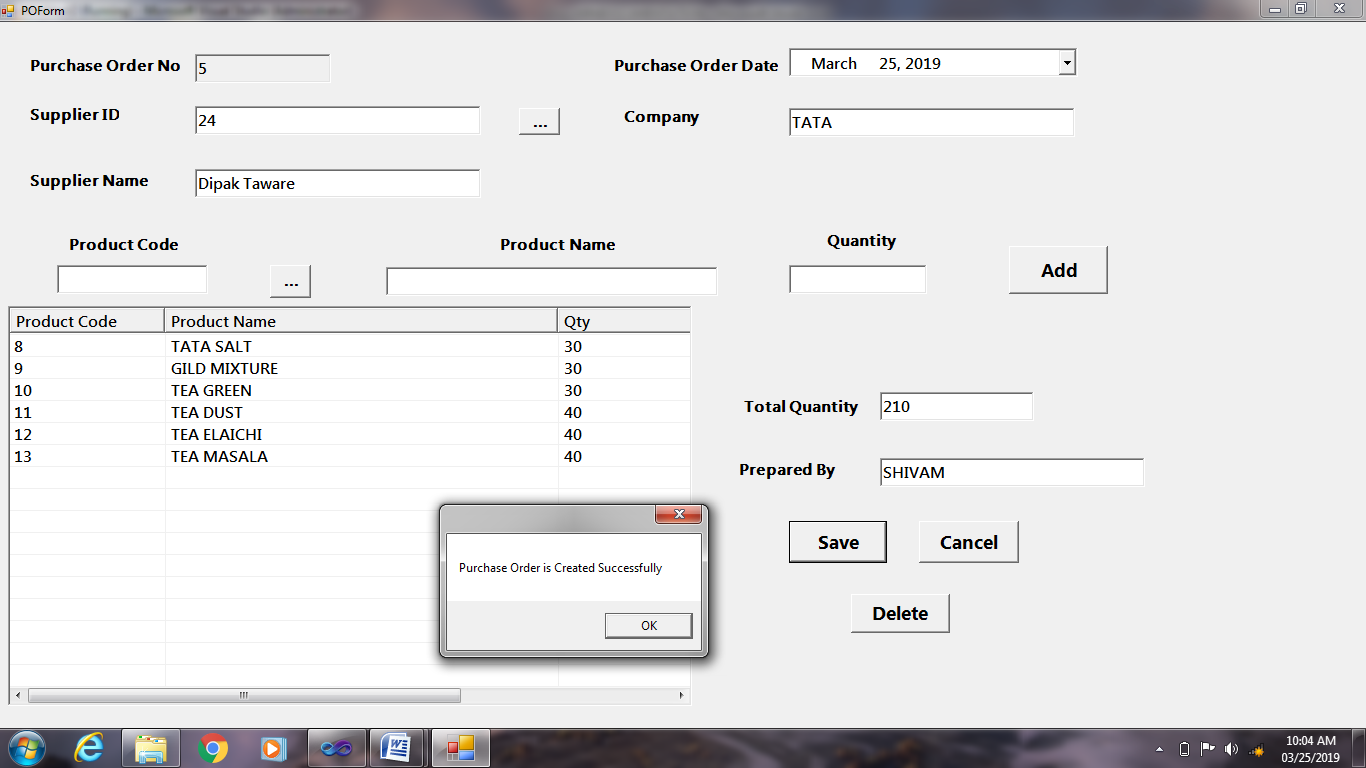
**Bill Done :**

****

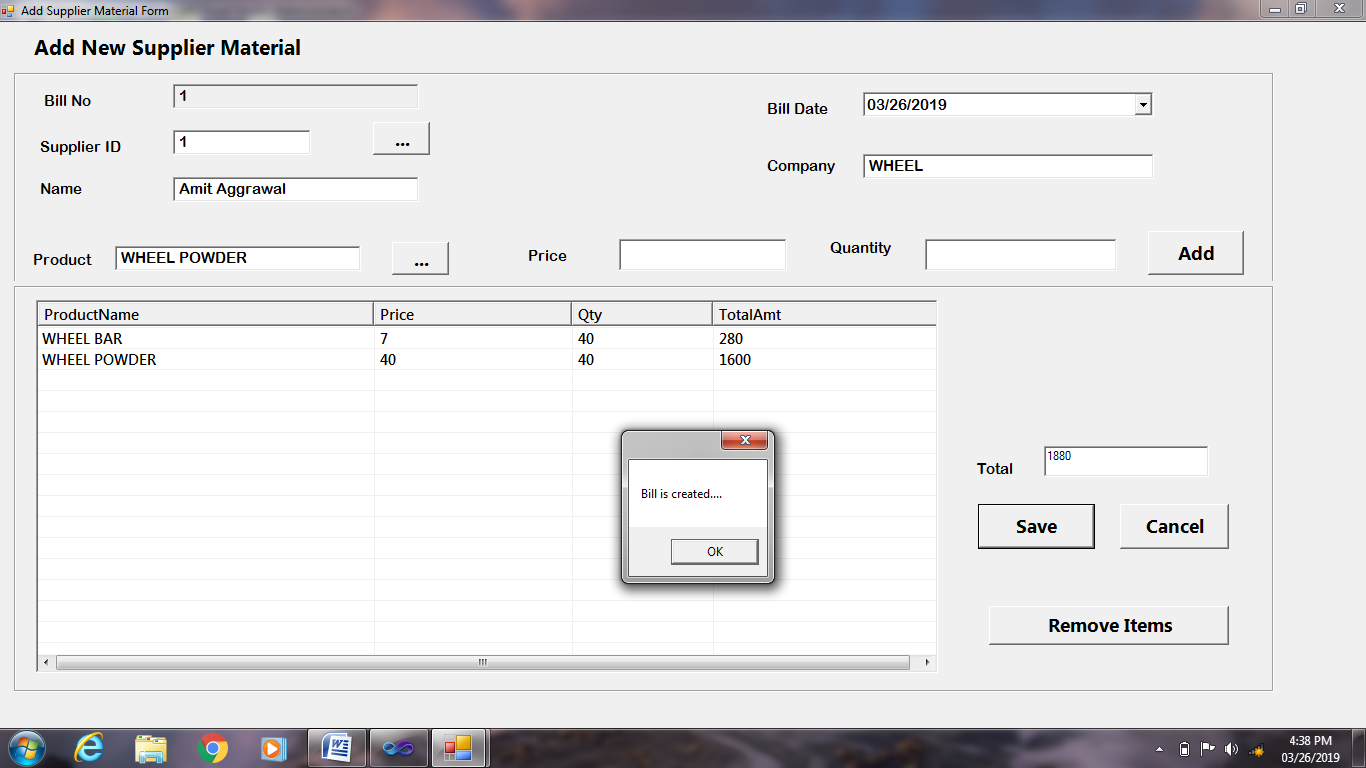
**Sales Return Done:**

****

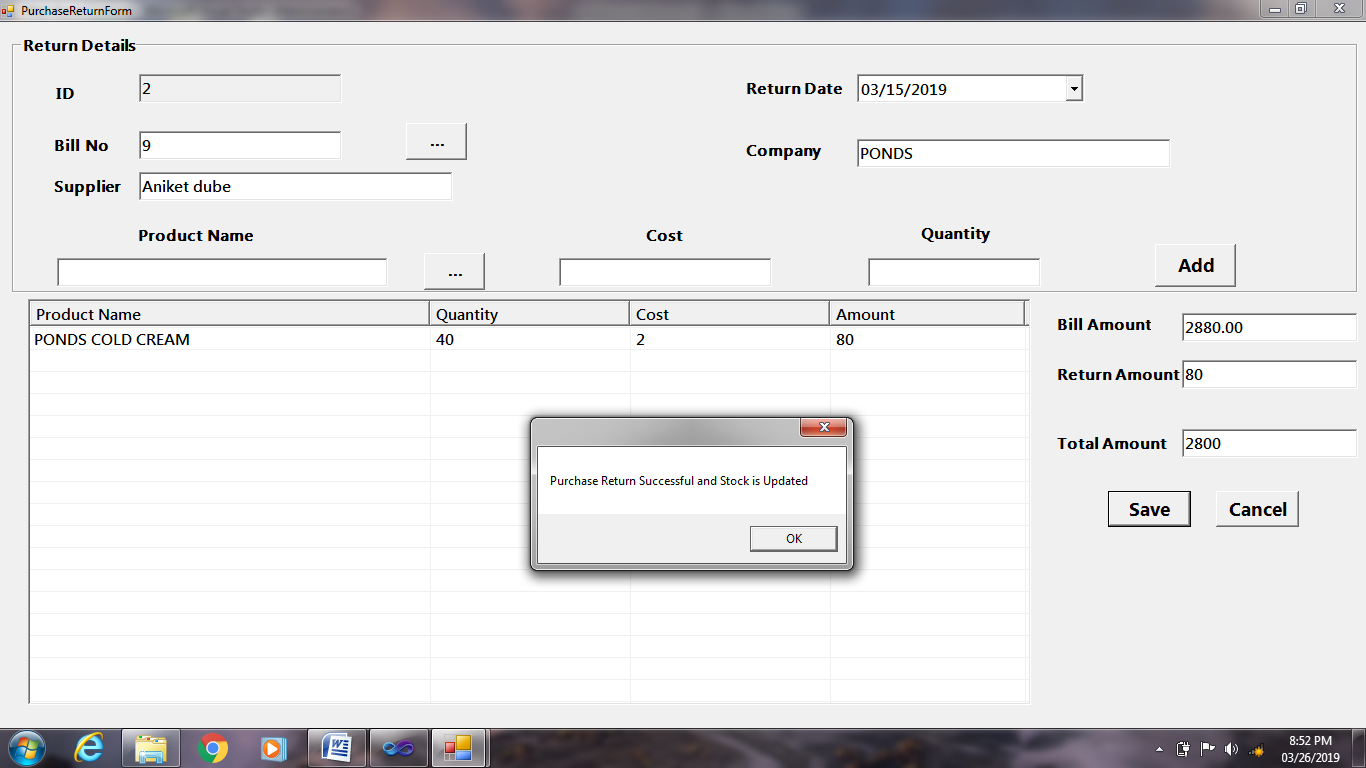
**Purchase Order Done:**

****

**Supplier Material Done:**

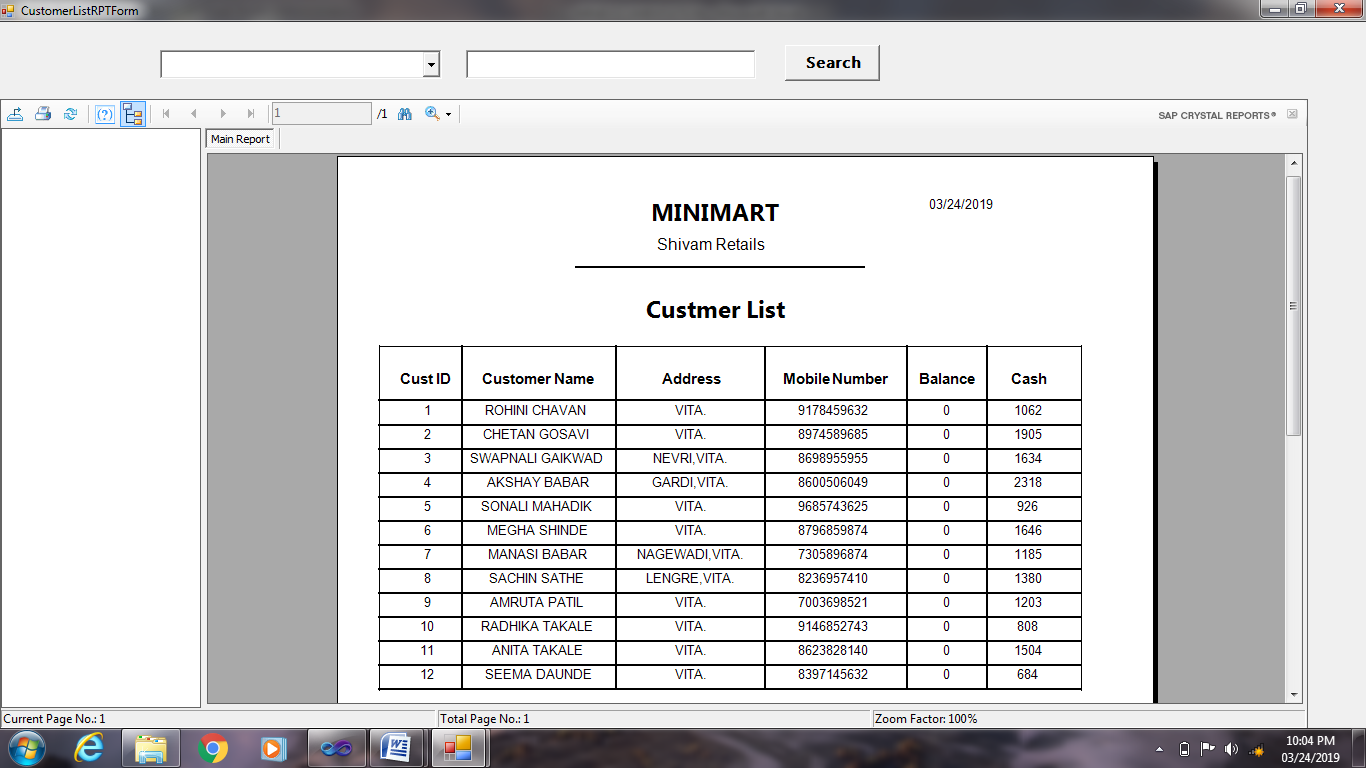
****

**Purchase Return Done:**

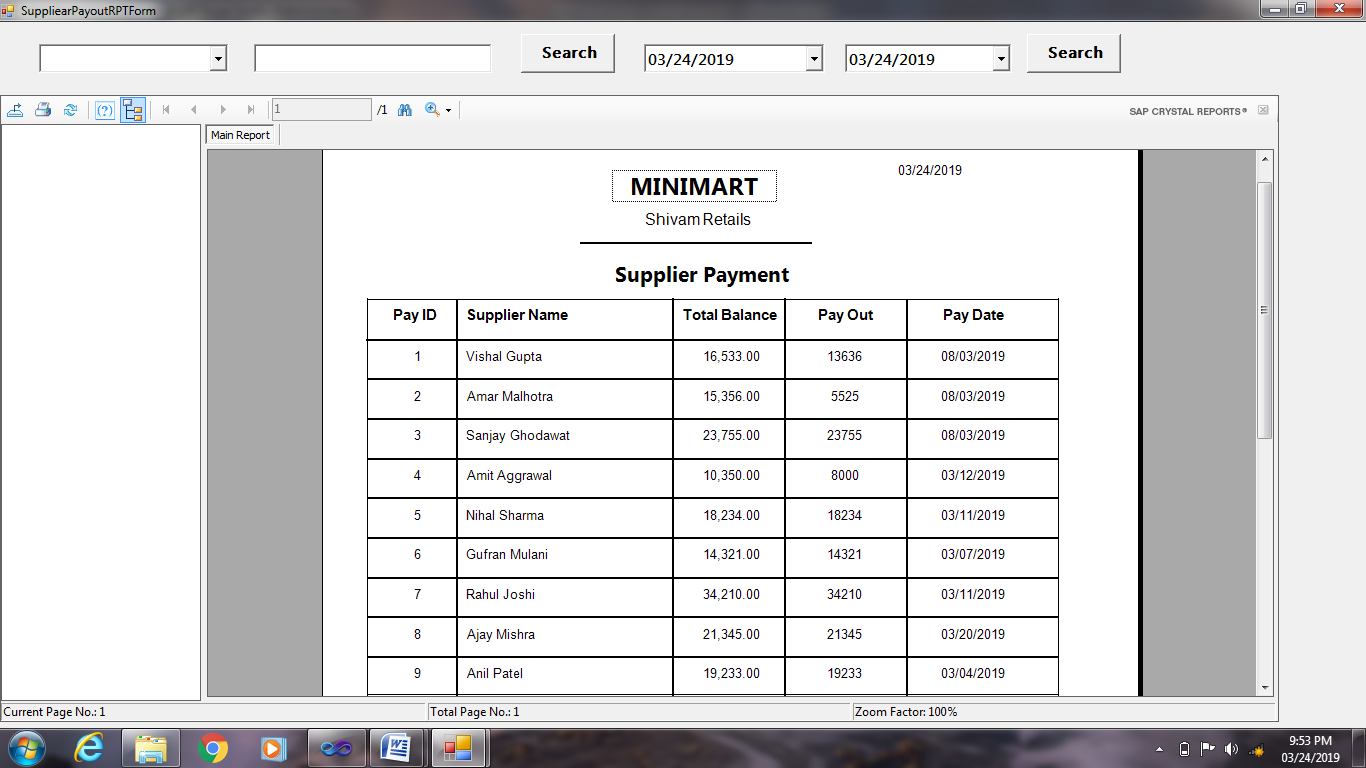
****

**REPORTS**

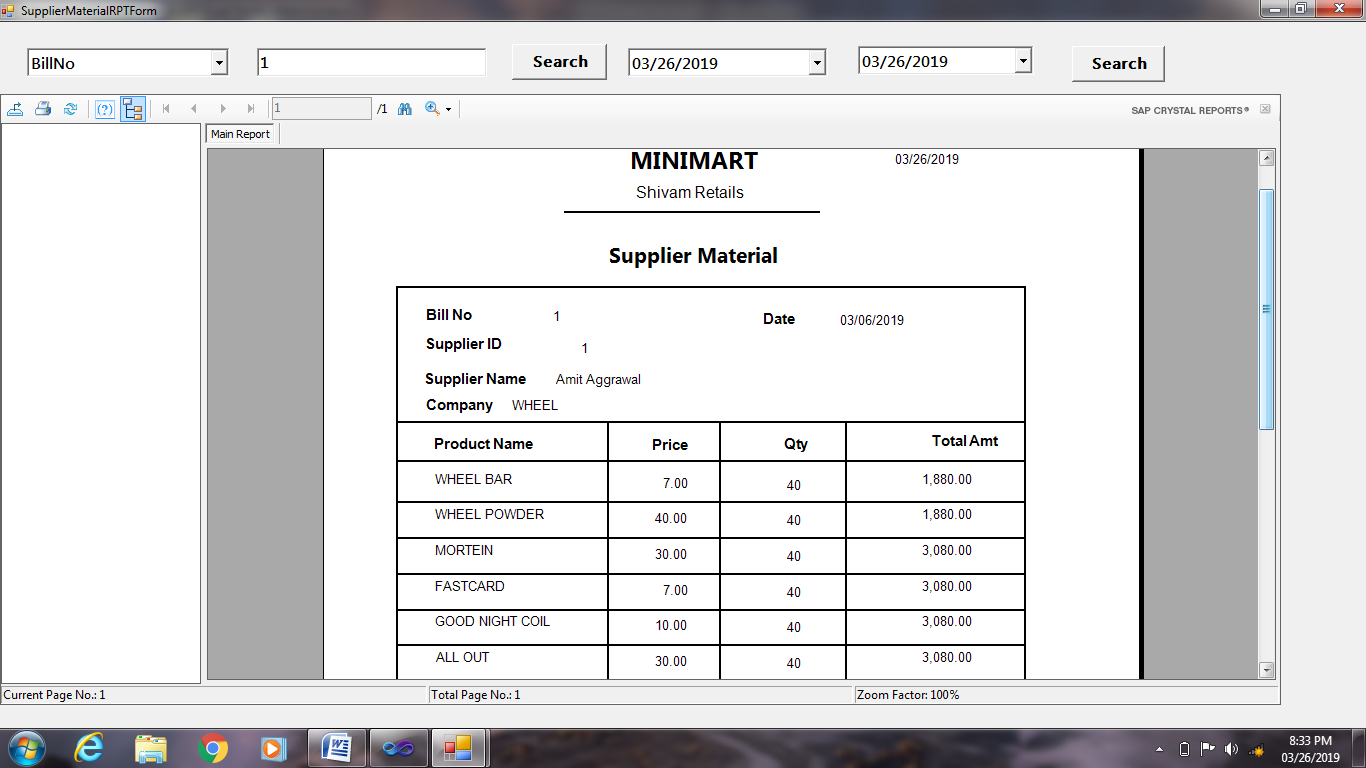
**Customer Records:**

****

**Supplier Payment Records:**

****

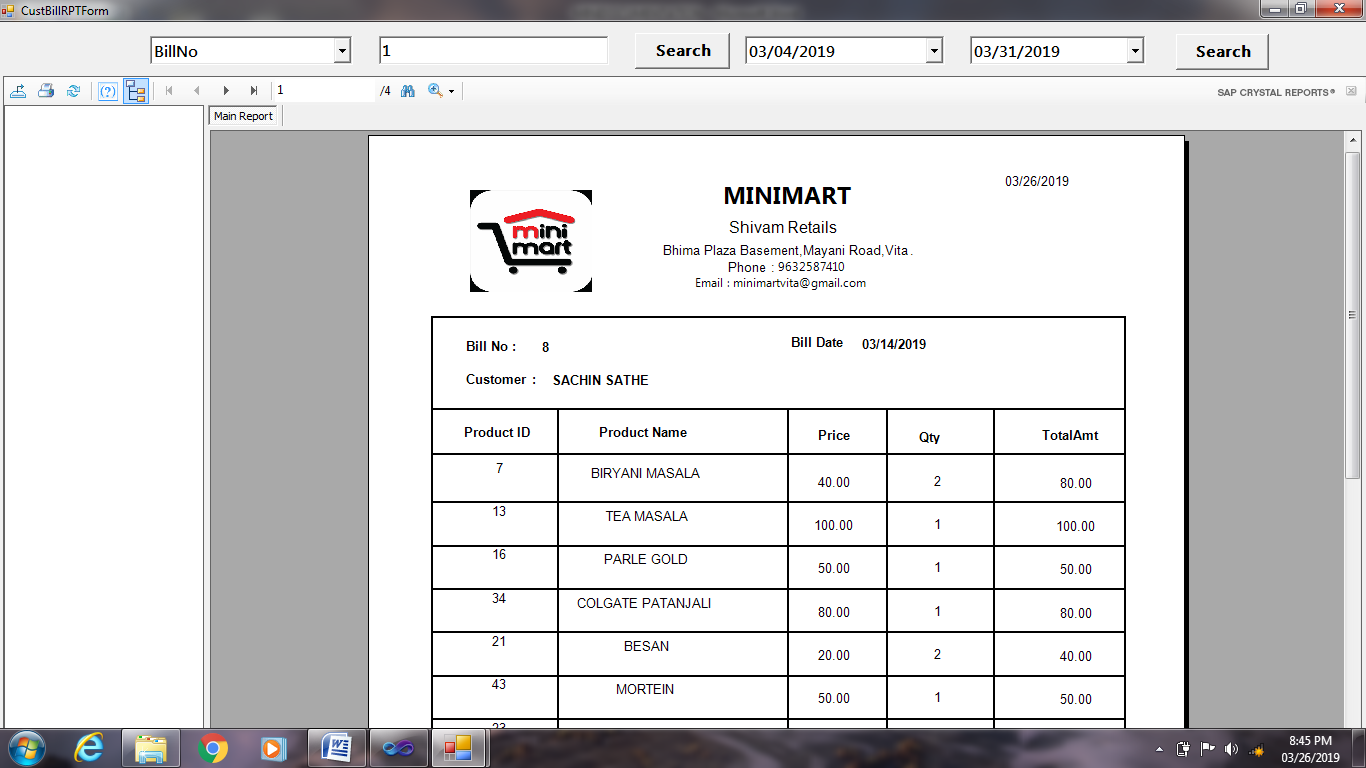
**Supplier Material Report:**

****

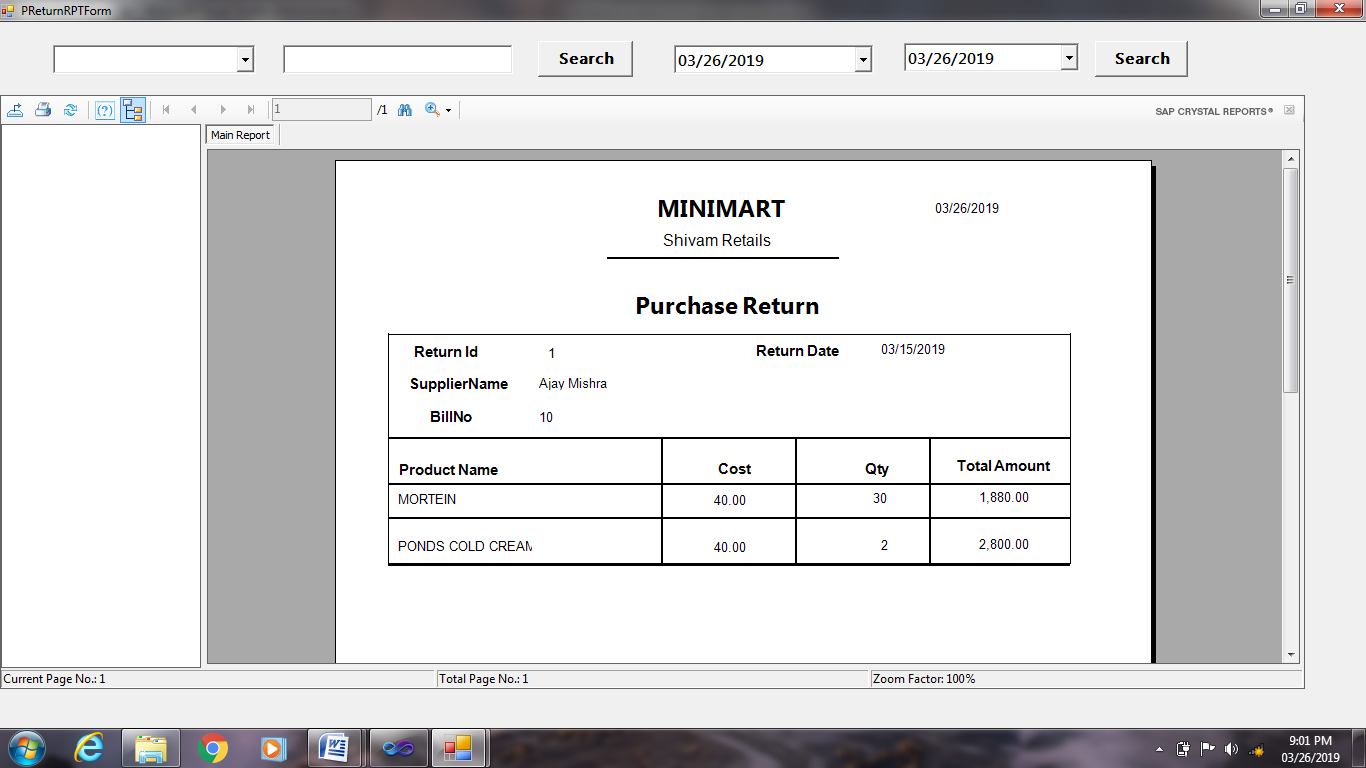
**Customer Bill Report:**

****

**Date Wise Report:**

****

**Purchase Return Report:**

****

**coding**

**Sales Return Code:**

Public Class AddSalesReturnForm

Dim BillNo As Integer

Private Sub CancelButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

End

End Sub

Private Sub AddSalesReturnForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

IDTextBox.Text = generateid("select MAX(ID) From AddSalesReturnTable")

End Sub

Sub CalculateTotal()

Dim total As Double = 0

For Each lt As ListViewItem In ListView1.Items

total += lt.SubItems(4).Text

Next

ReturnAmountTextBox.Text = total

TotalTextBox.Text = Val(BillAmtTextBox.Text) - total

End Sub

Private Sub BillButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BillButton.Click

SelectForm.SQL = "select \* from Bill"

SelectForm.FieldComboBox.Items.AddRange(New String() {"BillNo", "Customer", "BillAmt"})

SelectForm.ShowDialog()

BillNoTextBox.Text = SelectForm.row.Cells(0).Value

CustNameTextBox.Text = SelectForm.row.Cells(2).Value

BillAmtTextBox.Text = SelectForm.row.Cells(3).Value

SelectForm.Close()

End Sub

Public Sub clearcontrol1()

IDTextBox.Clear()

BillNoTextBox.Clear()

CustNameTextBox.Clear()

BillAmtTextBox.Clear()

ReturnAmountTextBox.Clear()

TotalTextBox.Clear()

End Sub

Private Sub Add\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Add.Click

Dim lt As New ListViewItem()

lt.Text = ProductIDTextbox.Text

lt.SubItems.Add(ProductNameTextBox.Text)

lt.SubItems.Add(PriceTextBox.Text)

lt.SubItems.Add(QtyTextBox.Text)

lt.SubItems.Add(PriceTextBox.Text \* QtyTextBox.Text)

ListView1.Items.Add(lt)

CalculateTotal()

ProductIDTextbox.Clear()

ProductNameTextBox.Clear()

PriceTextBox.Clear()

QtyTextBox.Clear()

QtyTextBox.Focus()

End Sub

Private Sub CodeButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CodeButton.Click

SelectForm.SQL = "select \* from BillItems"

SelectForm.FieldComboBox.Items.Clear()

SelectForm.FieldComboBox.Items.AddRange(New String() {"BillNo", "ProductID", "ProductName", "Price"})

SelectForm.ShowDialog()

ProductIDTextbox.Text = SelectForm.row.Cells(1).Value

ProductNameTextBox.Text = SelectForm.row.Cells(2).Value

PriceTextBox.Text = SelectForm.row.Cells(3).Value

QtyTextBox.Text = SelectForm.row.Cells(4).Value

End Sub

Private Sub ReturnButton\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ReturnButton.Click

ExecuteCommand(String.Format("Insert into AddSalesreturnTable (ID,Date,BillNo,Amount,ReturnAmount) values('{0}','{1}','{2}','{3}','{4}')", IDTextBox.Text, DateTimePicker1.Value.ToString("MM/dd/yyyy"), BillNoTextBox.Text, BillAmtTextBox.Text, ReturnAmountTextBox.Text))

For Each lt As ListViewItem In ListView1.Items

ExecuteCommand(String.Format("Insert into ReturndetailsTable values('{0}','{1}','{2}','{3}','{4}')", IDTextBox.Text, lt.Text, lt.SubItems(1).Text, lt.SubItems(2).Text, lt.SubItems(3).Text))

Dim dt As DataTable = gettable("select \* from billitems where BillNo=" & BillNoTextBox.Text)

For Each dr As DataRow In dt.Rows

ExecuteCommand("Update Stock set Quantity=Quantity+" & lt.SubItems(3).Text & " where HSNCode=" & dt.Rows(0)(0).ToString())

Next

Next

MessageBox.Show("Bill No " & BillNoTextBox.Text & " has refunded successfully...")

ListView1.Items.Clear()

ListView1.Refresh()

clearcontrol1()

End Sub

Private Sub CancelBtn\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CancelBtn.Click

Close()

End Sub

End Class

**Customer BillI Code:**

Public Class CustomerBillItemForm

Private Sub CancelButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RemoveButton.Click

ListView1.Items.RemoveAt(ListView1.FocusedItem.Index)

For Each lt As ListViewItem In ListView1.Items

Dim dt As DataTable = gettable("select HSNCode from Product where ProductId=" & lt.Text)

If dt.Rows.Count > 0 Then

ExecuteCommand("Update Stock set Quantity=Quantity+" & lt.SubItems(3).Text & " where HSNCode=" & dt.Rows(0)(0).ToString())

End If

Next

MessageBox.Show("Product is Removed....")

ListView1.Refresh()

CalculateTotal()

End Sub

Private Sub SaveButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveButton.Click

Dim Billno = ExecuteCommandId(String.Format("Insert into Bill (BillDate,Customer,BillAmt,GSTTax,TaxAmt,NetAmt) values('{0}','{1}','{2}','{3}','{4}','{5}') select @@identity", DateTimePicker1.Value.ToString("MM/dd/yyyy"), CustomerTextBox.Text, TotalTextBox.Text, TaxTextBox.Text, TaxAmtLabel.Text, NetAmtTextBox.Text))

For Each lt As ListViewItem In ListView1.Items

ExecuteCommand(String.Format("Insert into BillItems values('{0}','{1}','{2}','{3}','{4}')", Billno, lt.Text, lt.SubItems(1).Text, lt.SubItems(2).Text, lt.SubItems(3).Text))

Dim dt As DataTable = gettable("select HSNCode from Product where ProductId=" & lt.Text)

If dt.Rows.Count > 0 Then

ExecuteCommand("Update Stock set Quantity=Quantity-" & lt.SubItems(3).Text & " where HSNCode=" & dt.Rows(0)(0).ToString())

End If

Next

MessageBox.Show("Bill is created....")

clearcontrol1()

BillTextBox.Text = generateid("select MAX(BillNo) From Bill")

ListView1.Items.Clear()

ListView1.Refresh()

End Sub

Public Sub clearcontrol1()

CustomerTextBox.Clear()

ProductIDTextbox.Clear()

ProductNameTextBox.Clear()

PriceTextBox.Clear()

QtyTextBox.Clear()

TotalTextBox.Clear()

TaxTextBox.Clear()

TaxAmtLabel.Text = ""

NetAmtTextBox.Clear()

PaidTextBox.Clear()

QtyTextBox.Focus()

End Sub

Private Sub AddButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AddButton.Click

Dim lt As New ListViewItem()

lt.Text = ProductIDTextbox.Text

lt.SubItems.Add(ProductNameTextBox.Text)

lt.SubItems.Add(PriceTextBox.Text)

lt.SubItems.Add(QtyTextBox.Text)

lt.SubItems.Add(PriceTextBox.Text \* QtyTextBox.Text)

ListView1.Items.Add(lt)

CalculateTotal()

ProductIDTextbox.Clear()

ProductNameTextBox.Clear()

PriceTextBox.Clear()

QtyTextBox.Clear()

QtyTextBox.Focus()

End Sub

Sub CalculateTotal()

Dim total As Double = 0

For Each lt As ListViewItem In ListView1.Items

total += lt.SubItems(4).Text

Next

TotalTextBox.Text = total

End Sub

Private Sub TaxTextBox\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TaxTextBox.TextChanged

If Not String.IsNullOrEmpty(TaxTextBox.Text) Then

TaxAmtLabel.Text = TotalTextBox.Text \* (TaxTextBox.Text / 100)

NetAmtTextBox.Text = Val(TotalTextBox.Text) + Val(TaxAmtLabel.Text)

Else

NetAmtTextBox.Text = TotalTextBox.Text

End If

End Sub

Private Sub PaidTextBox\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles PaidTextBox.KeyPress

Dim ch As Char = e.KeyChar

If Not Char.IsDigit(ch) Then

If Not ch = vbBack Then

e.Handled = True

End If

End If

End Sub

Private Sub PaidTextBox\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PaidTextBox.TextChanged

If Not String.IsNullOrEmpty(PaidTextBox.Text) Then

RemainingLabel.Text = "Remaining " & (NetAmtTextBox.Text - PaidTextBox.Text)

Else

RemainingLabel.Text = "Remaining: 0"

End If

End Sub

Private Sub CustomerBillItemForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

BillTextBox.Text = generateid("select MAX(BillNo) From Bill")

ListView1.Refresh()

End Sub

Private Sub CodeButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CodeButton.Click

SelectForm.SQL = "select \* from Product"

SelectForm.FieldComboBox.Items.Clear()

SelectForm.FieldComboBox.Items.AddRange(New String() {"ProductID", "ProductName", "Price"})

SelectForm.ShowDialog()

ProductIDTextbox.Text = SelectForm.row.Cells(0).Value

ProductNameTextBox.Text = SelectForm.row.Cells(1).Value

PriceTextBox.Text = SelectForm.row.Cells(6).Value

Dim dt As DataTable = gettable("select HSNCode from Product where ProductId=" & ProductIDTextbox.Text)

If dt.Rows.Count > 0 Then

dt = gettable("select \* from Stock where HSNCode=" & dt.Rows(0)(0).ToString())

Label11.Text = dt.Rows(0)(1).ToString()

End If

QtyTextBox.Focus()

End Sub

Private Sub CustButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CustButton.Click

SelectForm.SQL = "select \* from Customer"

SelectForm.FieldComboBox.Items.AddRange(New String() {"CustID", "CustName"})

SelectForm.ShowDialog()

CustomerTextBox.Text = SelectForm.row.Cells(1).Value

End Sub

Private Sub NewCustButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewCustButton.Click

CustomerForm.ShowDialog()

End Sub

Private Sub TotalTextBox\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TotalTextBox.TextChanged

End Sub

Private Sub QtyTextBox\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles QtyTextBox.KeyPress

Dim ch As Char = e.KeyChar

If Not Char.IsDigit(ch) Then

If Not ch = vbBack Then

e.Handled = True

End If

End If

End Sub

Private Sub QtyTextBox\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles QtyTextBox.TextChanged

End Sub

End Class

**Customer Code:**

Imports System.Text.RegularExpressions

Public Class CustomerForm

Sub Refreshgrid()

Dim dt = DataModule.gettable("Select \* from Customer ")

DataGridView1.DataSource = dt

End Sub

Private Sub SaveButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveButton.Click

If Not Regex.IsMatch(MobTextBox.Text, "^\d{10}$") Then

MessageBox.Show("Mobile No should be of 10 digits")

Exit Sub

End If

If SaveButton.Text = "Save" Then

Dim sql As String

sql = String.Format("insert into Customer (CustID,CustName,Address,MobileNumber,Balance,Cash) values('{0}','{1}','{2}','{3}','{4}','{5}')", IDTextBox.Text, NameTextBox.Text, AddressTextBox.Text, MobTextBox.Text, BalanceTextBox.Text, CashTextBox.Text)

DataModule.ExecuteCommand(sql)

MessageBox.Show("Customer record is saved")

Refreshgrid()

clearcontrol1()

IDTextBox.Text = generateid("select MAX(CustID) From Customer")

Else

Dim sql As String

sql = String.Format("update Customer set CustName='{0}',Address='{1}',MobileNumber='{2}',Balance='{3}',Cash='{4}'Where CustID='{5}'", NameTextBox.Text, AddressTextBox.Text, MobTextBox.Text, BalanceTextBox.Text, CashTextBox.Text, IDTextBox.Text)

DataModule.ExecuteCommand(sql)

MessageBox.Show("Customer record is Updated")

Refreshgrid()

clearcontrol1()

IDTextBox.Text = generateid("select MAX(CustID) From Customer")

End If

End Sub

Private Sub EditButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles EditButton.Click

IDTextBox.Text = DataGridView1.CurrentRow.Cells(0).Value

NameTextBox.Text = DataGridView1.CurrentRow.Cells(1).Value

AddressTextBox.Text = DataGridView1.CurrentRow.Cells(2).Value

MobTextBox.Text = DataGridView1.CurrentRow.Cells(3).Value

BalanceTextBox.Text = DataGridView1.CurrentRow.Cells(4).Value

CashTextBox.Text = DataGridView1.CurrentRow.Cells(5).Value

SaveButton.Text = "Update"

End Sub

Private Sub DeleteButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DeleteButton.Click

If MessageBox.Show("Do you want to Delete?", "Confirm Delete", MessageBoxButtons.YesNo) = Windows.Forms.DialogResult.Yes Then

Dim CustID = DataGridView1.CurrentRow.Cells(0).Value

DataModule.ExecuteCommand("Delete from Customer where CustID=" & CustID)

Refreshgrid()

MessageBox.Show("Customer record is deleted")

IDTextBox.Text = generateid("select MAX(CustID) From Customer")

End If

End Sub

Public Sub clearcontrol1()

IDTextBox.Clear()

NameTextBox.Clear()

AddressTextBox.Clear()

MobTextBox.Clear()

BalanceTextBox.Clear()

CashTextBox.Clear()

NameTextBox.Focus()

End Sub

Private Sub CustomerForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

IDTextBox.Text = generateid("select MAX(CustID) From Customer")

Refreshgrid()

End Sub

Private Sub SearchTextBox\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SearchTextBox.TextChanged

Dim sql As String

sql = String.Format("select CustID,CustName,Address,Balance from Customer where {0} like '%{1}%'", FieldComboBox.Text, SearchTextBox.Text)

DataGridView1.DataSource = DataModule.gettable(sql)

End Sub

Private Sub NameTextBox\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles NameTextBox.KeyPress

Dim ch As Char = e.KeyChar

If Not Char.IsLetter(ch) Then

If Not Char.IsWhiteSpace(ch) Then

If Not ch = vbBack Then

e.Handled = True

MessageBox.Show("Please Enter Character Only")

End If

End If

End If

End Sub

Private Sub ResetButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RemoveButton.Click

Close()

End Sub

Private Sub DataGridView1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick

End Sub

Private Sub PrintButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PrintButton.Click

CustomerListRPTForm.Show()

CustomerListRPTForm.Customerlistreport()

End Sub

End Class

**Purchase Order Code:**

Public Class POForm

Dim sid As Integer

Private Sub SupplierButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SupplierButton.Click

SelectForm.SQL = "select \* from SupplierTable"

SelectForm.FieldComboBox.Items.AddRange(New String() {"SupplierID", "SupplierName", "Company"})

SelectForm.ShowDialog()

sid = SelectForm.row.Cells(0).Value

SupplierIDTextBox.Text = SelectForm.row.Cells(0).Value

SupplierNameTextBox.Text = SelectForm.row.Cells(1).Value

CompanyTextBox.Text = SelectForm.row.Cells(2).Value

End Sub

Private Sub CancelButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnCancel.Click

Close()

End Sub

Private Sub SaveButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveButton.Click

ExecuteCommand(String.Format("Insert into PurchaseOrderTable (PONo,PODate,SupplierID,PreparedBy) values('" & PONTextBox.Text & "','" & DateTimePicker1.Value.ToString("MM/dd/yyyy") & "'," & sid & ",'" & PBTextBox.Text & "')"))

Dim qry As String

Dim i As Integer

i = 0

For Each lt As ListViewItem In ListView1.Items

qry = String.Format("Insert into POrderItems (PONo,ProductNo,ProductName,Qty) values(" & PONTextBox.Text & "," & ListView1.Items(i).SubItems(0).Text & ",'" & ListView1.Items(i).SubItems(1).Text & "'," & ListView1.Items(i).SubItems(2).Text & ")")

ExecuteCommand(qry)

i = i + 1

Next

MessageBox.Show("Purchase Order is Created Successfully")

clearcontrol1()

PONTextBox.Text = generateid("select MAX(PONo) From PurchaseOrderTable")

ListView1.Items.Clear()

ListView1.Refresh()

End Sub

Public Sub clearcontrol1()

SupplierIDTextBox.Clear()

SupplierNameTextBox.Clear()

CompanyTextBox.Clear()

CodeTextBox.Clear()

NameTextBox.Clear()

QtyTextBox.Clear()

PBTextBox.Clear()

TotalTextBox.Clear()

QtyTextBox.Focus()

End Sub

Private Sub CodeButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CodeButton.Click

SelectForm.SQL = "select \* from Product"

SelectForm.FieldComboBox.Items.Clear()

SelectForm.FieldComboBox.Items.AddRange(New String() {"ProductID", "ProductName", "CategoryId", "CompanyName", "HSNCode"})

SelectForm.ShowDialog()

CodeTextBox.Text = SelectForm.row.Cells(0).Value

NameTextBox.Text = SelectForm.row.Cells(1).Value

QtyTextBox.Focus()

End Sub

Private Sub AddButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AddButton.Click

Dim lt As New ListViewItem()

lt.Text = CodeTextBox.Text

lt.SubItems.Add(NameTextBox.Text)

lt.SubItems.Add(QtyTextBox.Text)

ListView1.Items.Add(lt)

GetQty()

CodeTextBox.Clear()

NameTextBox.Clear()

QtyTextBox.Clear()

CodeTextBox.Focus()

End Sub

Sub GetQty()

Dim t = 0

For Each lt As ListViewItem In ListView1.Items

t = t + lt.SubItems(2).Text

Next

TotalTextBox.Text = t

End Sub

Private Sub DeleteButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DeleteButton.Click

If MessageBox.Show("Do you want to delete selected Purchase Order?", "Delete Order", MessageBoxButtons.YesNo, MessageBoxIcon.Question) = Windows.Forms.DialogResult.Yes Then

ExecuteCommand("delete from POrderItems where OrderNo=" & ListView1.FocusedItem.Text)

ExecuteCommand("delete from PurchaseOrderTable where OrderNo=" & ListView1.FocusedItem.Text)

FillListView(ListView1, "select \* from POView")

GetQty()

PONTextBox.Text = generateid("select MAX(PONo) From PurchaseOrderTable")

End If

End Sub

Private Sub POForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

PONTextBox.Text = generateid("select MAX(PONo) From PurchaseOrderTable")

End Sub

Private Sub QtyTextBox\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles QtyTextBox.KeyPress

Dim ch As Char = e.KeyChar

If Not Char.IsDigit(ch) Then

If Not ch = vbBack Then

e.Handled = True

End If

End If

End Sub

End Class

**Product Code:**

Public Class ProductForm

Dim id As String

Private Property Clear As String

Sub Refreshgrid()

Dim dt = DataModule.gettable("Select \* from Product")

DataGridView1.DataSource = dt

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BrowseButton.Click

AddHandler HSNForm.DataGridView1.MouseClick, AddressOf f1

HSNForm.ShowDialog()

End Sub

Sub f1(ByVal sender As Object, ByVal e As MouseEventArgs)

Dim dg As DataGridView = sender

HSNTextBox.Text = dg.CurrentRow.Cells(0).Value

HSNForm.Close()

End Sub

Private Sub Button5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveButton.Click

If SaveButton.Text = "Save" Then

Dim sql As String

sql = String.Format("insert into Product (ProductID,ProductName,Category,CompanyName,Unit,HSNCode,Price) values('{0}','{1}','{2}','{3}','{4}','{5}','{6}')", CodeTextBox.Text, NameTextBox.Text, CategoryComboBox.Text, CompanyComboBox.Text, UnitTextBox.Text, HSNTextBox.Text, PriceTextBox.Text)

DataModule.ExecuteCommand(sql)

Dim dt As DataTable = gettable("select \* from Stock where HSNCode='" & HSNTextBox.Text & "'")

If dt.Rows.Count = 0 Then

ExecuteCommand("Insert into Stock values('" & HSNTextBox.Text & "',1)")

Else

ExecuteCommand("Update Stock set Quantity=Quantity+1 where HSNCode='" & HSNTextBox.Text & "'")

End If

MessageBox.Show("Product record is saved")

Refreshgrid()

clearcontrol1()

CodeTextBox.Text = generateid("select MAX(ProductID) From Product")

Else

Dim sql As String

sql = String.Format("update Product set ProductName='{0}',Category='{1}',CompanyName='{2}',Unit='{3}',HSNCode='{4}',Price='{5}' where ProductID ='{6}' ", NameTextBox.Text, CategoryComboBox.Text, CompanyComboBox.Text, UnitTextBox.Text, HSNTextBox.Text, PriceTextBox.Text, CodeTextBox.Text)

DataModule.ExecuteCommand(sql)

MessageBox.Show("Product record is updated")

Refreshgrid()

clearcontrol1()

CodeTextBox.Text = generateid("select MAX(ProductID) From Product")

End If

End Sub

Public Sub clearcontrol1()

CodeTextBox.Clear()

NameTextBox.Clear()

UnitTextBox.Clear()

HSNTextBox.Clear()

PriceTextBox.Clear()

SaveButton.Text = "Save"

NameTextBox.Focus()

CategoryComboBox.SelectedIndex = 0

CompanyComboBox.SelectedIndex = 0

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles EditButton.Click

CodeTextBox.Text = DataGridView1.CurrentRow.Cells(0).Value

NameTextBox.Text = DataGridView1.CurrentRow.Cells(1).Value

CategoryComboBox.Text = DataGridView1.CurrentRow.Cells(2).Value

CompanyComboBox.Text = DataGridView1.CurrentRow.Cells(3).Value

UnitTextBox.Text = DataGridView1.CurrentRow.Cells(4).Value

HSNTextBox.Text = DataGridView1.CurrentRow.Cells(5).Value

PriceTextBox.Text = DataGridView1.CurrentRow.Cells(6).Value

SaveButton.Text = "Update"

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DeleteButton.Click

If MessageBox.Show("Do you want to Delete?", "Confirm Delete", MessageBoxButtons.YesNo) = Windows.Forms.DialogResult.Yes Then

Dim ProductID = DataGridView1.CurrentRow.Cells(0).Value

DataModule.ExecuteCommand("delete from Product where ProductId=" & ProductID)

Refreshgrid()

MessageBox.Show("Product record is deleted")

CodeTextBox.Text = generateid("select MAX(ProductID) From Product")

End If

End Sub

Private Sub GroupBox1\_Enter(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles GroupBox1.Enter

Refreshgrid()

FillCombo(CategoryComboBox, "select \* from ProductCategory", "CategoryName", "CategoryId")

FillCombo(CompanyComboBox, "select \* from Company", "CompanyName", "CompanyName")

End Sub

Private Sub SearchTextBox\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SearchTextBox.TextChanged

Dim sql As String

sql = String.Format("select ProductId,ProductName,CompanyName from Product where {0} like '%{1}%'", FieldComboBox.Text, SearchTextBox.Text)

DataGridView1.DataSource = DataModule.gettable(sql)

End Sub

Private Sub ProductForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

CodeTextBox.Text = generateid("select MAX(ProductID) From Product")

Refreshgrid()

End Sub

Private Sub PriceTextBox\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles PriceTextBox.KeyPress

Dim ch As Char = e.KeyChar

If Not Char.IsDigit(ch) Then

If Not ch = vbBack Then

e.Handled = True

End If

End If

End Sub

End Class

**Supplier Code:**

Imports System.Text.RegularExpressions

Public Class SupplierForm

Sub Refreshgrid()

Dim dt = DataModule.gettable("Select \* from SupplierTable")

DataGridView1.DataSource = dt

End Sub

Private Sub DeleteButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DeleteButton.Click

If MessageBox.Show("Do you want to Delete?", "Confirm Delete", MessageBoxButtons.YesNo) = Windows.Forms.DialogResult.Yes Then

Dim SupplierID = DataGridView1.CurrentRow.Cells(0).Value

DataModule.ExecuteCommand("delete from SupplierTable where SupplierID=" & SupplierID)

Refreshgrid()

MessageBox.Show("Supplier record is deleted")

SNoTextBox.Text = generateid("select MAX(SupplierID) From SupplierTable")

End If

End Sub

Private Sub SaveButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveButton.Click

If Not Regex.IsMatch(MNumberTextBox.Text, "^\d{10}$") Then

MessageBox.Show("Mobile No should be of 10 digits")

Exit Sub

End If

If SaveButton.Text = "Save" Then

Dim sql As String

sql = String.Format("insert into SupplierTable (SupplierID,SupplierName,CompanyName,Address,MobileNumber,Email) values('{0}','{1}','{2}','{3}','{4}','{5}')", SNoTextBox.Text, NameTextBox.Text, CNameComboBox.SelectedValue, AddressTextBox.Text, MNumberTextBox.Text, EmailTextBox.Text)

DataModule.ExecuteCommand(sql)

MessageBox.Show("Supplier record is saved")

Refreshgrid()

clearcontrol1()

SNoTextBox.Text = generateid("select MAX(SupplierID) From SupplierTable")

Else

Dim sql As String

sql = String.Format("update SupplierTable set SupplierName='{0}',CompanyName='{1}',Address='{2}',MobileNumber='{3}',Email='{4}' where SupplierID='{5}' ", NameTextBox.Text, CNameComboBox.SelectedValue, AddressTextBox.Text, MNumberTextBox.Text, EmailTextBox.Text, SNoTextBox.Text)

DataModule.ExecuteCommand(sql)

MessageBox.Show("Supplier record is saved")

Refreshgrid()

clearcontrol1()

SNoTextBox.Text = generateid("select MAX(SupplierID) From SupplierTable")

End If

End Sub

Public Sub clearcontrol1()

SNoTextBox.Clear()

NameTextBox.Clear()

CNameComboBox.SelectedIndex = 0

AddressTextBox.Clear()

MNumberTextBox.Clear()

EmailTextBox.Clear()

NameTextBox.Focus()

End Sub

Private Sub EditButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles EditButton.Click

SNoTextBox.Text = DataGridView1.CurrentRow.Cells(0).Value

NameTextBox.Text = DataGridView1.CurrentRow.Cells(1).Value

CNameComboBox.Text = DataGridView1.CurrentRow.Cells(2).Value

AddressTextBox.Text = DataGridView1.CurrentRow.Cells(3).Value

MNumberTextBox.Text = DataGridView1.CurrentRow.Cells(4).Value

EmailTextBox.Text = DataGridView1.CurrentRow.Cells(5).Value

SaveButton.Text = "Update"

End Sub

Private Sub CancelButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CancelBttn.Click

End

End Sub

Private Sub SearchTextBox\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SearchTextBox.TextChanged

Dim sql As String

sql = String.Format("select SupplierID,SupplierName,CompanyName,Address from SupplierTable where {0} like '%{1}%'", FieldComboBox.Text, SearchTextBox.Text)

DataGridView1.DataSource = DataModule.gettable(sql)

End Sub

Private Sub SupplierForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

SNoTextBox.Text = generateid("select MAX(SupplierID) From SupplierTable")

FillCombo(CNameComboBox, "select \* from Company", "ComnapyName", "CompanyName")

Refreshgrid()

End Sub

Private Sub NameTextBox\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles NameTextBox.KeyPress

Dim ch As Char = e.KeyChar

If Not Char.IsLetter(ch) Then

If Not Char.IsWhiteSpace(ch) Then

If Not ch = vbBack Then

e.Handled = True

MessageBox.Show("Please Enter Character Only")

End If

End If

End If

End Sub

Private Sub EmailTextBox\_Validating(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles EmailTextBox.Validating

Dim pattern As String = "^[a-z][a-z|0-9]\*([\_][a-z|0-9]+)\*([.][a-z|0-9]+([\_][a-z|0-9]+)\*)?@[a-z|0-9|]\*\.([a-z][a-z|0-9]\*(\.[a-z][a-z|0-9]\*)?)$"

Dim match As System.Text.RegularExpressions.Match = Regex.Match(EmailTextBox.Text.Trim(), pattern, RegexOptions.IgnoreCase)

If (match.Success) Then

MessageBox.Show("success", "checking")

Else

MessageBox.Show("Please enter a Valid email Id", "checking")

EmailTextBox.Clear()

End If

End Sub

End Class

**Supplier PayOut Code:**

Imports System.Data.SqlClient

Public Class SupplierPayOutForm

Sub Refreshgrid()

Dim dt = DataModule.gettable("Select \* from SupplierPayoutTable ")

DataGridView1.DataSource = dt

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveButton.Click

If SaveButton.Text = "Save" Then

Dim sql As String

sql = String.Format("insert into SupplierPayoutTable (PayID,SupplierName,TotalBalance,Payout,PayDate,PayDetails) values('{0}','{1}','{2}','{3}','{4}','{5}')", payIDTextBox.Text, SuppliernameComboBox.SelectedValue, TotalbalTextBox.Text, PayoutTextBox.Text, DateTimePicker1.Value, PaydetailsTextBox.Text)

DataModule.ExecuteCommand(sql)

MessageBox.Show("Supplier Payout record is saved")

Refreshgrid()

clearcontrol1()

payIDTextBox.Text = generateid("select MAX(PayID) From SupplierPayoutTable")

Else

Dim sql As String

sql = String.Format("update SupplierPayoutTable set SupplierName='{0}',TotalBalance='{1}',Payout='{2}',PayDate='{3}',PayDetails='{4}' where PayID='{5}' ", SuppliernameComboBox.SelectedValue, TotalbalTextBox.Text, PayoutTextBox.Text, DateTimePicker1.Value, PaydetailsTextBox.Text, payIDTextBox.Text)

DataModule.ExecuteCommand(sql)

MessageBox.Show("Supplier Payout record is updated")

Refreshgrid()

clearcontrol1()

payIDTextBox.Text = generateid("select MAX(PayID) From SupplierPayoutTable")

End If

End Sub

Private Sub SupplierPayOutForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

payIDTextBox.Text = generateid("select MAX(PayID) From SupplierPayoutTable")

FillCombo(SuppliernameComboBox, "select \* from SupplierTable", "SupplierName", "SupplierName")

Refreshgrid()

End Sub

Private Sub DeleteButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DeleteButton.Click

If MessageBox.Show("Do you want to Delete?", "Confirm Delete", MessageBoxButtons.YesNo) = Windows.Forms.DialogResult.Yes Then

Dim PayID = DataGridView1.CurrentRow.Cells(0).Value

DataModule.ExecuteCommand("delete from SupplierPayoutTable where PayID=" & PayID)

Refreshgrid()

MessageBox.Show("Supplier record is deleted")

payIDTextBox.Text = generateid("select MAX(PayID) From SupplierPayoutTable")

End If

End Sub

Public Sub clearcontrol1()

payIDTextBox.Clear()

SuppliernameComboBox.SelectedItem = -1

TotalbalTextBox.Clear()

PayoutTextBox.Clear()

PaydetailsTextBox.Clear()

SuppliernameComboBox.Focus()

End Sub

Private Sub CancelButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BttnCancel.Click

End

End Sub

Private Sub EditButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles EditButton.Click

payIDTextBox.Text = DataGridView1.CurrentRow.Cells(0).Value

SuppliernameComboBox.Text = DataGridView1.CurrentRow.Cells(1).Value

TotalbalTextBox.Text = DataGridView1.CurrentRow.Cells(2).Value

PayoutTextBox.Text = DataGridView1.CurrentRow.Cells(3).Value

DateTimePicker1.Value = DataGridView1.CurrentRow.Cells(4).Value

PaydetailsTextBox.Text = DataGridView1.CurrentRow.Cells(5).Value

SaveButton.Text = "Update"

End Sub

Private Sub SearchTextBox\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SearchTextBox.TextChanged

Dim sql As String

sql = String.Format("select PayID,SupplierName,TotalBalance from SupplierPayoutTable where {0} like '%{1}%'", FieldComboBox.Text, SearchTextBox.Text)

DataGridView1.DataSource = DataModule.gettable(sql)

End Sub

Private Sub TotalbalTextBox\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TotalbalTextBox.KeyPress

Dim ch As Char = e.KeyChar

If Not Char.IsDigit(ch) Then

If Not ch = vbBack Then

e.Handled = True

End If

End If

End Sub

Private Sub PayoutTextBox\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles PayoutTextBox.KeyPress

Dim ch As Char = e.KeyChar

If Not Char.IsDigit(ch) Then

If Not ch = vbBack Then

e.Handled = True

End If

End If

End Sub

Private Sub PaydetailsTextBox\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles PaydetailsTextBox.KeyPress

Dim ch As Char = e.KeyChar

If Not Char.IsLetter(ch) Then

If Not Char.IsWhiteSpace(ch) Then

If Not ch = vbBack Then

e.Handled = True

MessageBox.Show("Please Enter Character Only")

End If

End If

End If

End Sub

End Class

**Supplier Material Code:**

Public Class AddSupplierMaterialForm

Private Sub Save\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveButton.Click

Dim qry As String

qry = "Insert into AddSupplierMaterialTable (BillNo,SupplierID,SupplierName,Company,Date,TotalAmt) values(" + BillNoTextBox.Text + "," + SupplierIDTextBox.Text + ",'" + NameTextBox.Text + "','" + CompanyTextBox.Text + "','" + DateTimePicker1.Value.ToString("MM/dd/yyyy") + "'," + TotalTextBox.Text + ")"

ExecuteCommand(qry)

For Each lt As ListViewItem In ListView1.Items

ExecuteCommand(String.Format("Insert into SupplierMaterialItems values({0},{1},'{2}',{3},{4})", BillNoTextBox.Text, SupplierIDTextBox.Text, lt.SubItems(0).Text, lt.SubItems(1).Text, lt.SubItems(2).Text))

Next

MessageBox.Show("Bill is created....")

clearcontrol1()

BillNoTextBox.Text = generateid("select MAX(BillNo) From AddSupplierMaterialTable")

ListView1.Items.Clear()

ListView1.Refresh()

End Sub

Public Sub clearcontrol1()

BillNoTextBox.Clear()

SupplierIDTextBox.Clear()

NameTextBox.Clear()

PriceTextBox.Clear()

QtyTextBox.Clear()

TotalTextBox.Clear()

QtyTextBox.Focus()

End Sub

Private Sub Cancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Cancel.Click

End

End Sub

Private Sub Add\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Add.Click

Dim lt As New ListViewItem()

lt.Text = ProNameTextBox.Text

lt.SubItems.Add(PriceTextBox.Text)

lt.SubItems.Add(QtyTextBox.Text)

lt.SubItems.Add(PriceTextBox.Text \* QtyTextBox.Text)

ListView1.Items.Add(lt)

CalculateTotal()

PriceTextBox.Clear()

QtyTextBox.Clear()

PriceTextBox.Focus()

End Sub

Private Sub AddSupplierMaterialForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

BillNoTextBox.Text = generateid("select MAX(BillNo) From AddSupplierMaterialTable")

End Sub

Private Sub CodeButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CodeButton.Click

SelectForm.SQL = "select \* from SupplierTable"

SelectForm.FieldComboBox.Items.AddRange(New String() {"SupplierID", "SupplierName", "Company"})

SelectForm.ShowDialog()

SupplierIDTextBox.Text = SelectForm.row.Cells(0).Value

NameTextBox.Text = SelectForm.row.Cells(1).Value

CompanyTextBox.Text = SelectForm.row.Cells(2).Value

SelectForm.Close()

End Sub

Sub CalculateTotal()

Dim total As Double = 0

For Each lt As ListViewItem In ListView1.Items

total += lt.SubItems(3).Text

Next

TotalTextBox.Text = total

End Sub

Private Sub RemoveItemsButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RemoveItemsButton.Click

ListView1.Items.RemoveAt(ListView1.FocusedItem.Index)

ListView1.Refresh()

CalculateTotal()

End Sub

Private Sub TotalTextBox\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TotalTextBox.TextChanged

End Sub

Private Sub PriceTextBox\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles PriceTextBox.KeyPress

Dim ch As Char = e.KeyChar

If Not Char.IsDigit(ch) Then

If Not ch = vbBack Then

e.Handled = True

End If

End If

End Sub

Private Sub QtyTextBox\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles QtyTextBox.KeyPress

Dim ch As Char = e.KeyChar

If Not Char.IsDigit(ch) Then

If Not ch = vbBack Then

e.Handled = True

End If

End If

End Sub

**Purchase Return Code:**

Public Class PurchaseReturnForm

Private Sub AddButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AddButton.Click

Dim lt As New ListViewItem()

lt.Text = ProductNameTextBox.Text()

lt.SubItems.Add(QtyTextBox.Text)

lt.SubItems.Add(CostTextBox.Text)

lt.SubItems.Add(QtyTextBox.Text \* CostTextBox.Text)

ListView1.Items.Add(lt)

calTotal()

ProductNameTextBox.Focus()

clearcontrol1()

End Sub

Sub calTotal()

Dim s As Double = 0

For Each lt As ListViewItem In ListView1.Items

s = s + CDbl(lt.SubItems(3).Text)

Next

ReturnTextBox.Text = s

TotalTextBox.Text = Val(BillAmtTextBox.Text) - s

End Sub

Private Sub SaveButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveButton.Click

ExecuteCommandId("Insert into PurchaseReturn (ReturnID,ReturnDate,SupplierName,TotalAmount) values(" & IDTextBox.Text & ",'" & DateTimePicker1.Value.ToString("MM/dd/yyyy") & "','" & SupplierTextBox.Text & "'," & TotalTextBox.Text & ")")

For Each lt As ListViewItem In ListView1.Items

ExecuteCommand("Insert into POReturnsDetail values(" & IDTextBox.Text & "," & BillNoTextBox.Text & ",'" & lt.SubItems(0).Text & "'," & lt.SubItems(1).Text & "," & lt.SubItems(2).Text & ")")

Dim dt = gettable("select \* from Product where ProductName='" & lt.Text & "'")

ExecuteCommand("Update Stock set Quantity=Quantity-" & lt.SubItems(1).Text & " where HSNCode='" & dt.Rows(0)("HSNCode").ToString() & "'")

Next

MessageBox.Show("Purchase Return Successful and Stock is Updated")

clearcontrol1()

IDTextBox.Text = generateid("select MAX(ReturnID) From PurchaseReturn")

ListView1.Items.Clear()

ListView1.Refresh()

End Sub

Public Sub clearcontrol1()

ProductNameTextBox.Clear()

QtyTextBox.Clear()

CostTextBox.Clear()

QtyTextBox.Focus()

End Sub

Private Sub PurchaseReturnForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

IDTextBox.Text = generateid("select MAX(ReturnID) From PurchaseReturn")

End Sub

Private Sub CodeButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CodeButton.Click

SelectForm.SQL = "select \* from SupplierMaterialItems"

SelectForm.FieldComboBox.Items.Clear()

SelectForm.FieldComboBox.Items.AddRange(New String() {"BillNo", "SupplierID", "ProductName", "Price", "Qty"})

SelectForm.ShowDialog()

ProductNameTextBox.Text = SelectForm.row.Cells(2).Value

CostTextBox.Text = SelectForm.row.Cells(3).Value

QtyTextBox.Text = SelectForm.row.Cells(4).Value

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Close()

End Sub

Private Sub SupplierButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SupplierButton.Click

SelectForm.SQL = "select \* from AddSupplierMaterialTable"

SelectForm.FieldComboBox.Items.Clear()

SelectForm.FieldComboBox.Items.AddRange(New String() {"SupplierID", "SupplierName", "Company"})

SelectForm.ShowDialog()

BillNoTextBox.Text = SelectForm.row.Cells(0).Value

SupplierTextBox.Text = SelectForm.row.Cells(2).Value

CompanyTextBox.Text = SelectForm.row.Cells(3).Value

BillAmtTextBox.Text = SelectForm.row.Cells(5).Value

End Sub

End Class

**REPORT CODE**

**Customer Bill Report Code:**

Imports System.Data.SqlClient

Imports CrystalDecisions.Shared

Imports CrystalDecisions.CrystalReports.Engine

Public Class CustBillRPTForm

Dim bno As Integer

Sub BillItemslistreport(ByVal bno)

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter("select \*from billview where billno=" & bno, DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "billview")

DataModule.closeconnection()

Dim rpt As New CustBillRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

Private Sub RPTForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

BillItemslistreport(bno)

End Sub

Private Sub SearchButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SearchButton.Click

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter(String.Format("select \* from billview where {0}='{1}'", FieldComboBox.Text, SearchTextBox.Text), DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "billview")

DataModule.closeconnection()

Dim rpt As New CustBillRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

Private Sub ButtonSearch\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonSearch.Click

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter(String.Format("select \* from billview where BillDate between '" & DateTimePicker1.Value.ToString("MM/dd/yyyy") & "' and '" & DateTimePicker2.Value.ToString("MM/dd/yyyy") & "'"), DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "billview")

DataModule.closeconnection()

Dim rpt As New CustBillRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

End Class

**Customer List Report Code:**

Imports System.Data.SqlClient

Imports CrystalDecisions.Shared

Imports CrystalDecisions.CrystalReports.Engine

Public Class CustomerListRPTForm

Sub Customerlistreport()

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter("select \*from Customer", DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "Customer")

DataModule.closeconnection()

Dim rpt As New CustomerListRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

Private Sub CustomerListRPTForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Customerlistreport()

End Sub

Private Sub SearchButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SearchButton.Click

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter(String.Format("select CustID,CustName from Customer where {0} like '%{1}%'", FieldComboBox.Text, SearchTextBox.Text), DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "Customer")

DataModule.closeconnection()

Dim rpt As New CustomerListRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

End Class

**Purchase Return Report Code:**

Imports System.Data.SqlClient

Imports CrystalDecisions.Shared

Imports CrystalDecisions.CrystalReports.Engine

Public Class PReturnRPTForm

Sub PurchaseReturnlistreport()

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter("select \* from PurchaseReturnView", DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "PurchaseReturnView")

DataModule.closeconnection()

Dim rpt As New PurchaseReturnRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

Private Sub PReturnRPTForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

PurchaseReturnlistreport()

End Sub

Private Sub SearchButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SearchButton.Click

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter(String.Format("select \* from PurchaseReturnView where {0} like '%{1}%'", FieldComboBox.Text, SearchTextBox.Text), DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "PurchaseReturnView")

DataModule.closeconnection()

Dim rpt As New PurchaseReturnRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

Private Sub ButtonSearch\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonSearch.Click

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter(String.Format("select \* from PurchaseReturnView where Date between '" & DateTimePicker1.Value.ToString("MM/dd/yyyy") & "' and '" & DateTimePicker2.Value.ToString("MM/dd/yyyy") & "'"), DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "PurchaseReturnView")

DataModule.closeconnection()

Dim rpt As New PurchaseReturnRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

End Class

**Product List Report Code:**

Imports System.Data.SqlClient

Imports CrystalDecisions.Shared

Imports CrystalDecisions.CrystalReports.Engine

Public Class ProductListRPTForm

Sub Productlistreport()

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter("SELECT ProductID, ProductName, Category, CompanyName, Unit, HSNCode, Price FROM Product", DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "Product")

DataModule.closeconnection()

Dim rpt As New ProductListRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

Private Sub ProductListRPTForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Productlistreport()

End Sub

Private Sub SearchButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SearchButton.Click

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter(String.Format("select ProductID,ProductName from Product where {0} like '%{1}%'", FieldComboBox.Text, SearchTextBox.Text), DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "Product")

DataModule.closeconnection()

Dim rpt As New ProductListRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

End Class

**Sales Return Report Code:**

Imports System.Data.SqlClient

Imports CrystalDecisions.Shared

Imports CrystalDecisions.CrystalReports.Engine

Public Class SalesReturnRPTForm

Sub SalesReturnlistreport()

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter("select \*from SalesReturnView", DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "SalesReturnView")

DataModule.closeconnection()

Dim rpt As New SalesReturnRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

Private Sub SearchButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SearchButton.Click

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter(String.Format("select \*from SalesReturnView where {0} like '%{1}%'", FieldComboBox.Text, SearchTextBox.Text), DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "SalesReturnView")

DataModule.closeconnection()

Dim rpt As New SalesReturnRPT()

rpt.SetDataSource(ds.Tables(0))

CrystalReportViewer1.ReportSource = rpt

End Sub

Private Sub ButtonSearch\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonSearch.Click

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter(String.Format("select \*from SalesReturnView where Date between '" & DateTimePicker1.Value.ToString("MM/dd/yyyy") & "' and '" & DateTimePicker2.Value.ToString("MM/dd/yyyy") & "'"), DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "SalesReturnView")

DataModule.closeconnection()

Dim rpt As New SalesReturnRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

End Class

**Supplier Material Report Code:**

Imports System.Data.SqlClient

Imports CrystalDecisions.Shared

Imports CrystalDecisions.CrystalReports.Engine

Public Class SupplierMaterialRPTForm

Sub Suppliermateriallistreport()

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter("select \*from Suppl,ierMaterialView", DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "Suppl,ierMaterialView")

DataModule.closeconnection()

Dim rpt As New SupplierMaterialRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

Private Sub SupplierMaterialRPTForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Suppliermateriallistreport()

End Sub

Private Sub SearchButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SearchButton.Click

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter(String.Format("select \*from Suppl,ierMaterialView where {0} like '%{1}%'", FieldComboBox.Text, SearchTextBox.Text), DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "Suppl,ierMaterialView")

DataModule.closeconnection()

Dim rpt As New SupplierMaterialRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

Private Sub ButtonSearch\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonSearch.Click

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter(String.Format("select \* from Suppl,ierMaterialView where Date between '" & DateTimePicker1.Value.ToString("MM/dd/yyyy") & "' and '" & DateTimePicker2.Value.ToString("MM/dd/yyyy") & "'"), DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "Suppl,ierMaterialView")

DataModule.closeconnection()

Dim rpt As New SupplierMaterialRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

End Class

**Supplier Payout Report Code:**

Imports System.Data.SqlClient

Imports CrystalDecisions.Shared

Imports CrystalDecisions.CrystalReports.Engine

Public Class SupplierPayoutRPTForm

Sub SupplierPaymentlistreport()

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter("select \*from SupplierPayoutTable", DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "SupplierPayoutTable")

DataModule.closeconnection()

Dim rpt As New SupplierPaymentRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

Private Sub SuppliearPayoutRPTForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

SupplierPaymentlistreport()

End Sub

Private Sub SearchButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SearchButton.Click

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter(String.Format("select \* from SupplierPayoutTable where {0} like '%{1}%'", FieldComboBox.Text, SearchTextBox.Text), DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "SupplierPayoutTable")

DataModule.closeconnection()

Dim rpt As New SupplierPaymentRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

Private Sub ButtonSearch\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonSearch.Click

DataModule.ConnectToServer()

Dim da As New SqlDataAdapter(String.Format("select \* from SupplierPayoutTable where PayDate between '" & DateTimePicker1.Value.ToString("MM/dd/yyyy") & "' and '" & DateTimePicker2.Value.ToString("MM/dd/yyyy") & "'"), DataModule.con)

Dim ds As New DataSet

da.Fill(ds, "SupplierPayoutTable")

DataModule.closeconnection()

Dim rpt As New SupplierPaymentRPT()

rpt.SetDataSource(ds)

CrystalReportViewer1.ReportSource = rpt

End Sub

End Class

****

**SYSTEM TESTING**

**SYSTEM TESTING**

A common view of testing held by user is that it is performed To prove that there is no error in the program. A proper testing & implementation plan is necessary for the success of the system implementation. System testing is critical process which has to be done by every system developer in order to test the program error may occur due to incorrect programming or simply omission in the original design.

The testing plan suggested for the **“MINIMART MANAGEMENT SYSTEM”.** Is described in details. The strategies are as follows.

* **UNIT TESTING**
* **SUBSYSTEM TESTING**
* **SYSTEM TESTNG**
* **UNIT TESTING:**

In this type of testing procedure individual program will be tested with respect to desired output from the system .After completion o f some program some test data as per specification will be used for the testing of that program to see if it work as specified. Case will taken while entering the test data so as to check out the of the data element.

* **SUBSYSTEM TESTING:**

In one system there are many sub system. In this one complete module is treated as sub system & all the program coming under that module will be tested simultaneously with respect to individual performance as well as combined performance & the result or the output obtained will be verified

against the expected or the desired output.

In one sub system of the daily management system will not work properly then it works affect performance of the system as the whole.

* **SYSTEM TESTING :**

After completion of subsystem testing all the interconnected subsystem will be tested under this testing procedure.The system will be as whole & not in term of single molule testing orin terms of individual program testing.This procedure will test the integrity of the data while it transfer one process to another.

For testing purpose necessary test data as per specification is entered & whole system will be tested

**USER MANUAL**

**USER MANUAL**

**User manual:**

We will use 5 modules in this project. These are as follows:

**Module 1: Login–Id**

This module is made for the login of users. We know that we have 3 users so

login-id is for:

* Administrator
* Data Entry Operator
* Bill Calculating Operator

**Module 2: Apply taxes and commissions**

This module is for administrator who will

* Set the taxes for the products.
* Set the commissions for the products.

**Module 3: Check the Report**

This module is also for the administrator who can generate or check the report

of the product and how many products are sold on particular date or in a period

of time.

**Module 4: Enter the information about products**

This module is for data entry operator who will

* Enter which products come in the store.
* Prices and unit of the product.

**Module-5: Calculate the bill**

This module is for bill calculating operator who will

* Calculate the bill.
* Print it.

****

**Future enhancement**

**Future enhancement**

**Future enhancement:**

* To give Discount on shopping to festival.
* To give services to customer of debit and credit card.
* To buy gadget for swapping cards.
* To make website for online shopping.

****

**ADVANTAGES & LIMITATION OF SYSTEM**

**ADVANTAGES OF SYSTEM**

**Advantages:**

1. The proposed “**MINIMART MANAGEMENT SYSTEM**“ system is interactive & less time consumable.

2. The system gives result accurate & faster than the manual System.

3. The proposed system gives various reports about system tools.

4. The system is suitable for any organizational store.

5. The system is able to show reordered level, so not necessary to check stock every day.

**LIMITAtIONS OF SYSTEM**

**Limitation:**

1. As there is requirement of computer, the organization should have the computer.

2. Skilled person is required in organization .

****

**CONCLUSION**

**CONCLUSION**

**Conclusion:**

By all discussion we can conclude that computerized system developed by me is very useful for the organization. Computerized system reduced lots of paper work. The data can be stored in computer memory and we can take hard copies of data when we want.

The other thing is that by making some changes this software can be used for other organizational store.

****

**bibliography**

**bibliography**

For the completion of this system and documentation we have referred the following websites & other.

1. **MICROSOFT VISUAL STUDIO VB.NET— ManishaJadhav/Surywanshi**

**2 . www.codeproject.com**

