TABLE OF CONTENTS

List of Figures

SYNOPSIS

SYNOPSIS

Now a days we can see many restaurants are established which serves ready-made-food. All the activities such as food item list, menus, tables with different rents, customers ‘orders, daily collection reports., etc everything done in the manual system.

Then the biggest problem arises whenever many customer places the order at the same time with varieties of food items in different tables becomes difficult for the employees to handle the customers ‘orders. Whenever they want to alter their menu, it is also a challenge for them to do. But with the computerized system all this problem can be resolved in seconds.

RESTAUARANT MANAGEMENT SYSTEM is a desktop application. This system is developed to automate the day-to-day activities of a restaurant. This system can be used by the employees to handle the customers and their orders. It makes the easier way to generate accurate reports.

Scope of project in building a computerized system for silk route to handle billing restaurant records was to include the employees who are involved in the process of billing of a customer to storage of restaurant records and enables to view the records as desired. The employees are given limited access in order to safe guard the privacy and security of the records. The database is maintained in the whole project

INTRODUCTION

CHAPTER 1

INTRODUCTION

Restaurant Management system:

Restaurant management system is the system for managing the restaurant business. Restaurant management can vary across multiple management styles, however, there is always one common denominator when it comes to setting goals: maximizing a restaurant's profitability. In order to maximize a restaurant's profitability, one has to always examine and understand a restaurant's operational costs and how these relate to a restaurant's productivity and efficiency in delivering quality service to its customers. Management takes a very important role in controlling and manipulating the balance of costs and profitability.

An effective manager must always concern himself/herself with restaurant issues that pertain to including items, categories, pricing, order-taking, and much more. Oftentimes, a restaurant's profitability either rises or falls depending on how well it is being managed.

Managing a restaurant using a well-developed software minimizes the liabilities of mismanagement and productivity. The incorporation of a Restaurant Management Software in the managing of various business processes entails that your restaurant is competitive, innovative, well-managed, and up-to-date with the latest management and business trends.

This restaurant management system can be used by employees in a restaurant to handle the clients, their orders and can help them easily find tables.

The restaurant menu is organized by categories (south-Indian meal, soups, salads, Chinese dish, sides and drinks) of menu items. Each menu item has a name, price. It's easy to present the reports of the day- to- day activities.

SPECIFICATION REQUIREMENTS

+

CHAPTER 2

SPECIFICATION REQUIREMENTS

To be used efficiently, all computer software needs certain hardware components or the other software resources to be present on a computer. These pre-requisites are known as(computer) system requirements and are often used as a guideline as opposed to an absolute rule.

2.1. HARDWARE REQUIREMENTS: The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

HARDWARE REQUIREMENTS FOR PRESENT PROJECT:

PROCESSOR	: Intel Pentium 3 or above
RAM	: 1GB or above
HARD DISK	: 128 GB or above
MONITOR	: Standard colour monitor
KEYBOARD	: Standard keyboard

2.2. SOFTWARE REQUIREMENTS: Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed

SOFTWARE REQUIREMENTS FOR PRESENT PROJECT:

OPERATING SYSTEM	: WINDOWS 10
FRONT END	: VISUAL BASICS 6.0
BACK END	: SQL SERVER

Visual Basics 6

CHAPTER 3

VISUAL BASIC 6.0

Visual basic is an ideal program language, for developing sophisticated business application. It makes use of graphical interface for the creation of robust and powerful application. The GUI as the name suggests, uses illustrations for the text which enables its user to interact with the application. This feature makes it easier to comprehend the things in a quicker and easier way

Coding in a GUI environment is quite a transition to traditional – linear programming methods, where the user is guided through path of execution and is limited to a small set of operation. The number of options that are open to the user and developer features such as easier comprehension, user friendliness, faster access and many such other features make it an interesting tool to work with.

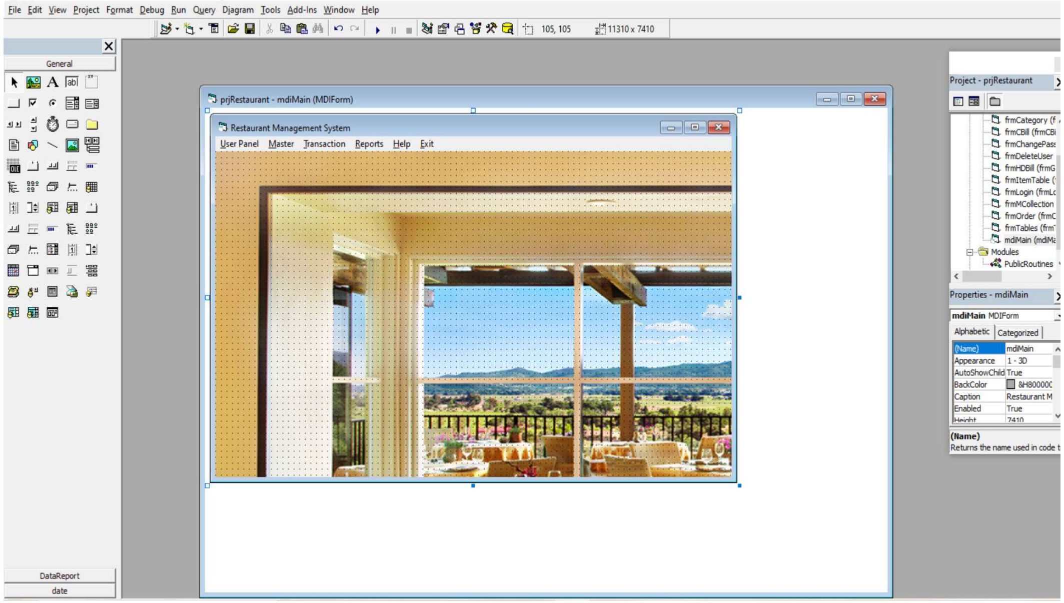
So what is Visual basic? The Visual path refers to the method used to create the Graphical User Interface (GUI). Rather than writing numerous lines of codes to describe the appearance and location of interface elements, you simply add pre-built objects into place on screen.

The “Basic” part refers to the BASIC (Beginners all-purpose symbolic instruction code) language a language used by more programmers rather than any other language in the history of computing. Visual basic has evolved from the original Basic language and now contains several hundred statements, functions and keywords many of which relate directly to the windows GUI. Whether your goal is to create a small utility for yourself or your work group, a large enterprise-wide system, or event distributed applications spanning the globe via Internet, Visual basic has the tools you need.

Data access features allow you to create databases, front-end applications and scalable sever side components for most popular database formats, including command prompt sever and other enterprise level database. ActiveX technologies allow you to use the functionally provided by the other applications, such as Microsoft word processor, Microsoft Excel spreadsheet, and other windows application. You can even automate application and objects created using the professional or Enterprise editions of visual basic.

This “Basic” part refers to the BASIC (Beginners All-purpose Symbolic Instruction code) language a language used by more programmers rather than any other language in the history of computing. Visual Basic has evolved from the original Basic language and now contains several hundred statements, functions and keywords many of which relate directly to the windows GUI. Whether your goal is to create a small utility for yourself or your work group, a large enterprise-wide system, or even distributed applications spanning the globe via Internet, Visual Basic has the tools you need. Data access features allow you to create database, front-end applications and scalable server-side components of most popular database formats, including Microsoft Access server side and other enterprise level database.

This chapter introduces the essential components of the visual basic language. After creating the interface to your application using forms and controls, you need to write the code that defines the application behavior. As with any programming language, Visual basic supports a number of common programming constructs and languages elements. Visual basic is an object oriented based programming language. The mere mention of objects may cause undue anxiety in many programmers. In this programming language you are dealing with objects and their handling. Once you understand a few basic concepts, object actually help to make programming easier than ever before.



(Name)	mdiMain
Appearance	1 - 3D
AutoShowChild	True
BackColor	&H800000
Caption	Restaurant M
Enabled	True
Height	7410

(Name)
Returns the name used in code t

SQL SERVER

CHAPTER 4

SQL SERVER

SQL Server is a database server by Microsoft. The Microsoft relational database management system is a software product which primarily stores and retrieves data requested by other applications. These applications may run on the same or a different computer.

SQL is **a specialized language for updating, deleting, and requesting information from databases**. SQL is an ANSI and ISO standard, and is the de facto standard database query language.

SQLCMD is a command line application that comes with Microsoft SQL Server, and exposes the management features of SQL Server. It allows SQL queries to be written and executed from the command prompt. It can also act as a scripting language to create and run a set of SQL statements as a script. Such scripts are stored as a sqlfile, and are used either for management of databases or to create the database schema during the deployment of a database.

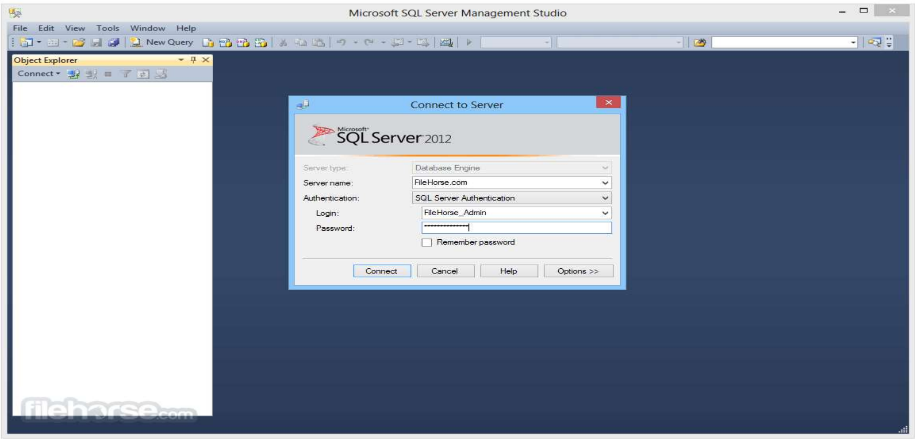
SQLCMD was introduced with SQL Server 2005 and has continued through SQL Server versions 2008, 2008 R2, 2012, 2014, 2016 and 2019. Its predecessor for earlier versions was OSQL and ISQL, which were functionally equivalent as it pertains to TSQL execution, and many of the command line parameters are identical, although SQLCMD adds extra versatility.

4.1 SQL Server Management Studio

SQL Server Management Studio is a GUI tool included with SQL Server 2005 and later for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. SQL Server Management Studio replaces Enterprise Manager as the primary management interface for Microsoft SQL Server since SQL Server 2005. A version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as *SQL Server Management Studio Express* (SSMSE).

A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyse query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyse , performance . It includes the query windows which provide a GUI based interface to write and execute queries.

Use it to deploy, monitor, and upgrade the data-tier components used by your applications, as well as build queries and scripts. Microsoft SQL Server belongs to "Databases" category of the tech stack, while Microsoft SQL Server Management Studio can be primarily classified under "Database Tools".



Server type: Database Engine
Server name: FileHorse.com
Authentication: SQL Server Authentication
Login: FileHorse_Admin
Password: ***************
□ Remember password

SQL Server Management Studio will help you to connect with four different Server Types

1. Database Engine: To work with relational databases. Here, we use the Transact SQL queries to communicate with the server.
2. Analysis Services: This is to work with SQL Server Analysis Services (SSAS).
3. Reporting Services: This Server type is to work with SQL Server Reporting Services (SSRS).
4. Integration Services: This Server type is to work with SQL Server Integration Services (SSIS).

SYSTEM DESIGN

CHAPTER 5

SYSTEM DESIGN

5.1 Data Flow Diagram (DFD)

Data flow diagram is a graphical representation of data movement, process files used in support of an information system. Unlike detail flow charts, DFDs do not supply detailed description of modules but graphically describe a system's data and how the data interact with the system. A data flow diagram represents the information at each processing points in the system and the direction it takes from the source and destination. To construct a data flow diagram, we use

- Arrows
- Circles
- Open-ended boxes
- Squares

An arrow identifies data flow or data in motion. Circle stands for a process that converts data into information. An open-ended box represents a data source or a temporary repository of data. A square defines a source or the destination of given data

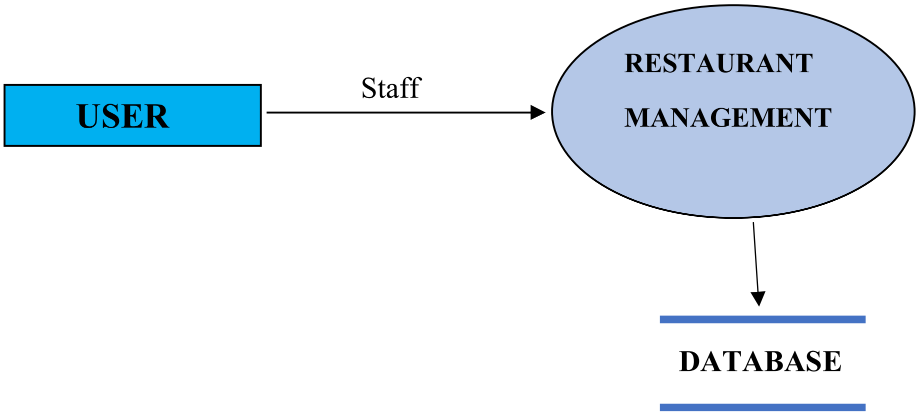
The following information rules govern construction of DFD

- Arrows should not cross each other
- Squares, circles, and files must bear names

DATA FLOW DIAGRAM

FOR

RESTAURANT MANAGEMENT SYSTE



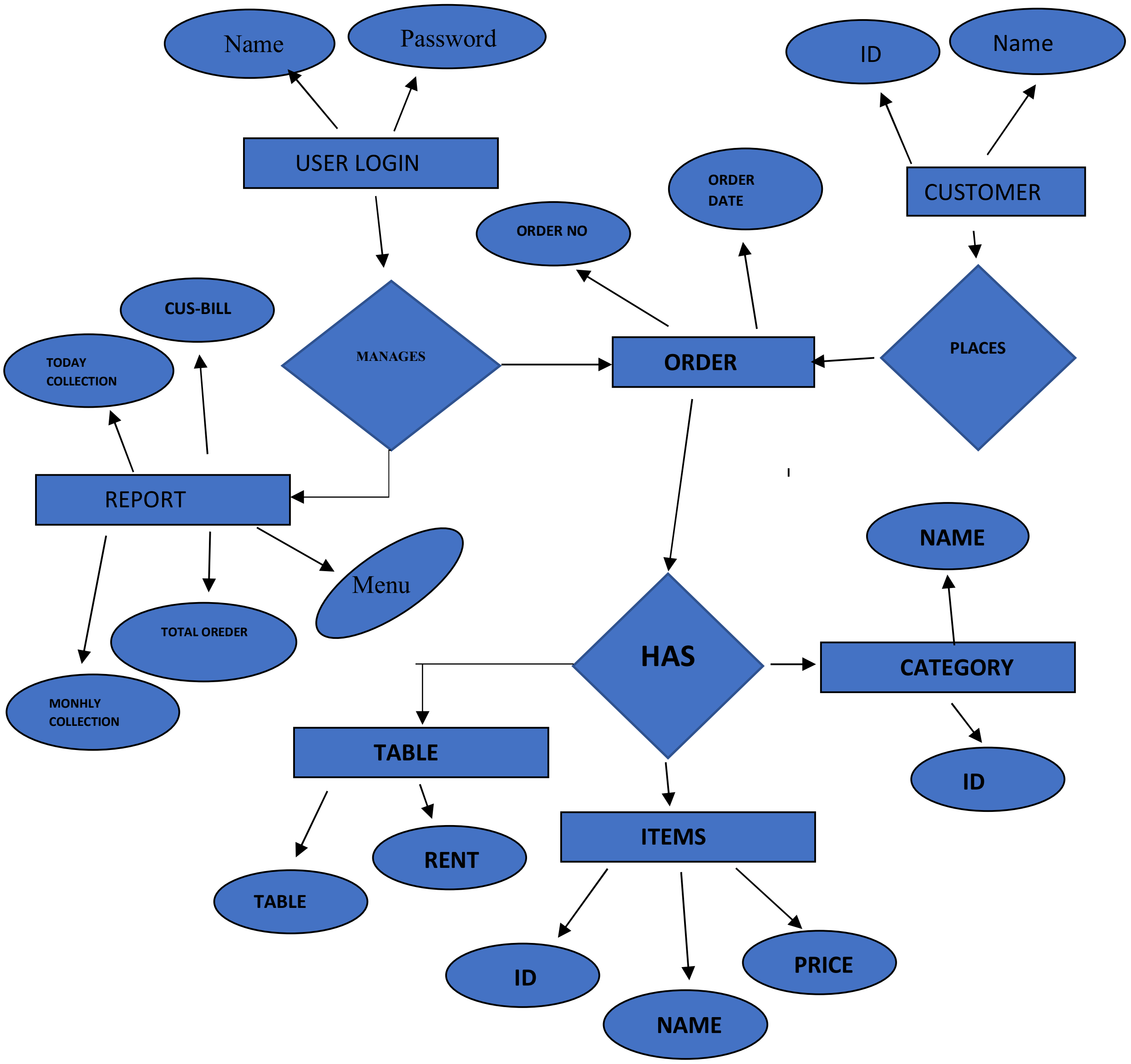
Level 0 DFD

5.2 ER Diagram

ER Diagram

For

Restaurant Management System



5.3 DATABASE DESIGN

Login Table

Data Field	Data Types
User_name	nvarchar(50)
Password	nvarchar(50)

Tables Table

Data Field	Data Types
Table_No	Int
Rent	Int

Category Table

Data Field	Data Types
Cat_Id	int
Cat_Name	nvarchar(30)

Customer Table

Data Field	Data Types
C_No	int
C_Name	nvarchar(50)

Item Table

Data Field	Data Types
Item_Name	nvarchar(50)
Rate	int
Cat_Id	int

Order Table

Data Field	Data Types
O_NO	Int
O_Date	Date
C_Name	nvarchar(50
Amount	Int
Discount	Int
BAmount	Int

Order Item Table

Data Field	Data Types
O_NO	int
O_Date	Date
C_Name	nvarchar(50)
O_Table	int
Item_Name	nvarchar(50)
Qty	int
Rate	int
ItemAmount	int
Amount	int
Discount	int
BAmount	int

SCREENSHOTS

CHAPTER 6

Screenshots

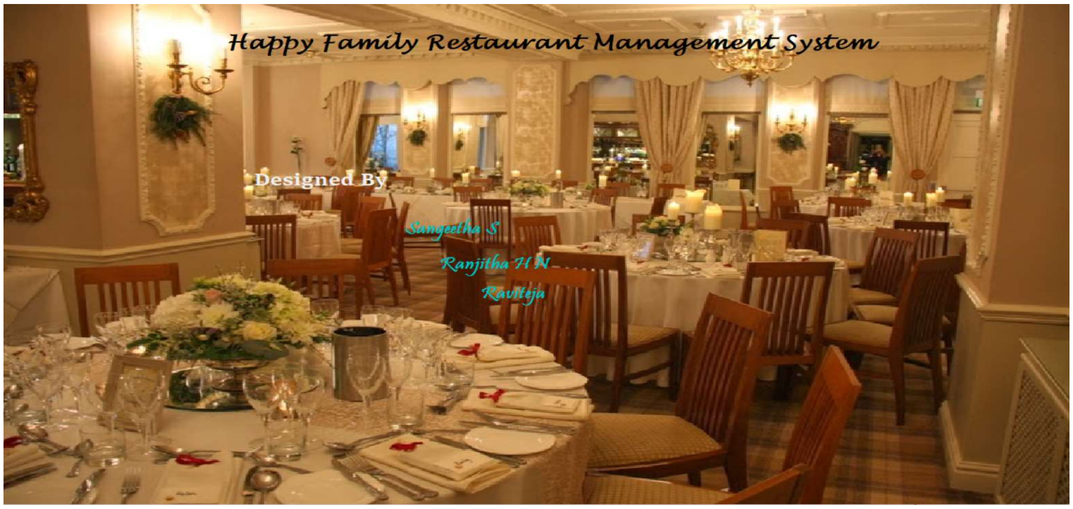


Fig 6.1: Splash Screen

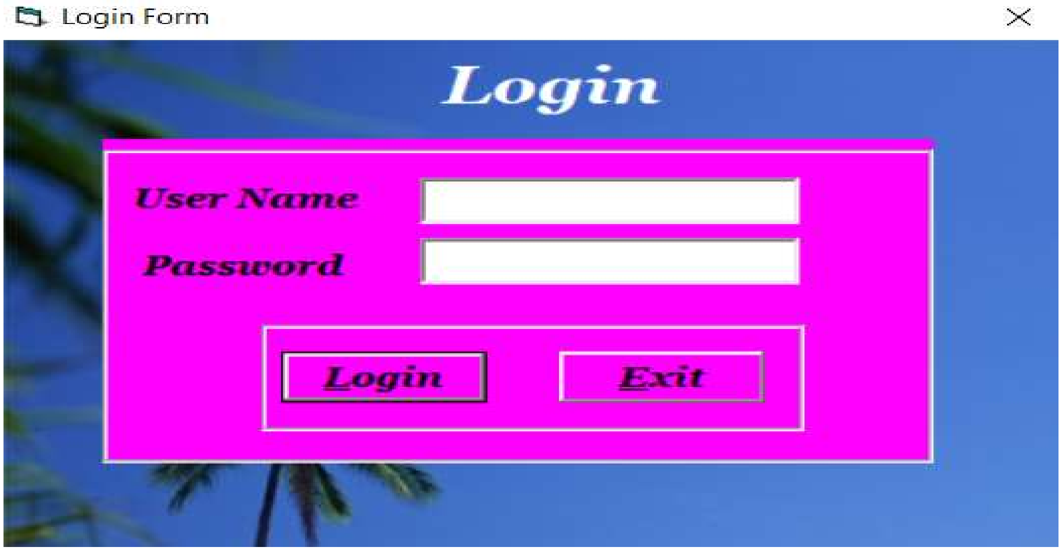
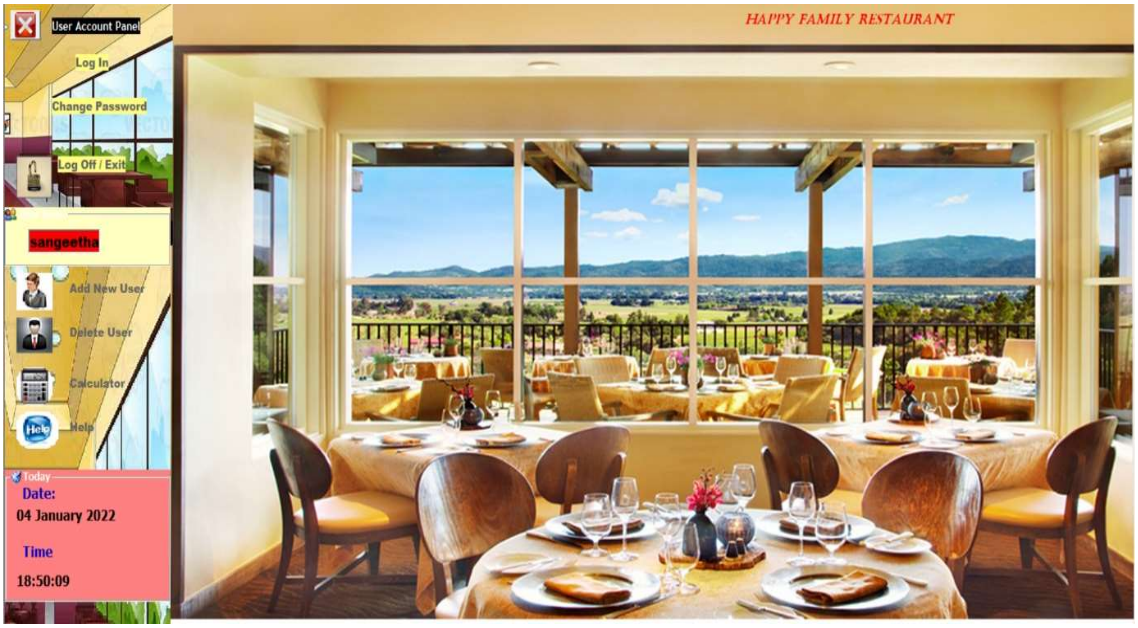


Fig6.2: Login Page

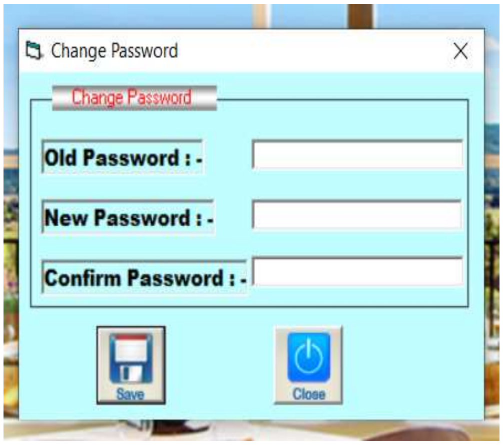


sangeetha

04 January 2022

18:50:09

Fig 6.3: Main Form



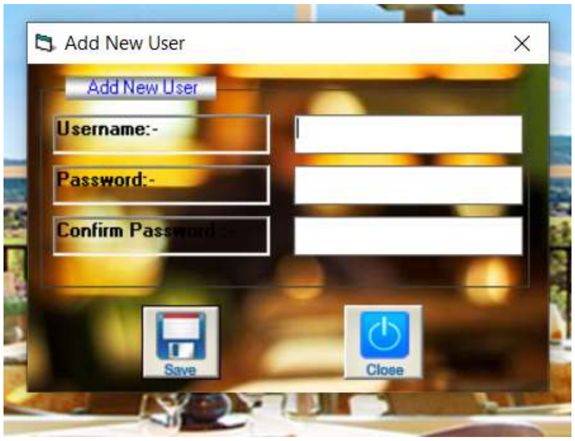
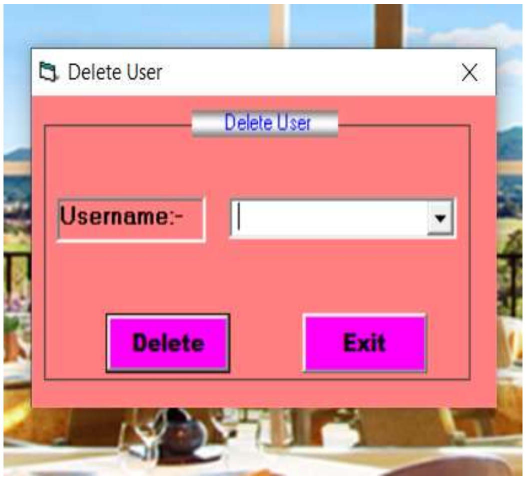


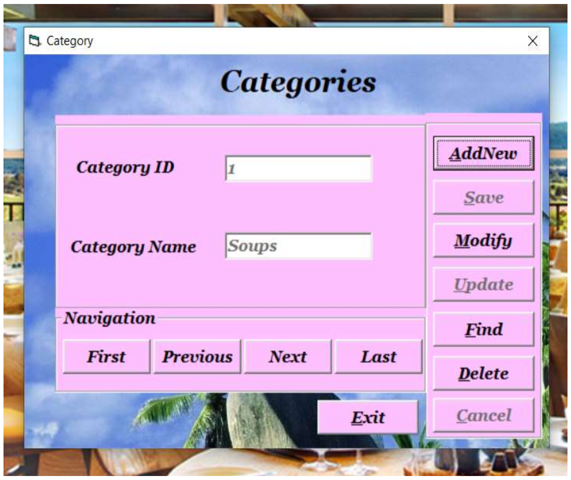
Fig 6.4: Form to Change Password

Fig 6.5: Form to Add New User



Delete User

Username:-



Category

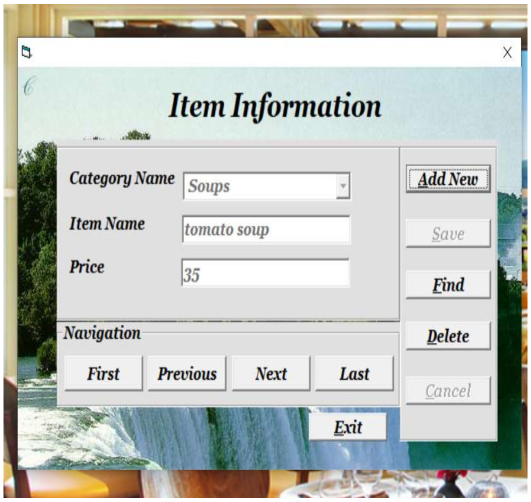
Categories

Category ID: 1

Category Name: Soups

Fig 6.6: Form to Delete User

Fig 6.7:CategoriesForm

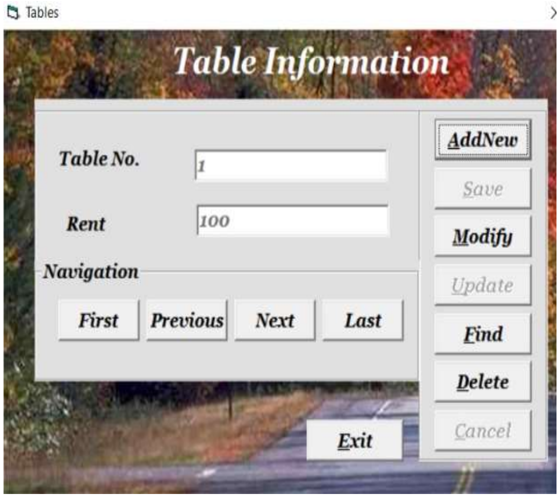


Item Information

Category Name: Soups

Item Name: tomato soup

Price: 35



Tables

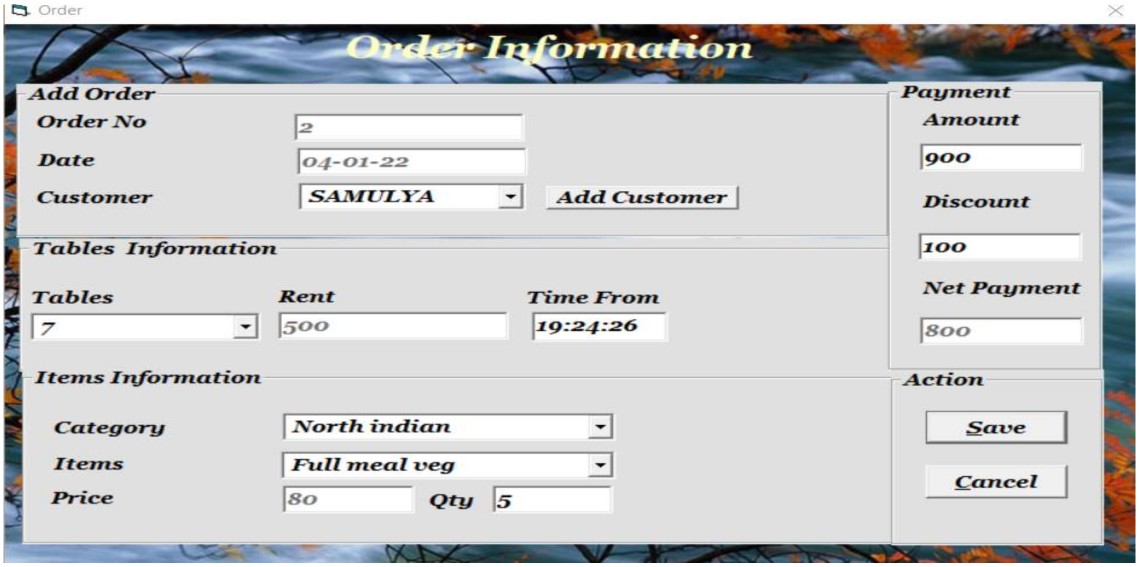
Table Information

Table No.: 1

Rent: 100

Fig 6.8: Item Information Form

Fig 6.9:Table Information Form



Order No: 2

Date: 04-01-22

Customer: SAMULYA

Tables: 7

Rent: 500

Time From: 19:24:26

Category: North indian

Items: Full meal veg

Price: 80

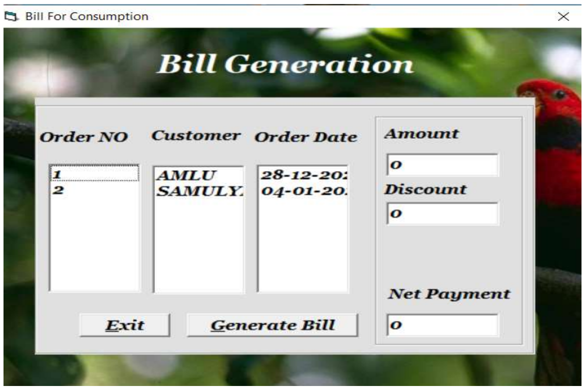
Qty: 5

Amount: 900

Discount: 100

Net Payment: 800

Fig 6.10: Order Information Form

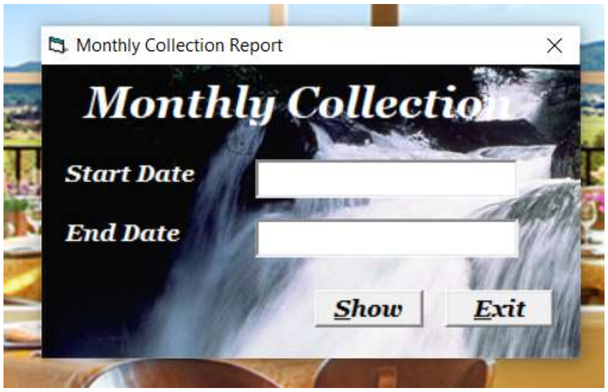


Order NO	Customer	Order Date
1	AMLU	28-12-20:
2	SAMULY.	04-01-20.

Amount: 0

Discount: 0

Net Payment: 0

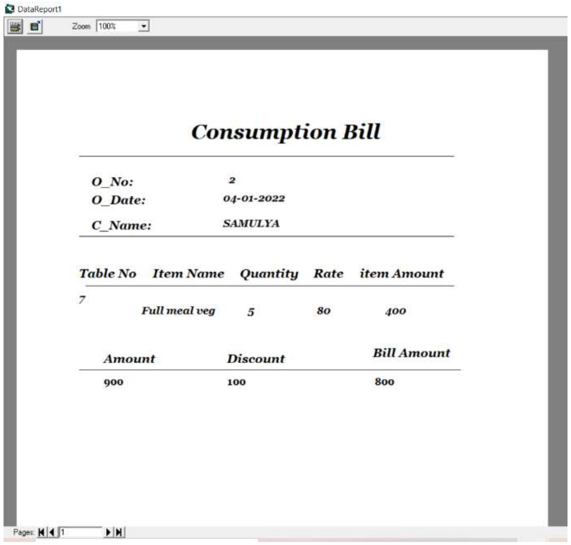


Start Date:

End Date:

Fig 6.11: Bill Generation Form

Fig6.12:Monthly Collection Form

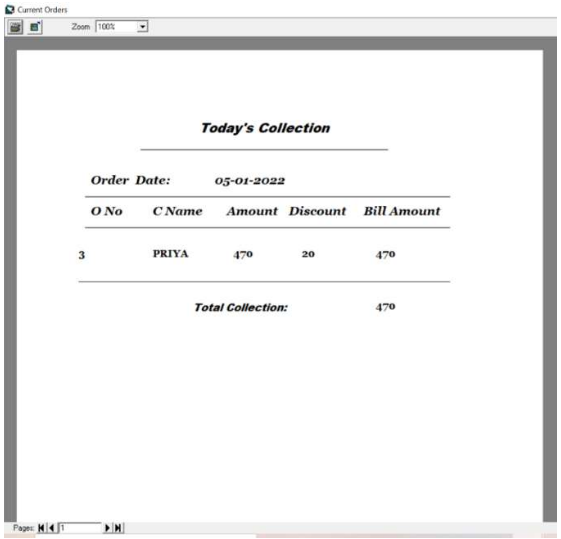


Consumption Bill

O_No: 2
O_Date: 04-01-2022
C_Name: SAMULYA

Table No	Item Name	Quantity	Rate	item Amount
7	Full meal veg	5	80	400

Amount	Discount	Bill Amount
900	100	800



Today's Collection

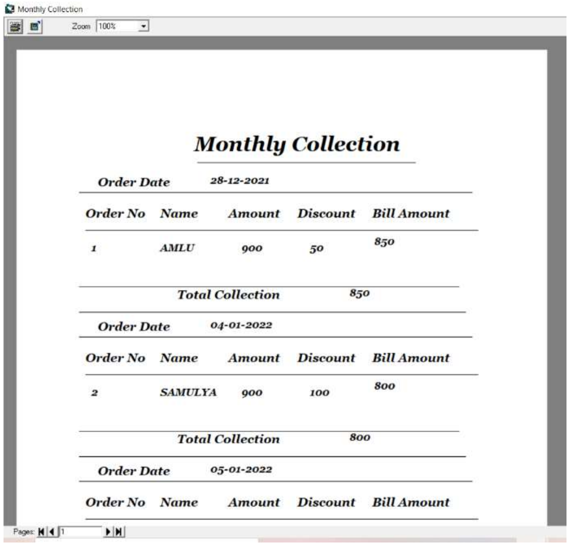
Order Date: 05-01-2022

O No	C Name	Amount	Discount	Bill Amount
3	PRIYA	470	20	470

Total Collection: 470

Fig 6.13:Consumption Bill Report

Fig6.14:Today'sCollectionReport



Monthly Collection

Order Date 28-12-2021

Order No	Name	Amount	Discount	Bill Amount
1	AMLU	900	50	850

Total Collection 850

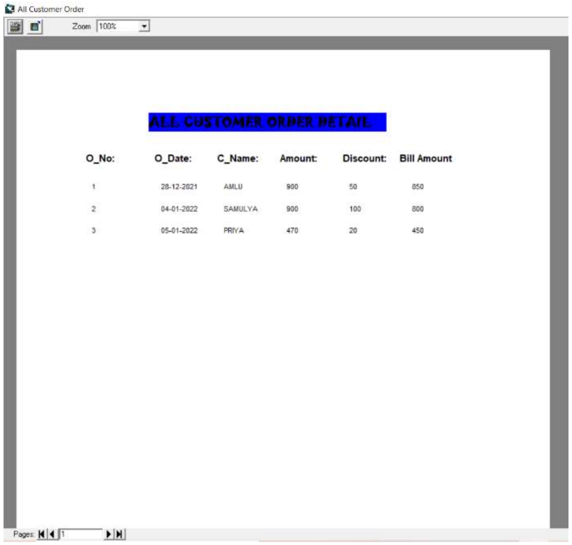
Order Date 04-01-2022

Order No	Name	Amount	Discount	Bill Amount
2	SAMULYA	900	100	800

Total Collection 800

Order Date 05-01-2022

Order No	Name	Amount	Discount	Bill Amount

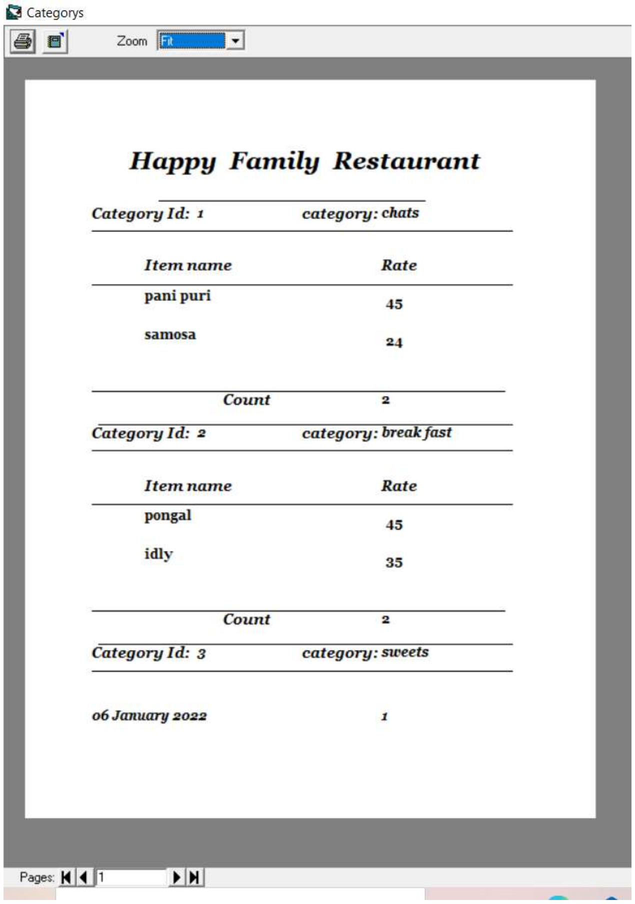


ALL CUSTOMER ORDER DETAIL

O_No:	O_Date:	C_Name:	Amount:	Discount:	Bill Amount
1	28-12-2021	AMLU	900	50	850
2	04-01-2022	SAMULYA	900	100	800
3	05-01-2022	PRIYA	470	20	450

Fig6.15: Monthly Collection Report

Fig6.16: All Customer OrderReport



Happy Family Restaurant

Category Id: 1 *category:* chats

Item name	*Rate*
pani puri	45
samosa	24

Count 2

Category Id: 2 *category:* break fast

Item name	*Rate*
pongal	45
idly	35

Count 2

Category Id: 3 *category:* sweets

06 January 2022 1

Fig 6.17 Menu Report

TESTING

CHAPTER 7

TESTING

SYSTEM TESTING

TESTING

Testing goes through the various stages, during testing the program to be tested has to be executed with a set of test cases, and ha the output of the program for the test case is evaluated to determine if the program is performing as expected. Due to its approach dynamic testing only ascertains the presence of error in the program. The exact nature of error is not usually decided by testing. Testing form is the first in determining error in the program.

Once the programs are tested individually then the system as a whole needs to be tested. During testing, the system is used experimentally to ensure that the software does not fail t.e. it will run according to its specification. The programs executed to check for any syntax or logical error. The error is corrected and test is made to determine whether the program is doing what it is supposed to do.

Various types of testing

7.1 Unit testing

Each component of the system is tested individually. The programmer does the testing. Testing is restrictive in nature i.e. programmer should try to test all individual conditions and see if the program breaks under any circumstance.

7.2 System testing

This is an integrated form of testing, which focuses on functionality and interface between units and team in a controlled environment does it.

7.3 Acceptance Testing

This is system testing done by the user of the application the only emphasis is functionally testing as the user is not aware of the technical aspect of the system. The testing is also done in a controlled environment with logging o all error based on the error found in the system, the user has to accept or reject the system.

7.4 Module Testing

This is an optional form of testing, which is done only for large system, which has a large number of modules.

7.5 Security Testing

Security testing will be done as a specialized form of testing if there is a high risk exposure in that area. If the risk exposure is not very high, then it can be done as part of the system testing. Typically, security testing would involve trying to break in to the system, trying to execute. transactions not allowed to person; to access areas on disk were the user is not allowed.

Testing is vital the success of the system. If it on this done successfully, this shows that the parts of the system are working correctly and all the goals are achieved.

CONCLUSION

CHAPTER 8

CONCLUSION

We were able to create a computerized system for Simple Route to maintain billing & Restaurant records. This system is able to store billing records securely and retrieve the records whenever needed easily. Data entering of customers, change of foods, menu etc are also included in this system along with the order and the billing process. Customers, restaurant records and employees are interconnected in order to maintain the accuracy of this system. This system can also be further improved adding many other features and including the other systems as well. Finally, we believe that we were able to launch an effective computerized system to the restaurant causing the restaurant to perform well in the future regarding the billing and restaurant records.

Each and every task of maintaining the restaurant activities are maintained very well and easily. All the activities are maintained efficiently

BIBLIOGRAPHY

CHAPTER 9

BIBLIOGRAPHY

Books:

Visual Programing

-Padma Geetha B.G

-Srikanth S

Websites:

(PDF) Restaurant Management System | rehana akhter - Academia.edu

SQL Server Management Studio (SSMS) (tutorialsteacher.com)

specify provider and connection string in VB6\Connect Database - YouTube

One-to-One, One-to-Many Table Relationships in SQL Server (tech-recipes.com)

database - Generate table relationship diagram from existing schema (SQL Server) - Stack Overflow

WinWorld: Microsoft Visual Basic 6.0 (winworldpc.com)

CODING

CHAPTER 10

CODING

Coding for Login Form:

```
Option Explicit
Dim rec As ADODB.Recordset
Private Sub cmdExit_Click()
End
End Sub
Private Sub cmdLogin_Click()
   If txtInput(0).Text = "" And txtInput(1).Text = "" Then
     MsgBox "Please Enter User Name and Password"
     Exit Sub
   End If
   If rec.RecordCount > 0 Then
     rec.MoveFirst
     rec.Find "User_name='" & txtInput(0).Text & "'"
     If rec.EOF Then
         MsgBox "Incorrect username", vbOKOnly
         txtInput(0).Text = ""
         txtInput(1).Text = ""
         txtInput(0).SetFocus
         Exit Sub
     Else
       If rec.Fields("password").Value <> txtInput(1).Text Then
         MsgBox "Incorrect password", vbOKOnly
         txtInput(0).Text = ""
         txtInput(1).Text = ""
         txtInput(0).SetFocus
Exit Sub
       End If
     End If
   End If
   Timer1.Enabled = True
   ProgressBar1.Visible = True
   Shape1.Visible = True
   Shape2.Visible = True
   Shape3.Visible = True
   Label2.Visible = True
   Label3.Visible = True
End Sub
Private Sub Form_Activate()
```

```
   txtInput(0).SetFocus
End Sub
Private Sub Form_Load()
   If Not OpenConnection Then
      MsgBox "Error while connecting with this database"
      End
   End If
   Set rec = New ADODB.Recordset
   rec.Open "Select * from Login", cn, adOpenKeyset, adLockOptimistic
End Sub

Private Sub Timer1_Timer()
    If Shape1.Visible Then
       Shape2.Visible = True
       Shape1.Visible = False
       Shape3.Visible = False
ElseIf Shape2.Visible Then
       Shape3.Visible = True
       Shape2.Visible = False
       Shape1.Visible = False
    ElseIf Shape3.Visible Then
       Shape1.Visible = True
       Shape2.Visible = False
       Shape3.Visible = False
    End If
   ProgressBar1.Value = ProgressBar1.Value + 5
   Label2.Caption = "Loading"
   Label3.Caption = ProgressBar1.Value & "%"
    If (ProgressBar1.Value = ProgressBar1.Max) Then
       Timer1.Enabled = False
       mdiMain.Show
       mdiMain.Label18.Caption = txtInput(0).Text
       Unload Me
    End If
End Sub
```

Coding For Add New User Form:

```
Private Sub cmdClose_Click()
Unload frmAddNewUser
End Sub
Private Sub cmdSave_Click()
   If txtUsername.Text = "" Then
      MsgBox "Enter UserName and Password ...", vbExclamation
      txtUsername.SetFocus
      Exit Sub
   End If
   If txtPassword.Text = "" Then
      MsgBox "Enter Password ...", vbExclamation
      txtPassword.SetFocus
      Exit Sub
   End If
   If txtConfirmPassword.Text = "" Then
   MsgBox "Enter confirpassword ...", vbExclamation
   txtConfirmPassword.SetFocus
   Exit Sub
   End If
   If txtPassword.Text <> txtConfirmPassword.Text Then
      MsgBox "Confirm password dosenot match with new password ...",
vbExclamation
      txtConfirmPassword.Text = ""
      txtPassword.Text = ""
      txtPassword.SetFocus
      Exit Sub
   End If
Set rs = cn.Execute("select * from Login where User_name='" +
txtUsername.Text + "' and Password='" + txtPassword.Text + "'")
If (Not rs.EOF) Then
   MsgBox "Sorry!! User already exists. Try another username", vbCritical
   txtPassword.Text = ""
   txtConfirmPassword.Text = ""
   txtUsername.Text = ""
   txtUsername.SetFocus
Else
   cn.Execute ("insert into Login values('" + txtUsername.Text + "','" +
txtPassword.Text + "')")
   MsgBox "User added sucessfully", vbInformation
   txtPassword.Text = ""
   txtConfirmPassword.Text = ""
   txtUsername.Text = ""
   txtUsername.SetFocus
  frmAddNewUser.Hide
  mdiMain.Show
End If
End Sub
Private Sub Form_Load()
OpenConnection
```

```
End Sub
```

Coding For Delete User Form

```
Private Sub cmdDeleteUser_Click()
cn.Execute ("delete from Login where User_name='" +
cmbUsername.Text + "'")
MsgBox "User deleted sucessfully!!", vbInformation
cmbUsername.Text = ""
Unload frmDeleteUser
mdiMain.Show
End Sub
Private Sub cmdExit_Click()
Unload frmDeleteUser
End Sub
Private Sub Form_Load()
OpenConnection
Set rs = cn.Execute("select * from Login")
While (Not rs.EOF)
   cmbUsername.AddItem rs(0)
   rs.MoveNext
Wend
End Sub
```

Coding For Table Form:

```
Dim rec As ADODB.Recordset
Private Function clearControls()
   txtInput(0).Text = ""
   txtInput(1).Text = ""
End Function
Private Function EDControls(mode As Boolean)
   txtInput(1).Enabled = mode
   cmdAddNew.Enabled = Not mode
   cmdCancel.Enabled = mode
   cmdModify.Enabled = Not mode
   cmdFind.Enabled = Not mode
   cmdDelete.Enabled = Not mode
End Function
Private Function EDNavigate(mode As Boolean)
   fraNavigator.Enabled = mode
End Function
Private Function showData()
   txtInput(0).Text = rec.Fields("Table_No").Value
   txtInput(1).Text = rec.Fields("Rent").Value
End Function
Private Sub cmdAddNew_Click()
```

```
    EDControls True
    EDNavigate False
    cmdSave.Enabled = True
    cmdUpdate.Enabled = False
    cmdCancel.Enabled = True
    clearControls
    Dim t As ADODB.Recordset
    Set t = New ADODB.Recordset
    t.Open "select * From Tables", cn, adOpenKeyset, adLockOptimistic
    If t.RecordCount > 0 Then
        t.MoveLast
        txtInput(0).Text = t.Fields(0).Value + 1
    Else
        txtInput(0).Text = 1
    End If
    Set t = Nothing
     txtInput(1).SetFocus
End Sub
Private Sub cmdCancel_Click()
    EDControls False
    EDNavigate True
    cmdSave.Enabled = False
    cmdUpdate.Enabled = False
    showData
End Sub
Private Sub cmdDelete_Click()
    Dim choice As Integer
    choice = MsgBox("Do you want to Delete the Record", vbYesNo +
vbQuestion, "confirmation")
    If choice = vbYes Then
        If rec.EOF = False And rec.BOF = False Then
            rec.Delete
            rec.MoveNext
            If rec.EOF Then rec.MoveLast
            showData
        End If
        End If
End Sub
Private Sub cmdExit_Click()
    Unload Me
End Sub
Private Sub cmdFind_Click()
    Dim s As String
```

```
   If rec.RecordCount > 0 Then
      rec.MoveFirst
      s = InputBox("Enter The table no")
      If s <> "" Then
         rec.Find "Table_No='" & s & "'"
         If Not rec.EOF Then
         Else
            MsgBox "Table Not Found"
            rec.MoveLast
         End If
      End If
      showData
   End If
End Sub
Private Sub cmdModify_Click()
   EDControls True
   EDNavigate False
   cmdSave.Enabled = False
   cmdUpdate.Enabled = True
   txtInput(1).SetFocus
End Sub
Private Sub cmdNavigate_click(Index As Integer)
   If Index = 0 Then
      rec.MovePrevious
      If (rec.BOF = True) Then
         MsgBox "You are already at the First Record"
      End If
      rec.MoveFirst
   ElseIf Index = 1 Then
rec.MovePrevious
      If rec.BOF Then
           MsgBox "You are already at the First Record"
           rec.MoveFirst
      End If
   ElseIf Index = 2 Then
      rec.MoveNext
      If rec.EOF Then
         MsgBox "you are already at the last record"
         rec.MoveLast
       End If
    Else
      rec.MoveNext
      If (rec.EOF = True) Then
```

```
            MsgBox "You are already at the last Record"
         End If
         rec.MoveLast
       End If
         showData
End Sub
Private Sub cmdSave_Click()
    If txtInput(0).Text = "" Or txtInput(1).Text = "" Then
        MsgBox "please provide the data"
        txtInput(1).SetFocus
        Exit Sub
    End If
    If Not IsNumeric(txtInput(1).Text) Then
        MsgBox "Price must be Numeric value"
        txtInput(1).Text = ""
        txtInput(1).SetFocus
        Exit Sub
    ElseIf txtInput(1).Text < 0 Then
         MsgBox "Price must be Positive value"
        txtInput(1).Text = ""
        txtInput(1).SetFocus
        Exit Sub
    End If
    EDControls False
    EDNavigate True
    cmdSave.Enabled = False
    rec.AddNew
    rec.Fields("Table_No").Value = txtInput(0).Text
    rec.Fields("Rent").Value = txtInput(1).Text
    rec.Update
End Sub
Private Sub cmdUpdate_Click()
     If txtInput(0).Text = "" Or txtInput(1).Text = "" Then
        MsgBox "Please provide the data"
        Exit Sub
    End If
        If Not IsNumeric((txtInput(1).Text) Or (txtInput(1).Text)) Then
        MsgBox "Price must be Numeric value"
        Exit Sub
    End If
    EDControls False
    EDNavigate True
    cmdUpdate.Enabled = False
    rec.Fields("Table_No").Value = txtInput(0).Text
    rec.Fields("Rent").Value = txtInput(1).Text
    rec.Update
End Sub
Private Sub Form_Load()
    Set rec = New ADODB.Recordset
    EDControls False
    cmdSave.Enabled = False
    cmdUpdate.Enabled = False
     txtInput(0).Enabled = False
End Sub
Private Sub Form_Activate()
    CenterWindow Me
    rec.Open "Select * from Tables", cn, adOpenKeyset, adLockOptimistic
    If rec.RecordCount > 0 Then
        rec.MoveFirst
        showData
    End If
End Sub
Private Sub Form_Unload(Cancel As Integer)
    rec.Close
    Set rec = Nothing
End Sub
```

Coding For Category Form:

```
Dim rec As ADODB.Recordset
Private Function clearControls()
    txtInput(0).Text = ""
    txtInput(1).Text = ""
End Function
Private Function EDControls(mode As Boolean)
    txtInput(1).Enabled = mode
    cmdAddNew.Enabled = Not mode
    cmdCancel.Enabled = mode
    cmdModify.Enabled = Not mode
    cmdFind.Enabled = Not mode
    cmdDelete.Enabled = Not mode
End Function
Private Function EDNavigate(mode As Boolean)
    fraNavigator.Enabled = mode
End Function
Private Sub cmdCancel_Click()
    EDControls False
    EDNavigate True
    cmdSave.Enabled = False
    cmdUpdate.Enabled = False
    showData
```

```
End Sub
Private Sub cmdDelete_Click()
   Dim t As ADODB.Recordset
   Set t = New ADODB.Recordset
   t.Open "Select * from Item_Table  where Cat_ID=" & txtInput(0).Text, cn,
adOpenKeyset, adLockOptimistic
   If t.RecordCount > 0 Then
         MsgBox "You Can not Delete the Record because Related Records Exists in
Item Table"
         Exit Sub
   End If
      Dim choice As Integer
      choice = MsgBox("Do you want to Delete the Record", vbYesNo + vbQuestion,
"confirmation")
      If choice = vbYes Then
         If rec.EOF = False And rec.BOF = False Then
            rec.Delete
            rec.MoveNext
            If rec.EOF Then rec.MoveLast
            showData
         End If
      End If
      Set t = Nothing
End Sub
Private Sub cmdExit_Click()
    Unload Me
End Sub
Private Sub cmdFind_Click()
Dim s As String
   If rec.RecordCount > 0 Then
      rec.MoveFirst
      s = InputBox("Enter The Category name")
      If s <> "" Then
         rec.Find "Cat_Name='" & s & "'"
         If Not rec.EOF Then
         Else
            MsgBox "Record Not Found"
            rec.MoveLast
         End If
      End If
      showData
   End If
End Sub
Private Sub cmdModify_Click()
    EDControls True
   EDNavigate False
   cmdSave.Enabled = False
   cmdUpdate.Enabled = True
   txtInput(1).SetFocus
End Sub
```

```
Private Sub cmdNavigate_click(Index As Integer)
    If Index = 0 Then
        rec.MovePrevious
        If (rec.BOF = True) Then
            MsgBox "You are already at the First Record"
        End If
        rec.MoveFirst
    ElseIf Index = 1 Then
        rec.MovePrevious
        If rec.BOF Then
              MsgBox "You are already at the First Record"
              rec.MoveFirst
        End If
    ElseIf Index = 2 Then
        rec.MoveNext
        If rec.EOF Then
            MsgBox "you are already at the last record"
            rec.MoveLast
        End If
    Else
        rec.MoveNext
        If (rec.EOF = True) Then
            MsgBox "You are already at the last Record"
        End If
        rec.MoveLast
    End If
        showData
End Sub
Private Sub cmdSave_Click()
     If txtInput(0).Text = "" Or txtInput(1).Text = "" Then
        MsgBox "please provide the data"
        Exit Sub
    End If
    If Not IsNumeric(txtInput(0).Text) Then
        MsgBox "Price must be Numeric value"
        Exit Sub
    End If
    EDControls False
    EDNavigate True
    cmdSave.Enabled = False
    rec.AddNew
    rec.Fields("cat_id").Value = txtInput(0).Text
    rec.Fields("cat_name").Value = txtInput(1).Text
    rec.Update
End Sub
Private Sub cmdUpdate_Click()
     If txtInput(0).Text = "" Or txtInput(1).Text = "" Then
        MsgBox "Please provide the data"
        Exit Sub
    End If
    If Not IsNumeric(txtInput(0).Text) Then
        MsgBox "Price must be Numeric value"
        Exit Sub
    End If
    EDControls False
    EDNavigate True
    cmdUpdate.Enabled = False
    rec.Fields("cat_id").Value = txtInput(0).Text
    rec.Fields("cat_name").Value = txtInput(1).Text
    rec.Update
End Sub
Private Sub Form_Load()
    Set rec = New ADODB.Recordset
    rec.Open "Select * from Category", cn, adOpenKeyset, adLockOptimistic
    EDControls False
    cmdSave.Enabled = False
    cmdUpdate.Enabled = False
    txtInput(0).Enabled = False
End Sub
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    Query
End Sub
Private Sub Form_Unload(Cancel As Integer)
    rec.Close
    Set rec = Nothing
End Sub
Private Function showData()
    txtInput(0).Text = rec.Fields("cat_id").Value
    txtInput(1).Text = rec.Fields("cat_name").Value
End Function
Private Sub cmdAddNew_Click()
    EDControls True
    EDNavigate False
    cmdSave.Enabled = True
    cmdUpdate.Enabled = False
    cmdCancel.Enabled = True
    clearControls
    Dim t As ADODB.Recordset
    Set t = New ADODB.Recordset
    t.Open "select * From Category", cn, adOpenKeyset, adLockOptimistic
    If t.RecordCount > 0 Then
        t.MoveLast
        txtInput(0).Text = t.Fields(0).Value + 1
    Else
        txtInput(0).Text = 1
    End If
    Set t = Nothing
   txtInput(1).SetFocus
End Sub
Private Function loadCategory()
```

```
    Dim t As ADODB.Recordset
    Set t = New ADODB.Recordset
    t.Open "select * from Category", cn, adOpenKeyset, adLockOptimistic
    If t.RecordCount > 0 Then
        While Not t.EOF
            txtInput(0).Text = t.Fields("cat_id").Value
            txtInput(1).Text = t.Fields("cat_name").Value
            t.MoveNext
        Wend
    End If
     Set t = Nothing
End Function
Private Sub Form_Activate()
    CenterWindow Me
    loadCategory
    If rec.RecordCount > 0 Then
        rec.MoveFirst
        showData
    End If
End Sub
Private Sub txtInput_KeyPress(Index As Integer, KeyAscii As Integer)
     If Index = 1 Then
        If Not ((KeyAscii >= 65 And KeyAscii <= 90) Or (KeyAscii >= 97 And KeyAscii
<= 122) Or KeyAscii = 8 Or KeyAscii = 32) Then
            KeyAscii = 0
            MsgBox "Please Enter Proper Category Name"
        End If
    End If
End Sub
```

Coding for Item Table Form

```
Dim rec As ADODB.Recordset
Private Function clearControls()
    txtInput(0).Text = ""
    txtInput(1).Text = ""
    If cboCatID.ListCount > 0 Then
        cboCatID.ListIndex = 0
    End If
End Function
Private Function EDControls(mode As Boolean)
    txtInput(0).Enabled = mode
    txtInput(1).Enabled = mode
    cboCatID.Enabled = mode
    cmdAddNew.Enabled = Not mode
    cmdCancel.Enabled = mode
    cmdFind.Enabled = Not mode
    cmdDelete.Enabled = Not mode
End Function
Private Function EDNavigate(mode As Boolean)
    fraNavigator.Enabled = mode
End Function
```

```
Private Function showData()
    Dim i As Integer
    txtInput(0).Text = rec.Fields("Item_name").Value
    txtInput(1).Text = rec.Fields("rate").Value
    For i = 0 To cboCatID.ListCount - 1
        If rec.Fields("Cat_id").Value = cboCatID.ItemData(i) Then
            cboCatID.ListIndex = i
            Exit For
        End If
    Next
End Function
Private Sub cmdAddNew_Click()
    EDControls True
    EDNavigate False
    cmdSave.Enabled = True
    cmdCancel.Enabled = True
    clearControls
    txtInput(0).SetFocus
End Sub
Private Sub cmdCancel_Click()
    EDControls False
    EDNavigate True
    cmdSave.Enabled = False
    showData
End Sub
Private Sub cmdDelete_Click()
    Dim choice As Integer
        choice = MsgBox("Do you want to Delete the Record", vbYesNo + vbQuestion,
"confirmation")
        If choice = vbYes Then
            If rec.EOF = False And rec.BOF = False Then
                rec.Delete
                rec.MoveNext
                If rec.EOF Then rec.MoveLast
                showData
            End If
        End If
End Sub
Private Sub cmdModify_Click()
    EDControls True
    EDNavigate False
    cmdSave.Enabled = False
    cmdUpdate.Enabled = True
    txtInput(0).SetFocus
End Sub
Private Sub cmdNavigate_click(Index As Integer)
      If Index = 0 Then
            rec.MovePrevious
            If (rec.BOF = True) Then
                MsgBox "You are already at the First Record"
            End If
            rec.MoveFirst
        ElseIf Index = 1 Then
            rec.MovePrevious
            If rec.BOF Then
                MsgBox "You are already at the First Record"
                rec.MoveFirst
            End If
        ElseIf Index = 2 Then
            rec.MoveNext
            If rec.EOF Then
            MsgBox "you are already at the last record"
            rec.MoveLast
            End If
        Else
            rec.MoveNext
        If (rec.EOF = True) Then
            MsgBox "You are already at the last Record"
        End If
            rec.MoveLast
        End If
            showData
End Sub
Private Sub cmdSave_Click()
    If txtInput(0).Text = "" Or txtInput(1).Text = "" Then
        MsgBox "please provide the data"
        Exit Sub
    End If
    If Not IsNumeric(txtInput(1).Text) Then
        MsgBox "Price must be Numeric value"
        Exit Sub
    End If
    EDControls False
    EDNavigate True
    cmdSave.Enabled = False
    rec.AddNew
    rec.Fields("Item_name").Value = txtInput(0).Text
    rec.Fields("Rate").Value = txtInput(1).Text
    rec.Fields("Cat_id").Value = cboCatID.ItemData(cboCatID.ListIndex)
    rec.Update
End Sub
Private Sub Form_Activate()
    CenterWindow Me
    loadCategory
    If rec.RecordCount > 0 Then
        rec.MoveFirst
        showData
    End If
End Sub
Private Sub Form_Load()
```

```
    Set rec = New ADODB.Recordset
    rec.Open "Select * from Item_Table", cn, adOpenKeyset, adLockOptimistic
    EDControls False
    cmdSave.Enabled = False
End Sub
Private Sub Form_Unload(Cancel As Integer)
    rec.Close
    Set rec = Nothing
End Sub
Private Function loadCategory()
    Dim t As ADODB.Recordset
    Set t = New ADODB.Recordset
    t.Open "select * from Category", cn, adOpenKeyset, adLockOptimistic
    cboCatID.Clear
    If t.RecordCount > 0 Then
        While Not t.EOF
            cboCatID.AddItem t.Fields("cat_name").Value
            cboCatID.ItemData(cboCatID.NewIndex) = t.Fields("cat_id").Value
            t.MoveNext
        Wend
    End If
     Set t = Nothing
End Function
Private Sub txtInput_KeyPress(Index As Integer, KeyAscii As Integer)
    If Index = 0 Then
        If Not ((KeyAscii >= 65 And KeyAscii <= 90) Or (KeyAscii >= 97 And KeyAscii <= 122) Or KeyAscii = 8 Or KeyAscii = 32) Then
            KeyAscii = 0
            MsgBox "Please Enter Proper Item Name"
        End If
    ElseIf Index = 1 Then
         If Not ((KeyAscii >= 48 And KeyAscii <= 57) Or (KeyAscii = 8) Or (KeyAscii = 46)) Then
            KeyAscii = 0
            MsgBox "Please Enter Numeric Value "
        End If
    End If
End Sub
```

Coding For Order Form

```
Dim FirstTime As Boolean
Dim rec As ADODB.Recordset
Private Sub cboCategory_click()
    loadItems
    txtPrice.Text = ""
End Sub
Private Sub cbocustomer_KeyPress(KeyAscii As Integer)
Call capsonly(KeyAscii)
End Sub
Private Sub cboTables_Click()
txtRent.Text = ""
```

```
Dim r As New Recordset
r.Open "select * from Tables where Table_No=" & Val(Trim(cboTables.Text)) & " ",
cn, adOpenDynamic, adLockOptimistic
If r.EOF = True Then
MsgBox "There is no table "
Exit Sub
Else
txtRent.Text = r.Fields("Rent").Value
End If
End Sub
Private Sub cmdAddCustomer_Click()
Dim cname As String, r As New Recordset, cid As Integer
cname = UCase(InputBox("Enter Customer name"))
cbocustomer.AddItem cname
r.Open "select * from Customer", cn, adOpenDynamic, adLockOptimistic
If r.EOF = True Then
cid = 1
cnn.Execute "insert into Customer(C_No,Cname) values(" & cid & " ,'" & cname &
"')"
Exit Sub
Else
r.MoveLast
cid = r.Fields("C_No").Value + 1
If cname = "" Then Exit Sub
cn.Execute "insert into Customer(C_No,C_name) values(" & cid & " ,'" & cname &
"')"
End If
r.Close
End Sub
End Sub
Private Sub cmdSave_Click()
   If Val(txtAmount.Text) <= Val(txtDiscount.Text) Then
      MsgBox "Please Enter Discount less than Amount"
      txtDiscount.SetFocus
      Exit Sub
    End If
  If cbocustomer.Text = "" Then
    MsgBox "Please Enter Customer Name"
    Exit Sub
  End If
  If txtDiscount.Text = "" Then
    MsgBox "Please Enter Discount"
    txtDiscount.SetFocus
    Exit Sub
  End If
  Dim rest As ADODB.Recordset
  Set rest = New ADODB.Recordset
  rest.Open "select * from morder", cn, adOpenKeyset, adLockOptimistic
  rest.AddNew
  rest.Fields("O_No").Value = txtOrderNo.Text
```

```
    rest.Fields("O_Date").Value = txtOrderDate.Text
    rest.Fields("C_Name").Value = cbocustomer.Text
    rest.Fields("Amount").Value = txtAmount.Text
    rest.Fields("Discount").Value = txtDiscount.Text
    'qty.Fields("O_Table").Value = cboTables.Text
    rest.Fields("BAmount").Value = Val(txtAmount.Text) - Val(txtDiscount.Text)
   rest.Update
   Dim qty As ADODB.Recordset
    Set qty = New ADODB.Recordset
    qty.Open "select * from Order_Items", cn, adOpenKeyset, adLockOptimistic
    qty.AddNew
    qty.Fields("O_No").Value = txtOrderNo.Text
    qty.Fields("O_Date").Value = txtOrderDate.Text
    qty.Fields("O_Table").Value = cboTables.Text
    qty.Fields("Item_name").Value = cboItems.Text
    qty.Fields("C_Name").Value = cbocustomer.Text
    qty.Fields("Rate").Value = txtPrice.Text
    qty.Fields("Qty").Value = txtqty.Text
    qty.Fields("ItemAmount").Value = Val(txtqty.Text) * Val(txtPrice.Text)
     qty.Fields("Amount").Value = txtAmount.Text
    qty.Fields("Discount").Value = txtDiscount.Text
    qty.Fields("BAmount").Value = Val(txtAmount.Text) - Val(txtDiscount.Text)
   qty.Update
   MsgBox "Save successfuly"
End Sub
    If txtDiscount.Text = "" Then
        MsgBox "Please Enter Discount"
        txtDiscount.SetFocus
        Exit Sub
        End If
    If Val(txtAmount.Text) <= Val(txtDiscount.Text) Then
        MsgBox "Please Enter Discount less than Amount"
        txtDiscount.SetFocus
        Exit Sub
        End If
End Sub
Public Sub capsonly(KeyAscii As Integer)
    If KeyAscii >= 97 And KeyAscii <= 122 Then
        KeyAscii = KeyAscii - 32
    Else
        KeyAscii = KeyAscii
    End If
End Sub
Private Sub cmsave_MouseMove(Button As Integer, Shift As Integer, x As Single, y
As Single)
 txtNetPayment.Text = Val(txtAmount.Text) - Val(txtDiscount.Text)
End Sub
Private Function loadTableNo()
    Dim t As ADODB.Recordset
    Dim h As Integer
```

```
  h = Hour(Time)
  Set t = New ADODB.Recordset
  t.Open "select * from Tables", cn, adOpenKeyset, adLockOptimistic
  cboTables.Clear
  If t.RecordCount > 0 Then
      While Not t.EOFm
        cboTables.AddItem t.Fields(0).Value
        cboTables.ItemData(cboTables.NewIndex) = t.Fields(1).Value
        t.MoveNext
      Wend
  End If
  Set t = Nothing
  cboTables.ListIndex = -1
  End Function
Private Function loadCategory()
  Dim t As ADODB.Recordset
  Set t = New ADODB.Recordset
  t.Open "select * from Category", cn, adOpenKeyset, adLockOptimistic
  cbocategory.Clear
  If t.RecordCount > 0 Then
    While Not t.EOF
      cbocategory.AddItem t.Fields("cat_name").Value
      cbocategory.ItemData(cbocategory.NewIndex) = t.Fields("cat_id").Value
      t.MoveNext
    Wend
  End If
   Set t = Nothing
   cbocategory.ListIndex = 0
  End Function
Private Function loadItems()
  Dim t As ADODB.Recordset
  Set t = New ADODB.Recordset
  If cbocategory.ListIndex >= 0 Then
    t.Open "select * from Item_Table where Cat_Id=" &
cbocategory.ItemData(cbocategory.ListIndex), cn, adOpenKeyset, adLockOptimistic
    cboItems.Clear
    If t.RecordCount > 0 Then
      While Not t.EOF
        cboItems.AddItem t.Fields(0).Value
        cboItems.ItemData(cboItems.NewIndex) = t.Fields(1).Value
        t.MoveNext
      Wend
    End If
  End If
   Set t = Nothing
   cboItems.ListIndex = -1
End Function
Private Sub cboItems_Click()
  If (cboItems.ListIndex >= 0) Then
    txtPrice.Text = Val(cboItems.ItemData(cboItems.ListIndex))
    Else
       txtPrice.Text = ""
    End If
 End Sub
Private Function loadCName()
    Dim t As ADODB.Recordset
    Set t = New ADODB.Recordset
    t.Open "select * from customer", cn, adOpenKeyset, adLockOptimistic
    cbocustomer.Clear
    If t.RecordCount > 0 Then
       While Not t.EOF
          cbocustomer.AddItem t.Fields("C_Name").Value
          cbocustomer.ItemData(cbocustomer.NewIndex) = t.Fields("C_No").Value
          t.MoveNext
       Wend
    End If
    Set t = Nothing
    cbocustomer.ListIndex = 0
 End Function
 Private Sub txtDiscount_KeyPress(KeyAscii As Integer)
     If Not ((KeyAscii >= 48 And KeyAscii <= 57) Or KeyAscii = 8) Then
       KeyAscii = 0
    End If
 End Sub
 Private Sub txtDiscount_LostFocus()
    txtNetPayment = Val(txtAmount.Text) - Val(txtDiscount.Text)
 End Sub
 Private Function clearControls()
     cbocustomer.ListIndex = -1
    optConsumed.Value = True
    LstTables.Clear
    lstStartTime.Clear
    lstItems.Clear
    lstquantity.Clear
    cboTables.ListIndex = -1
    cboTimeFrom.ListIndex = -1
    cbocategory.ListIndex = -1
    cboItems.ListIndex = -1
    txtRent.Text = ""
    txtHours.Text = ""
    txtPrice.Text = ""
    txtAmount.Text = ""
    txtDiscount.Text = ""
    txtNetPayment.Text = ""
 End Function
 Private Sub txtDiscount_MouseUp(Button As Integer, Shift As Integer, x As Single, y
 As Single)
 txtNetPayment = Val(txtAmount.Text) - Val(txtDiscount.Text)
 End Sub
 Private Sub txtqty_KeyPress(KeyAscii As Integer)
```

```
    If Not ((KeyAscii >= 48 And KeyAscii <= 57) Or KeyAscii = 8) Then
      KeyAscii = 0
      MsgBox "Please Enter Numerc Value"
      Exit Sub
   End If
End Sub
Private Sub generateOID()
   Dim r As ADODB.Recordset
   Set r = New ADODB.Recordset
r.Open "select * from morder", cn, adOpenKeyset, adLockOptimistic
If r.EOF = True Then
txtOrderNo.Text = 1
Exit Sub
Else
r.MoveLast
txtOrderNo.Text = r.Fields("O_No").Value + 1
End If
End Sub
Private Sub txtqty_LostFocus()
txtAmount.Text = ""
If txtqty.Text = "" Then
txtAmount.Text = Val(txtPrice.Text) * 1
Else
txtAmount.Text = Val(txtPrice.Text) * Val(txtqty.Text) + Val(txtRent.Text)
End If
End Sub
```

Coding For MDI Form:

```
Private Sub Command1_Click()
If Picture1.Visible = True Then
Picture1.Visible = False
mnuSideBar.Checked = False
End If
End Sub
Private Sub Image2_MouseMove(Button As Integer, Shift As Integer, x As Single, y
As Single)
        lblShortcut(0).ForeColor = &H80000006
        lblShortcut(1).ForeColor = &H80000006
        lblShortcut(2).ForeColor = &H80000006
        lblShortcut(3).ForeColor = &H80000006
        lblShortcut(4).ForeColor = &H80000006
        lblShortcut(5).ForeColor = &H80000006
        lblShortcut(6).ForeColor = &H80000006
        lblShortcut(0).FontUnderline = False
        lblShortcut(1).FontUnderline = False
        lblShortcut(2).FontUnderline = False
        lblShortcut(3).FontUnderline = False
        lblShortcut(4).FontUnderline = False
        lblShortcut(5).FontUnderline = False
        lblShortcut(6).FontUnderline = False
End Sub
```

```
Private Sub imgUserAccount_MouseMove(Button As Integer, Shift As Integer, x As
Single, y As Single)
        lblShortcut(0).ForeColor = &H80000006
        lblShortcut(1).ForeColor = &H80000006
        lblShortcut(2).ForeColor = &H80000006
        lblShortcut(3).ForeColor = &H80000006
        lblShortcut(4).ForeColor = &H80000006
        lblShortcut(5).ForeColor = &H80000006
        lblShortcut(6).ForeColor = &H80000006
        lblShortcut(0).FontUnderline = False
        lblShortcut(1).FontUnderline = False
        lblShortcut(2).FontUnderline = False
        lblShortcut(3).FontUnderline = False
        lblShortcut(4).FontUnderline = False
        lblShortcut(5).FontUnderline = False
        lblShortcut(6).FontUnderline = False
End Sub
Private Sub lblShortcut_Click(Index As Integer)
    Select Case (Index)
        Case 0:
        mdiMain.Hide
            frmLogin.Show
        Case 1:
              frmAddNewUser.Show , Me
              frmDeleteUser.Hide
              frmChangePassword.Hide
        Case 2:
               Shell "calc.exe", vbNormalFocus
        Case 3:
               cdlg.ShowHelp
        Case 4:
            frmChangePassword.Show , Me
            frmAddNewUser.Hide
            frmDeleteUser.Hide
        Case 5:
           iLogOutReply = MsgBox(UserName & ", Are You Sure You Wish To Log Out
Of Your Account?", vbYesNo + vbQuestion, "Log Out?")
            If iLogOutReply = vbYes Then
            End
            End If
        Case 6:
              frmDeleteUser.Show , Me
              frmAddNewUser.Hide
              frmChangePassword.Hide
            End Select
End Sub

Private Sub lblShortcut_MouseMove(Index As Integer, Button As Integer, Shift As
Integer, x As Single, y As Single)
    Select Case (Index)
```

```
Case 0:
    lblShortcut(0).ForeColor = &H800000
    lblShortcut(1).ForeColor = &H80000006
    lblShortcut(2).ForeColor = &H80000006
    lblShortcut(3).ForeColor = &H80000006
    lblShortcut(4).ForeColor = &H80000006
    lblShortcut(5).ForeColor = &H80000006
    lblShortcut(6).ForeColor = &H80000006
    lblShortcut(0).FontUnderline = True
    lblShortcut(1).FontUnderline = False
    lblShortcut(2).FontUnderline = False
    lblShortcut(3).FontUnderline = False
    lblShortcut(4).FontUnderline = False
    lblShortcut(5).FontUnderline = False
    lblShortcut(6).FontUnderline = False
Case 1:
    lblShortcut(0).ForeColor = &H80000006
    lblShortcut(1).ForeColor = &H800000
    lblShortcut(2).ForeColor = &H80000006
    lblShortcut(3).ForeColor = &H80000006
    lblShortcut(4).ForeColor = &H80000006
    lblShortcut(5).ForeColor = &H80000006
    lblShortcut(6).ForeColor = &H80000006
    lblShortcut(0).FontUnderline = False
    lblShortcut(1).FontUnderline = True
    lblShortcut(2).FontUnderline = False
    lblShortcut(3).FontUnderline = False
    lblShortcut(4).FontUnderline = False
    lblShortcut(5).FontUnderline = False
    lblShortcut(6).FontUnderline = False
Case 2:
    lblShortcut(0).ForeColor = &H80000006
    lblShortcut(1).ForeColor = &H80000006
    lblShortcut(2).ForeColor = &H800000
    lblShortcut(3).ForeColor = &H80000006
    lblShortcut(4).ForeColor = &H80000006
    lblShortcut(5).ForeColor = &H80000006
    lblShortcut(6).ForeColor = &H80000006
    lblShortcut(0).FontUnderline = False
    lblShortcut(1).FontUnderline = False
    lblShortcut(2).FontUnderline = True
    lblShortcut(3).FontUnderline = False
    lblShortcut(4).FontUnderline = False
    lblShortcut(5).FontUnderline = False
    lblShortcut(6).FontUnderline = False
Case 3:
    lblShortcut(0).ForeColor = &H80000006
    lblShortcut(1).ForeColor = &H80000006
    lblShortcut(2).ForeColor = &H80000006
    lblShortcut(3).ForeColor = &H800000
    lblShortcut(4).ForeColor = &H80000006
    lblShortcut(5).ForeColor = &H80000006
    lblShortcut(6).ForeColor = &H80000006
    lblShortcut(0).FontUnderline = False
    lblShortcut(1).FontUnderline = False
    lblShortcut(2).FontUnderline = False
    lblShortcut(3).FontUnderline = True
    lblShortcut(4).FontUnderline = False
    lblShortcut(5).FontUnderline = False
    lblShortcut(6).FontUnderline = False
Case 4:
    lblShortcut(0).ForeColor = &H80000006
    lblShortcut(1).ForeColor = &H80000006
    lblShortcut(2).ForeColor = &H80000006
    lblShortcut(3).ForeColor = &H80000006
    lblShortcut(4).ForeColor = &H800000
    lblShortcut(5).ForeColor = &H80000006
    lblShortcut(6).ForeColor = &H80000006
    lblShortcut(0).FontUnderline = False
    lblShortcut(1).FontUnderline = False
    lblShortcut(2).FontUnderline = False
    lblShortcut(3).FontUnderline = False
    lblShortcut(4).FontUnderline = True
    lblShortcut(5).FontUnderline = False
    lblShortcut(6).FontUnderline = False
Case 5:
    lblShortcut(0).ForeColor = &H80000006
    lblShortcut(1).ForeColor = &H80000006
    lblShortcut(2).ForeColor = &H80000006
    lblShortcut(3).ForeColor = &H80000006
    lblShortcut(4).ForeColor = &H80000006
    lblShortcut(5).ForeColor = &H800000
    lblShortcut(6).ForeColor = &H80000006
    lblShortcut(0).FontUnderline = False
    lblShortcut(1).FontUnderline = False
    lblShortcut(2).FontUnderline = False
    lblShortcut(3).FontUnderline = False
    lblShortcut(4).FontUnderline = False
    lblShortcut(5).FontUnderline = True
    lblShortcut(6).FontUnderline = False
Case 6:
    lblShortcut(0).ForeColor = &H80000006
    lblShortcut(1).ForeColor = &H80000006
    lblShortcut(2).ForeColor = &H80000006
    lblShortcut(3).ForeColor = &H80000006
    lblShortcut(4).ForeColor = &H80000006
    lblShortcut(5).ForeColor = &H80000006
    lblShortcut(6).ForeColor = &H800000
    lblShortcut(0).FontUnderline = False
    lblShortcut(1).FontUnderline = False
```

```
            lblShortcut(2).FontUnderline = False
            lblShortcut(3).FontUnderline = False
            lblShortcut(4).FontUnderline = False
            lblShortcut(5).FontUnderline = False
            lblShortcut(6).FontUnderline = True
End Select
End Sub
Private Sub MDIForm_Load()
Label2.Caption = Format(Date, "long date")
frmLogin.Show vbModel
End Sub
Private Sub MDIForm_QueryUnload(Cancel As Integer, UnloadMode As Integer)
Dim re As Variant
re = MsgBox("Do You Want Exit", vbYesNo)
If re = vbYes Then
End
Else
Cancel = 1
End If
End Sub
Private Sub mnuAbout_Click()
frmAbout.Show , Me
End Sub
Private Sub mnuAllCustomerOrder_Click()
rptAllCustomerOrder.Show
End Sub
Private Sub mnuCollection_Click()
    If DataEnvironment1.rscmdTCollection_Grouping.State Then
        DataEnvironment1.rscmdTCollection_Grouping.Close
    End If
    DataEnvironment1.cmdTCollection_Grouping Format(Date, "DD-MM-YY")
    DRTCollection.Show
End Sub
Private Sub mnuConsumption_Click()
    frmCBill.Show
End Sub
Private Sub mnuExit_Click()
If MsgBox("Are You Sure Exit System ?", vbYesNo + vbInformation, "Warning") =
vbYes Then
    End
    Unload frmSYSTRAYICON
End If
End Sub
Private Sub mnuHomeDelivery_Click()
    frmHDBill.Show
End Sub
Private Sub mnuMasterAdvanceBooking_Click()
    frmAdvancedBooking.Show
End Sub
Private Sub mnuMasterCategory_Click()
```

```
   frmCategory.Show
End Sub
Private Sub mnuMasterItemTable_Click()
   frmItemTable.Show
End Sub
Private Sub mnuMasterOrder_Click()
   frmOrder.Show
End Sub
Private Sub mnuMasterTables_Click()
   frmTables.Show
End Sub
Private Sub mnuMonthlyCollection_Click()
   frmMCollection.Show
End Sub
Private Sub mnuReportMenu_Click()
   DRPMenu.Show
End Sub
Private Sub mnuSideBar_Click()
If Picture1.Visible = True Then
Picture1.Visible = False
mnuSideBar.Checked = False
ElseIf Picture1.Visible = False Then
Picture1.Visible = True
mnuSideBar.Checked = True
End If
End Sub
Private Sub mnuViewHelp_Click()
cdlg.ShowHelp
End Sub
Private Sub Timer1_Timer()
Label1.Caption = DateTime.Time
End Sub
```

Code For Module(Database Connection Module):

```
Global rs As ADODB.Recordset
Public cnn As New Connection
Global cn As ADODB.Connection
Public cno As Long
Public loadOno As Long
Public LoadTime As Boolean
Public Function CenterWindow(frm As Form)
  ' frm.Left = (mdiMain.ScaleWidth - frm.Width) / 2
   'frm.Top = (mdiMain.ScaleHeight - frm.Height) / 2
End Function
Public Function OpenConnection() As Boolean
On Error GoTo HandleError
   Set cn = New ADODB.Connection
   cn.Provider = "SQLOLEDB.1;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=RESTAURANT DATABASE;Data
Source=DCRUZZ\SQLEXPRESS"
   cn.Open
   OpenConnection = True
   Exit Function
HandleError:
   OpenConnection = False
End Function
Public Function CloseConnection()
   cn.Close
   Set cn = Nothing
End Function
Public Function Query()
   Select Case UnloadMode
         Case vbFormCode
         Case vbFormControlMenu
         Case vbFormMDIForm
            MsgBox "First close the form"
            Cancel = True
         Case vbAppTaskManager
            MsgBox "First close the application then shut Down"
            Cancel = True
         Case vbAppWindows
            MsgBox "First close the Application"
            Cancel = True
   End Select
 End Function
```