**INTRODUCTION OF PROJECT 3**

**PLATFORM AND LANGUAGE 4**

**HARDWARE SPECIFICATION 5**

**OVERVIEW OF PROJECT 6**

**ANALYSIS OF PROJECT 7**

**SYSTEM ANALYSIS 8**

**SYSTEM DEVELOPMENT LIFE CYCLE 10**

**PRESENT SETUP 11**

**PROPOSED SYSTEM 12**

**FEASIBILITY STUDY 13**

**BEHAVIOURAL FEASIBILITY 14**

**SYSTEM DESIGN 15**

**CONTEXT DIAGRAM 16**

**E-R DIAGRAM 17**

**DATA FLOW DIAGRAM 19**

**SYSTEM TESTING AND IMPLEMENTATION 21**

**TABLE STRUCTURE 27**

**LOGIN SYSTEM 33**

**MENU SYSTEM 35**

**PROGRAM INPUT 36**

**PROGRAM OUTPUT 40**

**PROGRAM CODE 46**

**PIE CHART 104**

**BIBLIOGRAPHY 105**

**Resturent Management System**

**Introduction**

The objective of “**Resturent Management System”** is to assist and speed up the process of Resturent Management. Resturent Managementnot only maintains the database for future reference, but also plays an important role in making a paperless office.

Many steps are involved in between accepting Employee and Customer enquiry and supplying the requested information to the Resturent Management and processes in average with manual processing time of fifteen minutes to half an hour.

**Employee Entry** reduces most of the typing works involved specially in preparing entry of Employee records and Salary records in the resturent.

It improves the searching facility and easily updates the record. It also gives a detail report on employee information, customer information and order details in the Resturent which can improve the processing plan.

# Platform & Language Used

**Platform** :- **WINDOWS 98**, **WINDOWS XP**.

**Language** :- **Visual Basic(***Front-End***)**

**Oracle 10g(***Back-End***)**

**MINIMUM HARDWARE SPECIFICATION**

**PROCESSOR** : **PENTIUM-4**

**RAM** : **256MB**

**HARD DISK** : **1.2GB**

**MONITOR** : **COLOURED MONITOR**(SAMSUNG/LG)

**MOUSE** : **SCROLLABLE MOUSE**

**KEYBOARD** : **COMPATIBLE KEYBOARD**(102/103

KEYS)

**OPERATING SYSTEM** : **WINDOWS-Xp and above**

**Overview Of The Project**

The Manager scrutinizes the requirement and prepares work quotations based on enquiry, which contains detail of employee data like code no, name, address, dob, post, qualification and etc.

This software provides the feature to maintain the records of employee of the Resturent like Total number of Employee, Total salary detail, Employee’s attendence and Enquiry etc.

The Manager after the receipt of the information of employee and Enquiry makes the required report like attendence sheet of employee and salary list of employee

It maintains the final information of customer like customer details, phone, addresss, email etc.

It also maintains the order details like customer name, table number, waiter id, item datails etc.

**Analysis of the Resturent Management System:** -

* System Analysis – An introduction
* Present System
* Problem With Present System
* Purpose Of The Proposed System
* Benefits Of The Proposed System
* Feasibility Study –
  + Technical Feasibility
  + Behavioral Feasibility
  + Economical Feasibility
* Requirements Of The Proposed System

**SYSTEM ANALYSIS – AN INTRODUCTON**

System Analysis and Design refers to the process of examining a business situation with the intent of improving it through better methods and procedures.

System Analysis is the process of gathering and interpreting the facts, diagnosing the problems and using the information tool to recommends the improvements of the system. System designing is the process of planning a new business system or to replace or compliment the existing system.

The system Development Life Cycle (SDLC) method is classically thought of as the set of activities that analyzers, Designers or Users carry out to develop and implements the information system.

The nine Phases that make up the System Development

Life Cycle is: -

* REQUIREMENT SPECIFICATION
* REQUIREMENT DETERMINATION
* FEASIBILITY STUDY
* SYSTEM ANALYSIS
* HARDWARE STUDY
* SYSTEM DESIGN
* TESTING AND IMPLEMENTATION
* EVALUATION
* MAINTAINCE AND MODIFICATION

## SYSTEM DEVELOPMENT LIFE CYCLE DIAGRAM

Logical

System

Design

User

Feasibility Study

System

Revised

Requirement

Configuration

Test

Plan

Functional Specification

Decision to Design

**Restu. Management**

Revised prioritized Requirement Specification

Improve System

Physical

Requirement

### PRESENT SETUP

Before any big and complex project such as the proposed System is taken up, it is Vital to study the existing system in detail.

***Present Setup:***

The objective of Resturent Management System is to assist and speed up the process of Resturent information Processing in a Resturent of any size.

***PROBLEM WITH PRESENT SETUP:***

* The present system is very typical.
* It takes more-and-more time to do processing of this type of job.
* It takes more man power to prepare the employee enquiry register, maintain employee record, order detail & every customer detail records.
* It can maintain more than one file particularly.
* Integrity of data cannot be maintained.
* Redundancy is obstructing to maintain the data.
* If we have to find out a particular record, present system take more time.
* Modification in records is very difficult job.

So present system has many unavoidable problems.

**PURPOSE OF THE PROPOSED SYSTEM:**

The main purpose of the proposed system is to reduce the cost of man power and make system paperless. It takes less time to do their job. It is a user-friendly system ,any body can do this job who has learned computer operation. It does the job faster than the present system.

**BENEFITS OF PROPOSED SYSTEM:**

1. Reduce the cost.
2. Updating is easier.
3. Job is faster.
4. Easy to handle.
5. Integrity can be maintained.
6. Searching is so easy.
7. Reports are easily generated and more attractive.
8. We can generate and see any report in our own way.

#### FEASIBILITY STUDY

***Introduction:***

Before developing any computerized system it is imperative to examine the existing system to find out the drawbacks and requirements, which will be satisfied by the proposed system. It is necessary to examine the current procedure and information flow, to locate problem in the existing system, to identify what resources are used, to discuss with higher authorities the improvements that are necessary.

***Feasibility Study:***

Feasibility study is the procedure to identify, describe and evaluate proposed system and select the vast possible action for the job in terms of Economic, Technical and behavioral constraints.

***Cost benefit Analysis:***

The cost benefit analysis is to be considered as per given headings:

##### Technical Feasibility

Token Ring in the entire company makes the proposed system technically feasible to implement. The web server of the department can cater the request of any client through the Token Ring Network System thus fulfilling the basic requirement of the proposed system that is to compute data online from various processes.

##### Behavioural Feasibility

Regarding the behavioral feasibility, care had to be taken to provide a user- friendly system. It includes the training of the user on the proposed system. Care has been taken to provide the user as much facility as possible. The screen design is very much user friendly. Data entry jobs have been kept very easy and user friendly so that users can be trained very easily.

#### Requirement of the Proposed System

Requirement determination techniques involve studying the current system to find out how it works and identifying the area, which require changes. It may include alternative way to capture or process data.

After the initial study of the various activities, it was determined that the whole proposed system would consist of following modules:

###### System Design – An Introduction

System design is a solution for”How to approach the creation of a new system”. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Emphasis is on translating the performance requirements into design specifications. Design goes through logical and Physical stages of development.

The design covers the following: -

* Review of the current physical system – its data-flow, file content volume, frequencies etc.
* Preparation of output specifications- determination of format, content and frequency of reports etc.
* Preparation input specifications- determining the flow of the document from input data source to the actual input location, format and content of input functions.
* Preparations of edit, security and control specifications.
* Specification of plan to be implemented.

CONTEXT DIAGRAM

Context diagram represents the whole system as a single process.

It identifies the major input to the system and also identifies where the output of the system goes.

DEMAND ORDER

**DELIVERY NOTE**

**ENQUIRY**

QUOTATION

**Customer**

**PERMANENT**

**RECORDS**

E-R DIAGRAM

Entity – Relationship diagram for a pale ontological species occurrence data model.

The E-R (Entity – Relationship) data model views the real world as a set of basic objects (entities and relationships) among these objects.

It is intended primarily for the DB design process by allowing the specification of an enterprise scheme. This represents the overall logical structure of the DB.

In Resturent Management system has following entities:

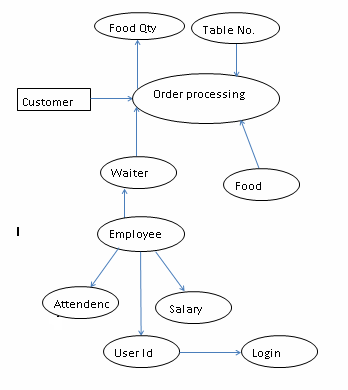
* EMPLOYEE
* CUSTOMER
* ORDER

**E-R DIAGRAM**

****

DATA FLOW DIAGRAM

Data flow diagram or DFD provides an easy, graphical means of modeling the flow of data through a system whether a manual, automated or mixture of both. It is a picture of the flow of the data through a system of any kind, showing the external entities that are sources or destinations of data, the processes that transform data and the places where data are stored.

****

**LEVEL 0 (ZERO) DFD:**

MASTER RECORDS

CUSTOMER

SERVICES

SYSTEM TESTING AND IMPLEMENTATION

***INTRODUCTION:***

As a step in system development life cycle, testing and implementation consist of final steps that put the new system into operation. An exhaustive test must be conducted to ascertain that the system produces right result. This chapter throws light on testing and implementation of the system.

***Testing:***

Testing involves executing the program (or part of it) using sample data and inferring from the output whether the software performs correctly or not. This can be done during module development (until testing 0 or when several modules are combined).

***Defect Testing:***

Defect Testing is testing for situation where the program does not meet its functional specification. Performance testing tests a system’s performance or reliability under realistic loads. This may go some way to ensuring that the program meets its nonfunctional requirements.

***Debugging:***

Debugging is a cycle of detection, location, repair and test. Debugging is a hypothesis testing process. When a bug is detected, the tester must form a hypothesis about the cause and location of the bug. Further execution of the program will usually take place to confirm the hypothesis. If the hypothesis is demonstrated to be incorrect, a new hypothesis must be formed. Debugging tools that show the state of the program are useful for this. But inserting print statement is often the only approach. Experienced debuggers use their knowledge of common and/or obscure bugs to facilitate the hypothesis testing process. After fixing a bug, the system must be retested to ensure that the key has worked and that no other bugs have been introduced. This is called regression testing. In principle, all tests should be performed again but this is often too expensive to do.

***Testing Planning:***

Testing needs to be planned to be cost and time effective. Planning is setting out standards for tests. Test plans out the context on which individual engineers can place their own work.

***Typical Test Plan contains:***

* Overview of testing process.
* Requirement tractability.
* List of items to be tested.
* Schedule.
* Recording procedure so that test results can be audited
* Hardware and Software requirements.

**Overview of Testing Strategies:**

***Alpha Testing And Beta Testing:***

The customer conducts the **Alpha Test** at the Developer’s site. The software is used in a natural setting with the developer” Looking over the shoulder” of the user and recording errors and usage problems. Alpha Tests is also known as Acceptance test since it is performed before the system is accepted as being fit for the intended users.

The **Beta test** is conducted at one more customer site by the end user of the software. Unlike Alpha testing, the developer is not present. Therefore, the beta test is ‘live’ application of the software in an environment that cannot be controlled by the developer. The customer records all problems (real or imaging) that are encountered during beta testing and reports these to the developer at regular intervals. System or different stages of the process.

***Top down testing:***

This approach tests high levels of system detailed components. This is appreciated when developing the systems top-down. Likely to show structural design errors early, validation (as distinct from verification) can begin early. Its’ disadvantages is that stubs needs to be generated (extra efforts) and might be Impracticable, if component is complex. Tests output may be difficult to observe.

***Bottom-up testing:***

This is opposite to top-down testing. This testing tests low-level unit that works up Hierarchy.

The advantage and disadvantage of bottom-up mirror those of top-down .In this testing there is need to write test-drive for each unit. These are reusable as the units itself. Combining top-down development with bottom-up testing means that all parts of systems must be implemented before testing can begin. Therefore it does not accord with incremental approach discussed above.

***Black-Box (functional) testing:***

Black Box Testing is concerned with the proper execution of the program specifications. In this testing, each function or sub-program used in the main program is first identified. Test cases are devised to test each function or sub-program separately. Test cases are decided solely on the basis of the requirements or specifications of the program and not on the basis of the coding. Black Box testing is complimentary to white box methods. This type of testing attempts to find the following errors:

1. Interface errors
2. Incorrect or missing functions
3. Errors in external database access
4. Performance errors
5. Initialization and termination errors

***White-Box (structural) testing:***

Core advantage is that structure of code can be used to find out how many test cases need to be performed. Knowledge of the algorithm (examination of the code) can be used to identify the testing based on knowledge of structure of component. Flow graphs are pictorial representation of the paths of control through program. Use flow graph to design cases that execute each path. Static tools may be used to make this easier in programs that have complex branching structure tool supports.

***Interface Testing:***

Usually done at integration stage when modules or sub-systems are combined. Objective is to detect errors or invalid assumptions about interfaces between modules. Reason these are not shown up to testing is that test case may perpetuate same incorrect assumption made by modular designer. Particularly important when OO development has been used.

***Stress testing:***

Test system’s ability to cope with a specified load (e.g. transactions per second). Test plans to increase load incrementally to go beyond design limit until system fails (this test particularly important for distributed systems).

***Back–to-back-testing:***

Comparison of test results from different versions of the system (e.g. compare with prototype, previous version or different configuration). Process runs first system, saving test case results. Run second time also saving its results. Compare results files. Note that no bugs imply any bugs.

Following testing has been done in eg. Learning management system:

***Installation test:***

The accepted system was installed in the environment in which it would be used and final installation test was performed to make sure that the system is working, as it should.

***User acceptance test:***

This task has been carried out in the implementation and testing of the application for various entities defined in the Resturent Management system. The various entities would be as under:

* + Customer,
  + Services,
  + Employee,
  + Bill Details,
  + MainBill,
  + Report

**TABLE STRUCTURE**

**Table Name : Customer\_ent**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Cusid | Text | 5 |
| Name | Text | 20 |
| Address | Text | 40 |
| Phno | Text | 12 |
| Email | Text | 25 |
| Doe | Date |  |
| Time | Text | 15 |

**Table Name : Emp\_ent**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Empid | Text | 5 |
| Name | Text | 20 |
| Fname | Text | 20 |
| Address | Text | 40 |
| Dob | Date |  |
| Phno | Number | 12 |
| Gender | Text | 7 |
| Doj | Date |  |
| Quali | Text | 18 |
| Depar | Text | 10 |
| Emptype | Text | 15 |
| Salary | Number | 6 |

**Table Name : Emp\_ent**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Empid | Text | 5 |
| Name | Text | 20 |
| Fname | Text | 20 |
| Address | Text | 40 |
| Dob | Date |  |
| Phno | Number | 12 |
| Gender | Text | 7 |
| Doj | Date |  |
| Quali | Text | 18 |
| Depar | Text | 10 |
| Emptype | Text | 15 |
| Salary | Number | 6 |

**Table Name : Attendence**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Attendencedt | Date |  |
| Empid | Text | 5 |
| Name | Text | 20 |
| Status | Text | 10 |
| Come | Text | 15 |
| Out | Text | 15 |
| Remark | Text | 40 |

**Table Name : Appetizers**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Mealnm | Text | 30 |
| Price | Number | 4 |
| Cate | Text | 100 |

**Table Name : Bar**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Mealnm | Text | 30 |
| Price | Number | 4 |
| Cate | Text | 100 |

**Table Name : Beverages**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Mealnm | Text | 30 |
| Price | Number | 4 |
| Cate | Text | 100 |

**Table Name : Desserts**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Mealnm | Text | 30 |
| Price | Number | 4 |
| Cate | Text | 100 |

**Table Name : Main\_course**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Mealnm | Text | 30 |
| Price | Number | 4 |
| Cate | Text | 100 |

**Table Name : Soups\_salad**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Mealnm | Text | 30 |
| Price | Number | 4 |
| Cate | Text | 100 |

**Table Name : Ordeer**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Order\_no | Number | 8 |
| Cname | Text | 20 |
| Wtrid | Text | 5 |
| Tblno | Number | 3 |
| Tot | Number | 5 |
| TX | Number | 6 |
| FTOT | Number | 7 |

**Table Name : Order\_qty**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Order\_no | Number | 8 |
| Qty | Number | 2 |
| Description | Text | 30 |
| Price | Number | 4 |
| Total | Number | 5 |

**Table Name : Sal\_ent**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Saldate | Date |  |
| Empid | Text | 5 |
| Name | Text | 20 |
| Ba | Number | 5 |
| Ta | Number | 5 |
| Da | Number | 5 |
| Hra | Number | 5 |
| Oa | Number | 5 |
| Pf | Number | 5 |
| Total | Number | 6 |

**Table Name : Sal\_set**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Ta | Number | 5 |
| Da | Number | 5 |
| Hra | Number | 5 |
| Oa | Number | 5 |
| Pf | Number | 5 |

**Table Name : Table\_Entry**

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Table\_no | Number | 3 |
| Status | Text | 4 |
| Cusid | Text | 10 |

**Table Name : User\_id**

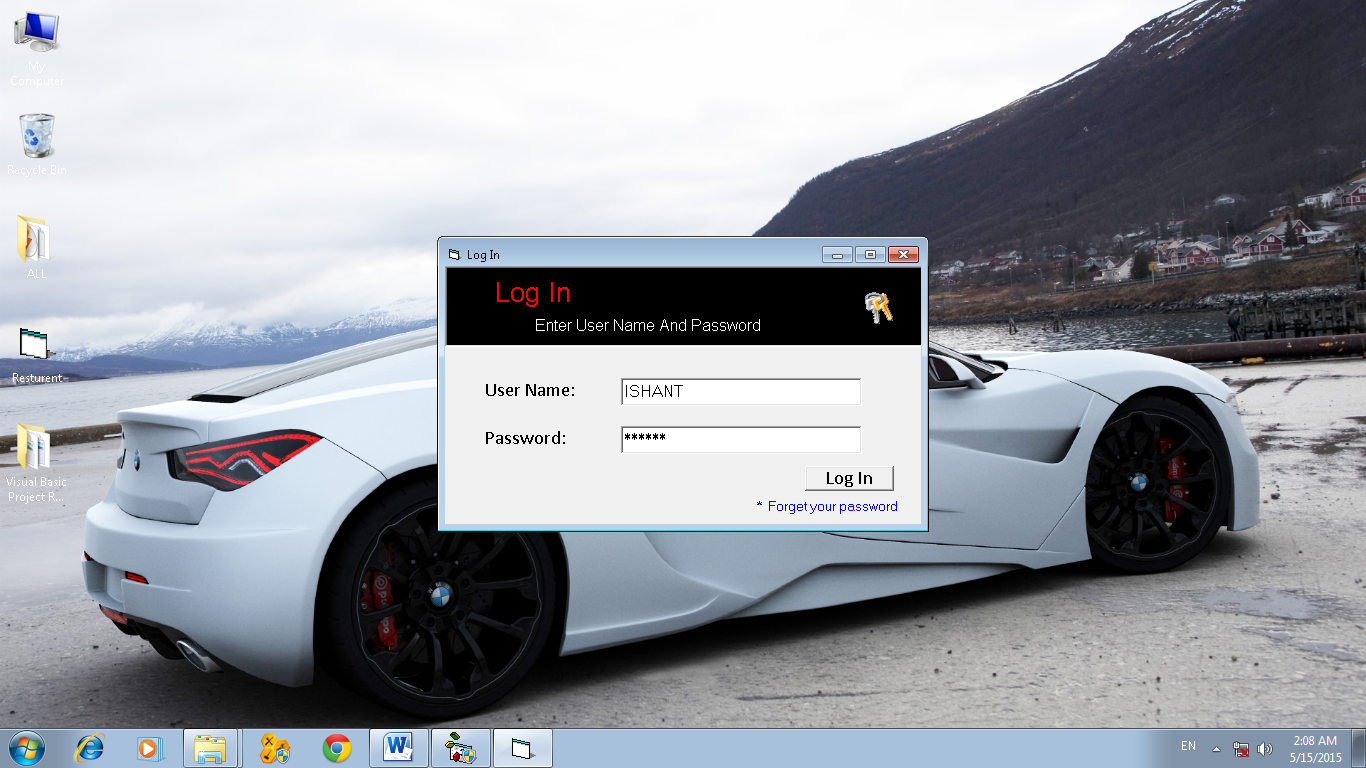
|  |  |  |
| --- | --- | --- |
| **Data Name** | **Type** | **Size** |
| Empid | Text | 20 |
| Userid | Text | 20 |
| Password | Text | 20 |
| Userlbl | Text | 15 |
| Secques | Text | 40 |
| Answer | Text | 20 |

**Login System**

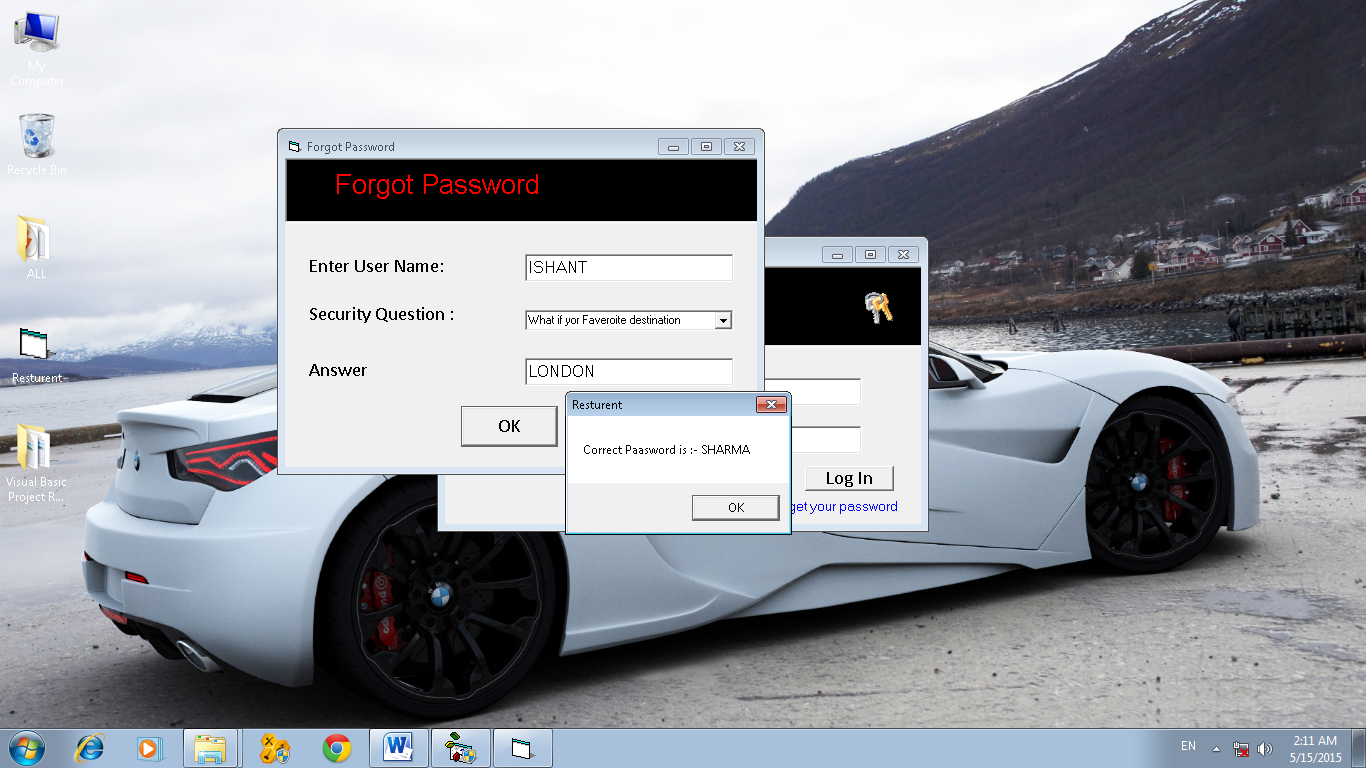
Splash Form



Login Form



Forgot Password Form

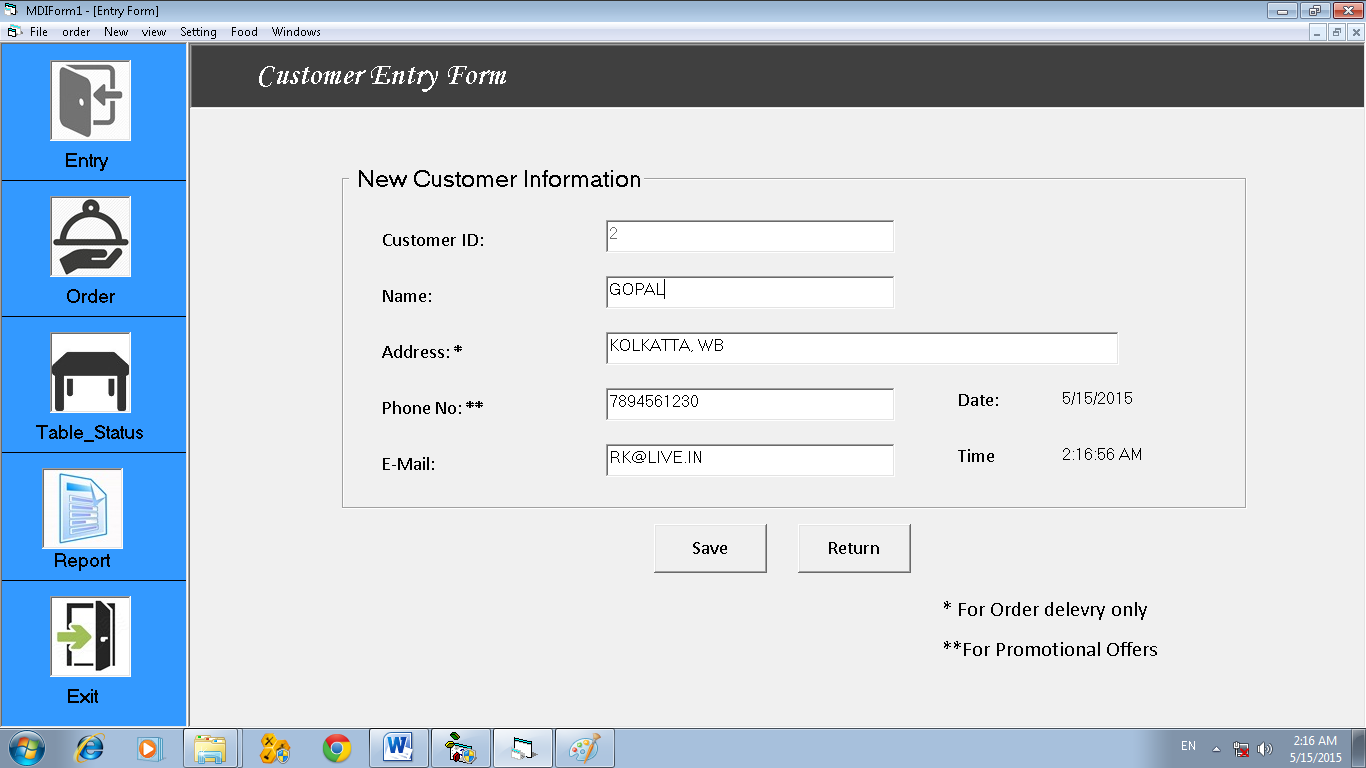


Mdi Main Form

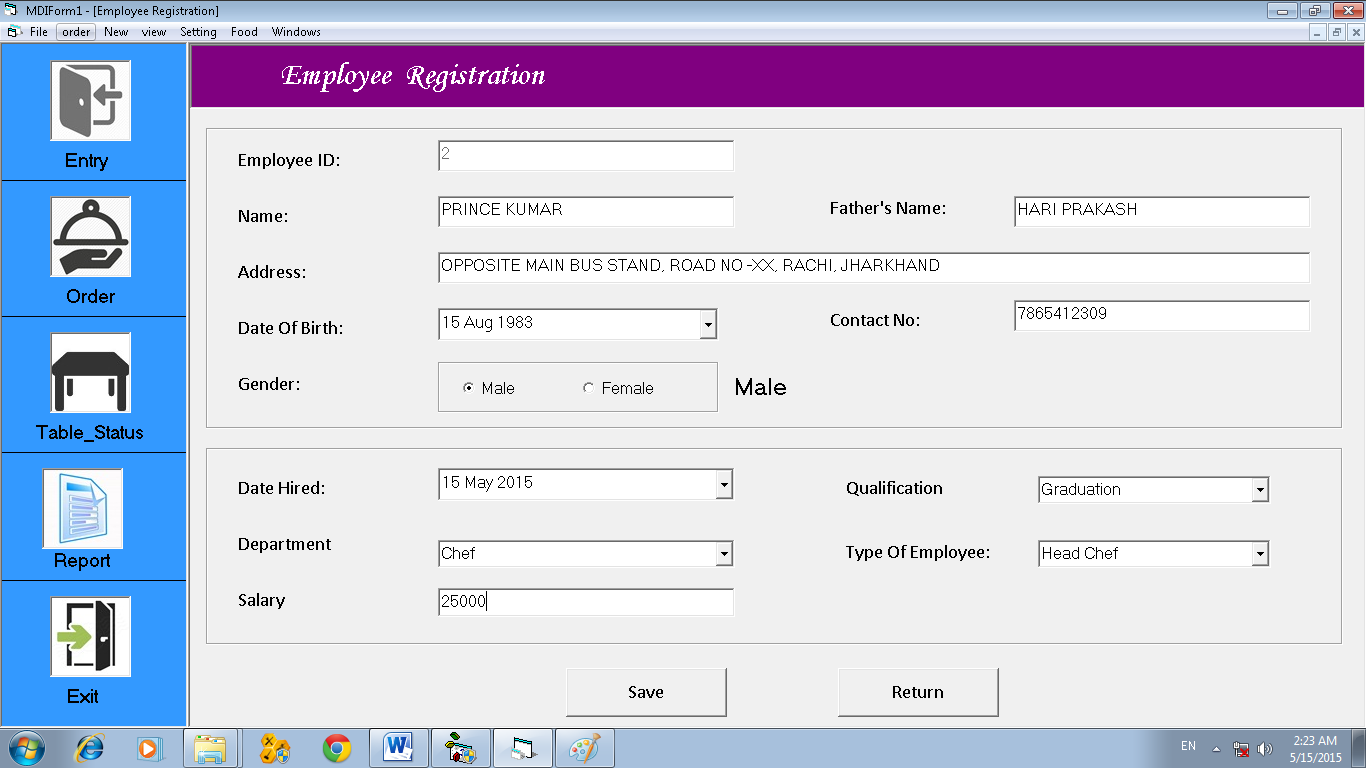


**Program Input**

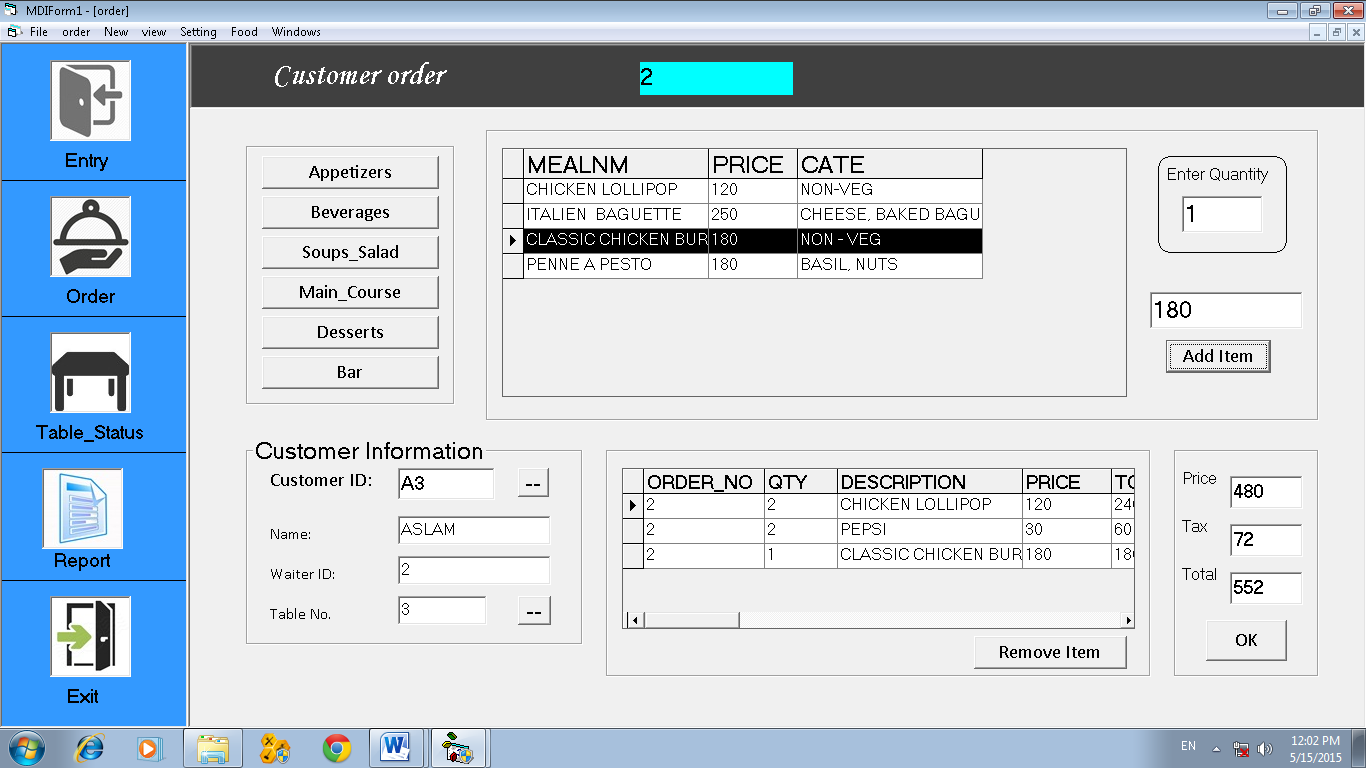
Customer Entry Form



Employee Registration



Order Form



Food Entry Form

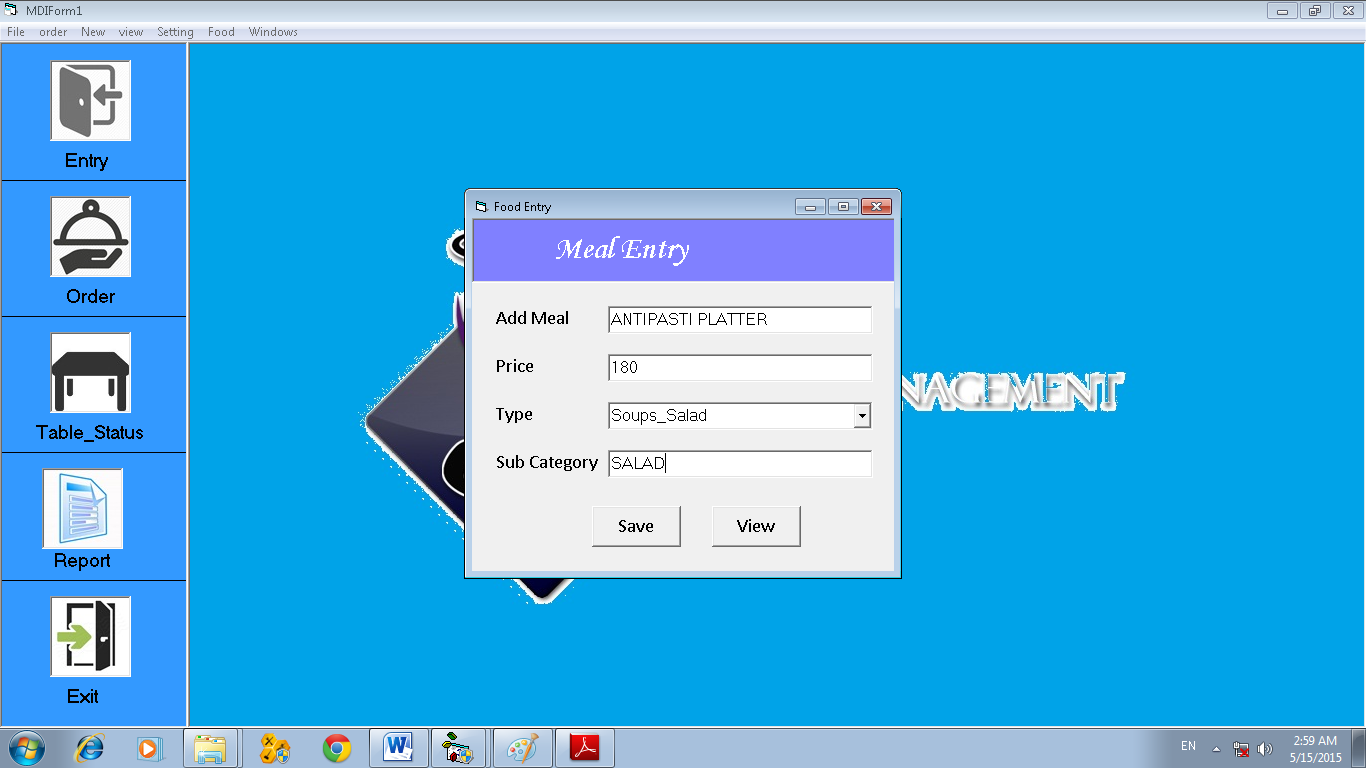
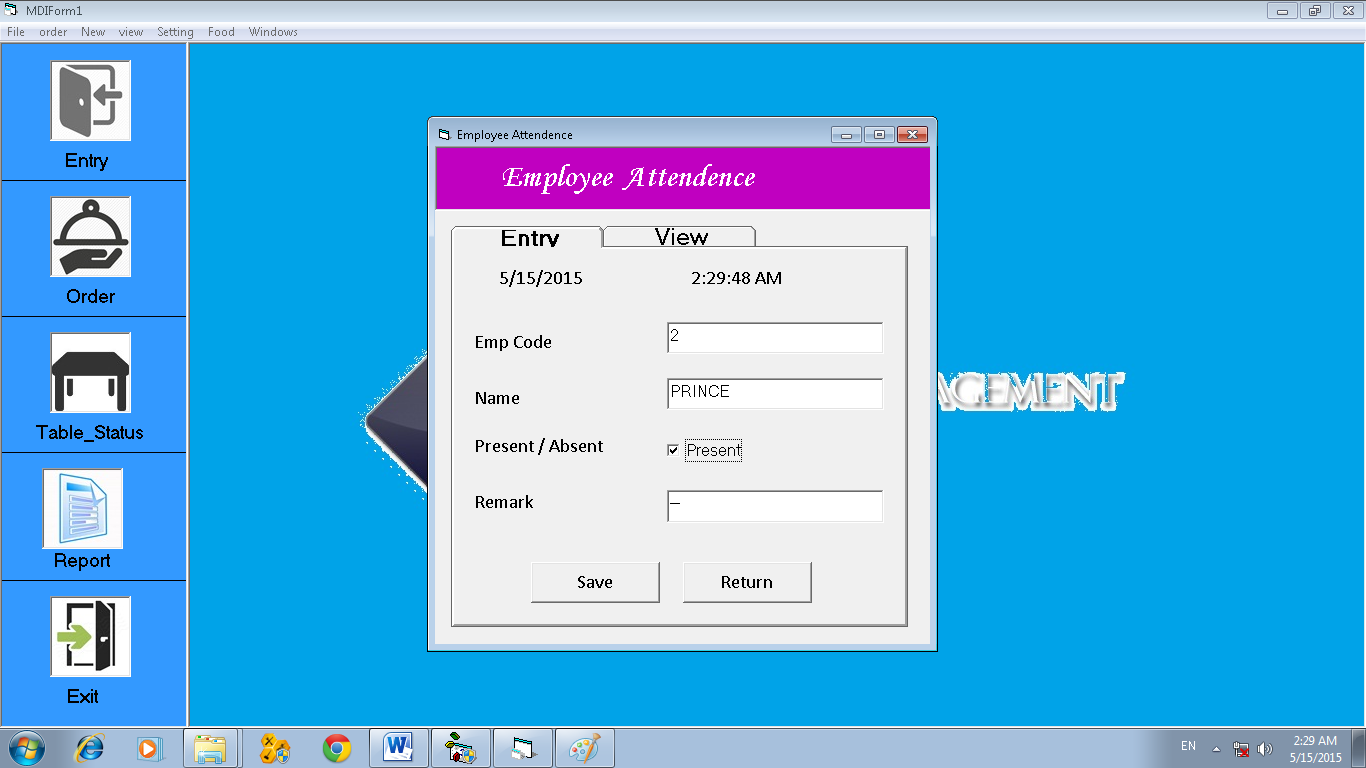


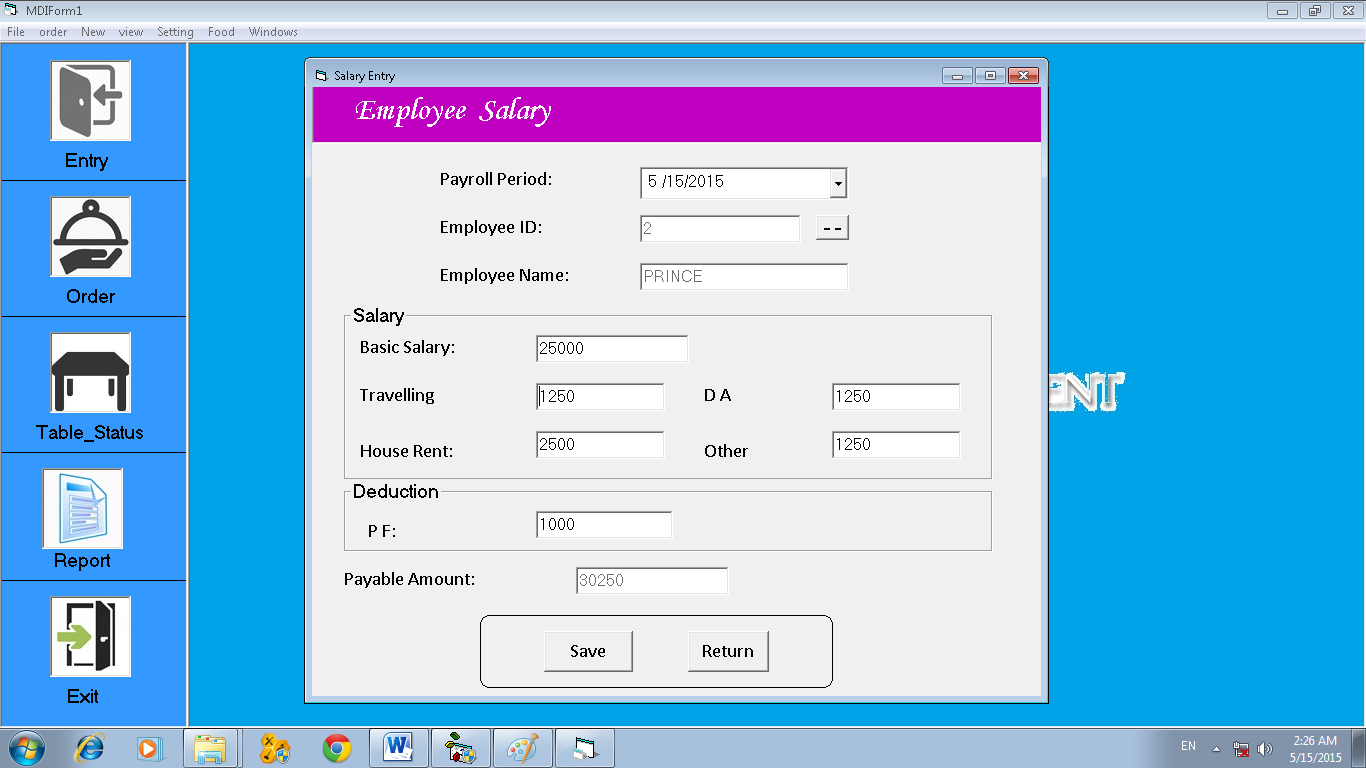
Table Entry Form



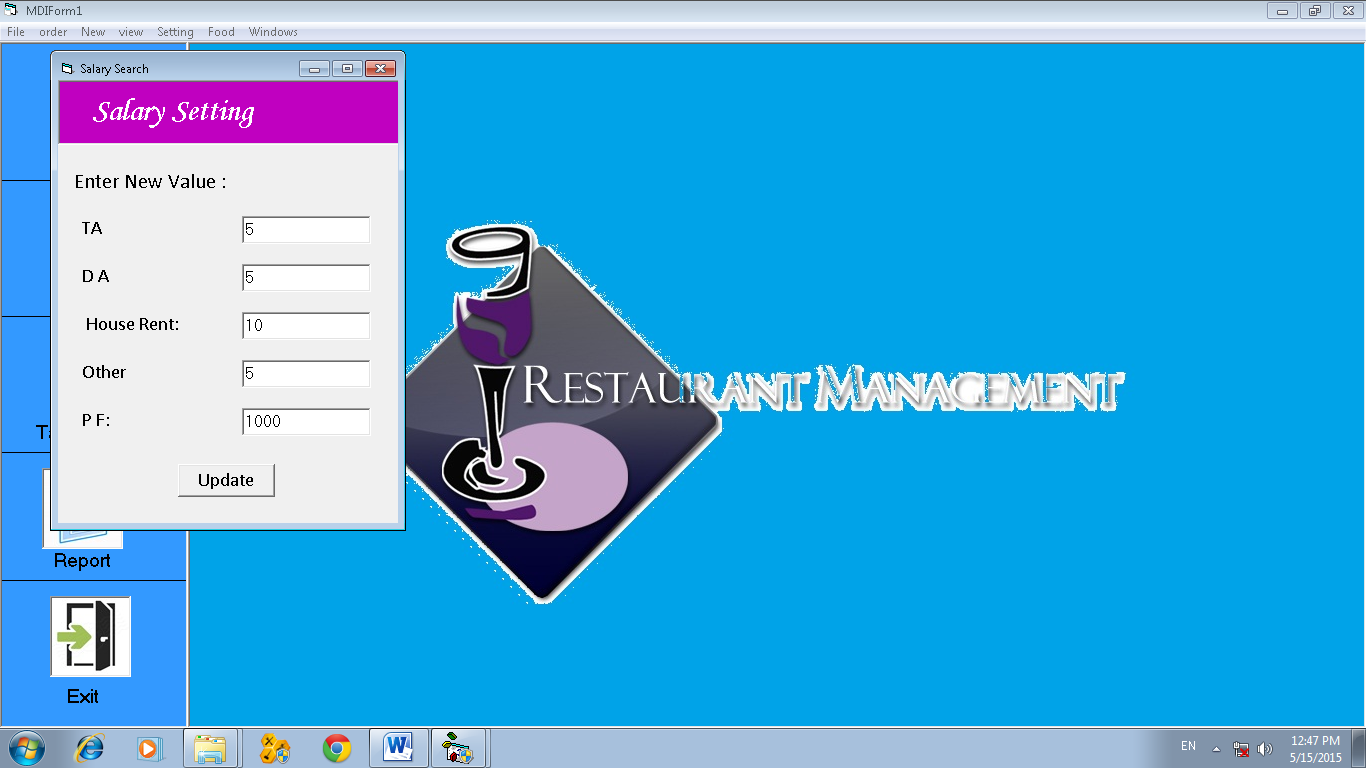
Employee Attendence form



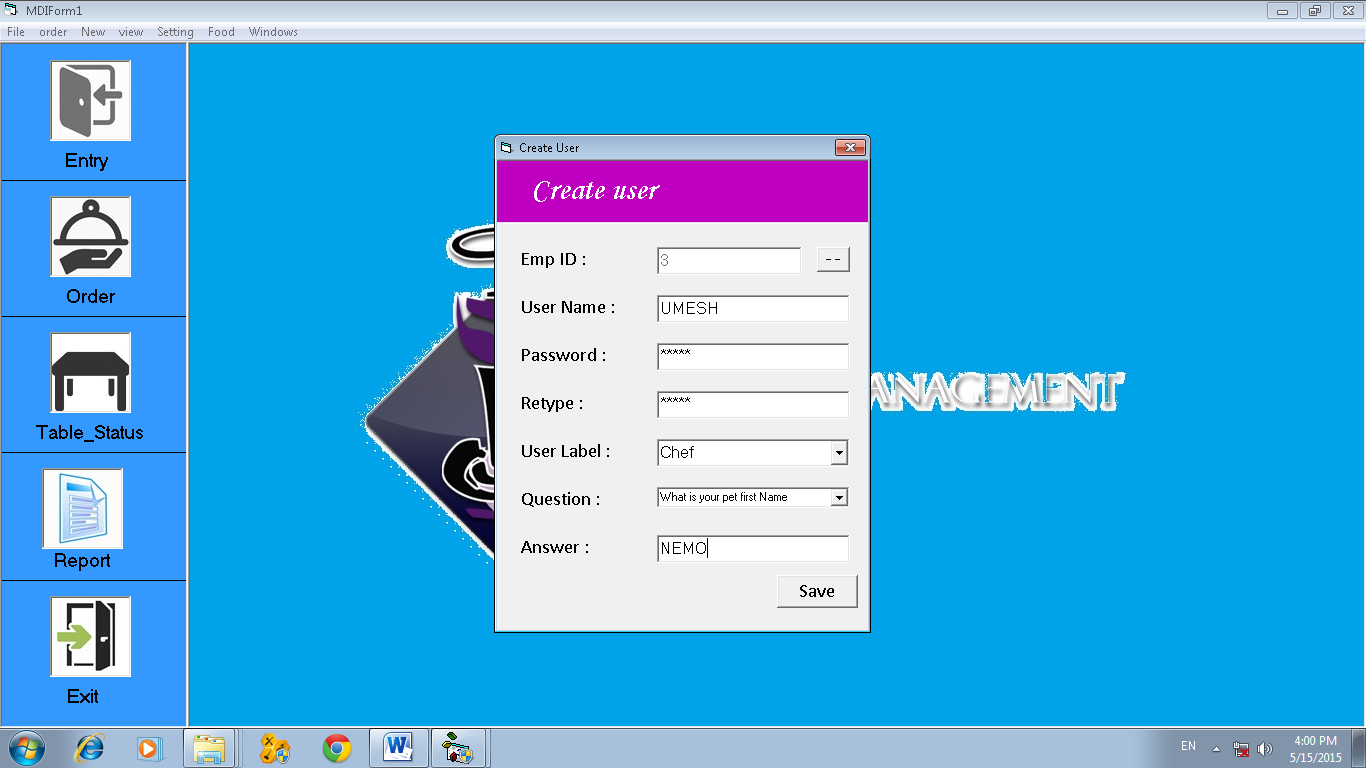
Salary Form



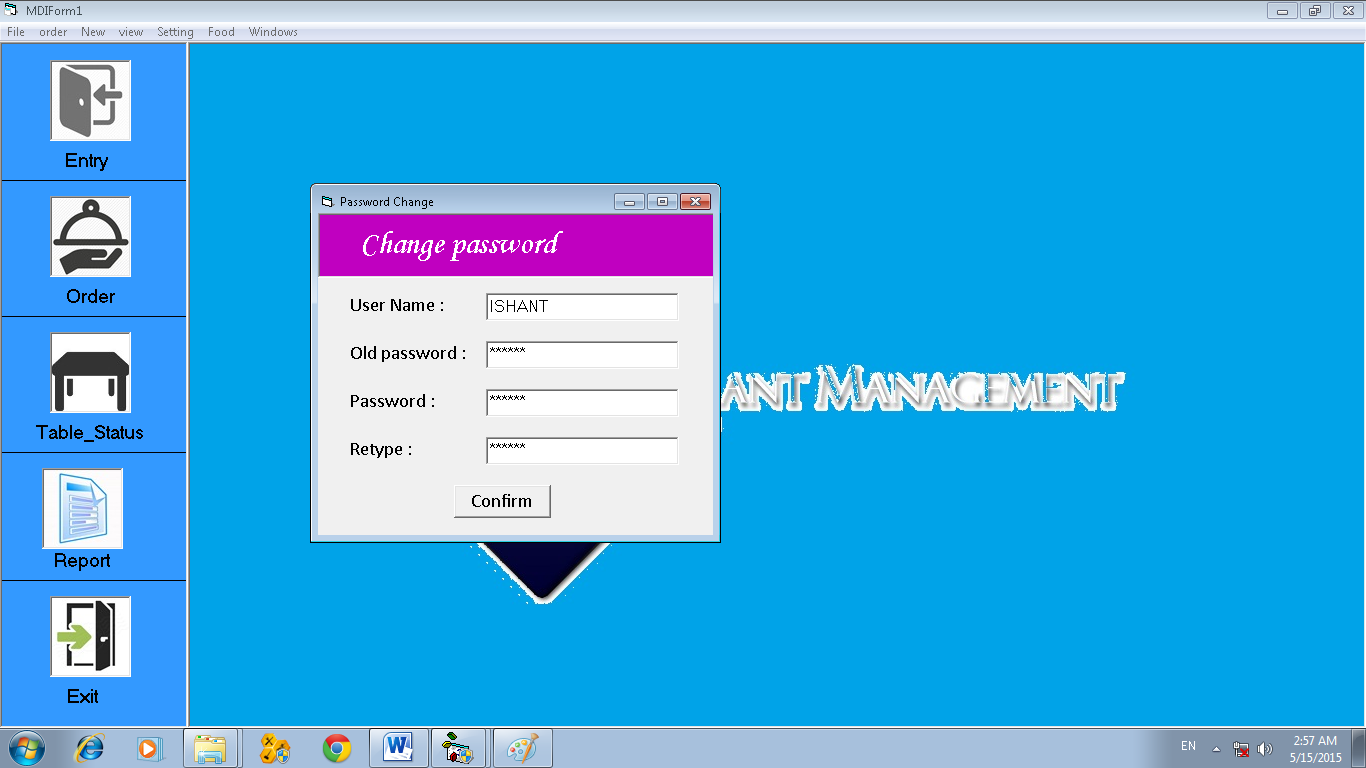
Salary Setting form



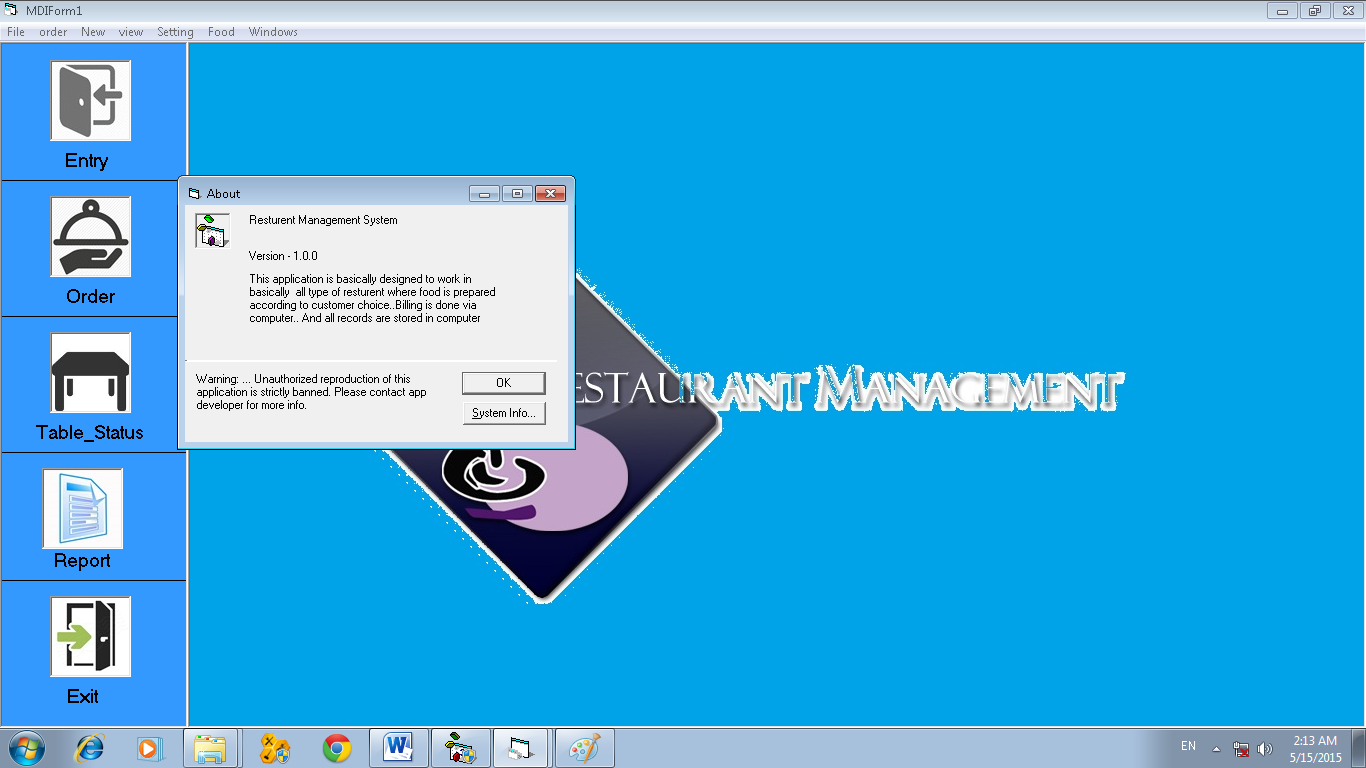
User Creation form



Password change form

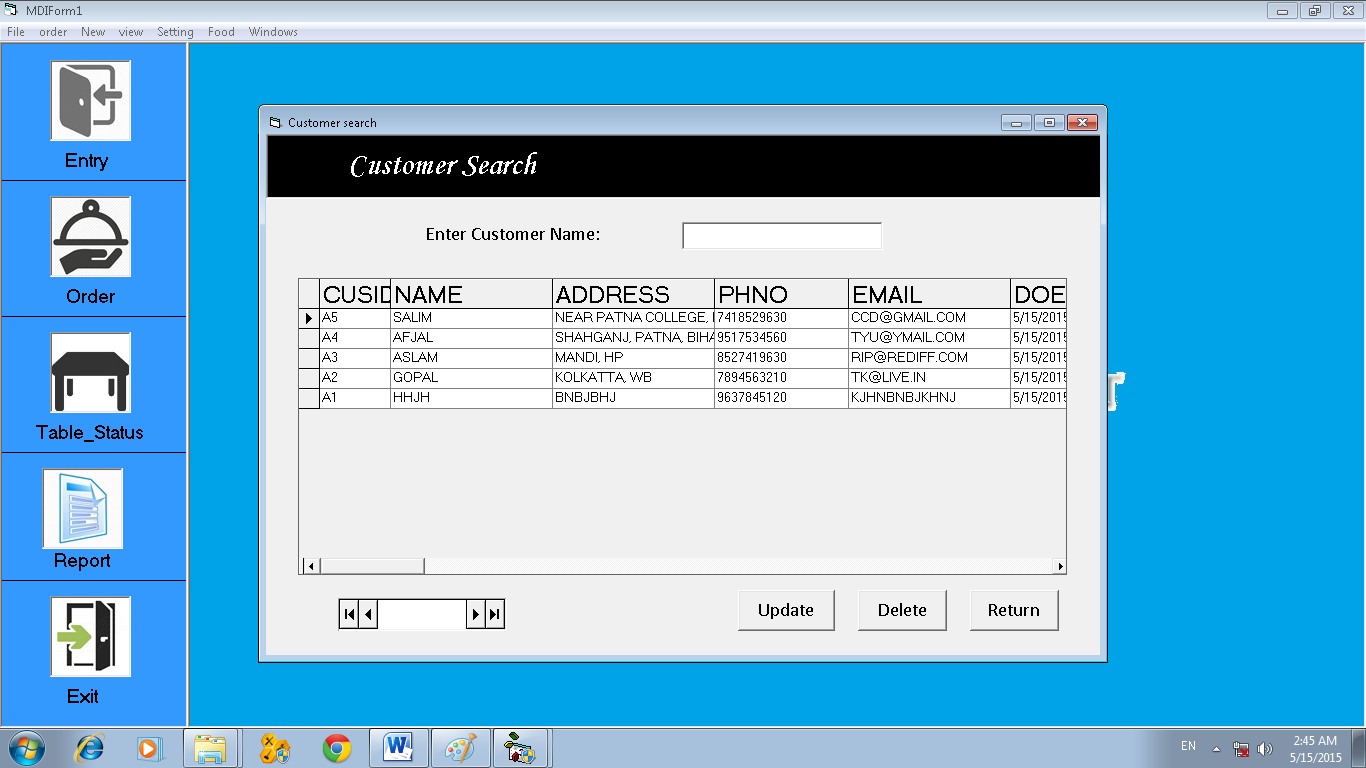


About Form

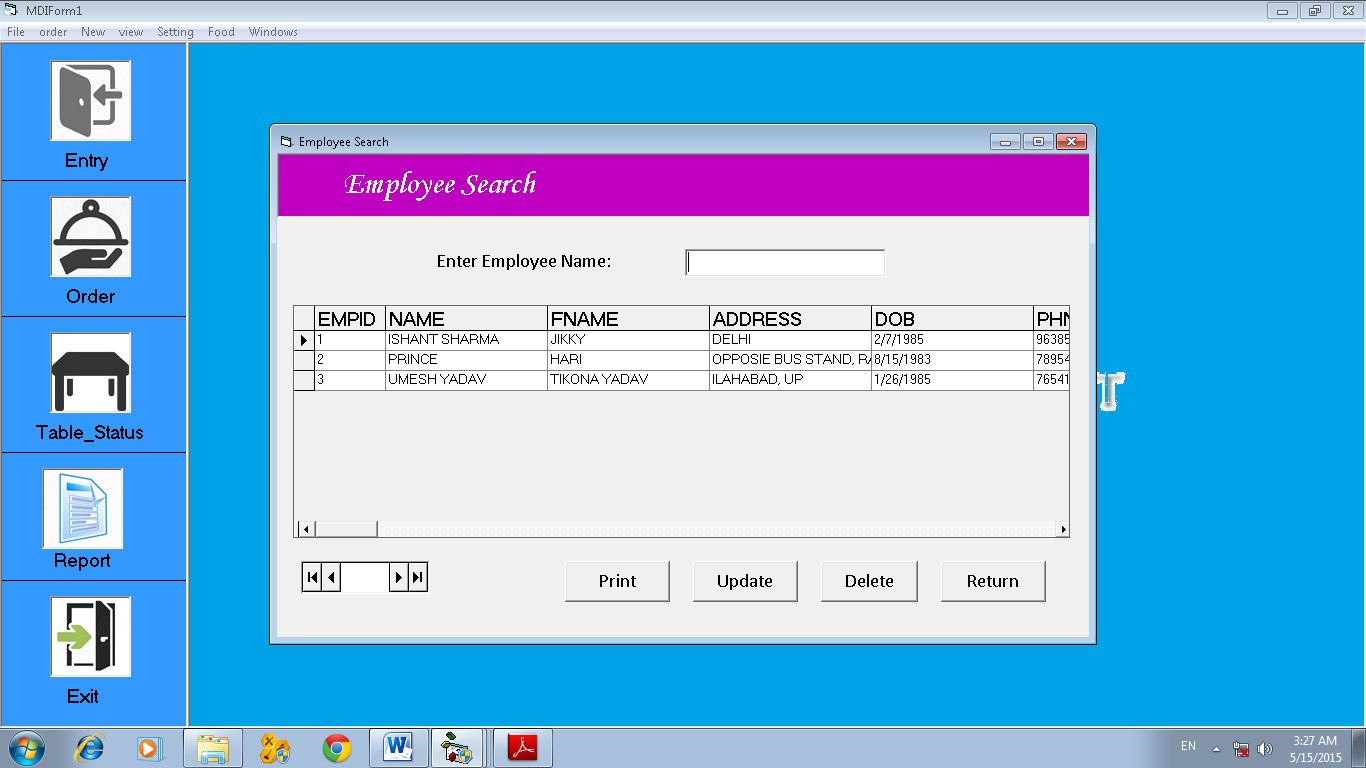


**Program Output**

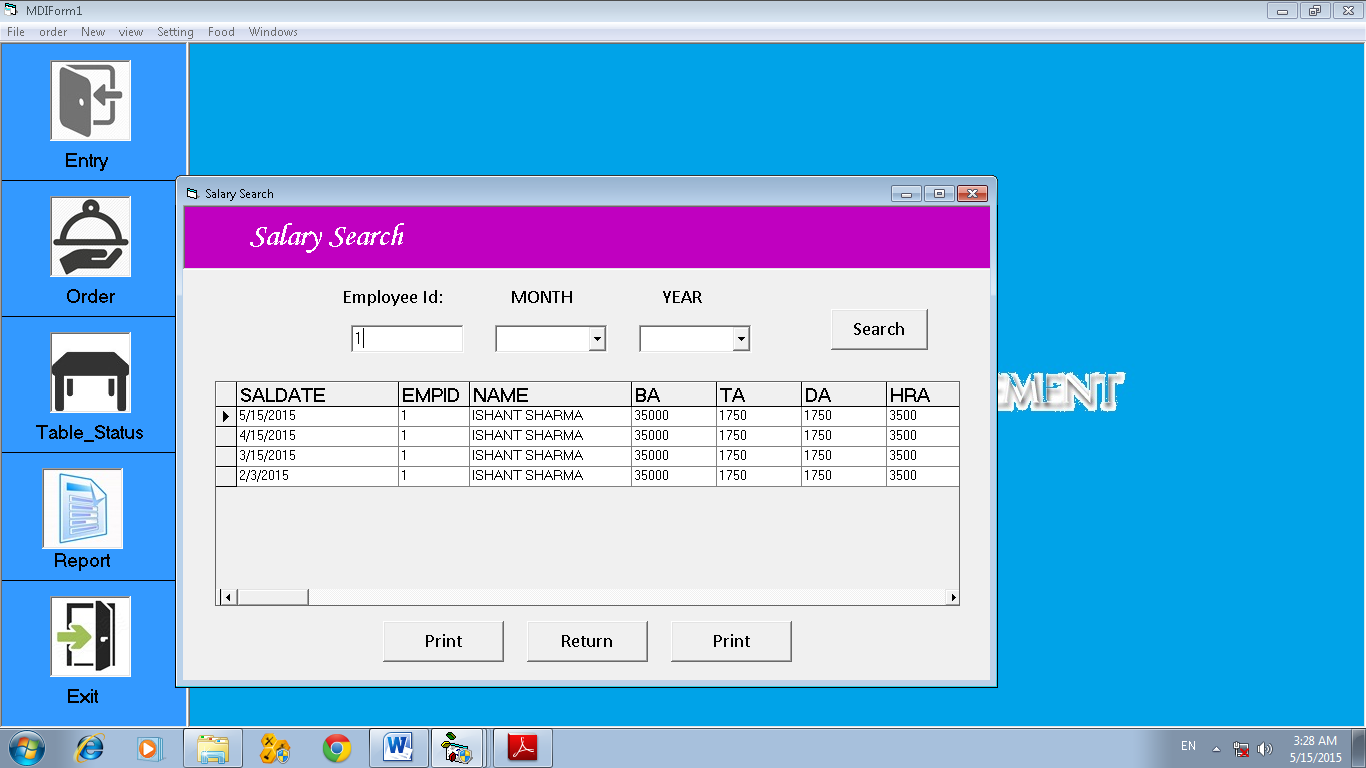
Customer search form



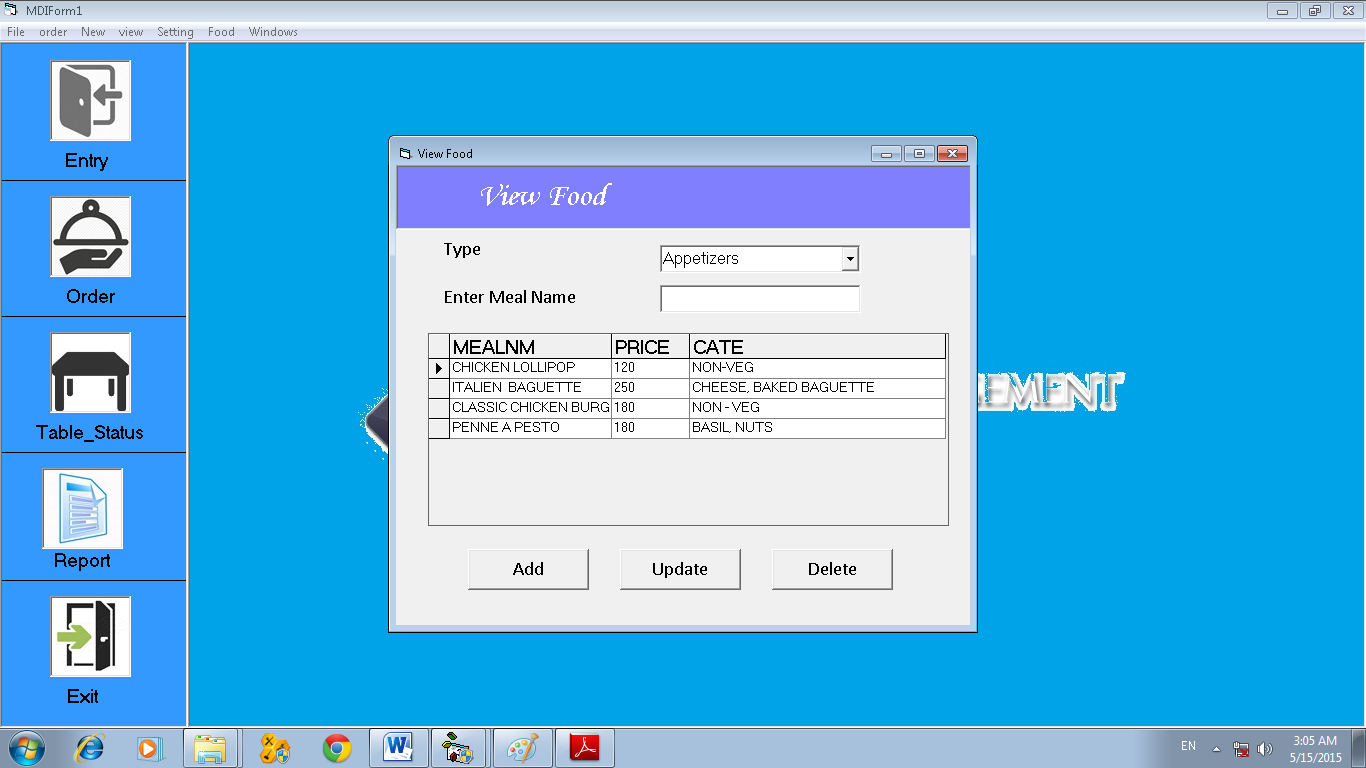
Employee Search Form



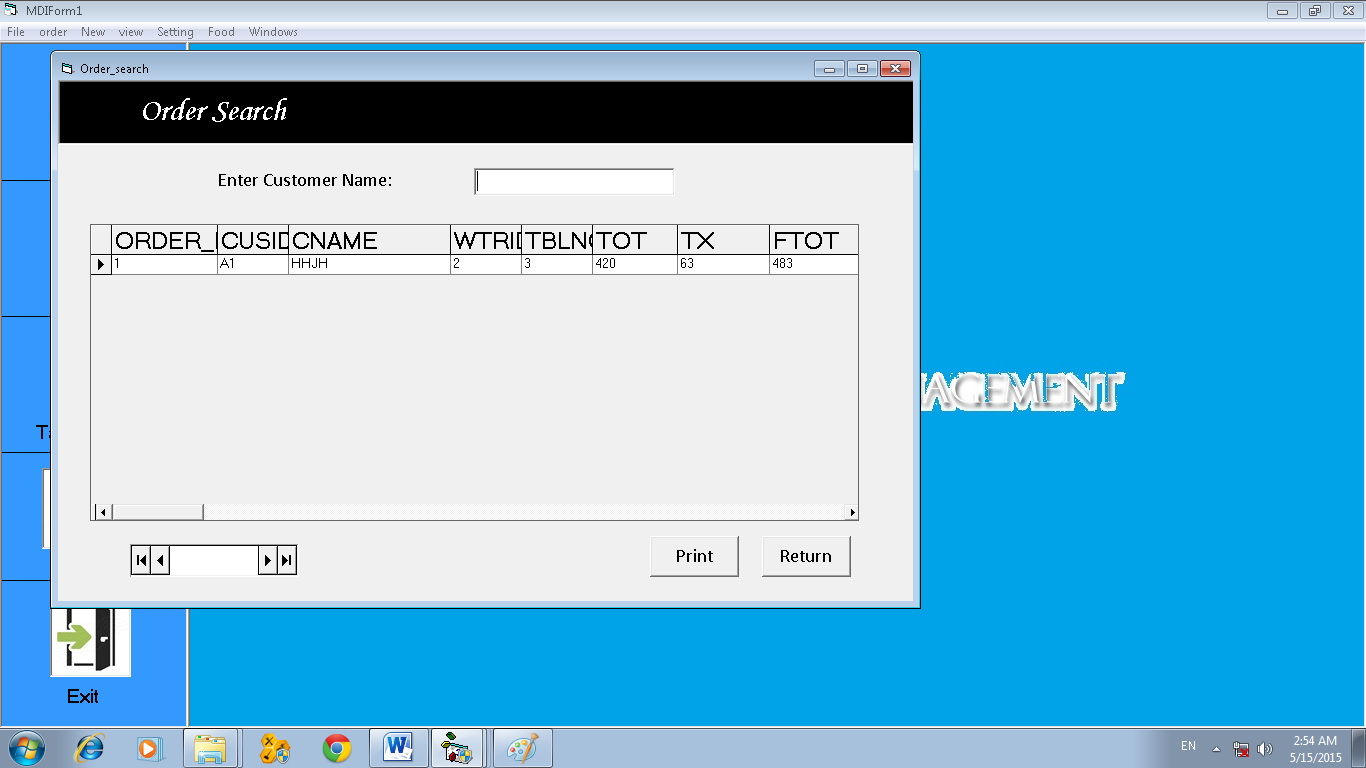
Salary Search Form



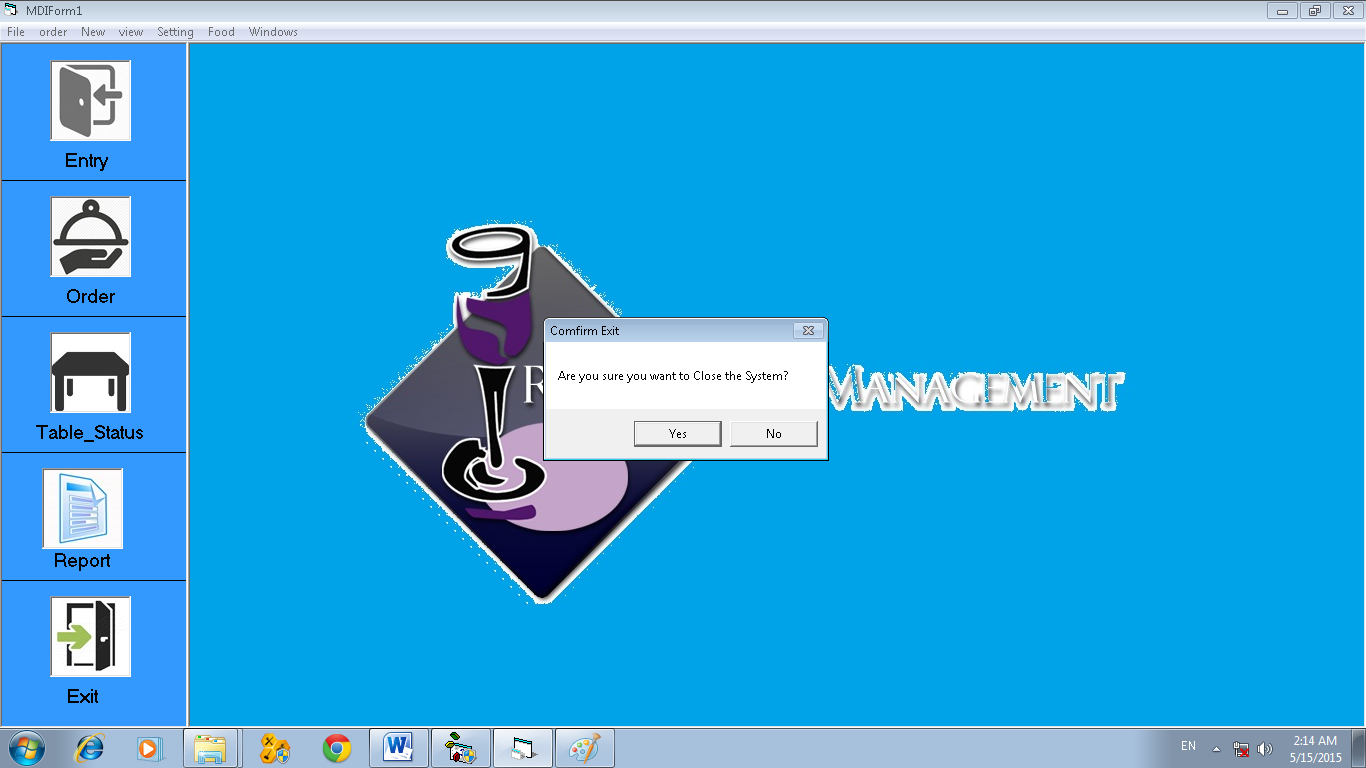
Food Search Form



Order Search Form



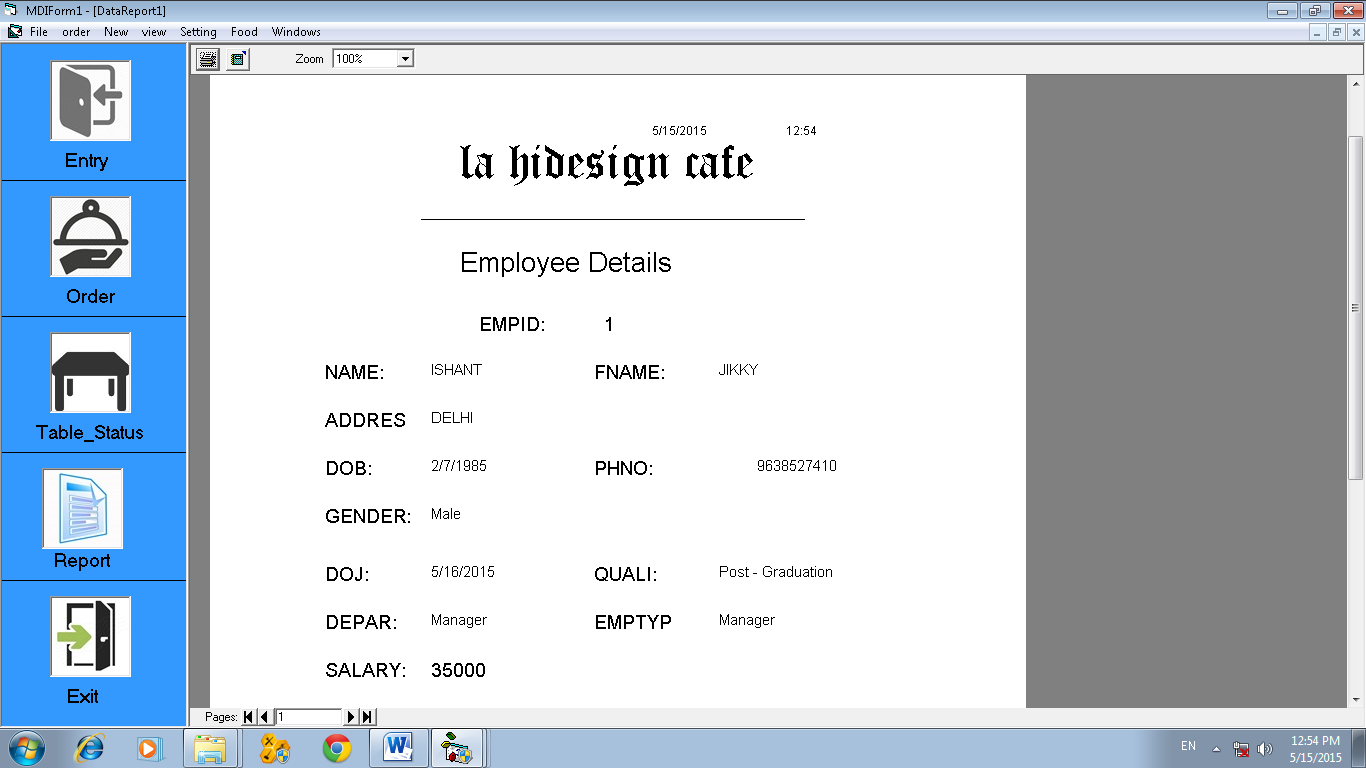
Close Form



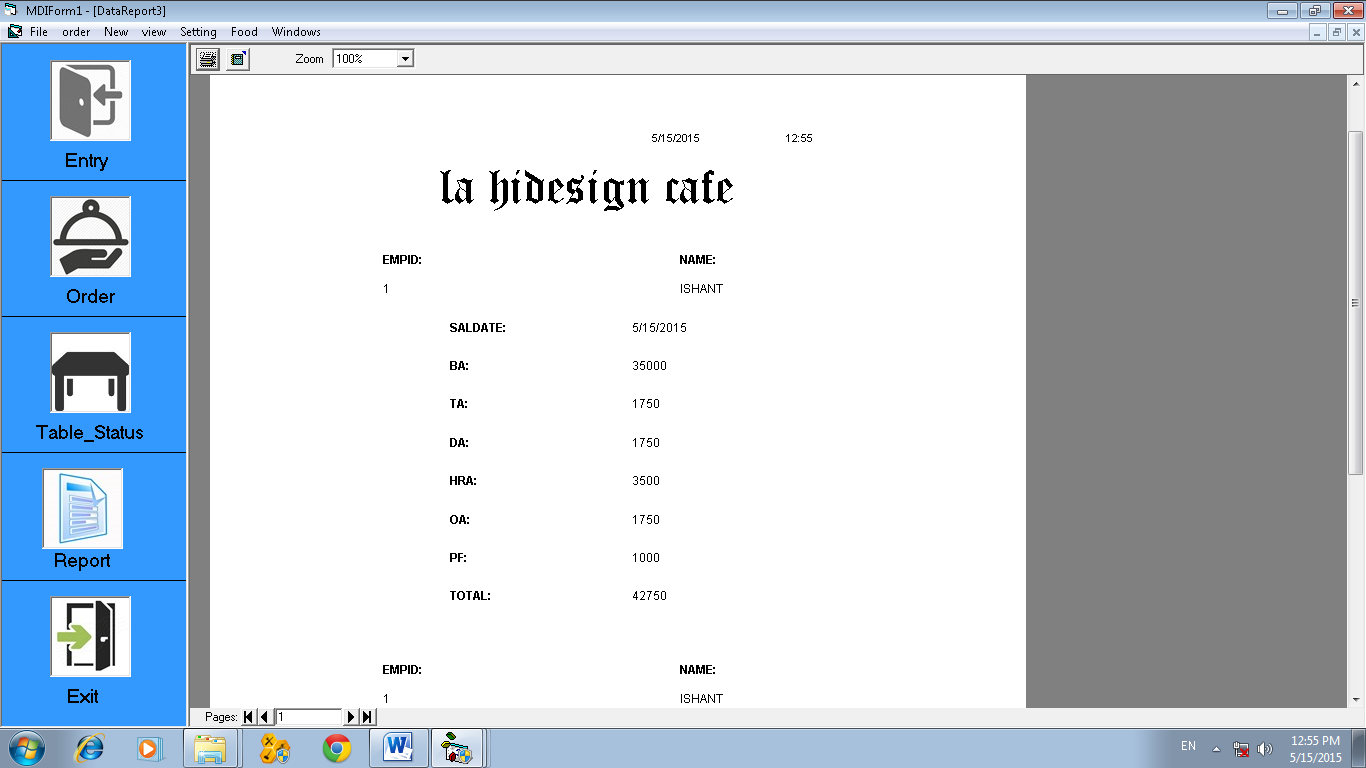
Order Report



Employee Report



Salary Report



**About**

Option Explicit

Private Sub cmdOK\_Click()

Unload Me

End Sub

**Create User**

Option Explicit

Private Sub Command1\_Click()

Employee\_Search.update.Visible = False

Employee\_Search.delete.Visible = False

Employee\_Search.close.Visible = False

Employee\_Search.Select.Visible = True

Employee\_Search.Show vbModal

'Text2.Text = r.Fields(1)

End Sub

Private Sub Command2\_Click()

On Error GoTo hell

If Text3.Text = Text4.Text And Text1.Text <> "" And Text2.Text <> "" And Text3.Text <> "" And Text4.Text <> "" And Text5.Text <> "" And Combo1.Text <> "" And Combo2.Text <> "" Then

conn

sql = "insert into user\_id values ('" + Text1.Text + "' , '" + Text2.Text + "' ,'" + Text3.Text + "' , '" + Combo1.Text + "', '" + Combo2.Text + "','" + Text5.Text + "')"

Set r = c.Execute(sql)

MsgBox "record saved"

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Combo1.Text = ""

Combo2.Text = ""

Text5.Text = ""

Text1.SetFocus

Else

MsgBox "Password Not match OR Field Blank"

Text3.Text = ""

Text4.Text = ""

Text3.SetFocus

End If

hell:

If Err.Number = -2147467259 Then MsgBox " Data overflow", vbExclamation, "Ok"

End Sub

**Customer Search**

Option Explicit

Private Sub cmddelete\_Click()

Adodc1.Recordset.Delete

End Sub

Private Sub cmdupdate\_Click()

Entry\_form.cid.Text = DataGrid1.Columns(0)

Entry\_form.cname.Text = DataGrid1.Columns(1)

Entry\_form.address.Text = DataGrid1.Columns(2)

Entry\_form.cphone.Text = DataGrid1.Columns(3)

Entry\_form.email.Text = DataGrid1.Columns(4)

Entry\_form.cmdupdate.Visible = True

Entry\_form.cmdsave.Visible = False

Unload Me

Entry\_form.Show

End Sub

Private Sub Command1\_Click()

End Sub

Private Sub Command2\_Click()

Unload Me

End Sub

Private Sub Form\_Load()

End Sub

Private Sub Select\_Click()

Order.Text1.Text = DataGrid1.Columns(0).Value

Order.Text2.Text = DataGrid1.Columns(1).Value

Unload Me

Order.Show

End Sub

Private Sub Text1\_Change()

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

sql = "select \* from customer\_ent where name like '%" & Text1.Text & "%'"

r.CursorLocation = adUseClient

r.CursorType = adOpenStatic

r.LockType = adLockOptimistic

r.Open sql, c, , , adCmdText

Set DataGrid1.DataSource = r

DataGrid1.Refresh

End Sub

**Employee attendance**

Option Explicit

Private Sub close\_Click()

Unload Me

End Sub

Private Sub Cmdsave\_Click()

On Error GoTo hell

conn

sql = "insert into attendence values ('" + Format(atdate.Caption, "dd MMM yyyy") + "' , '" + ecode.Text + "' , '" + ename.Text + "' , '" + status.Caption + "' , '" + attime.Caption + "', ' ' , '" + remark.Text + "')"

Set r = c.Execute(sql)

MsgBox "record saved"

Adodc1.Refresh

DataGrid1.Refresh

hell:

If Err.Number = -2147467259 Then MsgBox " Data overflow", vbExclamation, "Ok"

End Sub

Private Sub Form\_Load()

atdate.Caption = Date

End Sub

Private Sub out\_Click()

conn

sql = " update ATTENDENCE SET OUT = '" + attime.Caption + "' where ATTENDENCEDT = '" + Format(atdate.Caption, "dd MMM yyyy") + "' AND EMPID= '" + DataGrid1.Columns(1) + "'"

Set r = c.Execute(sql)

MsgBox " Record Updated"

Adodc1.Refresh

DataGrid1.Refresh

End Sub

Private Sub present\_Click()

If present.Value = 1 Then

status.Caption = "present"

Else

status.Caption = "Absent"

End If

End Sub

Private Sub Timer1\_Timer()

attime.Caption = Time

End Sub

**Employee Registration**

Option Explicit

Dim que As String

Private Sub Cmdsave\_Click()

On Error GoTo hell

If ename.Text = "" Or fname.Text = "" Or address.Text = "" Or phone.Text = "" Or gender.Caption = "" Or quali.Text = "" Or depart.Text = "" Or etype.Text = "" Or salary.Text = "" Then

MsgBox "PLEASE ENTER ALL VALUES"

Else

conn

sql = "insert into emp\_entry values ('" + eid.Text + "' , '" + ename.Text + "' ,'" + fname.Text + "' , '" + address.Text + "' , '" + Format(dob.Value, "dd MMM yyyy") + "' , " + phone.Text + " , '" + gender.Caption + "' , '" + Format(doj.Value, "dd MMM yyyy") + "' , '" + quali.Text + "' , '" + depart.Text + "' , '" + etype.Text + "' , " + salary.Text + ")"

Set r = c.Execute(sql)

MsgBox "record saved"

eid.Text = ""

ename.Text = ""

fname.Text = ""

address.Text = ""

phone.Text = ""

quali.Text = ""

salary.Text = ""

ename.SetFocus

End If

hell:

If Err.Number = -2147467259 Then MsgBox " Data overflow", vbExclamation, "Ok"

End Sub

Private Sub cmdupdate\_Click()

On Error GoTo hell

conn

s = " update emp\_entry SET empid='" + eid.Text + "', name='" + ename.Text + "', fname='" + fname.Text + "', address='" + address.Text + "', dob='" + Format(dob.Value, "dd MMM yyyy") + "', phno= " + phone.Text + ", gender='" + gender.Caption + "', doj='" + Format(doj.Value, "dd MMM yyyy") + "', quali= '" + quali.Text + "', depar='" + depart.Text + "', emptype='" + etype.Text + "', salary=" + salary.Text + " where empid='" + eid.Text + "'"

Set r = c.Execute(s)

MsgBox " Record Updated "

hell:

If Err.Number = -2147467259 Then MsgBox " Data overflow", vbExclamation, "Ok"

End Sub

Private Sub Command2\_Click()

Unload Me

End Sub

Private Sub Form\_Load()

conn

que = "select count (EMPID) from EMP\_ENTRY"

Set r = c.Execute(que)

eid.Text = r.Fields(0) + 1

dob.Value = Date

doj.Value = Date

End Sub

Private Sub Option1\_Click()

gender.Caption = "Male"

End Sub

Private Sub Option2\_Click()

gender.Caption = "Female"

End Sub

Private Sub Text5\_Validate(Cancel As Boolean)

'Call onlynum(Text5)

End Sub

Private Sub salary\_LostFocus()

Call onlynum(salary)

End Sub

**Employee Search**

' when clicked on edit it sends to employee entry where we can edit and update the record

' when clicked on delete record get deleted

' when clicked on add it sends to employee entry to add a new record

'Dim c As ADODB.Connection

'Dim r As ADODB.Recordset

Option Explicit

Dim sql As String

Private Sub Command4\_Click()

Adodc1.Recordset.delete

End Sub

Private Sub Command1\_Click()

DataReport1.Show

End Sub

Private Sub close\_Click()

Unload Me

End Sub

Private Sub cmdprint\_Click()

If DataEnvironment1.rsCommand1.State = 1 Then DataEnvironment1.rsCommand1.close

DataEnvironment1.Command1 DataGrid1.Columns(0).Value

DataReport1.Show

End Sub

Private Sub delete\_Click()

Adodc1.Recordset.delete

End Sub

Private Sub Form\_Load()

Adodc1.Refresh

End Sub

Private Sub Select\_Click()

Create\_user.Text1.Text = DataGrid1.Columns(0).Value

Salary\_Entry.eid.Text = DataGrid1.Columns(0).Value

Salary\_Entry.ename.Text = DataGrid1.Columns(1).Value

Salary\_Entry.bsal.Text = DataGrid1.Columns(11).Value

Unload Me

End Sub

Private Sub Text1\_Change()

sql = "select \* from EMP\_ENTRY where name like '%" & Text1.Text & "%'"

If r.State = 1 Then r.close

r.Open sql, c

Set DataGrid1.DataSource = r

DataGrid1.Refresh

End Sub

Private Sub update\_Click()

Employee\_Registration.eid.Text = DataGrid1.Columns(0)

Employee\_Registration.ename.Text = DataGrid1.Columns(1)

Employee\_Registration.fname.Text = DataGrid1.Columns(2)

Employee\_Registration.address.Text = DataGrid1.Columns(3)

Employee\_Registration.dob.Value = DataGrid1.Columns(4)

Employee\_Registration.phone.Text = DataGrid1.Columns(5)

Employee\_Registration.gender.Caption = DataGrid1.Columns(6)

Employee\_Registration.doj.Value = DataGrid1.Columns(7)

Employee\_Registration.quali.Text = DataGrid1.Columns(8)

Employee\_Registration.depart.Text = DataGrid1.Columns(9)

Employee\_Registration.etype.Text = DataGrid1.Columns(10)

Employee\_Registration.salary.Text = DataGrid1.Columns(11)

Employee\_Registration.cmdUpdate.Visible = True

Employee\_Registration.cmdsave.Visible = False

Employee\_Registration.Show

End Sub

**Customer Entry**

Option Explicit

Dim que As String

Private Sub Cmdsave\_Click()

If cname.Text = "" Or address.Text = "" Or cphone.Text = "" Or email.Text = "" Then

MsgBox " Please Enter all Values"

Else

conn

sql = "insert into customer\_ent values ('" + cid.Text + "' , '" + cname.Text + "' ,'" + address.Text + "' , " + cphone.Text + " ,'" + email.Text + "','" + Format(cedate.Caption, "dd MMM yyyy") + "' , '" + cetime.Caption + "')"

Set r = c.Execute(sql)

MsgBox "record saved"

cid.Text = ""

cname.Text = ""

address.Text = ""

cphone.Text = ""

email.Text = ""

'cid.SetFocus

End If

End Sub

Private Sub Cmdclose\_Click()

Unload Me

End Sub

Private Sub cmdupdate\_Click()

conn

s = " update customer\_ent SET cusid='" + cid.Text + "', name='" + cname.Text + "', address='" + address.Text + "', phno= " + cphone.Text + ", email='" + email.Text + "', doe='" + Format(cedate.Caption, "dd MMM yyyy") + "',time= '" + cetime.Caption + "' where cusid='" + cid.Text + "'"

Set r = c.Execute(s)

MsgBox " Record Updated"

End Sub

Private Sub Form\_Load()

conn

que = "select count (cusid) from customer\_ent"

Set r = c.Execute(que)

cid.Text = r.Fields(0) + 1

End Sub

Private Sub Timer1\_Timer()

cedate.Caption = Date

cetime.Caption = Time

End Sub

**Add food**

Option Explicit

Dim m As String

Private Sub addml\_GotFocus()

tmpvar.Caption = addml.Text

End Sub

Private Sub Cmdsave\_Click()

On Error GoTo hell

If addml.Text = "" Or price.Text = "" Or mtype.Text = "" Or cate.Text = "" Then

MsgBox " Please Enter all Values"

Else

conn

sql = "insert into " + mtype.Text + " values ('" + addml.Text + "', " + price.Text + ", '" + cate.Text + "')"

Set r = c.Execute(sql)

MsgBox "record saved"

addml.Text = ""

price.Text = ""

cate.Text = ""

mtype.Text = ""

addml.SetFocus

End If

hell:

If Err.Number = -2147467259 Then MsgBox " Data overflow", vbExclamation, "Ok"

End Sub

Private Sub cmdupdate\_Click()

conn

s = " update " + mtype.Text + " SET mealnm='" + addml.Text + "', price=" + price.Text + ", cate='" + cate.Text + "' where mealnm = '" + tmpvar.Caption + "'"

Set r = c.Execute(s)

MsgBox " Record Updated"

End Sub

Private Sub Command2\_Click()

Unload Me

View\_Food.Show

End Sub

Private Sub price\_LostFocus()

Call onlynum(price)

End Sub

**Forgot password**

Dim c As New ADODB.Connection

Dim r As New ADODB.Recordset

Dim sql As String

Dim ans As String

Dim ques As String

Dim p As String

Option Explicit

Private Sub cmdgot\_Click()

On Error GoTo hell

If secq.Text = ques And answer.Text = ans Then

'If answer.Text = ans Then

MsgBox " Correct Paasword is :- " + p

Else

MsgBox "Wrong Answer Or Username"

'End If

End If

hell:

If Err.Number = -2147467259 Then MsgBox " Data overflow", vbExclamation, "Ok"

End Sub

Private Sub uname\_LostFocus()

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

sql = "select secques from user\_id where USERID='" + uname.Text + "'"

Set r = c.Execute(sql)

ques = r.Fields(0)

sql = "select answer from user\_id where USERID='" + uname.Text + "'"

Set r = c.Execute(sql)

ans = r.Fields(0)

sql = "select PASSWORD from user\_id where USERID='" + uname.Text + "'"

Set r = c.Execute(sql)

p = r.Fields(0)

End Sub

**Splash Form**

Option Explicit

Private Sub Form\_KeyPress(KeyAscii As Integer)

Unload Me

Log\_In.Show

End Sub

Private Sub Form\_Load()

'lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." & App.Revision

'lblProductName.Caption = App.Title

End Sub

Private Sub Frame1\_Click()

Unload Me

Log\_In.Show

End Sub

Private Sub Label1\_Click()

End Sub

Private Sub Timer1\_Timer()

ProgressBar1.Value = ProgressBar1.Value + 1

If ProgressBar1.Value = 100 Then

Unload Me

Log\_In.Show

End If

End Sub

**Log – In**

Option Explicit

Dim c As ADODB.Connection

Dim r As ADODB.Recordset

'Dim s As ADODB.Recordset

Dim sql As String

Dim p As String

Dim lbl As String

Dim lblcheck As String

Private Sub Cmdlogin\_Click()

If lblcheck = "Manager" Then

If pass.Text = p Then

Unload Me

MDIForm1.Show

Else

MsgBox "Wrong password"

End If

End If

If lblcheck = "Waiter" Then

If pass.Text = p Then

Unload Me

Call mdiodisable

MDIForm1.Show

MDIForm1.mnutblstatus.Enabled = True

Else

MsgBox "Wrong password"

End If

End If

If lblcheck = "chef" Then

If pass.Text = p Then

Unload Me

Call mdiodisable

MDIForm1.Show

MDIForm1.mnuaddfood.Enabled = False

MDIForm1.mnuViewfood.Enabled = False

Else

MsgBox "Wrong password"

End If

End If

End Sub

Private Sub Form\_Load()

End Sub

Private Sub Label5\_Click()

Forgot\_Password.Show vbModal

End Sub

Private Sub uid\_LostFocus()

On Error GoTo hell

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

sql = "select PASSWORD from user\_id where USERID='" + uid.Text + "'"

Set r = c.Execute(sql)

p = r.Fields(0)

lbl = "select USERLBL from user\_id where USERID='" + uid.Text + "'"

Set r = c.Execute(lbl)

lblcheck = r.Fields(0)

Exit Sub

hell:

If Err.Number = 3021 Then MsgBox "Please Enter a Correct Username First", vbExclamation, "Ok"

End Sub

**Mdi Form**

Option Explicit

Dim Cancel As Boolean

Private Sub Label1\_Click()

Entry\_form.Show

End Sub

Private Sub Label2\_Click()

Table\_Status.Show vbModal

End Sub

Private Sub Label3\_Click()

Report.Show

End Sub

Private Sub Label4\_Click()

conn

Dim msg As String

msg = MsgBox("Are you sure you want to Close the System?", vbYesNo, "Comfirm Exit")

If msg = vbYes Then

Call mdiform1exit

Unload Me

Else

Cancel = 1

End If

End Sub

Private Sub MDIForm\_Unload(Cancel As Integer)

conn

Dim msg As String

msg = MsgBox("Are you sure you want to Close the System?", vbYesNo, "Comfirm Exit")

If msg = vbYes Then

Call mdiform1exit

Else

Cancel = 1

End If

End Sub

Private Sub mnuAbout\_Click()

About.Show

End Sub

Private Sub mnuaddfood\_Click()

Food\_Entry.Show vbModal

End Sub

Private Sub mnucascade\_Click()

Me.Arrange vbCascade

End Sub

Private Sub mnucustentry\_Click()

Entry\_form.Show

End Sub

Private Sub mnucustsearch\_Click()

Customer\_Search.Show vbModal

End Sub

Private Sub mnuEmpatten\_Click()

Employee\_Attendence.Show vbModal

End Sub

Private Sub mnuempentry\_Click()

Employee\_Registration.Show

End Sub

Private Sub mnuempsearch\_Click()

Employee\_Search.Show

End Sub

Private Sub mnuExitDatabase\_Click()

Dim msg As String

msg = MsgBox("Are you sure you want to Close the System?", vbYesNo, "Comfirm Exit")

If msg = vbYes Then

Call mdiform1exit

End

Else

Cancel = 1

End If

End Sub

Private Sub mnulogOut\_Click()

Unload Me

Log\_In.Show

End Sub

Private Sub mnumaximized\_Click()

Me.WindowState = 2

End Sub

Private Sub mnumini\_Click()

Me.WindowState = 1

End Sub

Private Sub order\_Click()

Order.Show

End Sub

Private Sub mnuodrsearch\_Click()

Order\_search.Show

End Sub

Private Sub mnuorder\_Click()

Order.Show

End Sub

Private Sub mnupasschange\_Click()

Password\_Change.Show vbModal

End Sub

Private Sub mnuRegister\_Click()

Create\_user.Show vbModal

End Sub

Private Sub mnusalentry\_Click()

Salary\_Entry.Show vbModal

End Sub

Private Sub mnusalsearch\_Click()

Sal\_Search.Show

End Sub

Private Sub mnuSalsetting\_Click()

Salary\_Setting.Show vbModal

End Sub

Private Sub mnutblentry\_Click()

Table\_Entry.Show vbModal

End Sub

Private Sub mnutblstatus\_Click()

Table\_Status.Show vbModal

End Sub

Private Sub mnuViewfood\_Click()

View\_Food.Show vbModal

End Sub

Private Sub orderlbl\_Click()

Order.Show

End Sub

Private Sub Picture1\_Click()

Order.Show

End Sub

Private Sub Picture2\_Click()

Table\_Status.Show vbModal

End Sub

Private Sub Picture3\_Click()

Entry\_form.Show

End Sub

Private Sub Picture4\_Click()

Report.Show

End Sub

**Order**

Option Explicit

Dim que As String

Dim t As Single

Dim st As Single

Dim tot As String

Dim S1 As String

Dim TAX As Double

Private Sub Command1\_Click()

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

sql = "select \* from appetizers"

r.CursorLocation = adUseClient

r.CursorType = adOpenStatic

r.LockType = adLockOptimistic

r.Open sql, c, , , adCmdText

'Set r = c.Execute(sql)

Set DataGrid1.DataSource = r

End Sub

Private Sub Command11\_Click()

Adodc1.Recordset.delete

Adodc1.Refresh

End Sub

Private Sub Command12\_Click()

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

s = " update table\_entry SET status= 'Occupied' where table\_no = '" + Text4.Text + "'"

Set r = c.Execute(s)

MsgBox " Record Updated"

End Sub

Private Sub Command2\_Click()

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

sql = "select \* from beverages"

r.CursorLocation = adUseClient

r.CursorType = adOpenStatic

r.LockType = adLockOptimistic

r.Open sql, c, , , adCmdText

'Set r = c.Execute(sql)

Set DataGrid1.DataSource = r

End Sub

Private Sub Command3\_Click()

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

sql = "select \* from Soups\_Salad"

r.CursorLocation = adUseClient

r.CursorType = adOpenStatic

r.LockType = adLockOptimistic

r.Open sql, c, , , adCmdText

'Set r = c.Execute(sql)

Set DataGrid1.DataSource = r

End Sub

Private Sub Command4\_Click()

Table\_Status.close.Visible = False

Table\_Status.out.Visible = False

Table\_Status.select.Visible = True

Table\_Status.Show vbModal

' this botton redirect table no. to order form

End Sub

Private Sub Command5\_Click()

Customer\_Search.cmdupdate.Visible = False

Customer\_Search.cmddelete.Visible = False

Customer\_Search.Command2.Visible = False

Customer\_Search.select.Visible = True

Customer\_Search.Show vbModal

End Sub

Private Sub Command6\_Click()

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

sql = "select \* from Main\_Course"

r.CursorLocation = adUseClient

r.CursorType = adOpenStatic

r.LockType = adLockOptimistic

r.Open sql, c, , , adCmdText

'Set r = c.Execute(sql)

Set DataGrid1.DataSource = r

End Sub

Private Sub Command7\_Click()

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

sql = "select \* from desserts"

r.CursorLocation = adUseClient

r.CursorType = adOpenStatic

r.LockType = adLockOptimistic

r.Open sql, c, , , adCmdText

'Set r = c.Execute(sql)

Set DataGrid1.DataSource = r

End Sub

Private Sub Command8\_Click()

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

sql = "select \* from bar"

r.CursorLocation = adUseClient

r.CursorType = adOpenStatic

r.LockType = adLockOptimistic

r.Open sql, c, , , adCmdText

Set DataGrid1.DataSource = r

End Sub

Private Sub Command9\_Click()

t = Text6.Text \* DataGrid1.Columns(1).Value

Text10.Text = t

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

s = "insert into order\_qty values (" + order\_no.Caption + "," + Text6.Text + ",'" + DataGrid1.Columns(0).Value + "'," + DataGrid1.Columns(1).Value + "," + Text10.Text + ")"

Set r = c.Execute(s)

c.Execute "commit"

MsgBox s

Adodc1.RecordSource = "select \* from order\_qty where order\_no= " & order\_no.Caption & ""

Adodc1.Refresh

DataGrid2.Refresh

sql = "select sum(total) from order\_qty where order\_no like '%" & order\_no.Caption & "%'"

Set r = c.Execute(sql)

tot = r.Fields(0)

Text9.Text = tot

TAX = tot \* 15 / 100

Text8.Text = TAX

Text7.Text = Text9.Text + TAX

S1 = " update ordeer SET tot=" + Text9.Text + ", TX=" + Text8.Text + ", FTOT=" + Text7.Text + " where order\_no = '" & order\_no.Caption & "'"

Set r = c.Execute(S1)

End Sub

Private Sub Form\_Load()

conn

que = "select count (order\_no) from ordeer"

Set r = c.Execute(que)

order\_no.Caption = r.Fields(0) + 1

Adodc1.RecordSource = "select \* from order\_qty where order\_no=''"

Adodc1.Refresh

End Sub

Private Sub Text4\_LostFocus()

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

sql = "insert into ordeer values (" + order\_no.Caption + " , '" + Text1.Text + "' ,'" + Text2.Text + "' , '" + Text3.Text + "' , " + Text4.Text + "," + Text9.Text + "," + Text8.Text + "," + Text7.Text + ")"

Set r = c.Execute(sql)

End Sub

Private Sub Text9\_GotFocus()

End Sub

**Order search**

Option Explicit

Private Sub cmdprint\_Click()

If DataEnvironment1.rsCommand2.State = 1 Then DataEnvironment1.rsCommand2.close

DataEnvironment1.Command2 DataGrid1.Columns(0).Value

DataReport2.Show

End Sub

Private Sub Command2\_Click()

Unload Me

End Sub

Private Sub Text1\_Change()

conn

sql = "select \* from ordeer where order\_no like '%" & Text1.Text & "%'"

r.CursorLocation = adUseClient

r.CursorType = adOpenStatic

r.LockType = adLockOptimistic

r.Open sql, c, , , adCmdText

Set DataGrid1.DataSource = r

DataGrid1.Refresh

End Sub

**Change password**

Option Explicit

Dim p As String

Private Sub Command1\_Click()

On Error GoTo hell

If oldpass.Text = p Then

If newpass.Text = retype.Text Then

conn

s = " update user\_id SET password = '" + newpass.Text + "' where userid = '" + username.Text + "'"

Set r = c.Execute(s)

MsgBox " Record Updated "

End If

Else

MsgBox "WRONG PASSWORD"

End If

hell:

If Err.Number = -2147467259 Then MsgBox " Data overflow", vbExclamation, "Ok"

End Sub

Private Sub username\_LostFocus()

On Error GoTo hell

conn

sql = "select PASSWORD from user\_id where USERID='" + username.Text + "'"

Set r = c.Execute(sql)

p = r.Fields(0)

Exit Sub

hell:

If Err.Number = 3021 Then MsgBox "Please Enter a Correct Username First", vbExclamation, "Ok"

End Sub

**Report**

Option Explicit

Private Sub Command1\_Click()

If DataEnvironment1.rsCommand1.State = 1 Then DataEnvironment1.rsCommand1.close

DataEnvironment1.Command1 Text1.Text

DataReport1.Show

End Sub

Private Sub Command2\_Click()

If DataEnvironment1.rsCommand4.State = 1 Then DataEnvironment1.rsCommand4.close

DataEnvironment1.Command4 Text2.Text

DataReport3.Show

End Sub

Private Sub Command3\_Click()

If DataEnvironment1.rsCommand2.State = 1 Then DataEnvironment1.rsCommand2.close

DataEnvironment1.Command2 Text3.Text

DataReport2.Show

End Sub

Private Sub Form\_Load()

End Sub

**Salary search**

Option Explicit

Private Sub Command1\_Click()

conn

sql = "select \* from sal\_ent where empid like '%" & Text1.Text & "%' And TO\_CHAR(SALDATE,'MON')= '" + Combo1.Text + "' AND TO\_CHAR(SALDATE,'YYYY')= '" + Combo2.Text + "' "

r.CursorLocation = adUseClient

r.CursorType = adOpenStatic

r.LockType = adLockOptimistic

r.Open sql, c, , , adCmdText

Set DataGrid1.DataSource = r

DataGrid1.Refresh

End Sub

Private Sub Command2\_Click()

Unload Me

End Sub

Private Sub Command4\_Click()

If DataEnvironment1.rsCommand4.State = 1 Then DataEnvironment1.rsCommand4.close

DataEnvironment1.Command4 DataGrid1.Columns(1).Value

DataReport3.Show

End Sub

Private Sub Form\_Load()

End Sub

Private Sub Text1\_Change()

conn

sql = "select \* from sal\_ent where empid like '%" & Text1.Text & "%'"

r.CursorLocation = adUseClient

r.CursorType = adOpenStatic

r.LockType = adLockOptimistic

r.Open sql, c, , , adCmdText

Set DataGrid1.DataSource = r

DataGrid1.Refresh

End Sub

' sql = "select \* from sal\_ent where empid like '%" & Text1.Text & "%' And TO\_CHAR(SALDATE,'MON-YYYY')='JAN-2015'"

' "select \* from sal\_ent where empid like '%" & Text1.Text & "%' And TO\_CHAR(SALDATE,'MON')= '%" & Combo1.Text & "%' AND TO\_CHAR(SALDATE,'YYYY')= '%" & Combo2.Text & "%' "

' "select \* from sal\_ent where empid like '%" & Text1.Text & "%' And TO\_CHAR(SALDATE,'MON')= ' " + Combo1.Text + " ' AND TO\_CHAR(SALDATE,'YYYY')= ' " + Combo2.Text + " ' "

' "select \* from sal\_ent where empid like '%" & Text1.Text & "%' And TO\_CHAR(SALDATE,'MON-YYYY')= ' " + Combo1.Text + " - " + Combo2.Text + " ' "

**Salary Entry**

Option Explicit

Dim varta As Double

Dim varda As Double

Dim varhra As Double

Dim varoa As Double

Dim pf As Double

Dim bs As Double

Dim ta As Double

Dim da As Double

Dim hra As Double

Dim oa As Double

Dim tot As Double

Private Sub cmdgotoesearch\_Click()

Employee\_Search.update.Visible = False

Employee\_Search.delete.Visible = False

Employee\_Search.close.Visible = False

Employee\_Search.select.Visible = True

Employee\_Search.Show vbModal

End Sub

Private Sub cmdremove\_Click()

End Sub

Private Sub cmdupdate\_Click()

conn

s = " update sal\_set SET saldate='" + Format(payperiod.Value, "dd MMM yyyy") + "', empid='" + eid.Text + "' , name = '" + ename.Text + "' , ba=" + bsal.Text + ", ta=" + trall.Text + ", da=" + docall.Text + ", hra=" + hourall.Text + ", oa=" + oall.Text + ", pf=" + pfund.Text + ",total=" + pamount.Text + ""

Set r = c.Execute(s)

MsgBox " Record Updated"

End Sub

Private Sub Cmdsave\_Click()

conn

sql = "insert into SAL\_ENT values ('" + Format(payperiod.Value, "dd MMM yyyy") + "', '" + eid.Text + "' , '" + ename.Text + "' ," + bsal.Text + " , " + trall.Text + " , " + docall.Text + " , " + hourall.Text + " , " + oall.Text + " , " + pfund.Text + " ," + pamount.Text + ")"

Set r = c.Execute(sql)

MsgBox "record saved"

eid.Text = ""

ename.Text = ""

bsal.Text = ""

trall.Text = ""

docall.Text = ""

hourall.Text = ""

oall.Text = ""

pfund.Text = ""

pamount.Text = ""

End Sub

Private Sub Command10\_Click()

Unload Me

End Sub

Private Sub Command2\_Click()

End Sub

Private Sub Command9\_Click()

End Sub

Private Sub Form\_Load()

payperiod.Value = Date

eid.Text = ""

ename.Text = ""

bsal.Text = ""

End Sub

Private Sub trall\_GotFocus()

conn

sql = "select \* from sal\_set"

Set r = c.Execute(sql)

varta = r.Fields(0)

varda = r.Fields(1)

varhra = r.Fields(2)

varoa = r.Fields(3)

pf = r.Fields(4)

bs = bsal.Text

ta = bs \* varta / 100

da = bs \* varda / 100

hra = bs \* varhra / 100

oa = bs \* varoa / 100

tot = (bs + ta + da + hra + oa) - pf

trall.Text = ta

docall.Text = da

hourall.Text = hra

oall.Text = oa

pfund.Text = pf

pamount.Text = tot

End Sub

**Salary Setting**

Option Explicit

Private Sub Command1\_Click()

On Error GoTo hell

conn

s = " update sal\_set SET TA=" + Text1.Text + ", DA=" + Text2.Text + ", HRA=" + Text3.Text + ", OA=" + Text4.Text + ", PF=" + Text5.Text + ""

Set r = c.Execute(s)

MsgBox " Record Updated"

hell:

If Err.Number = -2147467259 Then MsgBox " Data overflow", vbExclamation, "Ok"

End Sub

Private Sub Form\_Load()

conn

sql = "select \* from sal\_set"

Set r = c.Execute(sql)

Text1.Text = r.Fields(0)

Text2.Text = r.Fields(1)

Text3.Text = r.Fields(2)

Text4.Text = r.Fields(3)

Text5.Text = r.Fields(4)

End Sub

Private Sub Text1\_LostFocus()

Call onlynum(Text1)

End Sub

Private Sub Text2\_LostFocus()

Call onlynum(Text2)

End Sub

Private Sub Text3\_LostFocus()

Call onlynum(Text3)

End Sub

Private Sub Text4\_LostFocus()

Call onlynum(Text4)

End Sub

Private Sub Text5\_LostFocus()

Call onlynum(Text5)

End Sub

**Table entry**

Option Explicit

Private Sub Command1\_Click()

conn

sql = "insert into table\_entry values ('" + Text1.Text + "','','')"

Set r = c.Execute(sql)

MsgBox "record saved"

Text1.Text = ""

Text1.SetFocus

End Sub

**Table status**

Option Explicit

Private Sub Form\_Load()

End Sub

Private Sub out\_Click()

conn

s = " update table\_entry SET status= 'Available' where table\_no = '" + DataGrid1.Columns(0) + "'"

Set r = c.Execute(s)

MsgBox " Record Updated"

Adodc1.Refresh

DataGrid1.Refresh

End Sub

Private Sub Select\_Click()

Order.Text4.Text = DataGrid1.Columns(0).Value

Unload Me

Order.Show

Order.Text4.SetFocus

End Sub

Private Sub close\_Click()

Unload Me

End Sub

**View food**

Option Explicit

Private Sub Combo1\_LostFocus()

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

sql = "select \* from " + Combo1.Text + " "

r.CursorLocation = adUseClient

r.CursorType = adOpenStatic

r.LockType = adLockOptimistic

r.Open sql, c, , , adCmdText

Set DataGrid1.DataSource = r

End Sub

Private Sub Command1\_Click()

Unload Me

Food\_Entry.Show

End Sub

Private Sub Command2\_Click()

Food\_Entry.addml.Text = DataGrid1.Columns(0).Value

Food\_Entry.price.Text = DataGrid1.Columns(1).Value

Food\_Entry.mtype.Text = Combo1.Text

Food\_Entry.cate.Text = DataGrid1.Columns(2).Value

Food\_Entry.cmdsave.Visible = False

Unload Me

Food\_Entry.Show

End Sub

Private Sub Form\_Load()

End Sub

Private Sub Text1\_Change()

conn

sql = "select \* from " + Combo1.Text + " where mealnm like '%" & Text1.Text & "%'"

r.CursorLocation = adUseClient

r.CursorType = adOpenStatic

r.LockType = adLockOptimistic

r.Open sql, c, , , adCmdText

Set DataGrid1.DataSource = r

DataGrid1.Refresh

End Sub

**Module 1**

Public c As New ADODB.Connection

Public r As New ADODB.Recordset

Public s As String

Public sql As String

Public Sub conn()

Set c = New ADODB.Connection

c.Open "Provider=MSDAORA.1;User ID=demo/project;Persist Security Info=True"

Set r = New ADODB.Recordset

End Sub

Public Sub mdiform1exit()

Unload Entry\_form

Unload Order

Unload Customer\_Search

Unload Log\_In

Unload Employee\_Registration

Unload Employee\_Search

Unload Employee\_Attendence

Unload Salary\_Entry

Unload Sal\_Search

Unload Food\_Entry

Unload View\_Food

Unload Table\_Status

Unload Create\_user

Unload Salary\_Setting

Unload Table\_Entry

Unload Forgot\_Password

Unload Report

Unload Password\_Change

Unload About

Unload Order\_search

End Sub

Public Sub mdiodisable()

MDIForm1.mnuorder.Enabled = False

MDIForm1.mnucustentry.Enabled = False

MDIForm1.mnucustsearch.Enabled = False

MDIForm1.mnuempentry.Enabled = False

MDIForm1.mnuempsearch.Enabled = False

MDIForm1.mnusalentry.Enabled = False

MDIForm1.mnuEmpatten.Enabled = False

MDIForm1.mnuSalsetting.Enabled = False

MDIForm1.mnuRegister.Enabled = False

MDIForm1.mnutblentry.Enabled = False

MDIForm1.mnuaddfood.Enabled = False

MDIForm1.mnuViewfood.Enabled = False

MDIForm1.mnutblstatus.Enabled = False

End Sub

'Option Explicit

Public Sub onlynum(tb As TextBox)

If Not IsNumeric(tb.Text) Then

MsgBox "Please enter numeric value only.", vbInformation

tb.Text = ""

Cancel = True

tb.SetFocus

End If

End Sub

Public Sub kypress(tb As TextBox)

If KeyAscii = 13 Or 32 Then

tb.SetFocus

End If

End Sub

# PIE CHART

The Pie chart shows the consumption of time by various phases of the System Development Life Cycle of ***“Resturent Management System”***.

**BIBLIOGRAPHY**

1. **MASTERING VISUAL BASIC 6**

**-EVANGELOS PETROUTSOS**

1. **VISUAL BASIC IN 12 EASY LESSONS**

**-GREG PERRY**

1. **VISUAL BASIC PROGRAMMING**

**(BLACK BOOK)**

**-STEVEN HOLZENER**

1. **DATABASE DEVELOPER’S GUIDE**

**WITH VISUAL BASIC 5**

**-ROGER JENNINGS**