

**Project Advisor**

................................................................

**G.Hema Kumar (Guide)**

**M.Tech., M.C.A., M.Com.,**

Lecturer in Computers

Department of Computers

Sri Sai Krishna @ SMT.Theresa Arts & Science Degree College, Kurnool.

**Project Examiner**

...................................................... ..................................................

**Examiner 1 Examiner 1**

Department of Computers Department of Computers

**ACKNOWLEDGEMENTS**

I would like to express my deepest sense of gratitude to my reverend teacher and supervisors **Mr. G. Hema Kumar**, Lecturer, Department of Computer Science, Sri Sai Krishna @ SMT. Theresa Arts & Science Degree College, Kurnool, and, for her untiring guidance, constant supervision, enthusiastic encouragement, sagacious advice and an effective surveillance throughout the entire period of my project & thesis work and preparation of the manuscript. I greatly say thank you. Wish to express my heart full thanks to all of my honorable lecturers of the Department of Computer Science, Sri Sai Krishna @ SMT. Theresa Arts & Science Degree College, Kurnool.

I am thankful to Mr. **G. Hema Kumar**, who is having a vast knowledge of Ms Access, System Analysis and Design & Programming concepts which are the building block of project.

I also want to express my appreciation to my classmates and friends who helped me in one way or another during the course of developing this project. They endured the long hours of my absence during the development of this project.

I also wish to thanks to the members of **our college**, for providing me some details for books software specifications and for their cordial behavior during my project & thesis work.

I would like to extend my gratitude to **Mr. S.M. Jhony Basha garu**, Principal of Sri Sai Krishna Degree College, Kurnool for their cooperation and valuable support in every way during the period of the project.

Finally, I thank the almighty-God, from the depth of my heart, which has been the unfailing source of strength, comfort & inspiration in the completion of this project work.

1. **K. Mahesh babu**

**(H.T. No :22497009019)**

Dept. of Computer Science

**Sri Sai Krishna @ SMT.Theresa Arts & Science Degree College, Kurnool.**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No** | **Title** | **Page No** |
| **INTRODUCTION** | | |
| **1** | 1.1 Synopsis | **1** |
| 1.2 Aim of this project | **2** |
| 1.3 Main Purpose | **2** |
| **SYSTEM ENVIRONMENT** | | |
| **2** | 2.1 Hardware Configuration | **3** |
| 2.2 Software Configuration | **3** |
| 2.3 Software Features | **3 - 26** |
| **SYSTEM DESIGN** | | |
| **3** | 3.1 Input Design | **27** |
| 3.2 Process Design | **28** |
| 3.3 Database Design | **28 - 30** |
| 3.4 Output Design | **31** |
| **4** | **SOURCE CODE** | **32 - 42** |
| **5** | **OUTPUT SCREENS** | **43 - 48** |
| **IMPLEMENTATION** | | |
| **6** | 6.1 User Training | **47** |
| 6.2 Security and Maintenance | **49** |
| **7** | **CONCLUSION** | **50** |
| **8** | **BIBLIOGRAPHY** | **51** |

****

**Introduction**

**CHAPTER 1**

**CHAPTER 1**

**INTRODUCTION**

In the dynamic and fast phased world of hospitality, efficient management is Essential for success. Introducing our restaurant management system, designed to streamline operations, enhance customer experience, and empower restaurant owners under staff with powerful tools for success.Our restaurant management system leverages the latest technology to provide a comprehensive solution tailored to the unique needs of restaurants of all sizes. With the user friendly interface and institute design, our system is easy to navigate for both front of house and back of house staff ensuring seamless integration into daily workflows. Whether you are a small independent E3 or a large chain restaurant, our restaurant management system is customization to suit your specific requirements.

* 1. **Synopsis**

This project is intended as restaurant management system, especially developed for restaurants to manage their billing system. This project is designed using Ms. Access as the front end and Ms. Access as a back end. Its platform is Windows XP,7,8,9,10.Modular design: other than the basic fields, user will have choice to include as many items in their menu. Based on the menu in that restaurant, one could decide on the options.Relational database: Ms. Access database is used at the back end for data management has powerful features of data management to access the data efficiently.

Client/server technology: And hence the performance of the application windows environment.

**1.2 Aim of this project**

The aim of the restaurant management system project it’s 2 develop a comprehensive software solution that enhances the efficiency and effectiveness of restaurant operations. The key object is include enhanced billing experience efficient resource utilization and scalability and flexibility. By achieving these aims, the restaurant management system aims to empower restaurant owners and managers with the tools and insights they need to bill in their business more efficiently.

**1.3 Main purpose**

A big boom has been witnessed in this system in recent times. A large number of new players have entered the market and are trying to gain market share in this rapidly improving market. The study deals with economic survey in focus and the various segments that it caters to. The study then goes on to evaluate and analyze the findings so as to present a clear picture of trends in the finance section.



**SYSTEM ENVIRONMENT**

**CHAPTER 2**

**SYSTEM ENVIRONMENT**

**CHAPTER 2**

**2.1 Hardware Configuration**

Processor : Intel Pentium III or later

Main Memory(RAM) : 256 MB

Cache Memory : 512 KB

Monitor : 14 in color monitor

Keyboard : 108 keys

Mouse : Optical Mouse

Hard Disk : 160 GB

**2.2 Software Configuration**

Front End/Language : Access Form

Back End/Database : Access Database

Operating systems : Windows 7, 8, 10, XP

**2.3 Software Features**

**Introduction to MS Access:**

Microsoft Access is an information management tool that helps you store information for reference, reporting, and analysis.  Microsoft Access helps you analyze large amounts of information, and manage related data more efficiently than Microsoft Excel or other spreadsheet applications.  This article shows you when to use Access, and how it can help make you more effective.

A quick review of basic database terms such as table, record, field, etc and an overview of the MS-Access DBMS. Describes the MS-Access program window as well as the purpose and characteristics of the common MS-Access database objects, including queries, forms, and reports.

Earlier MS-Access versions used a specialized dialogue box called the Database Window to show database objects such as tables, queries, etc. When Microsoft introduced the ribbon in MS-Access 2007, it also replaced the Database Window with the Navigation Pane, which performs the same function.

The Navigation pane displays each Database object (tables, queries, etc.) as an aggregate. The data content of an object is displayed in the larger Contents pane to the right of the Navigation Pane. The Contents pane becomes tabbed as multiple database objects are opened. The Navigation pane can be collapsed in order to grow the Contents pane and display more data.

**Store Related Information Together, Eliminate Duplication**

The value any database can provide is to store related information in one place, and then let you connect various different things together (sometimes called "entities" in database speak).  You store one version of the truth for any given thing, like a client, a DVD, or an order.  Some of the major benefits to using a database are:

1. **Fewer errors and inconsistencies.**Maintaining one version of the truth for the things you need to track minimizes the potential for duplication, errors, and inconsistent values.  In the Excel example above, customer names, contact phone numbers, and product names are misspelled, transposed, and abbreviated in different forms.  Imagine trying to create a report or graph in Excel that shows an accurate picture of your sales or other key performance indicators.
2. **Higher productivity.** You and others working in your database need only change a single record and all other related things in your database will automatically "see" the change.   In our Excel example, you would need to change every occurrence of a customer's name.  Find and replace wouldn't necessarily catch that a customer's name is spelled six different ways.  In a database, there is only one customer record to change.  Every other record (orders, contacts, etc) that relates to that customer record will never need to change because they are joined to the customer.
3. **Security and Control**.  Databases provide a central location to store, secure, and control your data.  Microsoft Access includes the ability to encrypt and password protect database files.  And with users relying on a database rather than separate Excel spreadsheets, you can have greater control over access to the information.
4. **Better Decisions and Insight**.  Perhaps most importantly, a single source of truth in a standardized format means you can gain better insight and make better decisions by reporting and analyzing your data in a database.  Users are familiar with Excel charting, and databases like Microsoft Access can provide similar capabilities, while going beyond Excel with more powerful historical trending, aggregation, and query filtering.

**Features of Ms Access**

* It allows us to create the framework (forms, tables and so on) for storing information in a database.
* Microsoft Access allows opening the table and scrolling through the records contained within it.
* Microsoft Access forms provide a quick and easy way to modify and insert records into your databases.
* Microsoft Access has capabilities to answer more complex requests or queries.
* Access queries provide the capability to combine data from multiple tables and place specific conditions on the data retrieved.
* Access provides a user-friendly forms interface that allows users to enter information in a graphical form and have that information transparently passed to the database.

Microsoft Access has the look and feel of other Microsoft Office products as far as its layout and navigational aspects are concerned, but MS Access is a database and, more specifically, a relational database.

* Before MS Access 2007, the file extension was \*.mdb, but in MS Access 2007 the extension has been changed to \*.accdb extension.
* Early versions of Access cannot read accdb extensions but MS Access 2007 and later versions can read and change earlier versions of Access.
* An Access desktop database (.accdb or .mdb) is a fully functional RDBMS.
* It provides all the data definition, data manipulation, and data control features that you need to manage large volumes of data.
* You can use an Access desktop database (.accdb or .mdb) either as a standalone RDBMS on a single workstation or in a shared client/server mode across a network.
* A desktop database can also act as the data source for data displayed on webpages on your company intranet.
* When you build an application with an Access desktop database, Access is the RDBMS.

**Data Definition**

Let us now understand what Data Definition is −

* In document or a spreadsheet, you generally have complete freedom to define the contents of the document or each cell in the spreadsheet.
* In a document, you can include paragraphs of text, a table, a chart, or multiple columns of data displayed with multiple fonts.
* In spreadsheet, you can have text data at the top to define a column header for printing or display, and you might have various numeric formats within the same column, depending on the function of the row.
* An RDBMS allows you to define the kind of data you have and how the data should be stored.
* You can also usually define rules that the RDBMS can use to ensure the integrity of your data.
* For example, a validation rule might ensure that the user can’t accidentally store alphabetic characters in a field that should contain a number.

**Data Manipulation**

Working with data in RDBMS is very different from working with data in a word processing or spreadsheet program.

* In a word processing document, you can include tabular data and perform a limited set of functions on the data in the document.
* You can also search for text strings in the original document and, with ActiveX controls, include tables, charts, or pictures from other applications.
* In a spreadsheet, some cells contain functions that determine the result you want, and in other cells, you enter the data that provides the source information for the functions.

An RDBMS provides you many ways to work with your data. For example,

* You can search a single table for information or request a complex search across several related tables.
* You can update a single field or many records with a single command.
* You can write programs that use RDBMS commands to fetch data that you want to display and allow the user to update the data.

Access uses the powerful SQL database language to process data in your tables. Using SQL, you can define the set of information that you need to solve a particular problem, including data from perhaps many tables.

**Data Control**

Spreadsheets and word processing documents are great for solving single-user problems, but they are difficult to use when more than one person needs to share the data.

* When you need to share your information with others, RDBMS gives you the flexibility to allow multiple users to read or update your data.
* An RDBMS that is designed to allow data sharing also provides features to ensure that no two people can change the same data at the same time.
* The best systems also allow you to group changes (which is also known as transaction) so that either all the changes or none of the changes appear in your data.
* You might also want to be sure that no one else can view any part of the order until you have entered all of it.
* Because you can share your Access data with other users, you might need to set some restrictions on what various users are allowed to see or update.

**Tables in Ms Access:**

When you create an Access database, you store your data in tables. Subject-based lists that contain rows and columns. For instance, you can create a Contacts table to store a list of names, addresses, and telephone numbers, or a Products table to store information about products.

Other database objects depend so heavily on tables, you should always start your design of a database by creating all of its tables and then creating any other objects. Before you create tables, carefully consider your requirements and determine all the tables that you need. For an introduction to planning and designing a database

A table is a database object that you use to store data about a particular subject, such as employees or products. A table consists of records and fields.

Each record contains data about one instance of the table subject, such as a particular employee. A record is also commonly called a row or an instance.

Each field contains data about one aspect of the table subject, such as first name or email address. A field is also commonly called a column or an attribute.

A record consists of field values, such as mokshesh, or mokshesh@gmail.com. A field value is also commonly called a fact.

Tables are used to store a collection of related records. Recall that a record is the collection of information from all fields which pertains to a single entity (a person, for example), and that fields are the areas within a database which hold a specific type of information (name, date of birth, height, for example).

There are two specialized views of tables:

* Design View
* Datasheet View

The Design view is used to create or modify a table's structure. A table's structure is defined by the size and data type of its fields. Modifying a table structure means to add or delete fields and make changes to their properties.

The Datasheet view, in which field contents are shown in columns and records are shown as rows, is used to display a table's data.

Datasheet view is also used to maintain data. Maintaining data means adding or deleting records and editing the information in their fields.

**Queries in Ms Access:**

Queries are the basis of power in a database. They give you the ability to ask questions, record the questions for later, and to take actions on the answers.

Queries are the second structure in Access. Tables hold the information, queries contain stored questions

When you want to review, add, change, or delete data from your database consider using a query.

Using a query, you can answer very specific questions about your data that would be difficult to answer by looking at table data directly. You can use queries to filter your data, to perform calculations with your data, and to summarize your data. You can also use queries to automate many data management tasks and to review changes in your data before you commit to those changes.

A query is a request for data results, for action on data, or for both. You can use a query to answer a simple question, to perform calculations, to combine data from different tables, or even to add, change, or delete table data. Queries that you use to retrieve data from a table or to make calculations are called select queries. Queries that add, change, or delete data are called action queries.

You can also use a query to supply data for a form or report. In a well-designed database, the data that you want to present by using a form or report is often located in several different tables. By using a query, you can assemble the data that you want to use before you design your form or report.

Queries are used work selectively with table data. Examples of selective data would include only employees who work in one particular department, or only those customers who reside in a particular state or postal code. There are different types of queries.

Select queries select only records which match one or more selection criteria. Select queries can display information contained in fields from one or multiple related tables. They can also perform and display calculations based on table data, averages, for example.

An update query is used to make changes to fields and records based upon a selection criteria. For example: increase wages by ten percent for all employees whose pay rate falls below a certain level.

A delete query is used to automatically delete records which match a selection criteria. For example, a delete query could be used to delete all customers whose most recent order was placed more than two years ago.

Like tables, queries are created and modified in design view, while their results are displayed in datasheet view. Queries can also serve as the basis for Forms and Reports.

**Relationships in Ms Access:**

Building tables in [Access 2007](http://blog.pluralsight.com/microsoft-office-access) is fairly simple. Creating relationships between these tables is a different story.

This is the part that most of us get confused with because Access relationships (as any other relationships) can get quite complicated.

And let's face it; we all get a little confused when it comes to Access. But don't worry, help is on the way!

**Forms in Ms Access:**

In Access 2007, a form is an object that generally serves three purposes:

1. To allow users to perform data entry. Data can be inserted, updated, or deleted from a table using a Form object.
2. To allow users to enter custom information, and based on that information perform a task. For example, you may want to ask a user for parameters before running a report.
3. To allow users a method of navigating through the system. For example, you may create a form where a user can select a form to load, a report to run, etc.

A form in Access is a database object that you can use to create a user interface for a database application. A "bound" form is one that is directly connected to a data source such as a table or query, and can be used to enter, edit, or display data from that data source. Alternatively, you can create an "unbound" form that does not link directly to a data source, but which still contains command buttons, labels, or other controls that you need to operate your application.

You can use bound forms to control access to data, such as which fields or rows of data are displayed. For example, certain users might need to see only several fields in a table with many fields. Providing those users with a form that contains only those fields makes it easier for them to use the database. You can also add command buttons and other features to a form to automate frequently performed actions.

Think of bound forms as windows through which people see and reach your database. An effective form speeds the use of your database, because people don't have to search for what they need. A visually attractive form makes working with the database more pleasant and more efficient, and it can also help prevent incorrect data from being entered.

Another way to populate a database is with the use of **forms**. An Access 2007 form helps you know exactly what data to enter. In this lesson, we'll address the **benefits of using forms** with a database, and we'll show you how to **set up a basic form** for your Access 2007 database, as well as how to **use the form** to populate or edit data in the database. You'll also learn how to **enhance a basic form with a drop-down list**.

Access 2007 forms tools include:

* The Form command makes a basic form, showing a single record at a time.
* The Split Form command creates a form showing one record on top and includes the Datasheet view of the entire source table on the bottom.
* The Multiple Items command creates a form that shows all records at once, which looks similar to the source table in Datasheet view.
* The Form Wizard is hidden under the More Forms command. It walks you through the process of creating more customized forms.

### Text box Control:

The text box is the standard control used for viewing and editing data on forms and reports. Many different types of data can be displayed in text boxes, and you can also use them to perform calculations.

A bound text box displays data from a field in a table or query. On a form, you can use a text box that is bound to an updatable record source to enter or edit data in a field. The changes that you make in the text box will be reflected in the underlying table.

An unbound text box is not connected to a field in a table or query. You can use an unbound text box to display the results of a calculation or to accept input that you don't want to store directly in a table. It is easiest to add an unbound text box in Design view.

**Label Control:**

Label controls are used to display text and cannot be edited by the user. They are used to identify objects on a form — provide a description of what a certain control will do if clicked, for example — or at run time, they can display information in response to an event or process in your application.

When we create a report in MS Access 2007, it will add labels to each control box; it places labels on page header when controls are in detail section. We need the labels in report header not in page header, if we delete or cut the label from page header and place is to report header the association with control has been lost.

If you create a text box in the detail section, it makes a label next to it. If you then go to the control layout, and click tabular, it moves the label to the page header, with the text box control to the detail section.

**Command Button Control:**

You use a command button on an Access form to start an action or a set of actions. For example, you can create a command button that opens another form. To make a command button perform an action, you write a macro or event procedure and attach it to the command button's On Click property. You can also embed a macro directly into the On Click property of the command button. This makes it possible to copy the button to other forms without losing the functionality of the button.

By using the Command Button Wizard, you can quickly create command buttons that do a variety of tasks, such as closing the form, opening a report, finding a record, or running a macro.

If you have already created and saved a macro, you can easily create a command button that runs the macro by dragging the macro from the Navigation Pane to a form that is open in Design view.

### Drop-down list or Combo box:

Using a drop-down list on a form can increase the integrity of the data in the database because drop-down lists force form users to select one of the preset options in the list to populate the field. These types of form controls are relatively easy to set up using the Combo Box.

When entering data on forms in Access desktop databases, it can be quicker and easier to select a value from a list than to remember a value to type. A list of choices also helps ensure that the value entered in a field is appropriate. A list control can connect to existing data, or it can display fixed values that you enter when you create the control. Read on to learn about the list controls available for Access forms, and how to create and customize them.

The combo box control provides a more compact way to present a list of choices; the list is hidden until you click the drop-down arrow. A combo box also gives you the ability to enter a value that is not in the list. In this way, the combo box control combines the features of a text box and a list box.

**List Box Control:**

The ListBox control on forms is a very powerful control that is used for inputting and presenting data. The ListBox control is a very versatile control in that it can be used to display a list of data to select from. This data can come from various sources: A Table, a Query or a Static List.

The Microsoft Access List Box Database template is a Microsoft Access Database that shows you how to implement some of the basic features of the List box in your database. The features included in this template database are listed below.

The list box control displays a list of values or choices. The list box contains rows of data, and is usually sized so that several rows are visible at all times. The rows can have one or more columns, which can appear with or without headings. If the list has more rows than can be displayed in the control, Access displays a scroll bar in the control. The user is limited to the choices given in the list box; it is not possible to type a value into a list box.

**Subform/Subreport Control:**

When you are working with relational data (related data that is stored in separate tables), you often need to view multiple tables or queries on the same form. For example, you might want to see customer data from one table and information about customer orders from another table at the same time. Subforms are a convenient tool for doing this, and Access provides several ways to help you create subforms quickly.

**VBA (Visual Basic for Applications) in Access:**

When you create a new database, you typically begin by creating several database objects such as tables, forms, and reports. Eventually, you reach a point where you have to add some programming to automate certain processes and tie your database objects together. This article helps orient you to the programming tools in Access.

In Access, programming is the process of adding functionality to your database by using Access macros or Visual Basic for Applications (VBA) code. For example, suppose that you have created a form and a report, and you want to add a command button to the form that, when clicked, opens the report. Programming, in this case, is the process of creating a macro or VBA procedure and then setting the command button's **OnClick** event property so that clicking the command button runs the macro or procedure. For a simple operation, such as opening a report, you can use the Command Button Wizard to do all the work, or you can turn off the wizard and do the programming yourself.

Objects (such as forms and reports) and controls (such as command buttons and text boxes) have various event properties to which you can attach macros or procedures. Each event property is associated with a specific event, such as clicking the mouse, opening a form, or modifying data in a text box. Events can also be triggered by factors outside of Access, such as system events, or by macros or procedures that are attached to other events. Your database can get complex if you add many macros or procedures to several event properties of many objects, but in most cases, you can achieve the results that you want by using very little programming.

**VBA Data Types**

When you declare a variable, you should also identify its data type. You're probably already very familiar with data types because you assign data types to table fields. VBA uses the same data types to define a variable.

The most important job of a data type is to ensure the validity of your data. Specifying a data type won't keep you from entering an invalid value, but it will keep you from entering an invalid type. If you omit the data type, VBA applies the Variant data type to your variable—it's the most flexible and VBA won't guess at what the data type should be.

### The Byte Data Type

Byte is VBA's smallest numeric data type and holds a numeric value from 0 to 255. This data type doesn't include any negative values. If you attempt to assign one, VBA returns an error.

### The Currency Data Type

Use the Currency numeric data type to store monetary values from –922,337,203,477.5808 to 922,337,203,685,477.5807. A Currency data type results in a scaled value with accuracy to 15 digits to the left of the decimal point and 4 digits to the right. Use this data type to avoid rounding errors when precision is of the utmost importance.

### The Date Data Type

The Date data type stores a specially formatted numeric value that represents both the date and time. You don't have to store both the date and time value. The Date data type accepts either the date or the time, or both. Possible values range from January 1, 100 to December 31, 9999.

### The Decimal Data Type

The Decimal data type is a sub type of Variant and not a truly separate data type all its own, accommodating values from - 79,228,162,514,264,337,593,543,950,335 to 79,228,162,514,264,337,593,543,950,335 if the value contains no decimal places. The data type maintains precision up to 28 decimal places with values from -7.922816251426433759354395 0335 to 7.9228162514264337593543950335.

### The Double Data Type

Use the Double data type to store precision floating point numbers from –1.797693 13486232E308 to –4.94065645841247E-324 or 1.79769313486232E308 to 4.9406564584124 7E -324.

### The Integer Data Type

This is probably the most common data type in use, besides String. Use this data type to store only whole numbers that range from –32,768 to 32,767.

### The Long Data Type

The Long data type is also an Integer data type storing only whole numbers, but the range is much larger than the traditional Integer data type. Use Long to store values from –2,147,483,648 to 2,147,486,647.

### The Object Data Type

An Object variable is actually a reference to an Access object, such as a form, report, or control. Or, the data type can reference an ActiveX component, or a class object created in a class module.

Class modules are covered briefly in "Introducing the VBA Modules," in Chapter 2 and in more depth in "Introducing Objects".

### The Single Data Type

The Single data type stores precision numbers—numbers with decimal places or fractional numbers. The data type is similar to Double, but the range is smaller. Use this data type to store values from –3402823E38 to –1.401298E–45 or from 1.401298E–45 to 3.402823E38.

### The String Data Type

String is another very common data type; it stores values or numbers, but treats them as text. There are two varieties: fixed and variable. A fixed string can handle from 1 to 65,400 characters. To declare a fixed string, use the Dim statement in the form

### The Variant Data Type

The Variant data type stores numeric and non-numeric values. This data type is the most flexible of the bunch because it stores very large values of almost any type (matches the Double numeric data type). Use it only when you're uncertain of the data's type or when you're accommodating foreign data and you're not sure of the data type's specifications.

**VBA Conditional Statements**

The If...Then statement examines the truthfulness of an expression. Structurally, its formula is:

**Syn:** If Condition Then Statement

The program will examine the Condition. This condition can be a simple expression or a combination of expressions. If the Condition is true, then the program will execute the Statement.

**If..Then..Else Statement**

The If...Then statement offers only one alternative: to act if the condition is true. Whenever you would like to apply an alternate expression in case the condition is false, use the If...Then...Else statement. The formula of this statement is:

If ConditionIsTrue Then

     Statement1

Else

     Statement2

End If

The condition, in this ConditionIsTrue, would be examined. If it produces a true result, then the first statement, in this case Statement1, would be executed. If the condition (ConditionIsTrue) is false, the second statement, in this case Statement2, would be executed

**If..Then..Elseif Statement**

The If...Then...ElseIf statement acts like the If...Then...Else, except that it offers as many choices as necessary. The formula is:

If Condition1 Then

    Statement1

ElseIf Condition2 Then

    Statement2

ElseIf Condition\_n Then

    Statement\_n

End If

The program will first examine the first condition, in this case Condition1. If Condition1 is true, then the program would execute the first statement, in this case Statment1, and stop examining conditions. But if Condition1 is false, then the program would examine Condition2 and act accordingly. Whenever a condition is false, the program would continue examining the conditions until it finds one that is true. Once a true condition has been found and its statement executed, the program would terminate the conditional examination at End If.

**VBA Loop Statements**

There may be a situation when you need to execute a block of code several number of times. In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on.

Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times.

**Do .. While loop**

A Do…While loop is used when we want to repeat a set of statements as long as the condition is true. The condition may be checked at the beginning of the loop or at the end of the loop.

**Syn**: do while condition

Statements

Loop

**For loop**

A for loop is a repetition control structure that allows a developer to efficiently write a loop that needs to be executed a specific number of times.

**Syn**: for counter = start to end [step step count]

Statements

Next

**For each loop**

A For Each loop is used to execute a statement or a group of statements for each element in an array or collection.

A For Each loop is similar to For Loop; however, the loop is executed for each element in an array or group. Hence, the step counter won't exist in this type of loop. It is mostly used with arrays or used in context of the File system objects in order to operate recursively.

**Syn**: for each element in group

Statements

Next

**While … wend loop**

In a While…Wend loop, if the condition is True, all the statements are executed until the Wend keyword is encountered.If the condition is false, the loop is exited and the control jumps to the very next statement after the Wend keyword.

**Syn**: while Condition(s)

Statement

Wend

**Do … until loop**

A Do…Until loop is used when we want to repeat a set of statements as long as the condition is false. The condition may be checked at the beginning of the loop or at the end of loop.

**Syn**: do until condition

Statements

Loop

**Exit for statement**

A Exit For statement is used when we want to exit the For Loop based on certain criteria. When Exit For is executed, the control jumps to the next statement immediately after the For Loop.

**Syn**: Exit for

**Exit Do statement**

An Exit Do Statement is used when we want to exit the Do Loops based on certain criteria. It can be used within both Do…While and Do...Until Loops.When Exit Do is executed, the control jumps to the next statement immediately after the Do Loop.

**Syn**: Exit Do

**User-Defined VBA Function & Sub Procedures**

In Excel Visual Basic, a set of commands to perform a specific task is placed into a procedure, which can be a Function procedure or a Sub procedure (also known as functions and subroutines).

The main difference between a VBA Function procedure and a Sub procedure is that a Function procedure returns a result, whereas a Sub procedure does not.

Therefore, if you wish to perform a task that returns a result (e.g. summing of a group of numbers), you will generally use a Function procedure, but if you just need a set of actions to be carried out (e.g. formatting a set of cells), you might choose to use a Sub procedure.

**Arguments**

VBA procedures can be passed data via arguments, which are declared in the procedure definition. For example, you could have a VBA Sub procedure that adds an Integer to every cell in the current selected range on a worksheet. You could supply the value of the integer to the Sub via an argument, as follows:

Sub Add To Cells(i As Integer)

**.**  
**.**  
**.**  
End Sub

### Optional Arguments

You can also define VBA procedures to have Optional arguments. These are arguments that the user can supply if they want, but if they are omitted, the procedure will assign a default value to them.

To return to the example above, if we wanted to make the supplied integer argument optional, this would be declared as follows:

Sub Add To Cells(Optional i As Integer = 0)

In this case, the supplied integer, i, has a default value of 0.

You can use multiple Optional arguments in a VBA procedure, as long the Optional arguments are all positioned at the end of the argument list.

### Passing Arguments By Value and By Reference

When arguments are passed to VBA procedures, they can be passed in two ways:

* ByVal - The argument is passed by Value. This means that just the value (i.e. a copy of the argument) is passed to the procedure and therefore, any changes that are made to the argument inside the procedure will be lost when the procedure is exited.
* ByRef - The argument is passed by Reference. This means that the address of the argument is passed to the procedure. Any changes that are made to the argument inside the procedure will be remembered when the procedure is exited.

**VBA Function Procedures**

The VBA editor recognizes a Function procedure, because the commands are positioned between the following start and end commands:

Function

**.**  
**.**  
**.**

End Function

As previously mentioned, VBA function procedures (unlike sub procedures) return a value. The return values have the following rules:

* The data type of the returned value must be declared in the Function header.
* The value to be returned must be assigned to a variable having the same name as the Function. This variable does not need to be declared, as it already exists as a part of the function.

### Calling VBA Function Procedures

If the above simple Function procedure is typed into a Module in the Visual Basic Editor, it will then be available to be called from other VBA procedures or to be used in the worksheets of your Excel workbook.

#### Calling a VBA Function Procedure From Within VBA

You can call a Function procedure from within your VBA program by simply assigning the Function to a variable.

**Events in VBA**

VBA, an event-driven programming can be triggered when you change a cell or range of cell values manually. Change event may make things easier, but you can very quickly end a page full of formatting. There are two kinds of events.

* Worksheet Events
* Workbook Events

**Worksheet Events**

* Worksheet Events are triggered when there is a change in the worksheet. It is created by performing a right-click on the sheet tab and choosing 'view code', and later pasting the code.
* The user can select each one of those worksheets and choose "Work Sheet" from the drop down to get the list of all supported Worksheet events.

**Workbook Events**

* Workbook events are triggered when there is a change in the workbook on the whole. We can add the code for workbook events by selecting the 'ThisWorkbook' and selecting 'workbook' from the dropdown as shown in the following screenshot. Immediately Workbook\_open sub procedure is displayed to the user as seen in the following screenshot.

**What is an event?**

Events are, well events that are associated with the objects in your development framework. In this case that is Access. In Access only the user interface elements (forms and report) have events. But the objects from additional libraries you can add as references may have events as well. An event occurs when a button is clicked, when a form is loaded or a textbox is updated with a new value.

There are hundreds of predefined events available in Access. And these event occurs nonstop during an access application is used by a user. While these events occur all the time they do not have any effect on your application unless you handle them. You can use either a predefined macro or you can write VBA code to handle an event. The procedures that are tied to events in VBA are called event procedures. The macro or code you bound to an event is executed each time the event occurs.

**Form Events:**

**On Open Event**

The Open Event occurs when a form is opened, but before the first record is displayed. Therefore, attach code here that you wish to run as soon as the form is opened. The Open event will not occur when you activate (move to a previously opened form), i.e. if you open a second form from the first form, then close the second form, the first form’s On Open Event will not occur as the first form has not been closed and re-opened, it has just been hidden behind the second form.

If the Form is based on a Query, the Query is run prior to the On Open Event.

**On Load Event**

Whereas the on Open Event occurs when the form is opened and before the first record is displayed, the On Load Event occurs when the first record is displayed.

**On Resize Event**

The Resize Event occurs when a form is opened or whenever the form’s size changes.

**On Activate Event**

The Activate Event occurs when a form receives the focus and becomes the active window. You make a form active by: Opening it; Clicking on form with your mouse, or clicking a control on the form; Invoking the Set Focus method. The On Activate event can only occur if the form is visible. If the form is not visible an error will occur. On Got Focus Event this rarely used Event only occurs when the form gets focus, but only if there are no visible enabled controls on the form. Which is unlikely.

**On Current Event**

The On Current event occurs when the focus moves to a new or different record making it the current record, or when the Form is Refreshed or Required.

This Event occurs when a form is opened, whenever the focus leaves one record and moves to another, and when the Form’s underlying Table or Query is required. This event is one of the more commonly used Events. If you wish to run code whenever a record is displayed, this is the place to put it.

**Feasibility Study:**

**Economic Feasibility**

Economic analysis is most frequently used for evaluation of the effectiveness of the system. More commonly knows as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions is made to design and implement the system. This part of feasibility study gives the top management the economic justification for the new system. This is an important input to the management the management, because very often the top management does not like to get confounded by the various technicalities that bound to be associated with a project of this kind. A simple economic analysis that gives the actual comparison of costs and benefits.

**Technical Feasibility**

Technical feasibility centers on the existing manual system of the test management process and to what extent it can support the system. According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs are identified. It is also one of the important phases of the system development activities. The system offers greater levels of user friendliness combined with greater processing speed. Therefore, the cost of maintenance can be reduced. Since, processing speeds very high and the work is reduced in the maintenance point of view management convince that the project is operationally feasible.

**Behavioral Feasibility**

People are inherently resistant to change and computer has been known to facilitate changes. An estimate should be made of how strong the user is likely to move towards the development of computerized system. These are various levels of users in order to ensure proper authentication and authorization and security of sensitive data of the organization.



**SYSTEM DESIGN**

**CHAPTER 3**

**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 Input Design**

The system design is divided in to two portions. The Administrator section and the User (Customer) section.

**Administrator**

1. The Administrator can allot IDs to the different products.
2. He can delete the product details.
3. He can control the product list.
4. He can edit the details of the products. He can change their details, edit and delete the product records.

A process of converting user originated inputs to a computer-based format. Input design is an important part of development process since inaccurate input data are the most common cause of errors in data processing. Erroneous entries can be controlled by input design. It consists of developing specifications and procedures for entering data into a system and must be in simple format. The goal of input data design is to make data entry as easy, logical and free from errors as possible. In input data design, we design the source document that capture the data and then select the media used to enter them into the computer.

There are two major approaches for entering data in to the computer. They are

* Menus.
* Dialog Boxes.

**Menus**

A menu is a selection list that simplifies computer data access or entry. Instead of remembering what to enter, the user chooses from a list of options. A menu limits a user choice of response but reduce the chances for error in data entry.

**Dialog Box**

Dialog boxes are windows and these windows are mainly popup, which appear in response to certain conditions that occur when a program is run. It allows the display of bitmaps and pictures. It can have various controls like buttons, text boxes, list boxes and combo boxes. Using these controls we can make a ‘dialog’ with the program.

The proposed system has three major inputs. They are Machine Registration, Machine Scheduling and Request Form.

**3.2 Process Design**

Process design plays an important role in project development. In order to understand the working procedure, process design is necessary. Data Flow Diagram and System Flow chart are the tools used for process design.

System Flow Chart is a graphical representation of the system showing the overall flow of control in processing at the job level; specifies what activities must be done to convert from a physical to logical model.

Data Flow Diagram is the logical representation of the data flow of the project. The DFD is drawn using various symbols. It has a source and a destination. The process is represented using circles and source and destination are represented using squares. The data flow is represented using arrows.

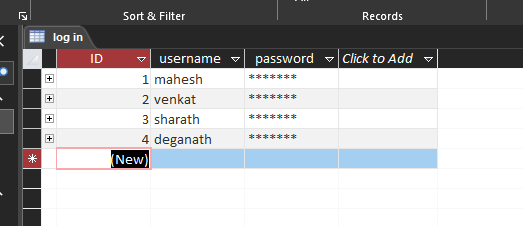
**3.3 Database Design**

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

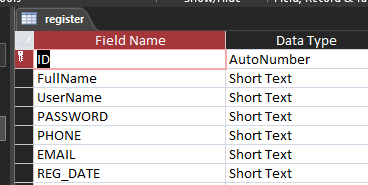
A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and aximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases.

The following are the tables that are involved in the proposed system

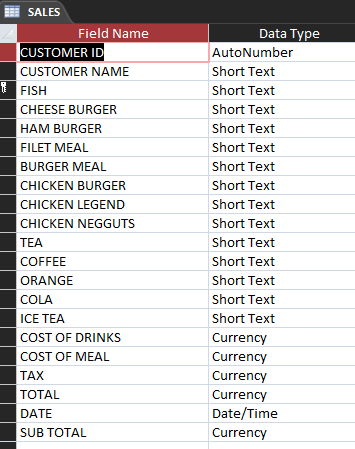
**Table: Admin log in**

****

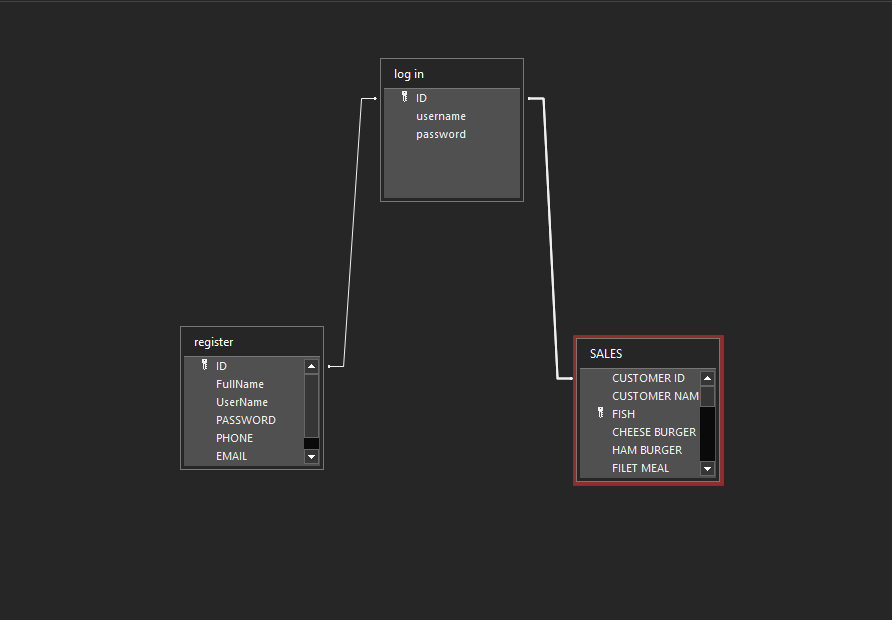
**Table : Register**

****

**Table : Sales**

****

**Tables :Relationships**

****

**3.4 Output Design**

Designing computer output should proceed in an organized, well throughout manner; the right output element is designed so that people will find the system whether or executed. When we design an output we must identify the specific output that is needed to meet the system. The usefulness of the new system is evaluated on the basis of their output.

Once the output requirements are determined, the system designer can decide what to include in the system and how to structure it so that the require output can be produced. For the proposed software, it is necessary that the output reports be compatible in format with the existing reports. The output must be concerned to the overall performance and the system’s working, as it should. It consists of developing specifications and procedures for data preparation, those steps necessary to put the inputs and the desired output, ie maximum user friendly. Proper messages and appropriate directions can control errors committed by users.

The output design is the key to the success of any system. Output is the key between the user and the sensor. The output must be concerned to the system’s working, as it should.

Output design consists of displaying specifications and procedures as data presentation. User never left with the confusion as to what is happening without appropriate error and acknowledges message being received. Even an unknown person can operate the system without knowing anything about the system.

****

****

**CHAPTER 4**

**SOURCE CODE**

**CHAPTER 4**

**SOURCE CODE**

Option Compare Database

Option Explicit

Dim firstnum As Double

Dim secondnum As Double

Dim answer As Double

Dim opera As String

Dim m(25) As Double 'meal

**DIABLE RIBBON(BOTTUN):-**

Private Sub btndisrib\_Click()

DoCmd.ShowToolbar "Ribbon", acToolbarNo

End Sub

**ENABLE RIBBON(BOTTUN):-**

Private Sub btnenbrib\_Click()

DoCmd.ShowToolbar "Ribbon", acToolbarYes

End Sub

**EXIT(BUTTON):-**

Private Sub btnExit\_Click()

Dim iExit As VbMsgBoxResult

iExit = MsgBox("Confirm if you want to exit", vbQuestion + vbYesNo, "Resturant Management System")

If iExit = vbYes Then

DoCmd.Close acForm, Me.Name

End If

End Sub

**COST OF ALL ITEMS(BUTTON):-**

Private Sub txtcost\_Click()

Dim Food\_Tax As Double

Food\_Tax = 0.35

m(0) = txtfish \* 50

m(1) = txtcheebur \* 50

m(2) = txthambur \* 50

m(3) = txtfiletmeal \* 60

m(4) = txtburgmeal \* 45

m(5) = txtchikenbur \* 70

m(6) = txtchickenleg \* 70

m(7) = txtchinuge \* 80

m(8) = txttea \* 10

m(9) = txtcoffee \* 10

m(10) = txtcola \* 20

m(11) = txtorg \* 30

m(12) = txticetea \* 40

m(13) = m(0) + m(1) + m(2) + m(3) + m(4) + m(5) + m(6) + m(7)

m(14) = m(8) + m(9) + m(10) + m(11) + m(12)

numericcod = m(14)

numericcom = m(13)

m(15) = ((m(13) + m(14)) \* Food\_Tax) / 100

m(16) = m(13) + m(14) + m(15)

numerictax = m(15)

numericSubtotal = m(13) + m(14)

numerictotal = m(16)

End Sub

**0(BUTTON):-**

Private Sub btn0\_Click()

If lblDisplay.Caption = "0" Then

lblDisplay.Caption = "0"

Else

lblDisplay.Caption = lblDisplay.Caption + "0"

End If

End Sub

**1(BUTTON):-**

Private Sub btn1\_Click()

If lblDisplay.Caption = "0" Then

lblDisplay.Caption = "1"

Else

lblDisplay.Caption = lblDisplay.Caption + "1"

End If

End Sub

**2(BUTTON):-**

Private Sub btn2\_Click()

If lblDisplay.Caption = "0" Then

lblDisplay.Caption = "2"

Else

lblDisplay.Caption = lblDisplay.Caption + "2"

End If

End Sub

**3(BUTTON):-**

Private Sub btn3\_Click()

If lblDisplay.Caption = "0" Then

lblDisplay.Caption = "3"

Else

lblDisplay.Caption = lblDisplay.Caption + "3"

End If

End Sub

**4(BUTTON):-**

Private Sub btn4\_Click()

If lblDisplay.Caption = "0" Then

lblDisplay.Caption = "4"

Else

lblDisplay.Caption = lblDisplay.Caption + "4"

End If

End Sub

**5(BUTTON):-**

Private Sub btn5\_Click()

If lblDisplay.Caption = "0" Then

lblDisplay.Caption = "5"

Else

lblDisplay.Caption = lblDisplay.Caption + "5"

End If

End Sub

**6(BUTTON):-**

Private Sub btn6\_Click()

If lblDisplay.Caption = "0" Then

lblDisplay.Caption = "6"

Else

lblDisplay.Caption = lblDisplay.Caption + "6"

End If

End Sub

**7(BUTTON):-**

Private Sub btn7\_Click()

If lblDisplay.Caption = "0" Then

lblDisplay.Caption = "7"

Else

lblDisplay.Caption = lblDisplay.Caption + "7"

End If

End Sub

**8(BUTTON):-**

Private Sub btn8\_Click()

If lblDisplay.Caption = "0" Then

lblDisplay.Caption = "8"

Else

lblDisplay.Caption = lblDisplay.Caption + "8"

End If

End Sub

**9(BUTTON):-**

Private Sub btn9\_Click()

If lblDisplay.Caption = "0" Then

lblDisplay.Caption = "9"

Else

lblDisplay.Caption = lblDisplay.Caption + "9"

End If

End Sub

**ADDITION(BUTTON):-**

Private Sub btnadd\_Click()

firstnum = lblDisplay.Caption

lblDisplay.Caption = ""

opera = "+"

End Sub

**ADDANDSUB(BUTTON):-**

Private Sub btnaddsub\_Click()

lblDisplay.Caption = -1 \* CDb1(lblDisplay.Caption)

End Sub

**SUBTRACTION(BUTTON):-**

Private Sub btnbs\_Click()

Dim strText As String

strText = lblDisplay.Caption

strText = Left(strText, Len(strText) - 1)

lblDisplay.Caption = strText

End Sub

**CANCEL(BUTTON)**:-

Private Sub btncan\_Click()

lblDisplay.Caption = "0"

End Sub

**COS(BUTTON):-**

Private Sub btncos\_Click()

lblDisplay.Caption = Format(Sin(lblDisplay.Caption), "##0.#########")

End Sub

**DIVISON(BUTTON):-**

Private Sub btndiv\_Click()

firstnum = lblDisplay.Caption

lblDisplay.Caption = ""

opera = "/"

End Sub

**DOT(BUTTON):-**

Private Sub btndot\_Click()

If InStr(lblblDisplay.Caption, ".") = 0 Then

lblDisplay.Caption = lblDisplay.Caption + "."

End If

End Sub

**MOD(BUTTON):-**

Private Sub btnmod\_Click()

firstnum = lblDisplay.Caption

lblDisplay.Caption = ""

opera = "mod"

End Sub

**MULTIPLICATION(BUTTON):-**

Private Sub btnmutl\_Click()

firstnum = lblDisplay.Caption

lblDisplay.Caption = ""

opera = "\*"

End Sub

**RESULT(BUTTON):-**

Private Sub btnres\_Click()

secondnum = lblDisplay.Caption

If opera = "+" Then

answer = firstnum + secondnum

lblDisplay.Caption = answer

ElseIf opera = "-" Then

answer = firstnum - secondnum

lblDisplay.Caption = answer

ElseIf opera = "\*" Then

answer = firstnum \* secondnum

lblDisplay.Caption = answer

ElseIf opera = "/" Then

answer = firstnum / secondnum

lblDisplay.Caption = answer

ElseIf opera = "mod" Then

answer = firstnum Mod secondnum

lblDisplay.Caption = answer

End If

End Sub

**SIN(BUTTON):-**

Private Sub btnsin\_Click()

lblDisplay.Caption = Format(Sin(lblDisplay.Caption), "##0.#########")

End Sub

**SQUARE(BUTTON):-**

Private Sub btnsqr\_Click()

lblDisplay.Caption = CDb1(lblDisplay.Caption) ^ 2

End Sub

**SUBTRACTION(BUTTON):-**

Private Sub btnsub\_Click()

firstnum = lblDisplay.Caption

lblDisplay.Caption = ""

opera = "-"

End Sub

**TAN(BUTTON):-**

Private Sub btntan\_Click()

lblDisplay.Caption = Format(Sin(lblDisplay.Caption), "##0.#########")

End Sub

**ALL ITEAM AFTER UPDATE SOURCE CODES :-**

Private Sub txtburgmeal\_AfterUpdate()

If Not IsNumeric(txtburgmeal.Value) Then

txtburgmeal = ""

End If

End Sub

Private Sub txtcheebur\_AfterUpdate()

If Not IsNumeric(txtcheebur.Value) Then

txtcheebur = ""

End If

End Sub

Private Sub txtchickenleg\_AfterUpdate()

If Not IsNumeric(txtchickenleg.Value) Then

txtchickenleg = ""

End If

End Sub

Private Sub txtchikenbur\_AfterUpdate()

If Not IsNumeric(txtchikenbur.Value) Then

txtchikenbur = ""

End If

End Sub

Private Sub txtchinuge\_AfterUpdate()

If Not IsNumeric(txtchinuge.Value) Then

txtchinuge = ""

End If

End Sub

Private Sub txtcoffee\_AfterUpdate()

If Not IsNumeric(txtcoffee.Value) Then

txtcoffee = ""

End If

End Sub

Private Sub txtcola\_AfterUpdate()

If Not IsNumeric(txtcola.Value) Then

txtcola = ""

End If

End Sub

Private Sub txtfiletmeal\_AfterUpdate()

If Not IsNumeric(txtfiletmeal.Value) Then

txtfiletmeal = ""

End If

End Sub

Private Sub txtfish\_AfterUpdate()

If Not IsNumeric(txtfish.Value) Then

txtfish = ""

End If

End Sub

Private Sub txthambur\_AfterUpdate()

If Not IsNumeric(txthambur.Value) Then

txthambur = ""

End If

End Sub

Private Sub txticetea\_AfterUpdate()

If Not IsNumeric(txticetea.Value) Then

txticetea = ""

End If

End Sub

Private Sub txtorg\_AfterUpdate()

If Not IsNumeric(txtorg.Value) Then

txtorg = ""

End If

End Sub

Private Sub txttea\_AfterUpdate()

If Not IsNumeric(txttea.Value) Then

txttea = ""

End If

End Sub

**ADMIN FORM :-**

Option Compare Database

**CANCEL(BUTTON):-**

Private Sub btncancel\_Click()

Dim iExit As VbMsgBoxResult

iExit = MsgBox("Confirm if you want to Cancel", vbQuestion + vbYesNo, "Resturant Management System")

If iExit = vbYes Then

DoCmd.Close acForm, Me.Name

End If

End Sub

**LOG IN(BUTTON):-**

Private Sub cmdlog\_Click()

DoCmd.Close

DoCmd.OpenForm "rest"

End Sub

**REGISTER(BUTTON):-**

Private Sub btnregis\_Click()

DoCmd.Close

DoCmd.OpenForm "register form"

End Sub

**REGISTER FORM :-**

Option Compare Database

**CANCEL(BUTTON):-**

Private Sub cmd\_cancel\_Click()

DoCmd.Close

DoCmd.OpenForm "admin\_form"

End Sub

**REGISTER(BUTTON):-**

Private Sub cmd\_register\_Click()

**REGISTER(BUTTON):-**

DoCmd.Save

DoCmd.GoToRecord , , acNewRec

DoCmd.Close

DoCmd.OpenForm "admin\_form"

End Sub

Private Sub Form\_Open(Cancel As Integer)

DoCmd.GoToRecord , , acNewRec

End Sub

**CUSTOMER FORM:-**

**CLOSE(BUTTON):-**

Private Sub btnclose\_close()

DoCmd.close

DoCmd.OpenForm "rest"

End Sub



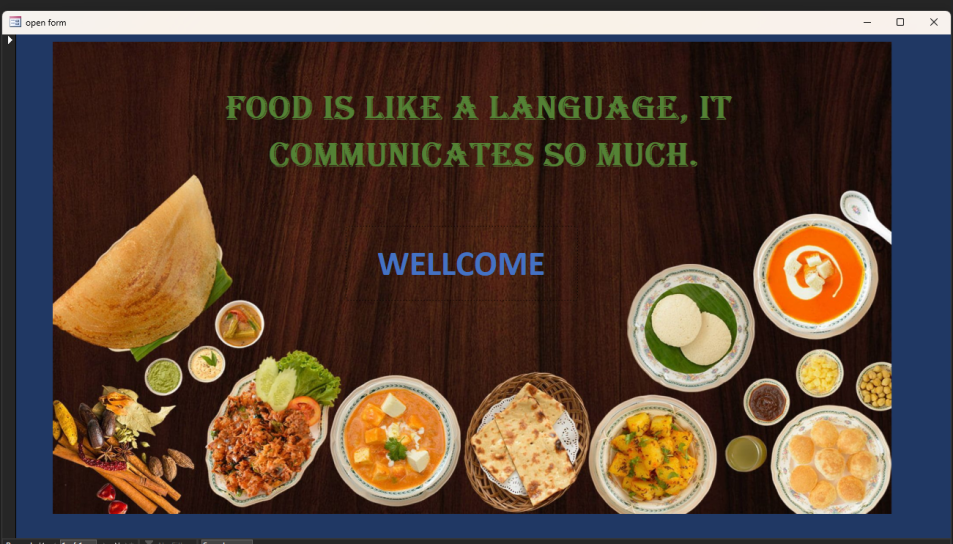
**CHAPTER 5**

**OUTPUT SCREENS**

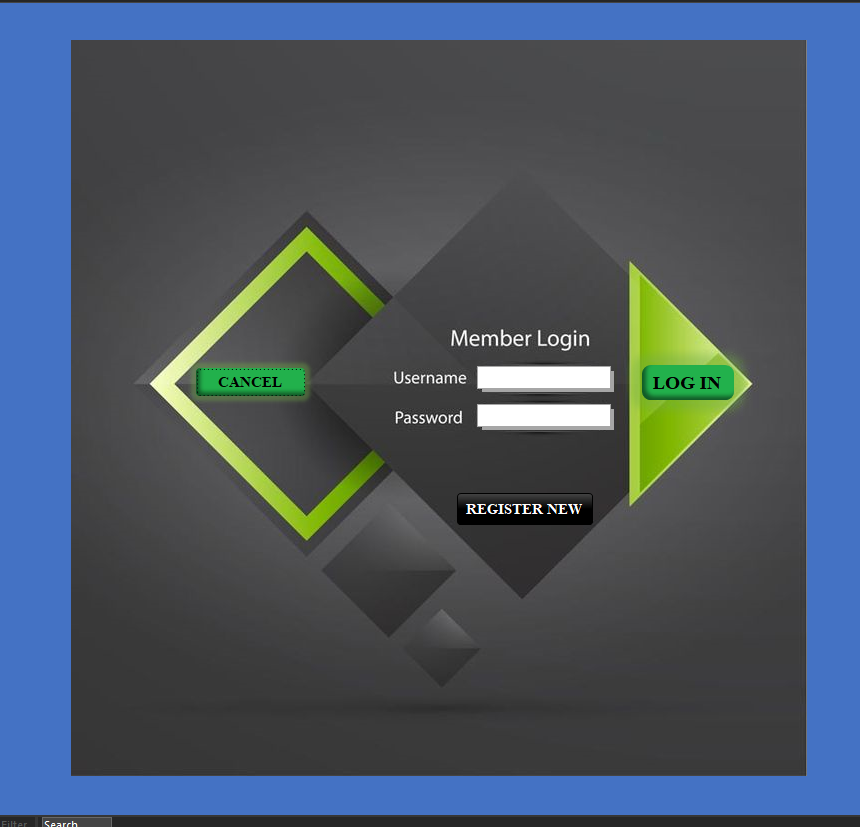
**CHAPTER 5**

**OUTPUT SCREENS**

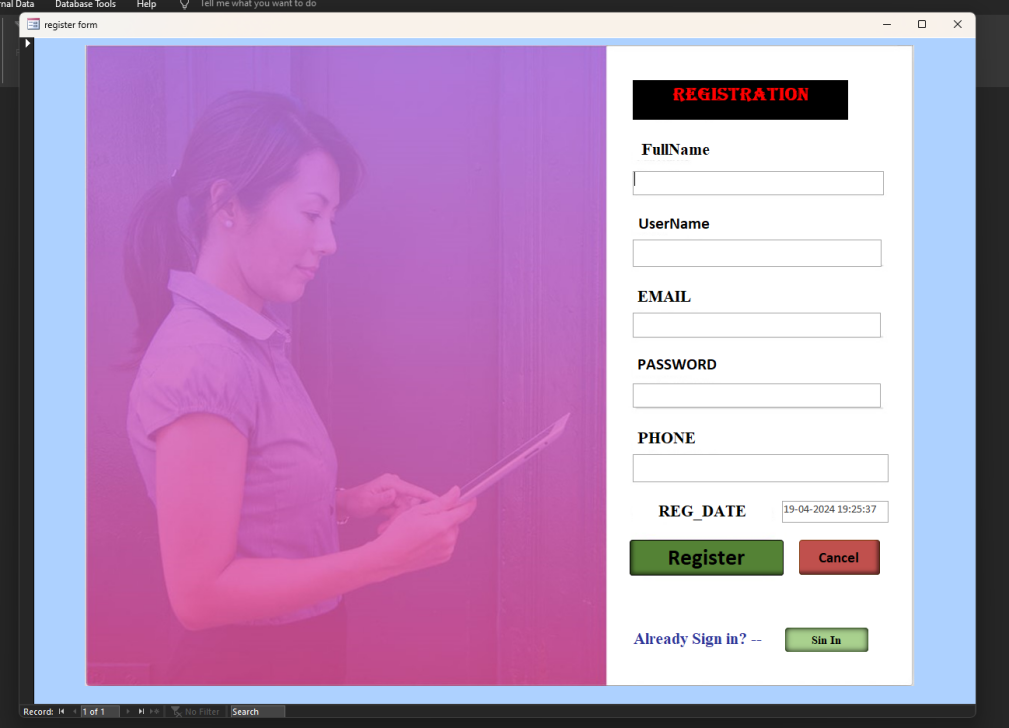
**Restaurant welcome form :**

****

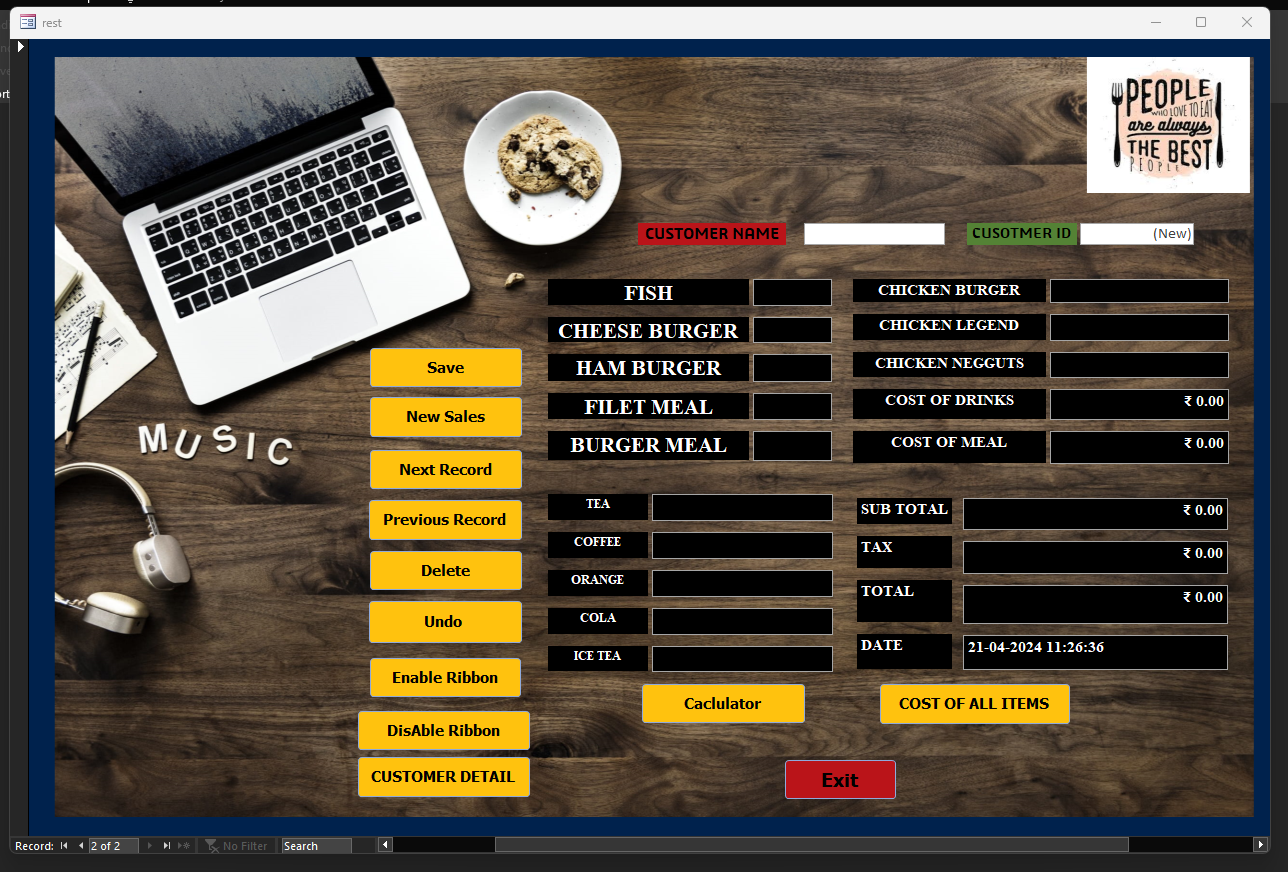
**Login output screen :**

****

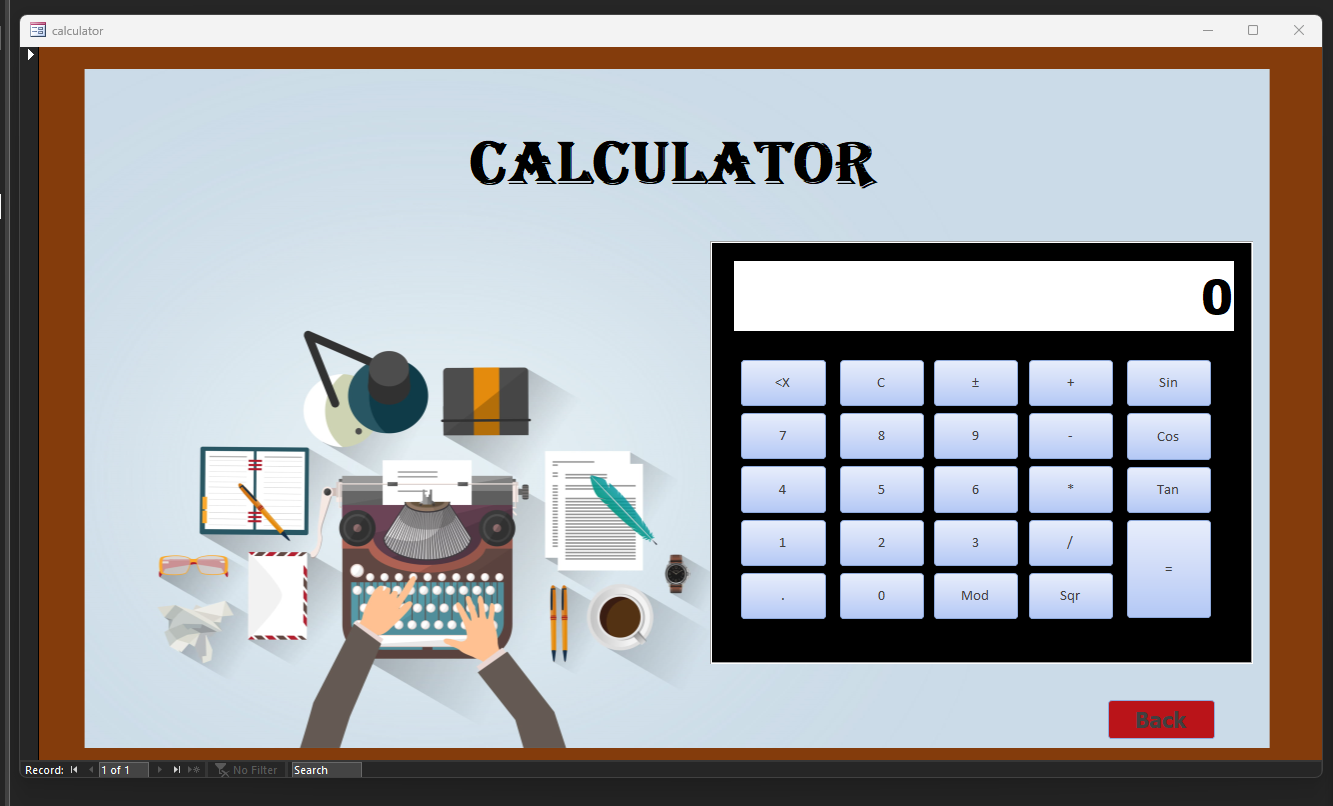
**Register form output screen :**

****

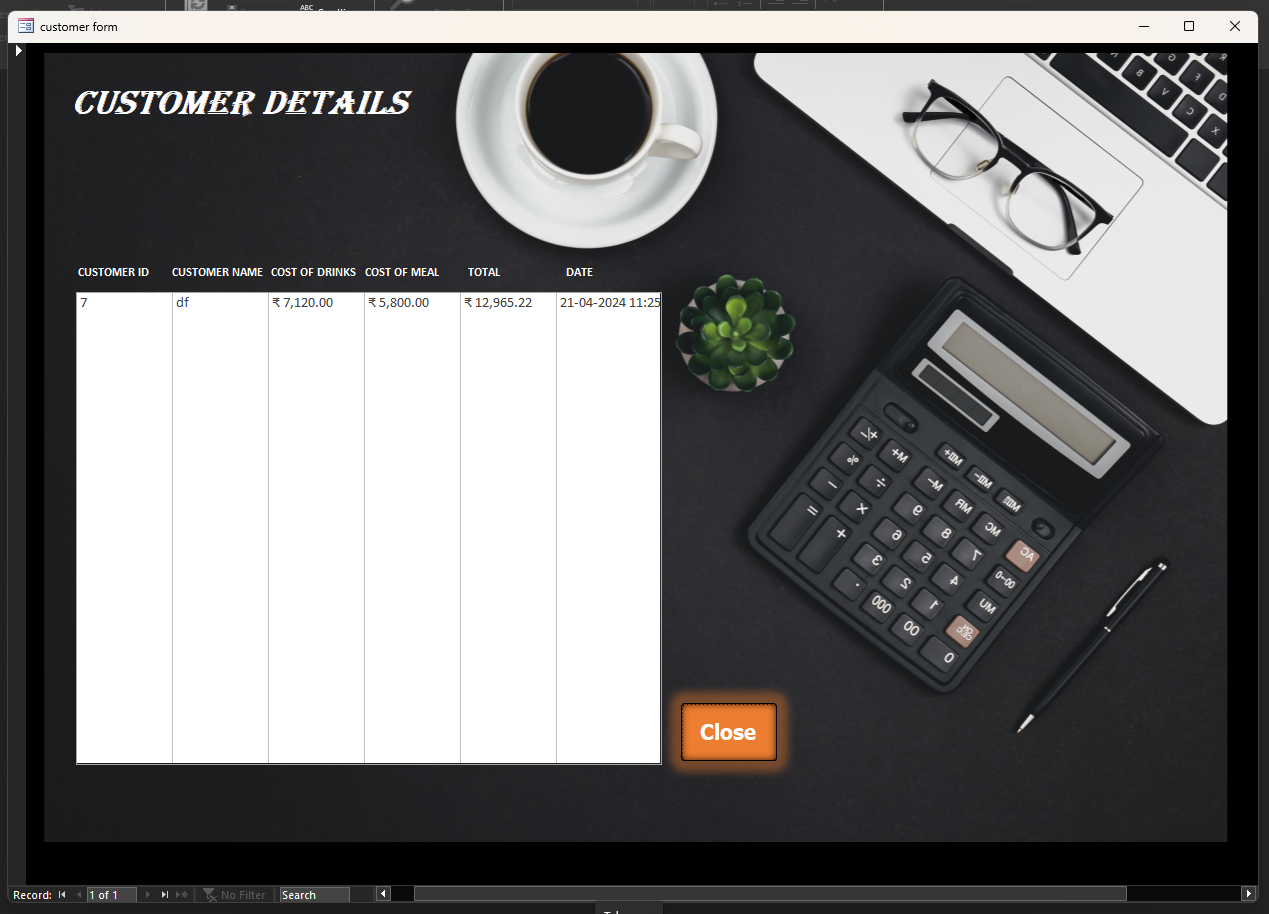
**Restaurant output screen :**

****

**Restaurant calculator screen:**

****

**CUSTOMER VIEW FORM:**

****



**IMPLEMENTATION**

**CHAPTER 6**

**CHAPTER 6**

**IMPLEMENTATION**

The system has been successfully implemented with the test values. At every stage of real data are fed into the system. The checklist is taken and data validation has been done. Until the data validation the data is error free.

Testing is a process of executing a program with the intent of finding error. During the testing process, each section of the project has been tested separately. In the design phase errors in the procedure were found and changes were made the user requirements. We verify that all the system elements have been properly integrated and perform the allocated functions. The step involves in acceptance testing.

Acceptance testing involves planning and execution of various types of steps in order to demonstrate that the implemented in software system satisfies the requirement of the user. The acceptance tests have been conducted with the user to find out whether the user is satisfied with the system.

The system has been implemented after completing the validation required to solve the errors found. The system is being given special emphasis on menu driven model. It is needs have been taken into account and all facilities have been provided in the system.

**6.1 User Training**

After the system is implemented successfully, training of the user is one of the most important subtasks of the developer. For this purpose user manuals are prepared and handled over to the user to operate the developed system. Thus the users are trained to operate the developed system. Both the hardware and software securities are made to run the developed systems successfully in future. In order to put new application system into use, the following activities were taken care of:

* Preparation of user and system documentation
* Conducting user training with demo and hands on
* Test run for some period to ensure smooth switching over the system

The users are trained to use the newly developed functions. User manuals describing the procedures for using the functions listed on menu are circulated to all the users. It is confirmed that the system is implemented up to users need and expectations.

**6.2 Security and Maintenance**

Maintenance involves the software industry captive, typing up system resources .It means restoring something to its original condition. Maintenance follows conversion to the extend that changes are necessary to maintain satisfactory operations relative to changes in the user’s environment. Maintenance often includes minor enhancements or corrections to problems that surface in the system’s operation. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

Any system developed should be secured and protected against possible hazards. Security measures are provided to prevent unauthorized access of the database at various levels. An uninterrupted power supply should be so that the power failure or voltage fluctuations will not erase the data in the files.

Password protection and simple procedures to prevent the unauthorized access are provided to the users .The system allows the user to enter the system only through proper user name and password

`

**CONCLUSION**

**CHAPTER 7**

**CHAPTER 7**

**CONCLUSION**

This was the first considerably large and important project undertaken by me during my B.C.A. course. It was an experience that changed the way I perceived project development. The coding could not be started before the whole system was completely finalized. Even then there were so many changes required and the coding needed to be changed. I attribute this to inadequate information gathering from the user. Though there were many meetings with the user and most of the requirements were gathered, a few misinterpretations of the requirements still crept in. It made me realize how important the systems analysis phase is. The project is a classic example for the adage that learning of concepts needs to be supplemented with application of that knowledge.

The given project titled “**RESTAURANT MANAGEMENT** **SYSTEM**” is successfully completed and the reports are also generated. Validation is done accordingly and the system is designed to be as interactive as possible. Listing them side by side provides all necessary option, any one of the desired option can be selected and the corresponding operation is performed. Finally the system is implemented in the client place.



**CHAPTER 8**

**BIBLIOGRAPHY**

**CHAPTER 8**

**BIBLIOGRAPHY**

**Books Referred**

* Microsoft Office Access 2007:

The Complete Reference by McGraw-Hill Education, March 2007 Edition

* Microsoft Access 2007:

Microsoft Access 2007 stop by step, Microsoft Press, 1 edition (January 2007)

* Visual Basic 6:

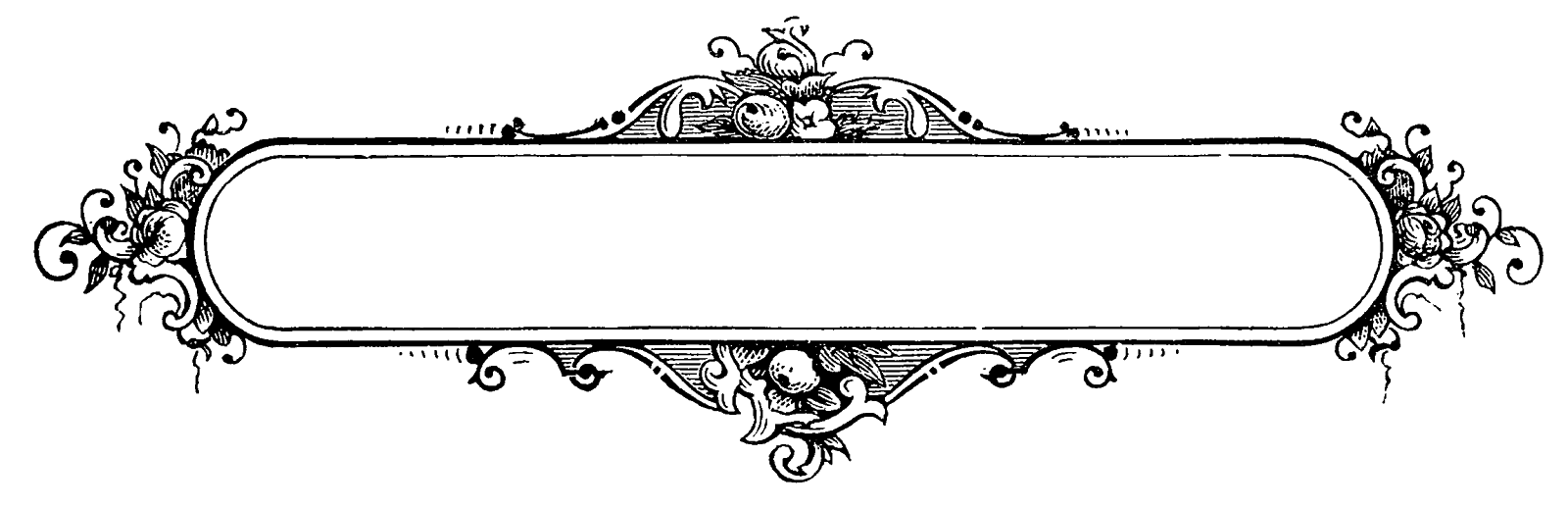
The Complete Reference by McGraw-Hill Education, 1 edition (April 1999)

* Visual Basic 6:

ADO Programming in Visual Basic 6 by Prentice Hall PTR (November 1999)

**References**

* https://www.zoho.com
* https://books.google.co.in
* https://www.techwalla.com
* https://www.informationvine.com



**Thank You**