

FERTILIZER SHOP

MANAGEMENT SYSTEM



A PROJECT REPORT ON
FERTILIZER SHOP MANAGEMENT SYSTEM
For BCA (Bachelor of computer application)
Academic year (2019-2022)



Arcade Business college
(Affiliated to Patliputra University, Patna)

Rajendra Nagar, Patna – 800016

GUIDED BY:

Anupam singh

Submitted By :- Abhiram Kumar

Mayank Kumar

Ashish Ranjan

Raushan Kumar

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

With my immense pleasure, I would like to express my special thanks of gratitude to **Mrs. Anupam Singh**, my project guide, for giving me this privilege working under her and completing this study.

I would like to thank our lab in charge Rajesh sir, Rahul sir who helped us directly or indirectly to complete my project whenever I found myself in problems.

DECLARATION

I hereby declare that the project work entitled "FERTILIZER SHOP MANAGEMENT SYSTEM" submitted to Arcade Business College is a record of an original work done by me under the guidance of Mrs. Anupam Singh. This project work has not performed on the basis for the award of any degree or diploma /associative or fellowship and similar project if any. The information and data given in the project is true and original to the best of own knowledge.

INDEX

Topic Title	Page No
Introduction	6
Limitation of Existing System	8
Objective of the Project	10
System analysis	12
Identification of Need	15
Preliminary Investigation	17
Feasibility Study	18
Project Planning	21
Software Requirement Specification(SRS)	22
Entity Relationship Model(ERD)	36
Module Description	43
Table Description	45
User Interface	52
Coding	69
Software Testing	212
Cost Estimation of The Project along with cost Estimation Model	214
Report Snaps	215
Future Scope and Future Enhancementof the Project	219
Bibliography	221

INTRODUCTION



- This is a Small Fertilizer Shop in village.
- The Registered Name of Shop is OM TRADERS.
- It is situated at Vill – Sasaur , Dis – Nalanda,Pin – 811102
- This shop owner Name is Ashok Kumar.
- This shop is established in 2016
- This shop provides fertilizer in village.

The manufacturer's objective is to keep fertilizer materials moving continuously out of the factory so that the level of stocks in the factory depot remains just sufficient to maintain the pipeline stocks and replenish stocks. An explanation of carry-over stocks is needed here. In general terms, they consist of supplies from producers and imports remaining at the end of the season. Working stocks account for a major part of the carry-over stocks and are required to ensure a smooth and uninterrupted flow of supplies from producer or importer, or both, to the final user, the farmer. Working stocks comprise stocks held by farmers, retailers, distributors, sub-distributors,

wholesalers, etc., including **quantities in transit** as well as those held by Government for sale through the public distribution system and for infra-seasonal supply for price stabilization.

Statistics are required to record the development of production capacities, production, imports, exports, deliveries, stocks, use in agriculture and prices. For the short-term these statistics are required to ensure that the types of fertilizers required by the farmer are available to him when and where they are needed. If fertilizers arrive too late in the season, they are useless to him, their food production potential is lost and in many cases some of the fertilizer will be lost or damaged before the next season. Statistics on fertilizer usage and stocks are needed for the organization of the acquisition, transport and storage of the supplies required by the farmer. Stock level statistics help the manufacturer to plan his production. Optimal use of fertilizers plays a key role in improving the productivity of various crops.

The production of fertilizer is continuous whereas the demand for fertilizer is seasonal. Over the years, it has been the endeavor of the factory to make available the required quantity of fertilizers to the farmers at right place, in right time to fulfill this basic objective, there is the need for proper estimation of demand for fertilizers, both on short-term and long-term basis.

LIMITATION OF THE EXISTING SYSTEM

The existing system is like a filing cabinet, slow and clumsy and we can lose records down the back. It can be like a file or register where certain types of information are stored like receipts, invoices especially organized for easier retrieval. There are several limitations of this system, these are:

- No proper security.
- Data loss: Records can be lost or misfiled making it hard to find them.
- Updating problem: If any changes have to be done on these records it would be done manually. Records can look messy and it would be very difficult to analyze them.
- Need for manual calculation.
- Lots of information has to be kept in mind while running the business.
- The records were never used to be in a systematic order, there used to be lots of difficulties in associating any particular transaction with a particular context.
- More paper work: Whole of the system have to be maintained with hands, thus the process of keeping, maintaining and

retrieving information was very tedious and lengthy.

- Difficulty in searching: If any information was to be found it was required to go through the different registers, documents, there would never exist anything like report generation.
- Time consuming: Unnecessary consumption of time while entering record and retrieving records.
- Backup issue: Difficult to make backup because every page would have to be rewritten or photocopied.

OBJECTIVE OF THE PROJECT

This project “Fertilizer shop management system” enables to manage and record the activities of Fertilizer Shop.

With the implementation of computerized system the task of keeping records in organized manner will be solved. The greatest of all is their retrieval which will be at the click of the mouse. So the proposed system helps in saving the time in different operations. The main objectives of this system are:

- Automatic updating: All the stock of the Fertilizer is updated automatically in the new system.
- User friendly: The system is user friendly and anyone can handle it as it is easy to operate.
- Secure: Provide security against unauthorized user by providing user id and password.
- Increase the efficiency of managing the shop.
- Easy manipulation: Editing, adding and updating of records are improved which result in proper resource management of Fertilizer shop data.
- Easy to make a backup in case of data loss.
- Better service with less time consumption
- In computer system the person has to fill the

various forms and number of copies of the forms can be easily generated at a time.



SYSTEM ANALYSIS



System analysis is a process of gathering and interpreting facts, diagnosing problems and the information about the Fertilizer Shop Management System to recommend improvements on the System . It is a problem solving activity that requires intensive communication between the system users and developers. System analysis or study is an important phase of any System development process. The System is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action. A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive at a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problems are re-identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighted with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is the loop that ends as soon as the user is satisfied with proposal. Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and developers. It does various feasibility studies. In these studies

a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

IDENTIFICATION OF NEED

The old manual system was suffering from a series of drawbacks. Since whole of the system was to be maintained with hands the process of keeping, maintaining and retrieving the information was very tedious and lengthy. The records were never used to be in a systematic order. There used to be lots of difficulties in associating any particular transaction with a particular context. If any information was to be found it was to be found it was required to go through the different registers, documents there would never exist anything like report generation. There would always be unnecessary consumption of time while entering records and retrieving records. One more problem was that it was very difficult to find errors while entering the records. Once the records were entered it was very difficult to update these records.

The reason behind it is that there is lot of information to be maintained and have to be kept in mind while running the business.

This project “Fertilizer shop management system” enables to manage and record the activities of whole Fertilizer shop of multi-facility skills only.

With the implementation of computerized system the task of keeping records in organized manner will be solved. The greatest of all is their retrieval which will be at the click of the mouse. So the proposed system helps in saving the time in different operations. The main objectives of this system are:

- All the stock of the Fertilizer is updated automatically in the new system.
- The system is user friendly and anyone can handle it as it is easy to operate.
- Provide security against unauthorized user by providing user id and password.
- Increase the efficiency of managing the shop.
- Editing, adding and updating of records are improved which result in proper resource management of Fertilizer shop data.

PRELIMINARY INVESTIGATION

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the system. The purpose of the preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the business system in all respect. Rather, it is the collecting of information that helps committee members to evaluate the merits of the project request and make an informed judgment about the feasibility of the proposed project.

Analysts working on the preliminary investigation should accomplish the following objectives:

- Clarify and understand the project request. Determine the size of the project.
- Assess costs and benefits of alternative approaches.
- Determine the technical and operational feasibility of alternative approaches.
- Report the findings to management, with recommendations outlining the acceptance or rejection of the proposal.
- Benefit to organization-The organization will obviously be able to gain benefits such as savings in operating cost, reduction in paperwork, better utilization of human resources and more presentable image increasing goodwill.

FEASIBILITY STUDY

After doing the project Fertilizer Management System study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible-given unlimited resources and infinite time.

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 includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

It is done to check how much system is feasible and the likelihood the system will be useful to the organization. The feasibility study is carried out by a small group of people who are familiar with information technique; understand the part of the business or in the system analysis and design and process.

Feasibility Study :-

- Economical Feasibility
- Technical Feasibility
- Operational Feasibility

Economical Feasibility :-

This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor.

- All hardware and software cost has to be borne by the organization.
- Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for system.

Technical Feasibility:-

The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the system, as described in the System Requirement Specification, and checked if everything was possible using different type of front-end and backend platform.

Operational Feasibility:-

The new system demands the expulsion of the people from the management. Application and technical client user can handle this software in a proper way so that it could be beneficial for the Fertilizer shop management system.

Application user can use this software in good way because these types of user having knowledge of application, so they use the system to access database and can work on the software in a good manner and positive approach. Technical user can also use this software but they must have proper and accurate technical skills. It can lead to the Indefinite strike to the shop also. No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that the feel comfortable with new system. As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

PROJECT PLANNING

PHASES	MEMBERS	TOTAL DAYS
1. ANALYSIS <ul style="list-style-type: none">• DATA GATHERING• FEASIBILITY STUDY• COSTBENEFIT ANALYSIS• PROJECT PROPOSAL	Abhiram Kumar Mayank Kumar Ashish Ranjan Raushan Kumar	TOTAL DAYS = 40
2. DESIGN	Abhiram Kumar Mayank Kumar Ashish Ranjan Raushan Kumar	TOTAL DAYS = 10
3. CODING	Abhiram Kumar Mayank Kumar Raushan Kumar	TOTAL DAYS = 25
4. TESTING	Ashish Ranjan Raushan Kumar	TOTAL DAYS = 20
5. IMPLEMENTATION	Abhiram Kumar Mayank Kumar Raushan Kumar	TOTAL DAYS =5
6. DOCUMENTATION	Abhiram Kumar Mayank Kumar	TOTAL DAYS =50

SOFTWARE REQUIREMENTS SPECIFICATION

Software Requirement:-

Software Environment is a technical specification of requirement of software product. This specifies the environment for development, operation and maintenance of the product.

Technologies used –

- VISUAL BASIC 6.0
- ORACLE 10G

VISUAL BASIC 6.0 :-

Visual Basic 6.0 is a third-generation event-driven programming language first released by Microsoft in 1991. It evolved from the earlier DOS version called BASIC. BASIC means Beginners' All-purpose Symbolic Instruction Code. Since then Microsoft has released many versions of Visual Basic, from Visual Basic 1.0 to the final version Visual Basic 6.0. Visual Basic is a user-friendly programming language designed for beginners, and it enables anyone to develop GUI window applications easily.

Visual Basic 6.0 is a highly interactive programming language. It has several features that allow the programmer to develop applications for MS-Windows in an easy and efficient manner. Visual Basic follows the Standard syntax of basic, except that some new language features have been added to it to give more flexibility to the programmer. Visual Basic makes use of GUI for creating robust and powerful applications.

The GUI enables users to interact with an application. This feature makes it easier to comprehend things in a quicker and easier way. Visual Basic 6.0 enables the rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using data Access Objects, Remote Data Objects, or ActiveX Data Objects, and Creation of ActiveX controls and objects.

The Integrated Development Environment :-

One of the most significant changes in Visual Basic 6.0 is the Integrated Development Environment (IDE). IDE is a term commonly used in the programming world to describe the interface and environment that we use to create our applications. It is called integrated because we can access virtually the entire development tool that we need from one screen called an interface. The IDE is also commonly referred to as the design environment, or the program. Integrated development is one in which we can develop, run, test and debug your applications.

The visual basic IDE is made up of a number of components:

- 1)** Menu Bar
- 2)** Tool Bar
- 3)** Project Explorer
- 4)** Properties window
- 5)** Form Layout Window
- 6)** Toolbox
- 7)** Form Designers
- 8)** Code Window

Menu Bar –The menu bar presents the Visual Basic menus. Here is a list of those menus and what they do.

- **File** – File handling and printing; also used to make EXE files.
- **Edit** – Standard editing functions, undo, searches.
- **View** – Displays or hides windows and toolbars.
- **Project** – Sets project properties, adds/removes forms and modules, and adds/removes references and components.
- **Format** – Aligns or sizes controls.
- **Debug** – Starts/stops debugging and stepping through programs.
- **Run** – Starts a program, or compiles and starts it.
- **Tools** – Adds procedures, starts the Menu Editor, sets IDE options.
- **Add-Ins** – Add-in manager lists add-ins like Application Wizard and APIViewer.
- **Window** – Arranges or selects open windows.
- **Help** – Handles Help and the about box

Toolbar – The main Visual Basic toolbar contains buttons matching popular menu items. Clicking the button is the same as selecting a menu item and can save some time. Besides the main toolbar, we can also display other dockable toolbars in Visual Basic: the Debug, Edit, and Form Editor toolbars. To display one of these toolbars, just select it using the Toolbars item in the View menu; the toolbar appears.

Project Explorer – Project Explorer Window allows us to coordinate the parts of our program into folders for easy manipulation. It gives us a valuable overview of our entire project, which is really very useful when a project gets large and contains many components.

Property Window – The properties window is docked under the projectexplorer window. It describes about the various characteristics of selected objects such as caption for label, text for text boxes. Here we can view and change the property of selected object.

Forms Layout Window – The form layout window is used to determine the initial positions of the forms in our application. This window is useful in applications that use multiple forms.

Toolbox – The toolbox contains a set of controls such as – text boxes, labels, list boxes, option buttons and much more. We can add these controls to our forms.

Form Designer – Form designers are really just windows in which a particular form appears. We can place controls into a form simply by drawing them after clicking the corresponding control's tool in the toolbox.

Code Window – We just place the code we want to attach to an object in the code window (to open an object's code in the code window, just double-click that object). There are two drop-down list boxes at the top of the code window: the left list lets us select the object to add code to, and the right list lets us select the procedure to add (all the methods the object supports appear in this list).

Programming in Visual Basic 6.0 –

Programming in VB 6.0 is a combination of visually arranging components or controls on a form, specifying attributes and actions for those components, and writing additional lines of code for more functionality.

Visual Basic supports several project types. Some of them are :-

- Standard Windows EXE programs
- ActiveX EXE files
- ActiveX DLLs
- Program written by the Visual Basic Application Wizard
- Data Projects
- IIS(the Microsoft Internet Information Server) applications
- Visual Basic add-ins
- ActiveX document DLLs
- ActiveX document EXE files
- DHTML applications
- VB Enterprise Edition Controls

This list of project types indicates that there is a vast scope for different types of application development in VB 6.0, but it is primarily used to develop Windows applications and to interface database systems. Forms are created in VB 6.0 using drag-and-drop techniques. A tool is used to place controls (e.g., text boxes, buttons, etc.) on the form (window). Controls have attributes and event handlers associated with

them. Defaultvalues are provided when the control is created, but may be changed by the programmer.

Many attribute values can be modified during run time based on user actions or changes in the environment, providing a dynamic application. When the user selects an element, an event handler is called that executes code that the programmer created to perform the action for that list item.

For example, code can be inserted into the form resize event handler to reposition a control so that it remains centered on the form, expands tofill up the form, etc. By inserting code into the event handler for a key press in a text box, the program can automatically translate the case of the text being entered, or even prevent certain characters from being inserted.

Visual Basic 6.0 provides user with a large library of utility objects, and the language provides basic support for object-oriented programming. Unlike many other programming languages, Visual Basic is generally not case- sensitive—though it transforms keywords into a standard case configuration and forces the case of variable names to conform to the case of the entry in the symbol table. String comparisons are case sensitive by default.

The Parts of a Visual Basic Project –

Forms –Forms are the templates we base windows on. Besides standard forms, Visual Basic also supports Multiple Document Interface (MDI) forms, as well as a whole number of predefined form.

Modules –Modules are collections of code and data that function something like objects in object-oriented programming (OOP), but without defining OOP characteristics like inheritance, polymorphism, and so on. The point behind modules is to enclose procedures and data in a way that hides them from the rest of the program. Breaking a large program into smaller, self-contained modules can be invaluable for creating and maintaining code.

Global Items –Global items are accessible to all modules and forms in a project, and we declare them with the Public keyword.

Control statements –Control statements are programming statements that affect program execution, based on the outcome of some logical comparison. There are two types of control statements. They are –

Decision Structures –These structures enable the program to make decisions on how it will operate based on test conditions. The decision structure can be of three types. They are – If-Then, If-Then-Else, Select case.

Looping Structures – These structures enable the program to execute a block of statement for a number of times based on a condition. They are while-wend, do-while loop, for-next. In addition to all these, Visual Basic 6.0 also supports the concept of different types of operators such as arithmetic operator, logical operators etc. It also supports the concept of array . Overall, Visual Basic 6.0 provides a programmer with a healthy environment to develop almost all kind of application.

ORACLE 10g :-

Every business enterprise maintains a large volume of data for its operations. With more and more people accessing these data for their work, the need to maintain its integrity and relevance increases. Normally, with the traditional methods of storing data and information in files, the chances that the data loses its integrity, consistency and data redundancy are very high. So, there is always a need of more secured and computerized way of storing, accessing and manipulating data. DBMS serves this purpose.

DBMS stands for Database Management System. It is application software. It is collection of hardware and software. Hardware refers storage media (i.e. Hard Disk) for storage of Database. Whereas, software defines set of programs required to provide access to Database and their management.

Types of DBMS –

- a. File Oriented DB
- b. Database Management System (DBMS)
- c. Relational Database Management System (RDBMS)
- d. Object Oriented Database Management System (OODBMS)
- e. Distributed Database Management System (DDBMS)
- f. Knowledge Based Database Management System (KDBMS)

Models of DBMS –

- 1. 1st Model – Hierarchical Model
- 2. 2nd Model – Network Model
- 3. 3rd Model – ER Model
- 4. 4th Model – Relational Model
- 5. 5th Model – Object oriented Model

Oracle was originally developed by Lawrence Ellison (Larry Ellison) and his two friends and former co-worker in 1977. Oracle DB runs on the most major platforms like Windows, UNIX, Linux and Mac OS. Oracle 10g is an Object Oriented Database Management System (OODBMS) having both the features of Object Oriented Programming and RDBMS. Oracle Database XE provides an organized mechanism for storing, managing, and retrieving information.

Tables are the basic storage structure for holding data. The Oracle Database offers a wide range of options and features in the areas of Availability, Scalability, Analytics, Performance, Security, Management, Developers and Integration. These aim to enhance and complement existing database functionality to meet customer-specific requirements.

Oracle products are based on a concept known as the 'client/server technology'. This concept involves segregating the processing of an application between two systems. One performs all activities related to the database (server) and the other performs activities that help the user to interact with the application (clients).

A client or front-end database application also interacts with the database by requesting and receiving information from the database server. It acts as an interface between the user and the database. Further, it also checks for the validation against the data entered by the user. Oracle 10g can perform wide array of tasks. It acts as a transparent interface between the physical storage and the logical presentation of data. It provides a set of flexible and sophisticated tools for handling information.

Facilities provided by Oracle are:

- Defining Database.

- Creating Table in Database
- Query Processing.
- Add, edit and delete data from.
- Modify the structure of the database.
- Secure data from public access.
- Communicates with in the network, export and import data.

Oracle uses SQL (Structured Query Language) as its native language to enable users to instruct DBMS components.

SQL is a 4th Generation Language. It contains command based language which is known as Query. Each query contains its executable program.

The name SQL stands for Structural Query Language. SQL is a data access language, like any other language, it is used for communication. SQL communicates with database manager. The database manager could be Oracle, Informix, DB2 and SQL database. SQL is easy to learn. Despite the fact that SQL is a computer programming language, it is much simpler than traditional programming language like COBOL, BASIC, FORTRAN or API.

This is due to the fact that SQL is a nonprocedural language.

A database management system requires a query language to enable users to access data. Structured Query Language (SQL – pronounced ‘sequel’) is the language used by most relational database systems. IBM developed the SQL language in a prototype relational database management system –System R – in the mid-1970s. In 1979, Oracle Corporation introduced the first commercially available implementation of SQL.

Features of SQL –

SQL is a non-procedural language. You specify what information you require, not how to get it. In other words, SQL does not require you to specify the access method to the data. All SQL statements use the query

optimizer – a part of the RDBMS – to determine the fastest means of retrieving the specified data. This feature makes it easier for you to concentrate on obtaining the desired result.

SQL processes sets of records rather than a single record at a time. The most common form of a set of records is a table.

A range of user including DBA's, application programmers, management personnel, and many other types of end users can use SQL.

SQL provides commands for a variety of tasks including:

- Querying data.
- Inserting, updating and deleting rows in a table.
- Creating, modifying and deleting database objects
- Controlling access to the database and database objects
- Guaranteeing database consistency.

All statements or query of SQL is divided into 4 sub languages according to access operation –

Data Definition Language – Creates, Alter, Drop, and Commands.

Data Manipulation Language – Insert, Select, Delete and Update.

Transaction Control Language – Commit, Save point and rollback

Data Control Language – Grant and Revoke commands.

Oracle greatly supports RDBMS features. Also it supports high security to the data and faster accessing capability. It can be run on a variety of platforms and operating systems. One can develop an application easily by providing user-friendly environment.

The features of oracle are portability and compatibility.

WORKING ENVIRONMENT

➤ HARDWARE REQUIREMENTS:-

Processor	Pentium based system with at least 533 MHz Speed or higher.
RAM	Recommended 256 MB or above.
Machine Architecture	32-bit
Hard Disk	Minimum 16 GB or above
Monitor	Color
Mouse	Windows Compatible
Keyboard	Compatible keyboard (102/103keys)
CD-ROM	52x or above

SOFTWARE REQUIREMENTS :-

- Windows XP, 7
- Visual Basic 6.0 As Front End (For Interface)
- Oracle 10G As Back End (For Database)

E R DIAGRAM

(ENTITY RELATIONSHIP DIAGRAM)

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

Uses of entity relationship diagrams

- Database design: ER diagrams are used to model and design relational databases, in terms of logic and business rules (in a logical data model) and in terms of the specific technology to be implemented (in a physical data model.) In software engineering, an ER diagram is often an initial step in determining requirements for an information systems project. It's also later used to model a particular database or databases. A relational database has an equivalent relational table and can potentially be expressed that way as needed.
- Database troubleshooting: ER diagrams are used to analyze existing databases to find and resolve problems in logic or deployment. Drawing the diagram should reveal where it's going wrong.

- Business information systems: The diagrams are used to design or analyze relational databases used in business processes. Any business process that uses fielded data involving entities, actions and interplay can potentially benefit from a relational database. It can streamline processes, uncover information more easily and improve results.
- Business process re-engineering (BPR): ER diagrams help in analyzing databases used in business process re-engineering and in modeling a new database setup.
- Education: Databases are today's method of storing relational information for educational purposes and later retrieval, so ER Diagrams can be valuable in planning those data structures.
- Research: Since so much research focuses on structured data, ER diagrams can play a key role in setting up useful databases to analyze the data.

The components and features of an ER diagram

ER Diagrams are composed of entities, relationships and attributes. They also depict cardinality, which defines relationships in terms of numbers. Here's a glossary:

Entity

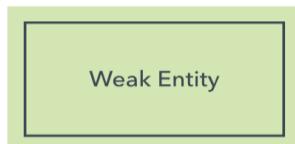
A definable thing—such as a person, object, concept or event—that can have data stored about it. Think of entities as nouns. Examples: a customer, student, car or product. Typically shown as a rectangle.



Entity type: A group of definable things, such as students or athletes, whereas the entity would be the specific student or athlete. Other examples: customers, cars or products.

Entity set: Same as an entity type, but defined at a particular point in time, such as students enrolled in a class on the first day. Other examples: Customers who purchased last month, cars currently registered in Florida. A related term is instance, in which the specific person or car would be an instance of the entity set.

Entity categories: Entities are categorized as strong, weak or associative. A strong entity can be defined solely by its own attributes, while a weak entity cannot. An associative entity associates entities (or elements) within an entity set.



Entity keys: Refers to an attribute that uniquely defines an entity in an entity set. Entity keys can be super, candidate or primary. Super key: A set of attributes (one or more) that together define an entity in an entity set. Candidate key: A minimal super key, meaning it has the least possible number of attributes to still be a super key. An entity set may have more than one candidate key. Primary key: A candidate key chosen by the database designer to uniquely identify the entity set. Foreign key: Identifies the relationship between entities.

Relationship

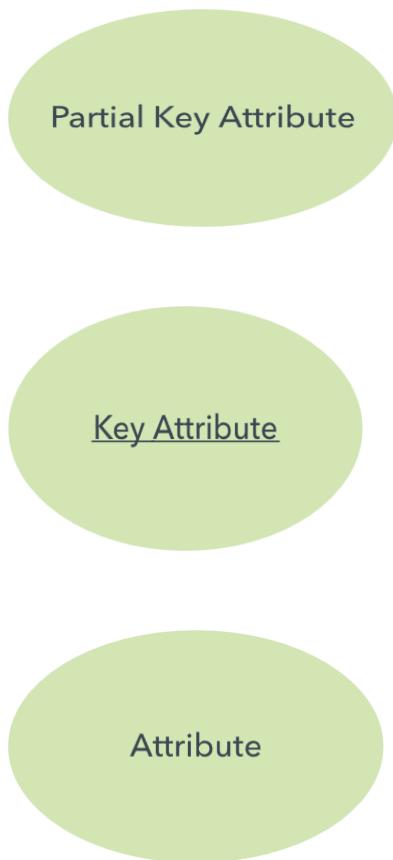
How entities act upon each other or are associated with each other. Think of relationships as verbs. For example, the named student might register for a course. The two entities would be the student and the course, and the relationship depicted is the act of enrolling, connecting the two entities in that way. Relationships are typically shown as diamonds or labels directly on the connecting lines.



Recursive relationship: The same entity participates more than once in the relationship.

Attribute

A property or characteristic of an entity. Often shown as an oval or circle.



Descriptive attribute: A property or characteristic of a relationship (versus of an entity.)

Attribute categories: Attributes are categorized as simple, composite, derived, as well as single-value or multi-value. Simple: Means the attribute value is atomic and can't be further divided, such as a phone number. Composite: Sub-attributes

spring from an attribute. Derived: Attributed is calculated or otherwise derived from another attribute, such as age from a birth date.

Multi-value: More than one attribute value is denoted, such as multiple phone numbers for a person.

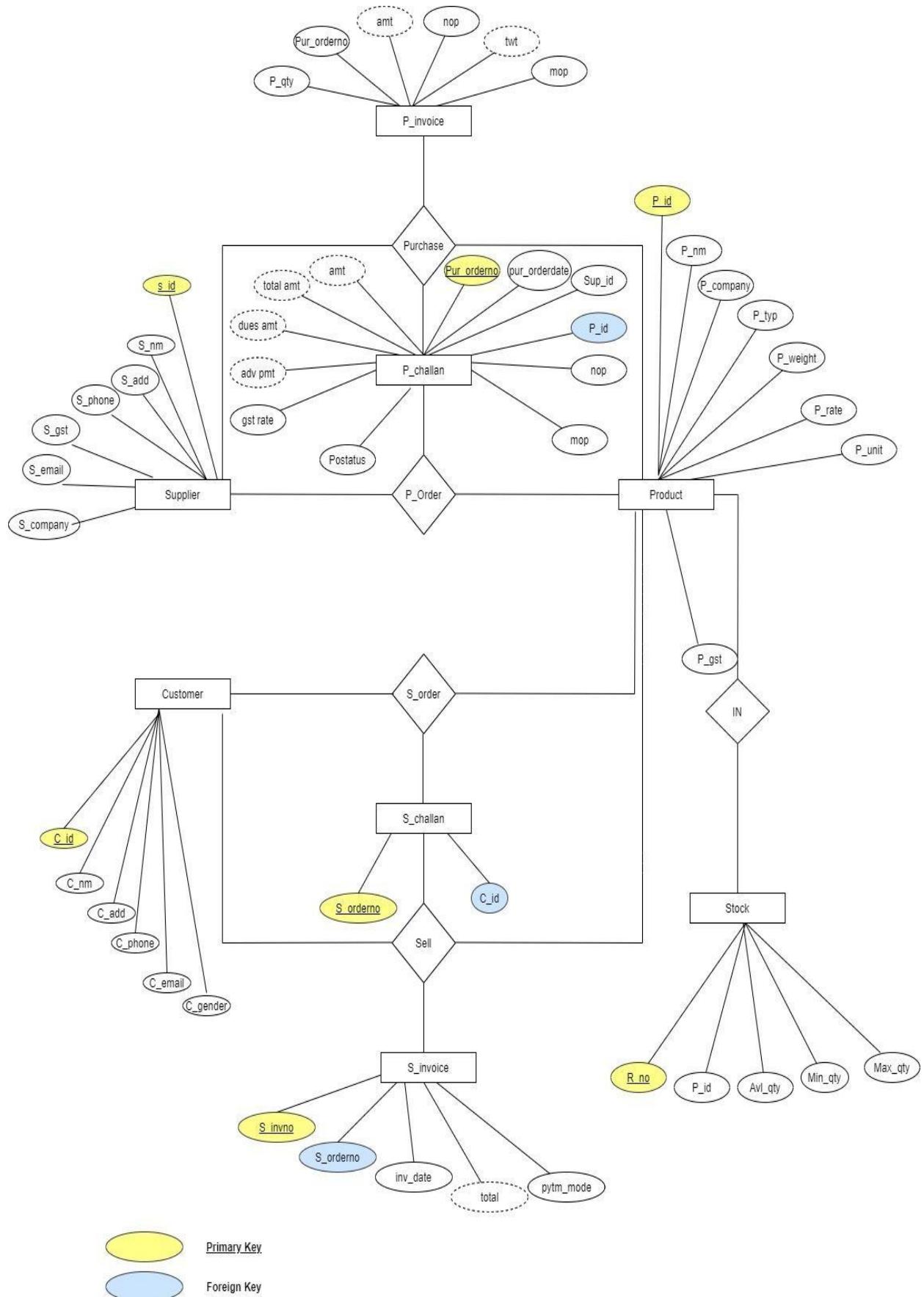
Single-value: Just one attribute value. The types can be combined, such as: simple single-value attributes or composite multi-value attributes.

Cardinality

Defines the numerical attributes of the relationship between two entities or entity sets. The three main cardinal relationships are one-to-one, one-to-many, and many-many. A one-to-one example would be one student associated with one mailing address. A one-to-many example (or many-to-one, depending on the relationship direction): One student registers for multiple courses, but all those courses have a single line back to that one student. Many-to-many example: Students as a group are associated with multiple faculty members, and faculty members in turn are associated with multiple students.

Cardinality views: Cardinality can be shown as look-across or same-side, depending on where the symbols are shown.

Cardinality constraints: The minimum or maximum numbers that apply to a relationship.



MODULE DESCRIPTION

“FERTILIZER SHOP MANAGEMENT SYSTEM” is a step towards offering more or less the similar features. This system enables to manage and record the activities of whole Shop. All records can be kept in organized manner. The greatest of all is their retrieval which will be at the click of the mouse.

➤ **SUPPLIER MANAGEMENT MODULE:-**

In this module functions related to supplier are done such as addition of new supplier, modification of existing supplier, and their updation etc. Also searching of particular supplier on the basis of their name, state, company name etc can be easily done. Here we can even randomly select a record and update it. This module handles the database of all suppliers with their full information such as their name, address number etc.

➤ **PRODUCT MANAGEMENT MODULE:-**

This module will handle all the records of fertilizer along with their details. In this module function related to addition of new product, modification of existing product, generation of product list and their updation is done.

➤ **STOCK MANAGEMENT MODULE:-**

In this module function related to stock such as quantity available, quantity for a particular batch, reorder level, where the Fertilizer is kept is done. Here we can update the stock, we can remove expired product directly by searching for expired products, we can order for the products which has reached reorder level easily. It will handle information about whether Fertilizer is available in the stock or not.

➤ PURCHASE MANAGEMENT MODULE:-

This module consist of list of items ordered , and item quantity and rate of product ordered, its date and updating stock and various reports as per required.

➤ CUSTOMER MANAGEMENT MODULE:-

In this module function related to customer are done, such as addition of new customer, modification of existing customer are done, generation of customer list and their updation with the customer such as customername, contact no and etc. We can search the customer according to their phone no, through their name easily. Here we have a unique features of sales count which tells how many sales a particular customer have done till date . It handles the database of all customer with their mobile no, name, age, address, dues amount etc.

➤ SALES MANAGEMENT MODULE:-

This module will take care of all the processes of sales such as pricing and taxation, sales order processing, etc.

TABLE DESCRIPTION

Login (login details):-

Field name	Data type	Size	Constraints	Description
User_id	Varchar	20	Foreign key	User id
Password	Varchar	20	Foreign key	Password of user
Mobile	Number	10	-	Mobile num of user

Product (Product Table):-

Field name	Data type	Size	Constraints	Description
P_id	Varchar	10	Primary key	Product id
P_nm	Varchar	30	Not Null	Name of Product
P_type	Char	10	Not Null	Type of Product
P_comp	Char	20	Not Null	Company of Product
P_wt	Decimal	(5,2)	Not Null	Weight of Product
P_gst	Decimal	(4,2)	Not Null	Rate of GST
P_rate	Decimal	(6,2)	Not Null	Rate of Product
P_unit	Varchar	5	Not Null	Unit of product
P_HSN	Varchar	8	Not Null	HSN of product

Supplier(Supplier Table):-

Field name	Data type	Size	Constraints	Description
Sup_id	Varchar	10	Primary key	Supplier id
Sup_nm	Varchar	20	Not Null	Name of Supplier
Sup_mob	Number	13	Not Null	Contact num of Supplier

Sup_location	Varchar	50	Not Null	Address of Supplier
Sup_State	Varchar	10	Not Null	State of Supplier
Sup_City	Varchar	10	Not Null	City of Supplier
Sup_Pin code	Number	6	Not Null	Pin code of Supplier
comp	Varchar	30	Not Null	Company Name of Supplier
Sup_GST No	Varchar	30	Not Null	GST NO of Supplier
Sup_email	Varchar	15	Not Null	Email of Supplier
Status	Varchar	15	-	Status

Supplierprd(product detail of supplier):-

Field name	Data type	Size	Constraints	Description
Sup_id	Varchar	10	Foreign key	Name of Supplier
P_id	Varchar	20	Foreign key	Name of product
Rate	Number	6,2	-	Rate of product

Customer(Customer Table):-

Field name	Data type	Size	Constraints	Description
C_id	Varchar	10	Primary key	Customer id
C_nm	Varchar	30	Not null	Name of Customer
C_mob	Varchar	10	Not null	Mobile No of Customer
C_add	Varchar	50	Null	Address of Customer
C_Gender	Varchar	15	Not Null	Gender of Customer
C_email	Varchar	30	Null	Email Address

Dues	Number	(8,2)	-	Dues of Customer
------	--------	-------	---	------------------

Purordetail (Table of purchase order):-

Field name	Data type	Size	Constraints	Description
Pur_order no	Varchar2	10	Primary key	Purchase order no
Pur_orderdate	Date	-	Not Null	Date of purchase
Sup_id	Varchar2	10	Foreign key	Supplier id
noofproduct	Number	2	Not Null	Date of delivery
modeofpayment	Varchar2	8	Not Null	Product id
chqno	Varchar2	16	-	Quantity of product
totalamoun	Number	(8,2)	Not Null	Amount
totalwithtax	Number	(9,2)	Not Null	Total amount
advamount	Number	(9,2)	Not Null	Advance Amount
duesamount	Number	(8,2)	Not Null	Dues Amount
Poststatus	Varchar2	25	-	status

P_det(Table of purchase detail):-

Field name	Data type	Size	Constraints	Description
pur_order no	Varchar2	10	Not Null	Purchase order no
S_no	Number	2	Not Null	Serial no
P_id	Varchar2	10	Not Null	Product id
P_nm	Varchar2	40	Not Null	Name of Product
P_typ	Varchar2	30	Not Null	Type of Product
P_rate	Number	(6,2)	Not Null	Rate of Product
P_unit	Varchar2	10	Not Null	Unit of Product
Qty	Number	4	Not Null	Quantity of product
Price	Number	(9,2)	Not Null	Price of product
Cgstper	Number	(4,2)	Not Null	Percentage of Central GST
Cgstamt	Number	(6,2)	Not Null	Amount of Central GST
Sgstper	Number	(4,2)	Not Null	Percentage of State GST
Sgstamt	Number	(6,2)	Not Null	Amount of state GST
Total	Number	(9,2)	Not Null	Total Price

Orderdetail(Received detail of purchase):-

Field name	Data type	Size	Constraints	Description
Invoiceno	Varchar2	10	Not Null	Invoice no of purchase
Invoicedate	Date	-	Not Null	Invoice date of purchase
Order_no	Varchar2	10	Not Null	Order no of purchase
Challanno	Varchar2	10	Not Null	Challan no of purchase
Noofprd	Number	2	Not Null	Number of product

Pymtby	Varchar2	10	Not Null	Mode of payment
Chqno	Varchar2	10	Not Null	Cheque no
Withtaxamt	Number	(8,2)	Not Null	Amount with tax
Amtpaidinadv	Number	(8,2)	Not Null	Paid amount in advance
Netamt	Number	(8,2)	Not Null	Net amount

Stock(Stock Table):-

Field name	Data type	Size	Constraints	Description
rno	Varchar2	5	Primary key	Rack no
P_id	Varchar2	10	Foreign key	ID of product
Avl_qty	Number	5	Not Null	Available Quantity
Min_qty	Number	5	Not Null	Minimum Quantity
Max_qty	Number	5	Not Null	Maximum Quantity

Sell_details(Selling details)

Field name	Data type	Size	Constraints	Description
S_ono	Varchar2	10	Primary ket	Selling order no
s_date	Date	-	Not Null	Date of selling
C_id	Varchar2	10	Foreign key	id of customer
nop	Number	3	Not Null	Number of Product
S_pm	Varchar2	10	Not Null	Payment mode of selling
chque	Varchar2	15	Not Null	Cheque no
S_total	Number	(8,2)	Not Null	Total amount
S_twt	Number	(8,2)	Not Null	Total amount with tax
C_prevdues	Number	(8,2)	-	Previous dues of customer
S_amtpd	Number	(8,2)	Not Null	Paid Amount

Sold_pdet(Selling details):-

Field name	Data type	Size	Constraints	Description
S_ono	Varchar2	10	Foreign key	Selling order no
p_id	Varchar2	10	Foreign key	id of product
quant	Number	(6,2)	Not Null	Number of Quantity
amount	Number	(8,2)	Not Nul	Sold amount

Recvd_p_det(Received detail of purchase product):-

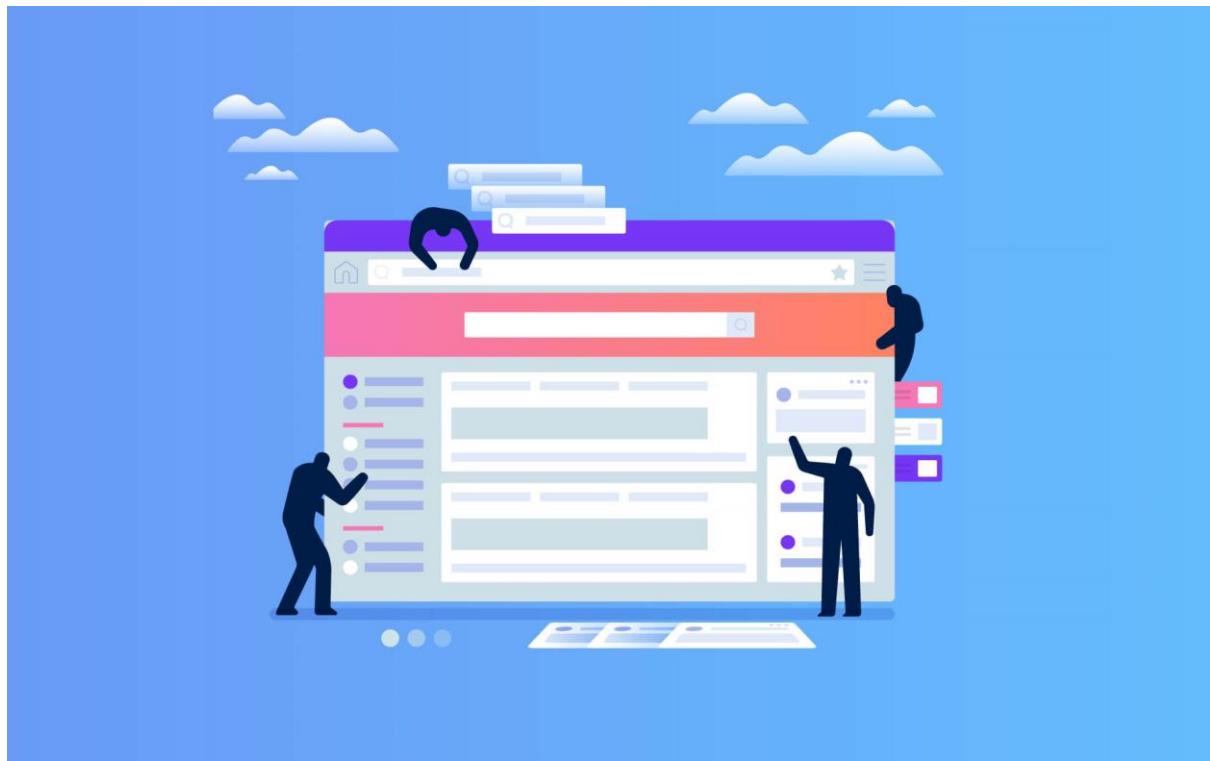
Field name	Data type	Size	Constraints	Description
pur_order no	Varchar2	10	Not Null	Purchase order no
S_no	Number	2	Not Null	Serial no

P_id	Varchar2	10	Not Null	Product id
P_nm	Varchar2	40	Not Null	Name of Product
P_typ	Varchar2	30	Not Null	Type of Product
P_rate	Number	(6,2)	Not Null	Rate of Product
P_unit	Varchar2	10	Not Null	Unit of Product
Qty	Number	4	Not Null	Quantity of product
Price	Number	(9,2)	Not Null	Price of product
Cgstper	Number	(4,2)	Not Null	Percentage of Central GST
Cgstamt	Number	(6,2)	Not Null	Amount of Central GST
Sgstper	Number	(4,2)	Not Null	Percentage of State GST
Sgstamt	Number	(6,2)	Not Null	Amount of state GST
Total	Number	(9,2)	Not Null	Total Price

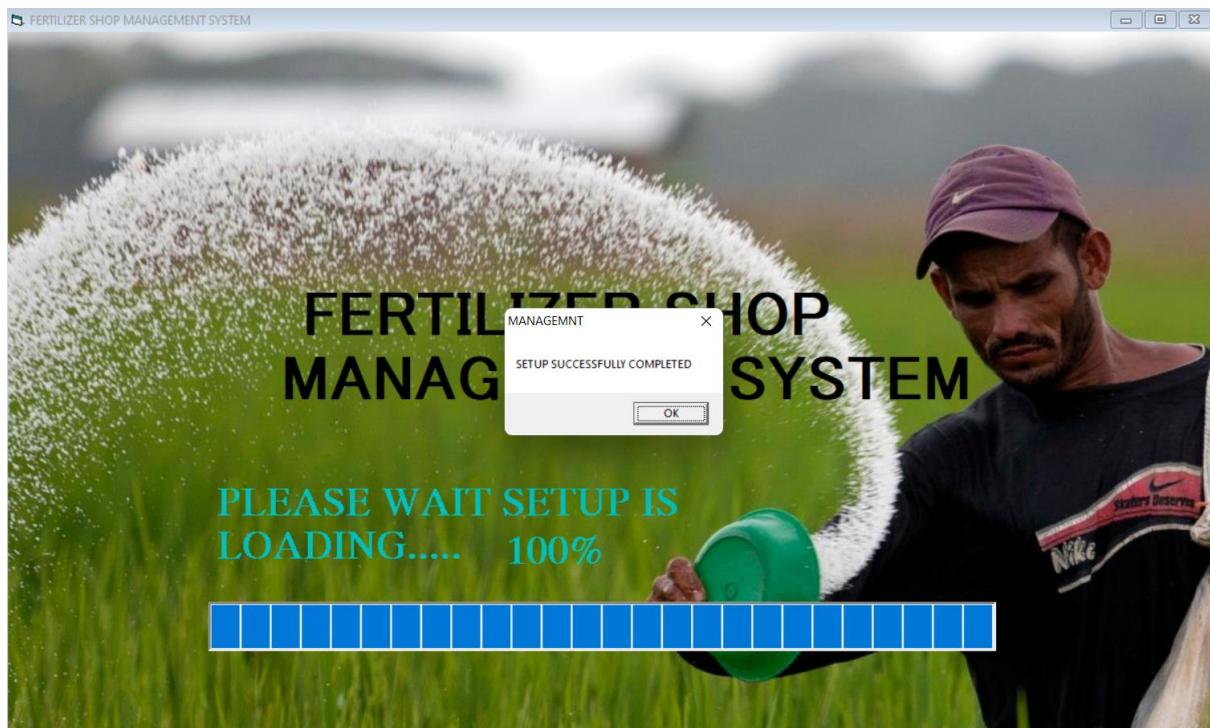
Sell_inv(selling invoice details:-)

Field name	Data type	Size	Constraints	Description
S_ono	Varchar2	10	Foreign key	Selling order no
Inv_no	Varchar2	10	Foreign key	Invoice no
Inv_date	Date	-	Not Null	Invoice date
Total	Number	(8,2)	Not Null	Total amount

USER INTERFACE



Splash Screen 1 :-



Splash Screen 2 :-



Login Form :-

The image shows a user login form window titled "USER LOGIN". Inside the window, there is a placeholder image of a person in a suit and tie, a placeholder image of a person working at a desk with a computer, and several text input fields and buttons.

USER LOGIN

USER ID

PASSWORD SHOW PASSWORD

LOGIN

Forgot Password?

Forgot Password :-

FORGOT PASSWORD ?

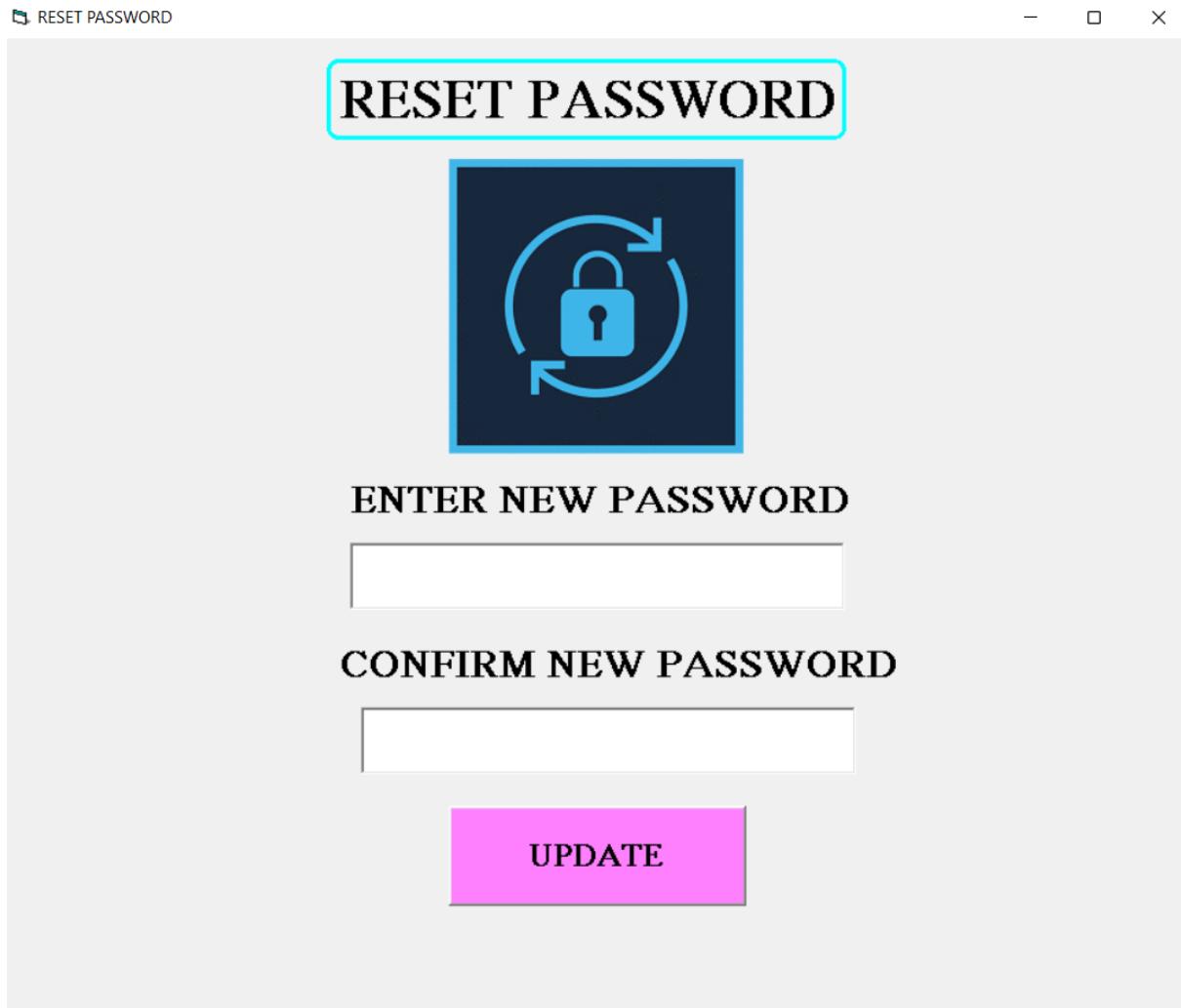


ENTER USER ID :-

REGISTERED PHONE NUMBER :-

RESET CANCEL

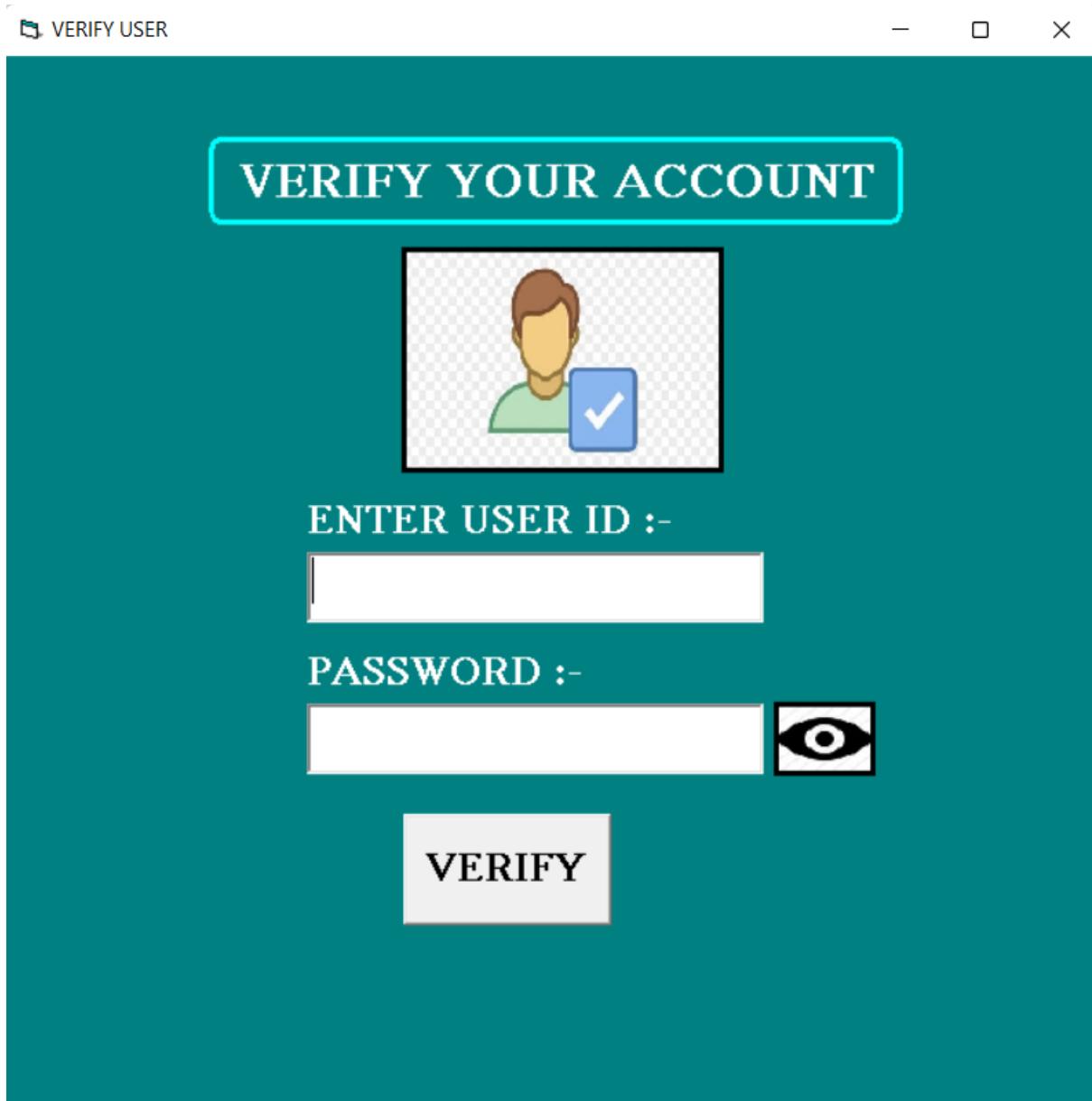
Reset Password :-



Add User :-



Verify User :-



Main Screen (MDI FORM) :-



Product Form :-

FERTILIZER SHOP - [PRODUCT]

ACCOUNT PRODUCT SUPPLIER PURCHASE CUSTOMER SALE REPORT ABOUT

FERTILIZER SHOP MANAGEMENT SYSTEM
02 July,2022 Saturday 12:07:14 AM

Product : Product id:

Product name * :-	<input type="text"/>	HSN Code * :-	<input type="text"/>																																																															
Product Type :-	<input type="text"/>																																																																	
Company Name * :-	<input type="text"/>																																																																	
Packet weight :-	<input type="text"/>																																																																	
G.S.T Rate :-	<input type="text"/> %																																																																	
Purchase Rate :-	<input type="text"/>																																																																	
Unit Of Measurement :-	<input type="text"/>																																																																	
<table border="1"> <thead> <tr> <th>P ID</th> <th>P NM</th> <th>P TYPE</th> <th>P COMP</th> <th>P WT</th> <th>P GST</th> <th>P RATE</th> <th>P UNIT</th> <th>P HSN</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				P ID	P NM	P TYPE	P COMP	P WT	P GST	P RATE	P UNIT	P HSN																																																						
P ID	P NM	P TYPE	P COMP	P WT	P GST	P RATE	P UNIT	P HSN																																																										
<input type="button" value="Add New"/> <input type="button" value="Save"/> <input type="button" value="Update"/> <input type="button" value="Search"/> <input type="button" value="Delete"/> <input type="button" value="Close"/>																																																																		

Stock Form :-

FERTILIZER SHOP - (STOCK)

ACCOUNT PRODUCT SUPPLIER PURCHASE CUSTOMER SALE REPORT ABOUT

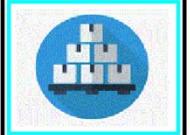
ORDER STOCK SUPPLIER CUSTOMER PRODUCT SALES REPORTS

FERTILIZER SHOP MANAGEMENT SYSTEM
02 July,2022 Saturday 12:06:42 AM

Stock :-

Rack No.	:-	<input type="text"/>	<input type="button" value="Add Rack"/>
Product Id	:-	<input type="text"/>	
Available Quantity	:-	<input type="text"/>	
Minimum Quantity	:-	<input type="text"/>	
Maximum Quantity	:-	<input type="text"/>	

Search



RNO	P_ID	AVL_QTY	MIN_QTY	MAX_QTY

Supplier Form :-

FERTILIZER SHOP - [SUPPLIER]

ACCOUNT PRODUCT PURCHASE CUSTOMER SALE REPORT ABOUT

ORDER STOCK SUPPLIER CUSTOMER PRODUCT SALES REPORTS

FERTILIZER SHOP MANAGEMENT SYSTEM
02 July,2022 Saturday 12:06:47 AM

SUPPLIER :

Supplier Id:

Supplier Name * :

Mobile No. * :

Supplier Location :

State :

City :

Pin no. :

Company Name :

Email Id :

G.S.T No. * :





Product Sold By Supplier

Product Id:	Rate:	
S.no	Product	Rate

Add **Delete**

SUP_ID	SUP_NM	SUP_MOB	SUP_LOCATION	SUP_STATE	SUP_CITY	SUP_PINCODE	COM	SUP_EMAIL	SUP_GSTNO	STATUS
S001	ABHIRAM	9572979465	PATNA	Bihar	PATNA	800026	KISAAN	sk@gmail.com	5555555555	active

Add New **Save** **Update** **Search** **Delete** **Close**

Customer Form :-

FERTILIZER SHOP - [CUSTOMER]

ACCOUNT PRODUCT PURCHASE CUSTOMER SALE REPORT ABOUT

ORDER STOCK SUPPLIER CUSTOMER PRODUCT SALES REPORTS

FERTILIZER SHOP MANAGEMENT SYSTEM
02 July,2022 Saturday 12:06:50 AM

Customer : Customer Id* :

Name : <input type="text"/>	
Mobile No* : <input type="text"/>	
Address : <input type="text"/>	
Gender : <input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Transgender	
Email Id : <input type="text"/>	

C ID	C NM	C MOB	C ADD	C GENDER	C EMAIL

Add New Save Update Search Delete Close

PURCHASE ORDER FORM :-

FERTILIZER SHOP - [ORDER]

ACCOUNT PRODUCT SUPPLIER PURCHASE CUSTOMER SALE REPORT ABOUT

FERTILIZER SHOP MANAGEMENT SYSTEM
02 July,2022 Saturday 10:06:26 AM

ORDER INFORMATION

Order no *	OD001	Supplier Name :	Order Date *	:	
Supplier ID *		Mobile No. :	Company Name :		Supplier Address :

ADD PRODUCT

Product Id:	Product Name:	Product Type	Product Weight	Unit:	Purchase rate	Quantity	Price:	GST Rate:	Total Price	%
Serial No.	Product Id	Product Name	Product Type	Purchase Rate	Unit	Quantity	Price	CGST %	SGST %	TOTAL PRICE

CALCULATION

No. of product :		Total :	With Tax Amount :	Place Order
Payment Mode :	<input type="radio"/> Cheque <input checked="" type="radio"/> Cash		Advance Payment :	
Cheque No. :			Remaining Amount :	Cancel

PURCHASE FORM :-

FERTILIZER SHOP - [PURCHASE]

ACCOUNT PRODUCT SUPPLIER PURCHASE CUSTOMER SALE REPORT ABOUT

FERTILIZER SHOP MANAGEMENT SYSTEM

02 July,2022 Saturday 10:07:05 AM

ORDER	STOCK	SUPPLIER	CUSTOMER	PRODUCT	SALES	REPORTS
Order no * :		Order Date :		Supply Date * :		Invoice Date :
Supplier ID :		Mobile No. :		Chalan No. * :		Invoice No. :
Supplier Name :		Company Name :		Supplier Address :		

PRODUCT ORDERED

Serial No.	Product Id	Product Name	Product Type	Purchase Rate	Quantity Ordered	Price	CGST		SGST		TOTAL PRICE
							%	Amount	%	Amount	

Final Amount :

RECEIVED PRODUCT

Product Id:	Product Name:	Product Type	Purchase rate:	Quantity Ordered:	Quantity Received * :	Price	CGST		SGST		TOTAL PRICE
							%	Amount	%	Amount	

Final Amount :

CALCULATION

No. of product :		STATUS :	<input checked="" type="radio"/> Received	<input type="radio"/> Not Received	<input type="radio"/> Partially Received	
Payment Mode :	<input type="radio"/> Cheque	<input checked="" type="radio"/> CASH				
Cheque No. :		With Tax Amount :		Amt paid in advance :		
				Net Amount :		
						Save
						Cancel

SALE ORDER FORM :-

FERTILIZER SHOP - (SALES)

ACCOUNT PRODUCT PURCHASE CUSTOMER SALE REPORT ABOUT

FERTILIZER SHOP MANAGEMENT SYSTEM

02 July,2022 Saturday 10:06:02 AM

Order Information:

Order No * : 50001 Mobile No. * : Order Date * :

Customer Id * : Add New Customer

Customer Name * : Customer Address : Gender: Male Female Transgender

Add Product:

Product Id:	Product Name:	Unit:	MRP :	Rack No.	Available	Quantity:	Price:	GST Rate:	Total Price	Qty %	Add
											Delete

Final Amount:

Calculation:

No. of product: Total: With Tax Amount: Net Amount: New Dues: Generate Invoice

Payment Mode : Cash Cheque Amount Paid: Cancel

Cheque No. :

Sale Invoice :-

SALE INVOICE



FERTILIZER
OM TRADERS

Sell Order No.: S0001
Invoice No.: INV001

SALE INVOICE

OM TRADERS, SASAUR
STATE-BIHAR
Mobile No.-9771392398

Save/Print

Invoice Date: 7/2/2022

Customer ID:
Customer Name:
Gender:

Address:
Mobile No.:

Product name	MRP	Quantity	Total Price	CGST	SGST	Net Amount

	Total:		CGST(Rs.)	SGST(Rs.)	
Total Tax Amount:					

Terms and Conditions:
Goods once sold will not be taken back or exchanged after 1 Week of purchase.

Previous Dues Amount(Rs.):
Sub Total(Rs.):
Amount Paid(Rs.):
New Dues Amount (Rs.):

Paid By:

Reports Form:-

FERTILIZER SHOP - (REPORTS)

ACCOUNT PRODUCT SUPPLIER PURCHASE CUSTOMER SALE REPORT ABOUT

ORDER STOCK SUPPLIER CUSTOMER PRODUCT SALES REPORTS

FERTILIZER SHOP MANAGEMENT SYSTEM
02 July,2022 Saturday 12:07:24 AM

REPORTS

CUSTOMER REPORT

STOCK REPORT

SUPPLIER REPORT

PRODUCT REPORT

PURCHASE REPORT

PURCHASE STATUS REPORT

SALES REPORT

CLOSE



FERTILIZER
OM TRADERS

CODING



Visual Basic Coding :-

Module Coding :-

Public C As New ADODB.Connection

Public R As New ADODB.Recordset

Public SQL As String

Public Function CONN()

Set C = New ADODB.Connection

C.Open "Provider=MSDAORA.1;User ID=PRJ2133F1/abhiram;Persist Security Info=True"

End Function

Login Form:-

Private Sub Check1_Click()

If (Text2.PasswordChar = "*") Then

Text2.PasswordChar = ""

Else

Text2.PasswordChar = "*"

End If

End Sub

Private Sub Command1_Click()

```
Set R = New ADODB.Recordset  
SQL = "SELECT * FROM LOGIN WHERE USER_ID=""" + Text1.Text + """  
Set R = C.Execute(SQL)  
If (Text1.Text = R.Fields(0) And Text2.Text = R.Fields(1)) Then  
    Unload Me  
    MDIForm1.Show  
    Text1.Text = ""  
    Text2.Text = ""  
Elseif Text1.Text = "" And Text2.Text = "" Then  
    MsgBox ("PLEASE ENTER USER ID AND PASSWORD")  
    Text1.SetFocus  
Elseif Text2.Text = "" And Text1.Text <> R.Fields(0) Then  
    MsgBox ("PLEASE ENTER PASSWORD AND INPUTED USER ID IS  
    WRONG")  
    Text1.SetFocus  
Elseif Text1.Text = "" And Text2.Text <> R.Fields(1) Then  
    MsgBox ("PLEASE ENTER USER ID AND INPUTED PASSWORD IS  
    WORNG")  
    Text1.SetFocus  
Elseif Text2.Text = "" Then  
    MsgBox ("PLEASE ENTER PASSWORD")  
    Text2.SetFocus  
Elseif Text1.Text <> R.Fields(0) Then
```

```
MsgBox ("INVALID USER ID ENTER CORRECT USER ID")
Text1.SetFocus
ElseIf Text2.Text <> R.Fields(1) Then
    MsgBox ("INVALID PASSWORD ENTER CORRECT PASSWORD")
    Text2.SetFocus
Else
    MsgBox "PLEASE ENTER CORRECT USER ID AND PASSWORD"
    Text1.SetFocus
    Text1.Text = ""
    Text2.Text = ""
End If
End Sub

Private Sub Command3_Click()
FORGOT.Show
End Sub

Private Sub Form_Load()
CONN
Command1.Enabled = False
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then Text2.SetFocus  
End Sub
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then Command1.SetFocus  
End Sub
```

```
Private Sub Text1_Change()  
Command1.Enabled = Len(Text1.Text) > 0  
End Sub
```

All CODING:-

```
Private Sub add_Click()  
Set R = New ADODB.Recordset  
SQL = "SELECT * FROM LOGIN"  
Set R = C.Execute(SQL)  
If (Text1.Text = R.Fields(0)) Then  
    MsgBox "USER ID ALREADY EXIST", vbCritical, "INVALID USER ID"  
    Text1.Text = ""  
    Text1.SetFocus  
Else  
    If (Text2.Text = Text3.Text) Then  
        Set R = New ADODB.Recordset
```

```
SQL = "INSERT INTO LOGIN VALUES('" + Text1.Text + "','" + Text2.Text +  
"', '" + Text4.Text + "')"
```

```
MsgBox SQL
```

```
Set R = C.Execute(SQL)
```

```
MsgBox "USER CREATED", vbInformation, "USER CREATE"
```

```
Unload Me
```

```
Else
```

```
MsgBox "CHECK YOUR PASSWORD", vbCritical, "PASSWORD"
```

```
Text2.Text = ""
```

```
Text3.Text = ""
```

```
Text1.SetFocus
```

```
End If
```

```
End If
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
RES = MsgBox("DO YOU WANT TO CANCEL", vbQuestion +  
vbYesNoCancel, "FOR CLOSE")
```

```
If RES = vbYes Then
```

```
Unload Me
```

```
End If
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
CONN
```

```
MsgBox "Connected!"
```

```
End Sub
```

```
Private Sub Image2_Click()
```

```
If (Text2.PasswordChar = "*") Then
```

```
Text2.PasswordChar = ""
```

```
Else
```

```
Text2.PasswordChar = "*"
```

```
End If
```

```
End Sub
```

```
Private Sub Image3_Click()
```

```
If (Text3.PasswordChar = "*") Then
```

```
Text3.PasswordChar = ""
```

```
Else
```

```
Text3.PasswordChar = "*"
```

```
End If
```

```
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then Text2.SetFocus
End Sub
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then Text3.SetFocus
End Sub
```

```
Private Sub Text3_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then Text4.SetFocus
End Sub
```

```
Private Sub Text4_KeyPress(KeyAscii As Integer)
If KeyAscii >= 48 And KeyAscii <= 57 Or KeyAscii = 8 Then
    Text4.Locked = False
Else
    Text4.Locked = True
    If KeyAscii = 13 Then add.SetFocus
    If Len(Text4.Text) <> 10 Then
        MsgBox "ENTER 10 DIGIT PHONE NUMBER"
        Text4.Text = ""
        Text4.SetFocus
    End If
End If
```

End If

End Sub

Private Sub Command1_Click()

Set R = New ADODB.Recordset

SQL = "SELECT * FROM LOGIN"

Set R = C.Execute(SQL)

If (Text1.Text = R.Fields(1)) Then

SQL = "delete from LOGIN where USER_ID=''' + Combo1.Text + '''"

Set R = C.Execute(SQL)

MsgBox "USER DELETED"

Combo1.Clear

Text1.Text = ""

Else

MsgBox "PASSWORD NOT MATCHED"

End If

End Sub

Private Sub Command2_Click()

RES = MsgBox("DO YOU WANT TO CANCEL", vbQuestion + vbYesNoCancel, "FOR CLOSE")

```
If RES = vbYes Then  
    Unload Me  
End If  
End Sub  
  
Private Sub Form_Load()  
    CONN  
    Set R = New ADODB.Recordset  
    SQL = "select *from LOGIN"  
    Set R = C.Execute(SQL)  
    While R.EOF = False  
        Combo1.AddItem R.Fields(0)  
        R.MoveNext  
    Wend  
End Sub
```

```
Private Sub Form_Load()  
    Timer1.Enabled = True  
End Sub
```

```
Private Sub Timer1_Timer()
```

```
ProgressBar1.Value = ProgressBar1.Value + 20  
Label2.Caption = "PLEASE WAIT SETUP IS LOADING....."  
Label3.Caption = ProgressBar1.Value & "%"  
If ProgressBar1.Value = ProgressBar1.Max Then  
    Timer1.Enabled = False  
    MsgBox "SETUP SUCCESSFULLY COMPLETED"  
    Unload Me  
    frmSplash.Show  
End If  
End Sub
```

```
Private Sub Command1_Click()  
    Set R = New ADODB.Recordset  
    SQL = "SELECT * FROM LOGIN"  
    Set R = C.Execute(SQL)  
    If (Text1.Text = R.Fields(0) And Text2.Text = R.Fields(2)) Then  
        Unload Me  
        RESET.Show  
        Text1.Text = ""  
        Text2.Text = ""  
    Elseif Text1.Text = "" And Text2.Text = "" Then
```

```
MsgBox ("PLEASE ENTER USER ID AND PHONE NUMBER")
Text1.SetFocus
ElseIf Text2.Text = "" And Text1.Text <> R.Fields(0) Then
    MsgBox ("PLEASE ENTER PHONE NUMBER AND INPUTED USER ID IS
WRONG")
Text1.SetFocus
ElseIf Text1.Text = "" And Text2.Text <> R.Fields(2) Then
    MsgBox ("PLEASE ENTER USER ID AND INPUTED PHONE NUMBER IS
WORNG")
Text1.SetFocus
ElseIf Text2.Text = "" Then
    MsgBox ("PLEASE ENTER PHONE NUMBER")
Text2.SetFocus
ElseIf Text1.Text <> R.Fields(0) Then
    MsgBox ("INVALID USER ID ENTER CORRECT USER ID")
Text1.SetFocus
ElseIf Text2.Text <> R.Fields(2) Then
    MsgBox ("INVALID PHONE NUMBER ENTER CORRECT PHONE
NUMBER")
Text2.Text = ""
Text2.SetFocus
Else
    MsgBox "PLEASE ENTER CORRECT USER ID AND PHONE NUMBER"
```

```
Text1.SetFocus  
Text1.Text = ""  
Text2.Text = ""  
End If  
End Sub  
  
Private Sub Command2_Click()  
RES = MsgBox("DO YOU WANT TO CANCEL", vbQuestion +  
vbYesNoCancel, "FOR CLOSE")  
If RES = vbYes Then  
Unload Me  
End If  
End Sub  
  
Private Sub Form_Load()  
CONN  
End Sub  
  
Private Sub Text1_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then Text2.SetFocus  
End Sub  
  
  
Private Sub Text2_KeyPress(KeyAscii As Integer)  
If KeyAscii >= 48 And KeyAscii <= 57 Or KeyAscii = 8 Then  
Text2.Locked = False
```

```
Else
Text2.Locked = True
If KeyAscii = 13 Then Command1.SetFocus
End If
End Sub
```

```
Private Sub Form_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Unload Me
LOGINFORM.Show
End If
End Sub
```

```
Private Sub about_Click()
about.Show
End Sub
```

```
Private Sub AB_OUT_Click()
about.Show
End Sub
```

```
Private Sub Command1_Click()
```

```
Unload Me
```

```
Load PRODUCT
```

```
PRODUCT.Show
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Unload Me
```

```
Load stock
```

```
stock.Show
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
Unload Me
```

```
Load supplier
```

```
supplier.Show
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
Unload Me
```

```
Load customer
```

```
customer.Show
```

End Sub

Private Sub Command5_Click()

Unload Me

Load ORDER

ORDER.Show

End Sub

Private Sub Command6_Click()

Unload Me

Load sale

sale.Show

End Sub

Private Sub Command7_Click()

Unload Me

Load REPORT

REPORT.Show

End Sub

Private Sub CREATEEUSER_Click()

verify.Show

End Sub

Private Sub CUS_TOMER_Click()

Unload Me

Load customer

customer.Show

End Sub

Private Sub CUSTOMER_REPORT_Click()

Unload Me

Load REPORT

REPORT.Show

REPORT.Frame4.Visible = True

REPORT.Combo5.Clear

REPORT.Combo5.AddItem "ID"

REPORT.Combo5.AddItem "MOBILE"

REPORT.Combo5.AddItem "NAME"

End Sub

Private Sub DELETEUSER_Click()

DELETE_USER.Show

End Sub

```
Private Sub FORGETPASSWORD_Click()
FORGOT.Show
End Sub

Private Sub MDIForm_Load()
Label2.Caption = Format$(Now, "dd mmmm,yyyy" & Space(19) &
"DDDD")
End Sub
```

```
Private Sub MPRODUCT_Click()
Unload Me
Load PRODUCT
PRODUCT.Show
End Sub
```

```
Private Sub OR_DER_Click()
Unload Me
Load ORDER
ORDER.Show
End Sub
```

```
Private Sub PORDER_Click()
Unload Me
```

```
Load sale
```

```
sale.Show
```

```
End Sub
```

```
Private Sub PRODUCT_REPORT_Click()
```

```
Unload Me
```

```
Load REPORT
```

```
REPORT.Show
```

```
REPORT.Frame3.Visible = True
```

```
REPORT.Combo1.Clear
```

```
REPORT.Combo1.AddItem "ID"
```

```
REPORT.Combo1.AddItem "NAME"
```

```
REPORT.Combo1.AddItem "COMPANY"
```

```
REPORT.Combo1.AddItem "TYPE"
```

```
End Sub
```

```
Private Sub PU_RCHASE_Click()
```

```
Unload Me
```

```
Load purchase
```

```
purchase.Show
```

```
End Sub
```

```
Private Sub PURCHASE_REPORT_Click()
Unload Me
Load REPORT
REPORT.Show
REPORT.Frame9.Visible = True
REPORT.Combo9.Clear
REPORT.Combo9.AddItem "Order No"
REPORT.Combo9.AddItem "Supplier Id"
REPORT.Combo9.AddItem "Date"
REPORT.Combo9.AddItem "Month"
REPORT.Combo9.AddItem "Between Dates"
End Sub
```

```
Private Sub PURCHASE_STATUS_REPORT_Click()
Unload Me
Load REPORT
REPORT.Show
REPORT.Frame5.Visible = True
REPORT.Combo6.Clear
REPORT.Combo6.AddItem "Order No"
REPORT.Combo6.AddItem "Invoice No."
REPORT.Combo6.AddItem "Invoice Date"
```

```
REPORT.Combo6.AddItem "Month"  
REPORT.Combo6.AddItem "Between Dates"  
End Sub
```

```
Private Sub SALE_INVOICE_Click()  
saleinvoice.Show  
End Sub
```

```
Private Sub SALES_REPORT_Click()  
Unload Me  
Load REPORT  
REPORT.Show  
REPORT.Frame25.Visible = True  
REPORT.sale1.Clear  
REPORT.sale1.AddItem "Order Id"  
REPORT.sale1.AddItem "Customer Id"  
REPORT.sale1.AddItem "Date"  
REPORT.sale1.AddItem "Month"  
REPORT.sale1.AddItem "Between Dates"  
End Sub
```

```
Private Sub ST_OCK_Click()
```

Unload Me

Load stock

stock.Show

End Sub

Private Sub STOCK_REPORT_Click()

Unload Me

Load REPORT

REPORT.Show

REPORT.Frame2.Visible = True

REPORT.STK1.Clear

REPORT.STK1.AddItem "ID"

REPORT.STK1.AddItem "RACK NO"

End Sub

Private Sub SUPP_LIER_Click()

Unload Me

Load supplier

supplier.Show

End Sub

Private Sub SUPPLIER_REPORT_Click()

```
Unload Me  
Load REPORT  
REPORT.Show  
REPORT.Frame7.Visible = True  
REPORT.Combo12.Clear  
REPORT.Combo12.AddItem "ID"  
REPORT.Combo12.AddItem "NAME"  
REPORT.Combo12.AddItem "MOBILE NO."  
End Sub
```

```
Private Sub Timer1_Timer()  
Label3.Caption = Format$(Time$, "hh:mm:ss AM/PM")  
End Sub
```

```
Private Sub Combo1_Click()  
Set R = New ADODB.Recordset  
SQL = "select *from product where P_id=" + Combo1.Text + """  
Set R = C.Execute(SQL)  
Combo2.Text = R.Fields(1)  
TXT41.Text = R.Fields(2)
```

```
TXT4.Text = R.Fields(4)  
Text1.Text = R.Fields("p_unit")  
Text3.Text = R.Fields(5)  
SQL = "select *from supplierprd where P_id="" + Combo1.Text + """  
Set R = C.Execute(SQL)  
Text4.Text = R.Fields(2)  
End Sub
```

```
Private Sub Command1_Click()  
id.AddItem Combo1.Text  
nm.AddItem Combo2.Text  
typ.AddItem TXT41.Text  
prate.AddItem Text4.Text  
unit.AddItem Text1.Text  
qty.AddItem TXT5.Text  
prc.AddItem Text2.Text  
cgst.AddItem Val(Text3.Text) / 2  
sgst.AddItem Val(Text3.Text) / 2  
cmt.AddItem (Val(Text2.Text) * Val(Text3.Text) / 100) / 2  
smt.AddItem (Val(Text2.Text) * Val(Text3.Text) / 100) / 2  
net.AddItem TXT71.Text  
If sr.ListCount = 0 Then
```

```
sr.AddItem 1  
Else  
    sr.AddItem (sr.ListCount + 1)  
End If  
Dim l As Long  
Dim lSum As Long  
For l = 0 To net.ListCount - 1  
    lSum = lSum + CLng(net.List(l))  
Next  
final.Text = lSum  
TXT7.Text = sr.ListCount  
TXT9.Text = final.Text  
  
Combo1.Text = " "  
Combo2.Text = " "  
TXT41.Text = " "  
Text4.Text = " "  
Text1.Text = " "  
TXT5.Text = " "  
Text2.Text = " "  
TXT4.Text = " "
```

```
Text3.Text = " "
```

```
TXT71.Text = " "
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
If sr.ListCount <> 0 Then Frame4.Enabled = False
```

```
TXT11.Text = ""
```

```
Text7.Text = ""
```

```
a = InputBox("Enter the Serial No. you want to remove:", "for delete")
```

```
If a = blank Then
```

```
MsgBox "Please enter serial no."
```

```
Else
```

```
TXT9.Text = Val(TXT9.Text) - net.List(a - 1)
```

```
final.Text = TXT9.Text
```

```
id.RemoveItem (a - 1)
```

```
nm.RemoveItem (a - 1)
```

```
unit.RemoveItem (a - 1)
```

```
typ.RemoveItem (a - 1)
```

```
qty.RemoveItem (a - 1)
```

```
prate.RemoveItem (a - 1)
```

```
prc.RemoveItem (a - 1)
```

```
If cgst.List(a - 1) <> "" Then cgst.RemoveItem (a - 1)
```

```
If cmt.List(a - 1) <> "" Then cmt.RemoveItem (a - 1)
If sgst.List(a - 1) <> "" Then sgst.RemoveItem (a - 1)
If smt.List(a - 1) <> "" Then smt.RemoveItem (a - 1)
net.RemoveItem (a - 1)
sr.Clear
TXT7.Text = id.ListCount
For i = 1 To id.ListCount
    sr.AddItem i
Next i
If sr.ListCount <> 0 Then Frame4.Enabled = False
If sr.ListCount = 0 Then Command2.Enabled = False
```

```
Dim l As Long
Dim lSum As Long
For l = 0 To prc.ListCount - 1
    lSum = lSum + CLng(prc.List(l))
Next
TXT8.Text = lSum
End If
End Sub
```

```
Private Sub Command3_Click()
```

```

If TXT1.Text = blank Or TXT3.Text = blank Or TXT2.Text = blank Or
TXT7.Text = blank Or TXT14.Text = blank Or TXT11.Text = blank Then
    MsgBox "Please fill all the details first!!"
Else
    answer = MsgBox("Do you want to place order ?", vbExclamation +
        vbYesNo, "add confirm")
    If answer = vbYes Then
        Set R = New ADODB.Recordset
        SQL = "insert into purordetail values('" + TXT1.Text + "','" + TXT2.Text +
            "','" + TXT3.Text + "','" + TXT7.Text + "','" + Text6.Text + "','" + TXT14.Text +
            "','" + TXT8.Text + "','" + TXT9.Text + "','" + TXT11.Text + "','" + Text7.Text +
            "','" + Label9.Caption + "')"
        Set R = C.Execute(SQL)
        MsgBox "order placed!!"
    Else
        MsgBox "Data not saved"
    End If

Dim i As Long
For i = 0 To sr.ListCount - 1
    SQL = "insert into p_det values('" + TXT1.Text + "','" + sr.List(i) + "','" +
        id.List(i) + "','" + nm.List(i) + "','" + typ.List(i) + "','" + prate.List(i) + "','" +
        unit.List(i) + "','" + qty.List(i) + "','" + prc.List(i) + "','" + cgst.List(i) + "','" +
        cmt.List(i) + "','" + sgst.List(i) + "','" + smt.List(i) + "','" + net.List(i) + "')"

```

```
Set R = C.Execute(SQL)
```

```
Next i
```

```
MsgBox "Data saved"
```

```
Unload Me
```

```
ORDER.Show
```

```
ORDER.Top = 0
```

```
ORDER.Left = 0
```

```
End If
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
End
```

```
End Sub
```

```
Private Sub final_Change()
```

```
Dim l As Long
```

```
Dim lSum As Long
```

```
For l = 0 To prc.ListCount - 1
```

```
    lSum = lSum + CLng(prc.List(l))
```

```
Next
```

```
TXT8.Text = lSum
```

End Sub

Private Sub Form_Load()

CONN

Dim a As String

Command1.Enabled = True

Command3.Enabled = False

Set R = New ADODB.Recordset

SQL = "select

max(to_number(SUBSTR(PUR_ORDERNO,5,LENGTH(PUR_ORDERNO)))
)from purordetail"

Set R = C.Execute(SQL)

If IsNull(R.Fields(0)) Then

TXT1.Text = "OD" & "00" & 1

Else

TXT1.Text = "OD" & "00" & R.Fields(0) + 1

a = TXT1.Text

End If

If (a = "OD0010") Then

Set R = New ADODB.Recordset

```
SQL = "select  
max(to_number(SUBSTR( PUR_ORDERNO,4,LENGTH(PUR_ORDERNO)))  
)from purordetail"  
  
Set R = C.Execute(SQL)  
  
TXT1.Text = "OD" & "0" & R.Fields(0) + 1  
  
End If  
  
Text6.Visible = False  
  
MonthView1.Visible = False  
  
Set R = New ADODB.Recordset  
  
SQL = "select *from supplier"  
  
Set R = C.Execute(SQL)  
  
While R.EOF = False  
  
TXT3.AddItem R.Fields(0)  
  
R.MoveNext  
  
Wend  
  
'auto_combo  
  
MonthView1.Refresh  
  
End Sub
```

```
Private Sub MonthView1_DateClick(ByVal DateClicked As Date)
    TXT2.Text = Format(MonthView1, "dd-mmm-yyyy")
    MonthView1.Visible = False
End Sub
```

```
Private Sub Text4_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then TXT5.SetFocus
End Sub
```

```
Private Sub Text7_Change()
    Command3.Enabled = True
End Sub
```

```
Private Sub Txt5_KeyPress(KeyAscii As Integer)
    0 If KeyAscii = 13 Then Text3.SetFocus
End Sub
```

```
Private Sub Text3_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then Command1.SetFocus
End Sub

Private Sub TXT11_LostFocus()
Text7.Text = Val(TXT9.Text) - Val(TXT11.Text)
End Sub

Private Sub Txt11_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then Text7.SetFocus
End Sub

Private Sub Text7_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then Command3.SetFocus
End Sub

Private Sub TXT131_Click()
Text6.Text = TXT131.Caption
TXT14.Text = "Cash"
Label24.Visible = False
Label7.Visible = True
End Sub
```

```
Private Sub TXT132_Click()
Text6.Text = TXT132.Caption
Label7.Visible = False
```

```
Label24.Visible = True
```

```
End Sub
```

```
Private Sub TXT2_Click()
```

```
MonthView1.Visible = True
```

```
End Sub
```

```
Private Sub TXT3_Click()
```

```
Set R = New ADODB.Recordset
```

```
SQL = "select *from supplier where SUP_ID=""" + TXT3.Text + """
```

```
Set R = C.Execute(SQL)
```

```
TXT31.Text = R.Fields(1)
```

```
TXT32.Text = R.Fields(2)
```

```
TXT33.Text = R.Fields(7)
```

```
TXT34.Text = R.Fields(3)
```

```
Combo1.Clear
```

```
Set R = New ADODB.Recordset
```

```
SQL = "select *from supplierprd where sup_id=""" + TXT3.Text + """
```

```
Set R = C.Execute(SQL)
```

```
While R.EOF = False
```

```
Combo1.AddItem R.Fields(1)
```

```
R.MoveNext
```

```
Wend
```

```
End Sub
```

```
Public Function auto_combo()
```

```
Set R = New ADODB.Recordset
```

```
SQL = "select *from product"
```

```
Set R = C.Execute(SQL)
```

```
While R.EOF = False
```

```
Combo1.AddItem R.Fields(0)
```

```
R.MoveNext
```

```
Wend
```

```
End Function
```

```
Private Sub Txt5_Change()
```

```
Text2.Text = Val(Text4.Text) * Val(TXT5.Text)
```

```
TXT71.Text = Val(Text2.Text) + (Val(Text2.Text) * Val(Text3.Text) / 100)
```

```
End Sub
```

```
Private Sub add_Click()
```

```
Dim a As String
```

```
Set R = New ADODB.Recordset
```

```
SQL = "select max(to_number(SUBSTR(p_id,4,LENGTH(p_id))))from product"
Set R = C.Execute(SQL)
If IsNull(R.Fields(0)) Then
    Label12.Caption = "P" & "00" & 1
Else
    Label12.Caption = "P" & "00" & R.Fields(0) + 1
    a = Label12.Caption
End If
If (a = "P0010") Then
    Set R = New ADODB.Recordset
    SQL = "select max(to_number(SUBSTR(p_id,3,LENGTH(p_id))))from product"
    Set R = C.Execute(SQL)
    Label12.Caption = "P" & "0" & R.Fields(0) + 1
End If
TX2.SetFocus
End Sub
```

```
Private Sub Close_Click()
    Unload Me
End Sub
```

```
Private Sub Combo1_Click()
Set R = New ADODB.Recordset
SQL = "select *from product where p_id=""" + Combo1.Text + """
Set R = C.Execute(SQL)
Label12.Caption = R.Fields(0)
TX2.Text = R.Fields(1)
Tx3.Text = R.Fields(2)
tx4.Text = R.Fields(3)
tx5.Text = R.Fields(4)
tx6.Text = R.Fields(5)
tx7.Text = R.Fields(6)
Tx8.Text = R.Fields(7)
Tx9.Text = R.Fields(8)
End Sub
```

```
Private Sub close2_Click()
searchby.Visible = False
End Sub
```

```
Private Sub delete_Click()
Set R = New ADODB.Recordset
SQL = "delete from product where p_id=""" + Combo1.Text + """
```

```
Set R = C.Execute(SQL)
MsgBox "record deleted"
Adodc1.Refresh
Combo1.Clear
refresh_search
Label12.Caption = " "
TX2.Text = " "
Tx3.Text = " "
tx4.Text = " "
tx5.Text = " "
tx6.Text = " "
tx7.Text = " "
Tx8.Text = " "
Tx9.Text = " "
```

```
End Sub
```

```
Private Sub Form_Load()
CONN
searchby.Visible = False
Adodc1.Visible = False
refresh_search
```

```
End Sub
```

```
Private Sub save_Click()
If Label12.Caption = blank Or TX2.Text = " " Or Tx3.Text = " " Or tx4.Text
= " " Or tx5.Text = " " Or tx6.Text = " " Or tx7.Text = " " Or Tx8.Text = " "
Or Tx9.Text = " " Then
    MsgBox "please fill all the data"
Else
    Set R = New ADODB.Recordset
    SQL = "insert into product values(" + Label12.Caption + ",'" + TX2.Text
    + "','" + Tx3.Text + "','" + tx4.Text + "','" + tx5.Text + "','" + tx6.Text + "','" +
    tx7.Text + "','" + Tx8.Text + "','" + Tx9.Text + "')"
    Set R = C.Execute(SQL)
    MsgBox "record saved"
End If
Adodc1.Refresh
Combo1.Clear
refresh_search
Label12.Caption = " "
TX2.Text = " "
Tx3.Text = " "
tx4.Text = " "
tx5.Text = " "
```

```
tx6.Text = " "
```

```
tx7.Text = " "
```

```
Tx8.Text = " "
```

```
Tx9.Text = " "
```

```
TX2.SetFocus
```

```
End Sub
```

```
Private Sub search_Click()
```

```
searchby.Visible = True
```

```
End Sub
```

```
Private Sub Tx2_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Or KeyAscii = 9 Then Tx3.SetFocus
```

```
End Sub
```

```
Private Sub tx2_LostFocus()
```

```
TX2.Text = UCASE(Mid(TX2.Text, 1, 1)) & Mid(TX2.Text, 2,  
Len(TX2.Text))
```

```
End Sub
```

```
Private Sub Tx3_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Or KeyAscii = 9 Then tx4.SetFocus  
End Sub
```

```
Private Sub tx3_LostFocus()  
Tx3.Text = UCASE(Mid(Tx3.Text, 1, 1)) & Mid(Tx3.Text, 2, Len(Tx3.Text))  
End Sub
```

```
Private Sub Tx4_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Or KeyAscii = 9 Then tx5.SetFocus  
End Sub
```

```
Private Sub tx4_LostFocus()  
tx4.Text = UCASE(Mid(tx4.Text, 1, 1)) & Mid(tx4.Text, 2, Len(tx4.Text))  
End Sub
```

```
Private Sub Tx5_KeyPress(KeyAscii As Integer)  
If KeyAscii >= 48 And KeyAscii <= 57 Or KeyAscii = 8 Then  
    tx5.Locked = False  
Else  
    tx5.Locked = True  
End If  
If KeyAscii = 13 Then tx6.SetFocus
```

End Sub

```
Private Sub Tx6_KeyPress(KeyAscii As Integer)
If KeyAscii >= 48 And KeyAscii <= 57 Or KeyAscii = 8 Then
    tx6.Locked = False
Else
    tx6.Locked = True
End If
If KeyAscii = 13 Then tx7.SetFocus
End Sub

Private Sub Tx7_KeyPress(KeyAscii As Integer)
If KeyAscii >= 48 And KeyAscii <= 57 Or KeyAscii = 8 Then
    tx7.Locked = False
Else
    tx7.Locked = True
End If
If KeyAscii = 13 Then Tx8.SetFocus
End Sub

Private Sub Tx8_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Or KeyAscii = 9 Then Tx9.SetFocus
End Sub
```

```
Private Sub tx8_LostFocus()
Tx8.Text = UCASE(Mid(Tx8.Text, 1, 1)) & Mid(Tx8.Text, 2, Len(Tx8.Text))
End Sub
```

```
Private Sub tx9_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Or KeyAscii = 9 Then save.SetFocus
End Sub
```

```
Public Function refresh_search()
Set R = New ADODB.Recordset
SQL = "select *from product"
Set R = C.Execute(SQL)
While R.EOF = False
Combo1.AddItem R.Fields(0)
R.MoveNext
Wend
End Function
```

```
Private Sub update_Click()
Set R = New ADODB.Recordset
```

```
SQL = "update product set p_nm=""" + TX2.Text + """,p_type=""" +
Tx3.Text + "", p_comp=""" + tx4.Text + "", p_wt=""" + tx5.Text + ",p_gst=""" +
tx6.Text + ",p_rate=""" + tx7.Text + ",p_unit=""" + Tx8.Text + ",p_hsn=""" +
Tx9.Text + "" where p_id=""" + Combo1.Text + """"

Set R = C.Execute(SQL)

MsgBox "record updated"

Adodc1.Refresh

Label12.Caption = " "

TX2.Text = " "

Tx3.Text = " "

tx4.Text = " "

tx5.Text = " "

tx6.Text = " "

tx7.Text = " "

Tx8.Text = " "

Tx9.Text = " "

End Sub

Private Sub Combo1_Click()

Set R = New ADODB.Recordset

SQL = "select *from product where P_id=""" + Combo1.Text + """"

Set R = C.Execute(SQL)

Text1.Text = R.Fields(1)
```

```
TXT11.Text = R.Fields(2)  
gst.Text = R.Fields(5)  
SQL = "select *from supplierprd where P_id="" + Combo1.Text + """  
Set R = C.Execute(SQL)  
txt13.Text = R.Fields(2)
```

End Sub

```
Private Sub Command1_Click()  
a = InputBox("Enter the Serial No. you want to remove:", "for delete")  
If a = blank Then  
    MsgBox "Please enter serial no."  
Else  
    List8.RemoveItem (a - 1)  
    List2.RemoveItem (a - 1)  
    List4.RemoveItem (a - 1)  
    List5.RemoveItem (a - 1)  
    List6.RemoveItem (a - 1)  
    List7.RemoveItem (a - 1)  
    List12.RemoveItem (a - 1)  
    List10.RemoveItem (a - 1)  
    List3.RemoveItem (a - 1)
```

```
List11.RemoveItem (a - 1)
List9.RemoveItem (a - 1)
List1.Clear
For i = 1 To List8.ListCount
    List1.AddItem i
Next i
Dim l As Long
Dim lSum As Long
For l = 0 To List9.ListCount - 1
    lSum = lSum + CLng(List9.List(l))
Next
Text4.Text = lSum

End If

End Sub

Private Sub Form_Load()
CONN
Set R = New ADODB.Recordset
SQL = "select *from purordetail where postatus='Incomplete' or
postatus='Partially Received' "
```

```
Set R = C.Execute(SQL)
While R.EOF = False
    odno.AddItem R.Fields(0)
    R.MoveNext
Wend
MonthView1.Visible = False
MonthView2.Visible = False
gst.Visible = False
MonthView1.Refresh
MonthView2.Refresh
End Sub
```

```
Private Sub MonthView1_DateClick(ByVal DateClicked As Date)
    TXT6.Text = Format(MonthView1, "dd-mmm-yyyy")
    MonthView1.Visible = False
```

```
If TXT6.Text < TXT3.Text Then  
    MsgBox "invalid date"  
    TXT6.Text = " "  
End If  
End Sub
```

```
Private Sub MonthView2_DateClick(ByVal DateClicked As Date)  
    Text3.Text = Format(MonthView2, "dd-mmm-yyyy")  
    MonthView2.Visible = False  
    If Text3.Text < TXT3.Text Or Text3.Text > TXT6.Text Then  
        MsgBox "invalid date"  
        Text3.Text = " "  
    End If  
End Sub
```

```
Private Sub odno_Click()  
    Set R = New ADODB.Recordset  
    SQL = " select *from purordetail where pur_orderno=" + odno.Text +  
    " "  
    Set R = C.Execute(SQL)  
    TXT3.Text = R.Fields(1)
```

```
TXT1.Text = R.Fields("sup_id")
Txt20.Text = R.Fields(5)
txt22.Text = R.Fields(8)
final.Text = R.Fields(7)

Set R = New ADODB.Recordset
SQL = "select *from supplier where sup_id=""" + TXT1.Text + """
Set R = C.Execute(SQL)

TXT2.Text = R.Fields(1)
TXT4.Text = R.Fields(2)
TXT5.Text = R.Fields(7)
TXT8.Text = R.Fields(3)

Set R = New ADODB.Recordset
SQL = "select *from p_det where pur_orderno=""" + odno.Text + """
Set R = C.Execute(SQL)

sr.Clear
pid.Clear
pnm.Clear
ptyp.Clear
prate.Clear
qty1.Clear
prc.Clear
cgst.Clear
```

```
camt.Clear  
sgst.Clear  
samt.Clear  
totprc.Clear  
Combo1.Clear  
While R.EOF = False  
    sr.AddItem R.Fields(1)  
    pid.AddItem R.Fields(2)  
    pnm.AddItem R.Fields(3)  
    ptyp.AddItem R.Fields(4)  
    prate.AddItem R.Fields(5)  
    qty1.AddItem R.Fields(7)  
    prc.AddItem R.Fields(8)  
    cgst.AddItem R.Fields(9)  
    camt.AddItem R.Fields(10)  
    sgst.AddItem R.Fields(11)  
    samt.AddItem R.Fields(12)  
    totprc.AddItem R.Fields(13)  
    Combo1.AddItem R.Fields(2)  
    R.MoveNext  
Wend
```

End Sub

```
Private Sub ok_Click()
List8.AddItem Combo1.Text
List2.AddItem Text1.Text
List4.AddItem TXT11.Text
List5.AddItem txt13.Text
List6.AddItem TXT15.Text
List7.AddItem Val(txt13.Text) * Val(TXT15.Text)
List12.AddItem Val(gst.Text) / 2
List3.AddItem Val(gst.Text) / 2
List10.AddItem (Val(Val(TXT15.Text) * Val(txt13.Text)) * Val(gst.Text) /
100) / 2
List11.AddItem (Val(Val(TXT15.Text) * Val(txt13.Text)) * Val(gst.Text) /
100) / 2
List9.AddItem (Val(Val(TXT15.Text) * Val(txt13.Text)) * Val(gst.Text) /
100) + (Val(txt13.Text) * Val(TXT15.Text))
```

Dim l As Long

Dim lSum As Long

```
For I = 0 To List9.ListCount - 1  
    lSum = lSum + CLng(List9.List(I))  
Next  
Text4.Text = lSum  
  
  
If List1.ListCount = 0 Then  
    List1.AddItem 1  
Else  
    List1.AddItem (List1.ListCount + 1)  
End If  
TXT19.Text = List1.ListCount  
TXT21.Text = Text4.Text  
txt23.Text = Val(TXT21.Text) - Val(txt22.Text)  
Text1.Text = " "  
TXT11.Text = " "  
txt13.Text = " "  
TXT14.Text = " "  
TXT15.Text = " "  
Combo1.Text = " "  
End Sub
```

```

Private Sub save_Click()
If odno.Text = blank Or TXT6.Text = blank Or TXT7.Text = blank Or
Text2.Text = blank Or Text3.Text = blank Or TXT19.Text = blank Or
Txt20.Text = blank Or TXT21.Text = blank Or Text6.Text = blank Then
MsgBox "Please fill the details first !"
Else
Set R = New ADODB.Recordset
SQL = "insert into ordetails values (" + Text2.Text + "," + Text3.Text +
"," + odno.Text + "," + TXT7.Text + "," + TXT19.Text + "," + Text5.Text +
"," + Txt20.Text + "," + TXT21.Text + "," + txt22.Text + "," +
txt23.Text + ")"
Set R = C.Execute(SQL)
MsgBox "Record saved!"

Dim i As Long
For i = 0 To List1.ListCount - 1
SQL = "insert into recvd_p_det values(" + odno.Text + "," + List1.List(i) +
"," + List8.List(i) + "," + List2.List(i) + "," + List4.List(i) + "," +
List5.List(i) + "," + List6.List(i) + "," + List7.List(i) + "," +
List12.List(i) + "," + List10.List(i) + "," + List3.List(i) + "," +
List11.List(i) + "," + List9.List(i) + ")"
Set R = C.Execute(SQL)
Next i

```

```
MsgBox "data saved"
```

```
SQL = "update purordetail set postatus = '" + Text6.Text + "' where  
pur_orderno=''" + odno.Text + "' "
```

```
Set R = C.Execute(SQL)
```

```
Dim k As Long
```

```
For k = 0 To List1.ListCount - 1
```

```
SQL = "update stock set avl_qty= avl_qty+" + List6.List(k) + " where  
p_id=''" + List8.List(k) + """"
```

```
Set R = C.Execute(SQL)
```

```
Next k
```

```
MsgBox "stock updated.....!"
```

```
End If
```

```
Unload Me
```

```
purchase.Show
```

```
End Sub
```

```
Private Sub ST01_Click()
```

```
Text6.Text = ST01.Caption
```

```
End Sub
```

```
Private Sub ST02_Click()  
Text6.Text = ST02.Caption  
End Sub
```

```
Private Sub ST03_Click()  
Text6.Text = ST03.Caption  
End Sub
```

```
Private Sub ST121_Click()  
Text5.Text = ST121.Caption  
End Sub
```

```
Private Sub ST122_Click()  
Text5.Text = ST122.Caption  
Txt20.Text = ST122.Caption  
End Sub
```

```
Private Sub Text3_Click()  
MonthView2.Visible = True  
End Sub
```

```
Private Sub txt14_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then TXT15.SetFocus
End Sub
```

```
Private Sub txt15_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then ok.SetFocus
End Sub
```

```
Private Sub TXT6_Click()
MonthView1.Visible = True
End Sub
```

```
Private Sub Combo1_Click()
If Combo1.Text = "ID" Then
    Combo2.Clear
    auto_p_id
ElseIf Combo1.Text = "NAME" Then
    Combo2.Clear
    auto_P_nm
ElseIf Combo1.Text = "COMPANY" Then
    Combo2.Clear
```

```
auto_P_comp  
ElseIf Combo1.Text = "TYPE" Then  
    Combo2.Clear  
    auto_P_type  
End If
```

```
End Sub
```

```
Private Sub Combo10_CLICK()  
    Command17.Enabled = True  
    Command17.SetFocus  
    Command22.Enabled = True  
End Sub
```

```
Private Sub Combo12_CLICK()  
    If Combo12.Text = "ID" Then  
        Combo11.Clear  
        auto_sup_id  
    ElseIf Combo12.Text = "NAME" Then  
        Combo11.Clear  
        auto_sup_nm
```

```
ElseIf Combo12.Text = "MOBILE NO." Then  
    Combo11.Clear  
    auto_sup_mob  
End If  
End Sub
```

```
Private Sub Combo5_Click()  
If Combo5.Text = "ID" Then  
    Combo4.Clear  
    auto_cust_id  
  
ElseIf Combo5.Text = "MOBILE" Then  
    Combo4.Clear  
    auto_cust_mob  
  
ElseIf Combo5.Text = "NAME" Then  
    Combo4.Clear  
    auto_cust_nm
```

End If

End Sub

```
Private Sub Combo6_Click()
If Combo6.Text = "Order No" Then
    Combo7.Clear
    auto_orderno
ElseIf Combo6.Text = "Invoice No." Then
    Combo7.Clear
    auto_invno
ElseIf Combo6.Text = "Between Dates" Then
    Combo7.Clear
    DATE3.Enabled = True
    Date4.Enabled = True
ElseIf Combo6.Text = "Invoice Date" Then
    Combo7.Clear
    auto_invdate
ElseIf Combo6.Text = "Month" Then
```

```
Combo7.Clear  
Set R = New ADODB.Recordset  
SQL = "select distinct upper(to_char(invdate,'MON')) from ordetails"  
Set R = C.Execute(SQL)  
While R.EOF = False  
    Combo7.AddItem R.Fields(0)  
    R.MoveNext  
Wend  
End If  
  
End Sub
```

```
Private Sub Combo9_Click()  
If Combo9.Text = "Order No" Then  
    Combo17.Clear  
    auto_pur_orderno  
ElseIf Combo9.Text = "Supplier Id" Then  
    Combo17.Clear  
    auto_pur_sup_id  
ElseIf Combo9.Text = "Between Dates" Then  
    Combo17.Clear  
    DATE1.Enabled = True
```

```
DATE2.Enabled = True  
ElseIf Combo9.Text = "Date" Then  
    Combo17.Clear  
    auto_pur_date  
ElseIf Combo9.Text = "Month" Then  
    Combo17.Clear  
    Set R = New ADODB.Recordset  
    SQL = "select distinct upper(to_char(pur_orderdate,'MON')) from  
    purordetail"  
    Set R = C.Execute(SQL)  
    While R.EOF = False  
        Combo17.AddItem R.Fields(0)  
        R.MoveNext  
    Wend  
End If  
End Sub
```

```
Private Sub Command1_Click()  
    Frame3.Visible = True  
    Frame25.Visible = False  
    Frame7.Visible = False
```

```
Frame4.Visible = False  
Frame5.Visible = False  
Frame2.Visible = False  
'Frame11.Visible = False  
Combo1.Clear  
Frame9.Visible = False  
Combo1.AddItem "ID"  
Combo1.AddItem "NAME"  
Combo1.AddItem "COMPANY"  
Combo1.AddItem "TYPE"  
End Sub
```

```
Private Sub Command101_Click()
```

```
If Combo9.Text = "Between Dates" Then  
    DataEnvironment2.Connection1.Open  
    DataEnvironment2.Command14 DATE1.Value, DATE2.Value  
    PUR_BTW_DATE.Show  
    PUR_BTW_DATE.Refresh  
    DataEnvironment2.Connection1.Close
```

```
ElseIf Combo9.Text = blank Or Combo17.Text = blank Then  
    MsgBox "Please select the parameters first.!!"  
Else  
    If Combo9.Text = "Order No" Then  
  
        DataEnvironment2.Connection1.Open  
        DataEnvironment2.Command10 Combo17.Text  
        PUR_ORDERNO.Show  
        PUR_ORDERNO.Refresh  
        DataEnvironment2.Connection1.Close  
  
    ElseIf Combo9.Text = "Supplier Id" Then  
        DataEnvironment2.Connection1.Open  
        DataEnvironment2.Command11 Combo17.Text  
        PUR_SUP_ID.Show  
        PUR_SUP_ID.Refresh  
        DataEnvironment2.Connection1.Close  
  
    ElseIf Combo9.Text = "Date" Then  
        DataEnvironment2.Connection1.Open  
        DataEnvironment2.Command12 Combo17.Text  
        PUR_DATE.Show
```

```
PUR_DATE.Refresh  
DataEnvironment2.Connection1.Close  
  
ElseIf Combo9.Text = "Month" Then  
    DataEnvironment2.Connection1.Open  
    DataEnvironment2.Command13 Combo17.Text  
    pur_month.Show  
    pur_month.Refresh  
    DataEnvironment2.Connection1.Close  
  
End If  
End If  
  
End Sub  
  
Private Sub Command12_Click()  
    allprd_report.Show  
End Sub  
  
Private Sub Command14_Click()  
    allstock_report.Show  
End Sub
```

```
Private Sub Command15_Click()
If Combo12.Text = blank Or Combo11.Text = blank Then
    MsgBox "Please select the parameters first.!!"
Else
    If Combo12.Text = "ID" Then
        DataEnvironment2.Connection1.Open
        DataEnvironment2.Command3 Combo11.Text
        SUPPL_REPORT.Show
        SUPPL_REPORT.Refresh
        DataEnvironment2.Connection1.Close
    ElseIf Combo12.Text = "NAME" Then
        DataEnvironment2.Connection1.Open
        DataEnvironment2.Command1 Combo11.Text
        suppl_report_nm.Show
        suppl_report_nm.Refresh
        DataEnvironment2.Connection1.Close
    ElseIf Combo12.Text = "MOBILE NO." Then
        DataEnvironment2.Connection1.Open
        DataEnvironment2.Command2 Combo11.Text
```

```
suppl_Report_mob.Show  
suppl_Report_mob.Refresh  
DataEnvironment2.Connection1.Close  
  
End If  
End If  
End Sub
```

```
Private Sub Command152_Click()  
If STK1.Text = blank Or STK2.Text = blank Then  
    MsgBox "Please select the parameters first.!!"  
Else  
    If STK1.Text = "ID" Then  
        DataEnvironment2.Connection1.Open  
        DataEnvironment2.Command8 STK2.Text  
        Stock_pid.Show  
        Stock_pid.Refresh  
        DataEnvironment2.Connection1.Close  
  
    Elseif STK1.Text = "RACK NO" Then  
        DataEnvironment2.Connection1.Open
```

```
DataEnvironment2.Command9 STK2.Text  
stock_report_rackno.Show  
stock_report_rackno.Refresh  
DataEnvironment2.Connection1.Close  
End If  
End If  
  
End Sub
```

```
Private Sub Command153_Click()  
  
If Combo6.Text = "Between Dates" Then  
  
    DataEnvironment2.Connection1.Open  
    DataEnvironment2.Command22 DATE3.Value, Date4.Value  
    purst_btwdtate.Show  
    purst_btwdtate.Refresh  
    DataEnvironment2.Connection1.Close  
  
ElseIf Combo6.Text = blank Or Combo7.Text = blank Then  
    MsgBox "Please select the parameters first.!!"
```

Else

If Combo6.Text = "Order No" Then

DataEnvironment2.Connection1.Open

DataEnvironment2.Command20 Combo7.Text

PURST_ORDNO.Show

PURST_ORDNO.Refresh

DataEnvironment2.Connection1.Close

Elself Combo6.Text = "Invoice No." Then

DataEnvironment2.Connection1.Open

DataEnvironment2.Command18 Combo7.Text

purst_invno.Show

purst_invno.Refresh

DataEnvironment2.Connection1.Close

Elself Combo6.Text = "Invoice Date" Then

DataEnvironment2.Connection1.Open

DataEnvironment2.Command19 Combo7.Text

purst_invdate.Show

```
purst_invdate.Refresh  
DataEnvironment2.Connection1.Close  
  
ElseIf Combo6.Text = "Month" Then  
  
    DataEnvironment2.Connection1.Open  
    DataEnvironment2.Command21 Combo7.Text  
    purst_month.Show  
    purst_month.Refresh  
    DataEnvironment2.Connection1.Close
```

```
End If  
End If  
End Sub
```

```
Private Sub Command155_Click()  
If Combo5.Text = blank Or Combo4.Text = blank Then  
    MsgBox "Please select the parameters first.!!"  
Else  
    If Combo5.Text = "ID" Then  
        DataEnvironment2.Connection1.Open
```

```
DataEnvironment2.Command15 Combo4.Text  
cust_Report_id.Show  
cust_Report_id.Refresh  
DataEnvironment2.Connection1.Close  
  
ElseIf Combo5.Text = "MOBILE" Then  
    DataEnvironment2.Connection1.Open  
    DataEnvironment2.Command17 Combo4.Text  
    cust_report_mob.Show  
    cust_report_mob.Refresh  
    DataEnvironment2.Connection1.Close  
  
ElseIf Combo5.Text = "NAME" Then  
    DataEnvironment2.Connection1.Open  
    DataEnvironment2.Command16 Combo4.Text  
    cust_report_nm.Show  
    cust_report_nm.Refresh  
    DataEnvironment2.Connection1.Close  
End If  
End If  
End Sub
```

```
Private Sub Command16_Click()
If Combo1.Text = blank Or Combo2.Text = blank Then
    MsgBox "Please select the parameters first.!!"
Else
    If Combo1.Text = "ID" Then
        DataEnvironment2.Connection1.Open
        DataEnvironment2.Command4 Combo2.Text
        prd_report_id.Show
        prd_report_id.Refresh
        DataEnvironment2.Connection1.Close

    ElseIf Combo1.Text = "NAME" Then
        DataEnvironment2.Connection1.Open
        DataEnvironment2.Command5 Combo2.Text
        prd_report_nm.Show
        prd_report_nm.Refresh
        DataEnvironment2.Connection1.Close

    Elseif Combo1.Text = "COMPANY" Then
        DataEnvironment2.Connection1.Open
        DataEnvironment2.Command6 Combo2.Text
        prd_report_comp.Show
        prd_report_comp.Refresh
```

```
DataEnvironment2.Connection1.Close

ElseIf Combo1.Text = "TYPE" Then
    DataEnvironment2.Connection1.Open
    DataEnvironment2.Command7 Combo2.Text
    prd_report_type.Show
    prd_report_type.Refresh
    DataEnvironment2.Connection1.Close
End If
End If
End Sub
```

```
Private Sub Command17_Click()
If sale1.Text = "Between Dates" Then
    DataEnvironment2.Connection1.Open
    DataEnvironment2.Command27 DT1.Value, DT2.Value
    selReport_btwn_date.Show
    selReport_btwn_date.Refresh
    DataEnvironment2.Connection1.Close

```

```
ElseIf sale1.Text = blank Or Combo10.Text = blank Then
    MsgBox "Please select the parameters first.!!"
```

```
Else
If sale1.Text = "Order Id" Then
    DataEnvironment2.Connection1.Open
    DataEnvironment2.Command23 Combo10.Text
    sel_report_ordno.Show
    sel_report_ordno.Refresh
    DataEnvironment2.Connection1.Close

Elseif sale1.Text = "Customer Id" Then
    DataEnvironment2.Connection1.Open
    DataEnvironment2.Command24 Combo10.Text
    selReport_cust_id.Show
    selReport_cust_id.Refresh
    DataEnvironment2.Connection1.Close

Elseif sale1.Text = "Date" Then
    DataEnvironment2.Connection1.Open
    DataEnvironment2.Command25 Combo10.Text
    selReport_date.Show
    selReport_date.Refresh
    DataEnvironment2.Connection1.Close
```

```
ElseIf sale1.Text = "Month" Then  
    DataEnvironment2.Connection1.Open  
    DataEnvironment2.Command26.Combo10.Text  
    selReport_month.Show  
    selReport_month.Refresh  
    DataEnvironment2.Connection1.Close  
End If  
End If
```

End Sub

```
Private Sub Command2_Click()
```

```
Frame4.Visible = True  
Frame25.Visible = False  
Frame7.Visible = False
```

```
Frame5.Visible = False
```

```
Frame2.Visible = False
```

```
Frame3.Visible = False
```

```
Frame9.Visible = False
```

```
Combo5.Clear
```

```
Combo5.AddItem "ID"
```

```
Combo5.AddItem "MOBILE"
```

```
Combo5.AddItem "NAME"
```

```
End Sub
```

```
Private Sub Command20_Click()
```

```
DataReport2.Show
```

```
End Sub
```

```
Private Sub Command201_Click()
```

```
allpurord_report.Show
```

```
End Sub
```

```
Private Sub Command21_Click()
```

```
allcust_report.Show
```

```
End Sub
```

```
Private Sub Command22_Click()
```

```
Allsel_report.Show
```

```
End Sub
```

```
Private Sub Command24_Click()  
All_ordetail_report.Show  
End Sub
```

```
Private Sub Command29_Click()  
Frame9.Visible = True  
Frame25.Visible = False  
Frame7.Visible = False  
Frame4.Visible = False  
Frame5.Visible = False  
Frame2.Visible = False  
'Frame11.Visible = False  
Frame3.Visible = False
```

```
Combo9.Clear  
Combo9.AddItem "Order No"  
Combo9.AddItem "Supplier Id"  
Combo9.AddItem "Date"  
Combo9.AddItem "Month"  
Combo9.AddItem "Between Dates"
```

End Sub

Private Sub Command3_Click()

Frame7.Visible = True

Frame25.Visible = False

Frame4.Visible = False

Frame5.Visible = False

Frame2.Visible = False

'Frame11.Visible = False

Frame3.Visible = False

Frame9.Visible = False

Combo12.Clear

Combo12.AddItem "ID"

Combo12.AddItem "NAME"

Combo12.AddItem "MOBILE NO."

End Sub

Private Sub Command30_Click()

Frame5.Visible = True

Frame25.Visible = False

```
Frame7.Visible = False
```

```
Frame4.Visible = False
```

```
Frame2.Visible = False
```

```
'Frame11.Visible = False
```

```
Frame3.Visible = False
```

```
Frame9.Visible = False
```

```
Combo6.Clear
```

```
Combo6.AddItem "Order No"
```

```
Combo6.AddItem "Invoice No."
```

```
Combo6.AddItem "Invoice Date"
```

```
Combo6.AddItem "Month"
```

```
Combo6.AddItem "Between Dates"
```

```
End Sub
```

```
Private Sub Command33_Click()
```

```
AllSuppl_report.Show
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
Frame2.Visible = True
```

```
Frame25.Visible = False
```

```
Frame7.Visible = False
```

```
Frame4.Visible = False
```

```
Frame5.Visible = False
```

```
'Frame11.Visible = False
```

```
Frame3.Visible = False
```

```
Frame9.Visible = False
```

```
STK1.Clear
```

```
STK1.AddItem "ID"
```

```
STK1.AddItem "RACK NO"
```

```
End Sub
```

```
Private Sub COMMAND44_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Command45_Click()
```

```
'Frame11.Visible = True
```

```
Frame25.Visible = False
```

```
Frame7.Visible = False
```

```
Frame4.Visible = False
```

```
Frame5.Visible = False
```

```
Frame2.Visible = False
```

```
Frame3.Visible = False
```

```
Frame9.Visible = False
```

```
End Sub
```

```
Private Sub Command48_Click()
```

```
cust_report_dues.Show
```

```
End Sub
```

```
Private Sub Command5_Click()
```

```
Frame25.Visible = True
```

```
Frame7.Visible = False
```

```
Frame4.Visible = False
```

```
Frame5.Visible = False
```

```
Frame2.Visible = False
```

```
'Frame11.Visible = False
```

```
Frame3.Visible = False  
Frame9.Visible = False  
sale1.Clear  
sale1.AddItem "Order Id"  
sale1.AddItem "Customer Id"  
sale1.AddItem "Date"  
sale1.AddItem "Month"  
sale1.AddItem "Between Dates"
```

End Sub

```
Private Sub Command6_Click()  
Frame6.Visible = True  
Frame25.Visible = False  
Frame7.Visible = False  
Frame4.Visible = False  
Frame5.Visible = False  
Frame2.Visible = False  
'Frame11.Visible = False
```

End Sub

```
Private Sub Form_Load()
    CONN
    Frame25.Visible = False
    Frame7.Visible = False
    Frame4.Visible = False
    Frame5.Visible = False
    Frame2.Visible = False
    'Frame11.Visible = False
    Frame3.Visible = False
    Frame9.Visible = False
    If (DataEnvironment2.Connection1.State) Then
        DataEnvironment2.Connection1.Close
    End If
```

```
End Sub
```

```
Public Function auto_sup_id()
    Set R = New ADODB.Recordset
```

```
SQL = "select *from supplier"  
Set R = C.Execute(SQL)  
While R.EOF = False  
    Combo11.AddItem R.Fields(0)  
    R.MoveNext  
Wend  
End Function  
  
Public Function auto_sup_nm()  
Set R = New ADODB.Recordset  
SQL = "select *from supplier"  
Set R = C.Execute(SQL)  
While R.EOF = False  
    Combo11.AddItem R.Fields(1)  
    R.MoveNext  
Wend  
End Function
```

```
Public Function auto_sup_mob()  
Set R = New ADODB.Recordset  
SQL = "select *from supplier"  
Set R = C.Execute(SQL)
```

```
While R.EOF = False  
  Combo11.AddItem R.Fields(2)  
  R.MoveNext  
Wend  
  
End Function
```

```
Public Function auto_p_id()  
  Set R = New ADODB.Recordset  
  SQL = "select *from product"  
  Set R = C.Execute(SQL)  
  While R.EOF = False  
    Combo2.AddItem R.Fields(0)  
    R.MoveNext  
  Wend  
End Function
```

```
Public Function auto_P_nm()  
  Set R = New ADODB.Recordset  
  SQL = "select *from product"  
  Set R = C.Execute(SQL)
```

```
While R.EOF = False  
    Combo2.AddItem R.Fields(1)  
    R.MoveNext  
Wend  
End Function  
  
Public Function auto_P_comp()  
    Set R = New ADODB.Recordset  
    SQL = "select p_comp from product"  
    Set R = C.Execute(SQL)  
    While R.EOF = False  
        Combo2.AddItem R.Fields(0)  
        R.MoveNext  
    Wend  
End Function  
  
Public Function auto_P_type()  
    Set R = New ADODB.Recordset  
    SQL = "select p_type from product"  
    Set R = C.Execute(SQL)  
    While R.EOF = False  
        Combo2.AddItem R.Fields(0)  
        R.MoveNext  
    Wend
```

End Function

Public Function auto_pur_orderno()

Set R = New ADODB.Recordset

SQL = "select *from purordetail order by pur_orderno"

Set R = C.Execute(SQL)

While R.EOF = False

Combo17.AddItem R.Fields(0)

R.MoveNext

Wend

End Function

Public Function auto_orderno()

Set R = New ADODB.Recordset

SQL = "select *from ordetails order by orderno"

Set R = C.Execute(SQL)

While R.EOF = False

Combo7.AddItem R.Fields(2)

R.MoveNext

Wend

End Function

Public Function auto_invno()

Set R = New ADODB.Recordset

```
SQL = "select *from ordetails order by invoiceNo"  
Set R = C.Execute(SQL)  
While R.EOF = False  
    Combo7.AddItem R.Fields(0)  
    R.MoveNext  
Wend  
End Function  
  
Public Function auto_invdate()  
Set R = New ADODB.Recordset  
SQL = "select *from ordetails order by invdate"  
Set R = C.Execute(SQL)  
While R.EOF = False  
    Combo7.AddItem R.Fields(1)  
    R.MoveNext  
Wend  
End Function  
  
Public Function auto_pur_date()  
Set R = New ADODB.Recordset  
SQL = "select *from purordetail order by pur_orderdate"  
Set R = C.Execute(SQL)  
While R.EOF = False  
    Combo17.AddItem R.Fields(1)
```

```
R.MoveNext  
Wend  
End Function  
  
Public Function auto_pur_sup_id()  
Set R = New ADODB.Recordset  
SQL = "select distinct sup_id from purordetail order by sup_id"  
Set R = C.Execute(SQL)  
  
While R.EOF = False  
Combo17.AddItem R.Fields(0)  
  
R.MoveNext  
Wend  
End Function  
  
Public Function auto_cust_id()  
Set R = New ADODB.Recordset  
SQL = "select *from customer"  
Set R = C.Execute(SQL)  
  
While R.EOF = False  
Combo4.AddItem R.Fields(0)  
  
R.MoveNext  
Wend  
End Function  
  
Public Function auto_cust_nm()
```

```
Set R = New ADODB.Recordset
SQL = "select *from customer"
Set R = C.Execute(SQL)
While R.EOF = False
    Combo4.AddItem R.Fields(1)
    R.MoveNext
Wend
End Function

Public Function auto_cust_mob()
Set R = New ADODB.Recordset
SQL = "select *from customer"
Set R = C.Execute(SQL)
While R.EOF = False
    Combo4.AddItem R.Fields(2)
    R.MoveNext
Wend
End Function
```

```
Private Sub sale1_Click()
If sale1.Text = "Order Id" Then
    Combo10.Clear
```

```
auto_sell_orderid  
Elseif sale1.Text = "Customer Id" Then  
    Combo10.Clear  
    auto_sell_cust_id  
Elseif sale1.Text = "Between Dates" Then  
    Combo10.Clear  
    DT1.Enabled = True  
    Format (DT1.Value = "dd-mmm-yyyy")  
    DT2.Enabled = True  
Elseif sale1.Text = "Date" Then  
    Combo10.Clear  
    auto_sell_date  
Elseif sale1.Text = "Month" Then  
    Combo10.Clear  
    Set R = New ADODB.Recordset  
    SQL = "select distinct upper (to_char(s_date, 'MON') ) from  
    sell_details"  
    Set R = C.Execute(SQL)  
    While R.EOF = False  
        Combo10.AddItem R.Fields(0)  
        R.MoveNext  
    Wend
```

End If

End Sub

Private Sub STK1_Click()

If STK1.Text = "ID" Then

STK2.Clear

auto_stock_id

ElseIf STK1.Text = "RACK NO" Then

STK2.Clear

auto_stock_rno

End If

End Sub

Public Function auto_stock_id()

Set R = New ADODB.Recordset

SQL = "select *from stock"

Set R = C.Execute(SQL)

While R.EOF = False

STK2.AddItem R.Fields(1)

```
R.MoveNext  
Wend  
End Function  
  
Public Function auto_stock_rno()  
Set R = New ADODB.Recordset  
SQL = "select *from stock"  
Set R = C.Execute(SQL)  
While R.EOF = False  
STK2.AddItem R.Fields(0)  
R.MoveNext  
Wend  
End Function
```

```
Public Function auto_sell_orderid()  
  
Set R = New ADODB.Recordset  
SQL = "select *from sell_details"  
Set R = C.Execute(SQL)  
While R.EOF = False  
Combo10.AddItem R.Fields(0)  
R.MoveNext  
Wend
```

End Function

Public Function auto_sell_cust_id()

Set R = New ADODB.Recordset

SQL = "select *from sell_details"

Set R = C.Execute(SQL)

While R.EOF = False

Combo10.AddItem R.Fields(2)

R.MoveNext

Wend

End Function

Public Function auto_cust_dues()

Set R = New ADODB.Recordset

SQL = "select *from customer"

Set R = C.Execute(SQL)

While R.EOF = False

Combo4.AddItem R.Fields(6)

R.MoveNext

Wend

End Function

Public Function auto_sell_date()

```
Set R = New ADODB.Recordset  
SQL = "select *from sell_details"  
Set R = C.Execute(SQL)  
While R.EOF = False  
    Combo10.AddItem R.Fields(1)  
    R.MoveNext  
Wend  
End Function
```

```
Private Sub Form_Load()  
CONN  
End Sub
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then UPDATE.SetFocus  
End Sub
```

```
Private Sub update_Click()
```

```
Set R = New ADODB.Recordset
SQL = "SELECT * FROM LOGIN"
Set R = C.Execute(SQL)
If (Text1.Text = Text2.Text) Then
    SQL = "UPDATE LOGIN SET PASS=""" + Text1.Text + """
    Set R = C.Execute(SQL)
    MsgBox "PASSWORD UPDATED"
    Unload Me
    LOGINFORM.Show
Else
    MsgBox "PASSWORD NOT MATCHED"
    Text1.Text = ""
    Text2.Text = ""
End If
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then Text2.SetFocus
End Sub
```

```
Private Sub Combo1_Click()
Set R = New ADODB.Recordset
```

```
SQL = "select *from product where p_id="" + Combo1.Text + """
Set R = C.Execute(SQL)
Combo2.Text = R.Fields(1)
Combo3.Text = R.Fields(7)
TXT8.Text = R.Fields(6)
TX13.Text = R.Fields(5)
Set R = New ADODB.Recordset
SQL = "select *from stock where p_id="" + Combo1.Text + """
Set R = C.Execute(SQL)
Tx9.Text = R.Fields(0)
Tx10.Text = R.Fields(2)
End Sub
```

```
Private Sub Combo4_Click()
Set R = New ADODB.Recordset
SQL = "select *from customer where c_nm="" + Combo4.Text + """
Set R = C.Execute(SQL)
Combo6.Text = R.Fields(0)
TX2.Text = R.Fields(2)
tx4.Text = R.Fields(3)
Text2.Text = R.Fields(4)
txt18.Text = R.Fields(6)
```

```
End Sub
```

```
Private Sub Combo6_Click()
    Combo6.Refresh
    Set R = New ADODB.Recordset
    SQL = "select *from customer where c_id=" + Combo6.Text + ""
    Set R = C.Execute(SQL)
    Combo4.Text = R.Fields(1)
    TX2.Text = R.Fields(2)
    tx4.Text = R.Fields(3)
    Text2.Text = R.Fields(4)
    txt18.Text = R.Fields(6)
End Sub
```

```
Private Sub Command1_Click()
    customer.Show
End Sub
```

```
Public Function auto_c_id()
    Set R = New ADODB.Recordset
    SQL = "select *from customer order by c_id"
    Set R = C.Execute(SQL)
```

```
While R.EOF = False  
    Combo6.AddItem R.Fields(0)  
    R.MoveNext  
Wend  
End Function
```

```
Public Function auto_c_nm()  
    Set R = New ADODB.Recordset  
    SQL = "select *from customer order by c_nm"  
    Set R = C.Execute(SQL)  
    While R.EOF = False  
        Combo4.AddItem R.Fields(1)  
        R.MoveNext  
    Wend  
End Function  
  
Public Function auto_p_id()  
    Set R = New ADODB.Recordset  
    SQL = "select *from stock order by p_id"  
    Set R = C.Execute(SQL)  
    While R.EOF = False  
        Combo1.AddItem R.Fields(1)  
        R.MoveNext
```

Wend
End Function

```
Private Sub Command2_Click()
id.AddItem Combo1.Text
PN.AddItem Combo2.Text
unit.AddItem Combo3.Text
MRP.AddItem TXT8.Text
qty.AddItem TX11.Text
prc.AddItem TX12.Text
net.AddItem TX14.Text
cgst.AddItem Val(TX13.Text) / 2
sgst.AddItem Val(TX13.Text) / 2
cmt.AddItem ((Val(TX13.Text) / 2) * Val(TX12.Text)) / 100
smt.AddItem ((Val(TX13.Text) / 2) * Val(TX12.Text)) / 100
Combo1.Text = " "
Combo2.Text = " "
Combo3.Text = " "
TXT8.Text = " "
TX11.Text = " "
TX12.Text = " "
TX14.Text = " "
```

```
TX13.Text = " "
Tx9.Text = " "
Tx10.Text = " "
If sr.ListCount = 0 Then
    sr.AddItem 1
Else
    sr.AddItem (sr.ListCount + 1)
End If
Dim l As Long
Dim lSum As Long
For l = 0 To net.ListCount - 1
    lSum = lSum + CLng(net.List(l))
Next
final.Text = lSum
```

```
Dim f As Long
Dim fSum As Long
For f = 0 To cmt.ListCount - 1
    fSum = fSum + CLng(cmt.List(f))
Next
camnt.Text = fSum
```

```
TXT15.Text = sr.ListCount  
  
TXT19.Text = final.Text  
  
'TXT20.Text = Val(TXT19.Text)  
  
End Sub  
  
Private Sub Command5_Click()  
  
'If sr.ListCount = 0 Then  
  
'MsgBox "please add product first"  
  
a = InputBox("Enter the Serial No. you want to remove:", "for delete")  
  
If a = blank Then  
  
    MsgBox "Please enter serial no."  
  
Else  
  
    TXT19.Text = Val(TXT19.Text) - net.List(a - 1)  
  
    final.Text = Val(TXT19.Text)  
  
    Txt20.Text = Val(final.Text)  
  
    id.RemoveItem (a - 1)  
  
    PN.RemoveItem (a - 1)  
  
    unit.RemoveItem (a - 1)  
  
    MRP.RemoveItem (a - 1)  
  
    qty.RemoveItem (a - 1)  
  
    prc.RemoveItem (a - 1)  
  
    net.RemoveItem (a - 1)  
  
    cgst.RemoveItem (a - 1)
```

```
sgst.RemoveItem (a - 1)
cmt.RemoveItem (a - 1)
smt.RemoveItem (a - 1)
sr.Clear
For i = 1 To id.ListCount
    sr.AddItem i
Next i
Dim l As Long
Dim lSum As Long
For l = 0 To prc.ListCount - 1
    lSum = lSum + CLng(prc.List(l))
Next
TXT17.Text = lSum
TXT15.Text = sr.ListCount
Txt20.Text = Val(txt18.Text) + Val(Txt20.Text)
End If

End Sub
```

```
Private Sub Command6_Click()
If Text1.Text = blank Or Combo6.Text = blank Or TXT6.Text = blank Or
    TXT15.Text = blank Or Text4.Text = blank Or TXT16.Text = blank Or
    TXT17.Text = blank Or TXT21.Text = blank Then
```

```
MsgBox "Please fill the details first !!"  
Else  
Set R = New ADODB.Recordset  
SQL = "insert into sell_details values('' + Text1.Text + "", '' + TXT6.Text +  
'', '' + Combo6.Text + "", '' + TXT15.Text + "", '' + Text4.Text + "", '' +  
TXT16.Text + "", '' + TXT17.Text + "", '' + TXT19.Text + "", '' + Text3.Text + "",  
+ TXT21.Text + ")"
```

```
Set R = C.Execute(SQL)
```

```
Dim i As Long  
For i = 0 To sr.ListCount - 1  
SQL = "insert into sold_pdet values('' + Text1.Text + "", '' + id.List(i) + "",  
+ qty.List(i) + "", '' + net.List(i) + ")"  
Set R = C.Execute(SQL)
```

```
Next
```

```
Dim k As Long  
For k = 0 To sr.ListCount - 1  
SQL = "update stock set avl_qty= avl_qty- " + qty.List(k) + " where  
p_id= '' + id.List(k) + """  
Set R = C.Execute(SQL)  
Next k
```

```
SQL = "update customer set dues=" + Text3.Text + " where c_id="" +
Combo6.Text + """
Set R = C.Execute(SQL)
MsgBox "Sell Completed!!"
'Unload Me
'Sell.Show
SALE_INVOICE.Show
SALE_INVOICE.Top = 0
SALE_INVOICE.Left = 0
SALE_INVOICE.Text2.Text = Sell.Text1.Text
SALE_INVOICE.Text3.Text = Sell.Combo6.Text
SALE_INVOICE.TXT3.Text = Sell.Combo4.Text
SALE_INVOICE.TXT4.Text = Sell.Text2.Text
SALE_INVOICE.Text6.Text = Sell.tx4.Text
SALE_INVOICE.Text8.Text = Sell.TX2.Text
Dim m As Long
For m = 0 To sr.ListCount - 1
SALE_INVOICE.List1.AddItem Sell.PN.List(m)
SALE_INVOICE.List2.AddItem Sell.MRP.List(m)
SALE_INVOICE.List3.AddItem Sell.qty.List(m)
```

```
SALE_INVOICE.List4.AddItem Sell.prc.List(m)
SALE_INVOICE.List5.AddItem Sell.cmt.List(m)
SALE_INVOICE.List6.AddItem Sell.smt.List(m)
SALE_INVOICE.List7.AddItem Sell.net.List(m)
```

Next

```
SALE_INVOICE.Txt88.Text = Sell.TXT19.Text
SALE_INVOICE.Text13.Text = Sell.Txt20.Text
SALE_INVOICE.Text15.Text = Sell.TXT21.Text
```

```
SALE_INVOICE.paidby.Text = Sell.Text4.Text
SALE_INVOICE.Text10.Text = Sell.TXT17.Text
SALE_INVOICE.Text4.Text = Sell.txt18.Text
SALE_INVOICE.Text5.Text = Sell.Text3.Text
SALE_INVOICE.cgmt.Text = Sell.camnt.Text
SALE_INVOICE.sgmt.Text = Sell.camnt.Text
```

End If

End Sub

```
Private Sub Command7_Click()
```

Unload Me

End Sub

```
Private Sub final_Change()
Dim I As Long
Dim ISum As Long
For I = 0 To prc.ListCount - 1
    ISum = ISum + CLng(prc.List(I))
Next
TXT17.Text = ISum
End Sub
```

```
Private Sub Form_Load()
CONN
auto_c_id
auto_c_nm
auto_p_id

Dim a As String
Set R = New ADODB.Recordset
SQL = "select max(to_number(SUBSTR(s_ono,5,LENGTH(s_ono))))from
sell_details"
```

```
Set R = C.Execute(SQL)
If IsNull(R.Fields(0)) Then
    Text1.Text = "SO" & "00" & 1
Else
    Text1.Text = "SO" & "00" & R.Fields(0) + 1
    a = Text1.Text
End If
If (a = "SO0010") Then
    Set R = New ADODB.Recordset
    SQL = "select max(to_number(SUBSTR(s_ono,4,LENGTH(s_ono))))from
    sell_details"
    Set R = C.Execute(SQL)
    Text1.Text = "SO" & "0" & R.Fields(0) + 1
End If
Text2.Visible = False
MonthView1.Visible = False
camnt.Visible = False
Text4.Visible = False
End Sub
```

```
Private Sub MonthView1_DateClick(ByVal DateClicked As Date)
    Dim d1, d2 As String
    d1 = MonthView1.Value
    d2 = Date
    MonthView1.Visible = False
    If d1 > d2 Then
        MsgBox "Don't enter future date"
        TXT6.Text = ""
    Else
        TXT6.Text = Format(MonthView1, "dd-mmm-yyyy")
    End If
End Sub

Private Sub Text2_Change()
    If Text2.Text = "Male" Then
        Option1.Value = True
        Option2.Value = False
        Option3.Value = False
    End If
End Sub
```

```
ElseIf Text2.Text = "Female" Then  
    Option2.Value = True  
    Option1.Value = False  
    Option3.Value = False  
  
ElseIf Text2.Text = "Transgender" Then  
    Option3.Value = True  
    Option1.Value = False  
    Option2.Value = False  
  
End If  
  
End Sub
```

```
Private Sub TX11_Change()  
  
If Val(TX11.Text) > Val(Tx10.Text) Then  
    MsgBox "not enough quantity"  
    TX11.Text = ""  
  
End If  
  
    TX12.Text = Val(TXT8.Text) * Val(TX11.Text)  
    TX14.Text = Val(TX12.Text) + (Val(TX12.Text) * (Val(TX13.Text) / 100))  
  
End Sub
```

```
Private Sub TXT131_Click()
```

```
Label451.Visible = False
```

```
Label450.Visible = True
```

```
TXT16.Text = "CASH"
```

```
Text4.Text = "Cash"
```

```
End Sub
```

```
Private Sub TXT132_Click()
```

```
Label450.Visible = False
```

```
Label451.Visible = True
```

```
TXT16.Text = ""
```

```
Text4.Text = "Cheque"
```

```
End Sub
```

```
Private Sub TXT19_Change()
```

```
Txt20.Text = Val(txt18.Text) + Val(TXT19.Text)
```

```
End Sub
```

```
Private Sub TXT21_Change()
```

```
Text3.Text = Val(Txt20.Text) - Val(TXT21.Text)
```

```
End Sub
```

```
Private Sub TXT6_Click()
MonthView1.Visible = True
End Sub
```

```
Private Sub Command1_Click()
Set R = New ADODB.Recordset
SQL = "insert into sell_inv values('" + Text2.Text + "','" +
Format(TXT1.Text, "dd-MMM-yyyy") + "','" + Text13.Text + "','" +
Text1.Text + "')"
Set R = C.Execute(SQL)
MsgBox "Invoice Saved"
Me.PrintForm
End Sub
```

```
Private Sub Form_Load()
CONN
Text2.Text = sale.Text1.Text

Dim a As String
Set R = New ADODB.Recordset
SQL = "select
max(to_number(SUBSTR(inv_no,6,LENGTH(inv_no))))from sell_inv"
```

```
Set R = C.Execute(SQL)
If IsNull(R.Fields(0)) Then
    Text1.Text = "INV" & "00" & 1
Else
    Text1.Text = "INV" & "00" & R.Fields(0) + 1
    a = Text1.Text
End If
If (a = "INV0010") Then
    Set R = New ADODB.Recordset
    SQL = "select
        max(to_number(SUBSTR(inv_no,5,LENGTH(inv_no))))from sell_inv"
    Set R = C.Execute(SQL)
    Text1.Text = "INV" & "0" & R.Fields(0) + 1
End If
TXT1.Text = Date
End Sub
```

```
Private Sub close2_Click()
    Frame2.Visible = False
End Sub
```

```
Public Function auto_p_id()
```

```
Set R = New ADODB.Recordset  
SQL = "select *from stock"  
Set R = C.Execute(SQL)  
While R.EOF = False  
    Combo1.AddItem R.Fields(1)  
    R.MoveNext  
Wend  
End Function
```

```
Private Sub Combo1_Click()  
Set R = New ADODB.Recordset  
SQL = "select *from stock where P_ID=" + Combo1.Text + """  
Set R = C.Execute(SQL)  
Text1.Text = R.Fields(0)  
Combo2.Text = R.Fields(1)  
Text3.Text = R.Fields(2)  
Text4.Text = R.Fields(3)  
Text5.Text = R.Fields(4)  
End Sub
```

```
Private Sub Command2_Click()
```

```
If Text1.Text = blank Or Combo2.Text = blank Or Text3.Text = blank Or  
Text4.Text = blank Or Text5.Text = blank Then  
    MsgBox "Please enter the details first!!"  
Else  
    Set R = New ADODB.Recordset  
    SQL = "insert into stock values("" + Text1.Text + "", "" + Combo2.Text +  
        "", "" + Text3.Text + "", "" + Text4.Text + "", "" + Text5.Text + ")"  
    Set R = C.Execute(SQL)  
    MsgBox "data saved"  
    Unload Me  
    stock.Show  
    stock.Top = 0  
    stock.Left = 0  
  
End If  
End Sub
```

```
Private Sub Command3_Click()  
    Set R = New ADODB.Recordset  
    SQL = "update stock set RNO="" + Text1.Text + "", P_Id="" +  
        Combo2.Text + "", AVL_QTY="" + Text3.Text + ", min_qty="" + Text4.Text +  
        ", max_qty="" + Text5.Text + " where P_ID="" + Combo1.Text + """  
    Set R = C.Execute(SQL)
```

```
MsgBox "record updated"
```

```
Adodc1.Refresh
```

```
Combo3.Text = " "
```

```
Combo2.Text = " "
```

```
Text3.Text = " "
```

```
Text4.Text = " "
```

```
Text5.Text = " "
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
Frame2.Visible = True
```

```
End Sub
```

```
Private Sub Command6_Click()
```

```
Set R = New ADODB.Recordset
```

```
SQL = "Delete from stock where p_id=" + Combo1.Text + """"
```

```
Set R = C.Execute(SQL)
```

```
MsgBox "Record deleted"
```

```
Adodc1.Refresh
```

```
Combo3.Clear
```

```
auto_p_id
```

```
Combo2.Text = " "
```

```
Text3.Text = " "
```

```
Text4.Text = " "
```

```
Text5.Text = " "
```

```
End Sub
```

```
Private Sub Command7_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Command8_Click()
```

```
Dim i As String
```

```
'i = Combo3.ListCount
```

```
'Combo3.AddItem i + 1
```

```
Set R = New ADODB.Recordset
```

```
SQL = "select count (rno) from stock"
```

```
Set R = C.Execute(SQL)
```

```
i = R.Fields(0)
```

```
Text1.Text = i + 1
```

```
End Sub
```

```
Private Sub Form_Load()
CONN
Text1.Locked = True
Adodc1.Visible = False
Set R = New ADODB.Recordset
SQL = "select distinct p_id from product where p_id not in (select p_id
from stock)"
Set R = C.Execute(SQL)
While R.EOF = False
Combo2.AddItem R.Fields(0)
R.MoveNext
Wend
auto_p_id
'Text1.SetFocus
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text2.SetFocus
```

```
End If

End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text3.SetFocus
End If
End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text4.SetFocus
End If
End Sub

Private Sub Text4_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text5.SetFocus
End If
End Sub

Private Sub Text5_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Command2.SetFocus
End If
End Sub
```

```
Private Sub add_Click()
Dim a As String
Set R = New ADODB.Recordset
SQL = "select
max(to_number(SUBSTR(SUP_ID,4,LENGTH(SUP_ID))))from supplier"
Set R = C.Execute(SQL)
If IsNull(R.Fields(0)) Then
Label21.Caption = "S" & "00" & 1
Else
Label21.Caption = "S" & "00" & R.Fields(0) + 1
a = Label21.Caption
End If
If (a = "S0010") Then
Set R = New ADODB.Recordset
SQL = "select
max(to_number(SUBSTR(SUP_ID,3,LENGTH(SUP_ID))))from supplier"
Set R = C.Execute(SQL)
Label21.Caption = "S" & "0" & R.Fields(0) + 1
End If
TX2.SetFocus
TX2.Text = " "
Tx3.Text = " "
tx4.Text = " "
```

```
tx5.Text = " "
tx6.Text = " "
tx7.Text = " "
Tx8.Text = " "
Tx9.Text = " "
Tx10.Text = " "
End Sub
```

```
Private Sub Close_Click()
Unload Me
End Sub
```

```
Private Sub close2_Click()
searchby.Visible = False
End Sub
```

```
Private Sub Combo1_Click()
Set R = New ADODB.Recordset
SQL = "select *from supplier where SUP_ID=" + Combo1.Text + """
Set R = C.Execute(SQL)
Label21.Caption = R.Fields(0)
TX2.Text = R.Fields(1)
```

```
Tx3.Text = R.Fields(2)
```

```
tx4.Text = R.Fields(3)
```

```
tx5.Text = R.Fields(4)
```

```
tx6.Text = R.Fields(5)
```

```
tx7.Text = R.Fields(6)
```

```
Tx8.Text = R.Fields(7)
```

```
Tx9.Text = R.Fields(8)
```

```
Tx10.Text = R.Fields(9)
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
If sr.ListCount = 0 Then
```

```
sr.AddItem 1
```

```
Else
```

```
sr.AddItem (sr.ListCount + 1)
```

```
End If
```

```
id.AddItem Combo2.Text
```

```
rate.AddItem Text1.Text
```

```
Combo2.Text = " "
```

```
Text1.Text = " "
```

```
End Sub
```

```
Private Sub Command2_Click()
a = InputBox("Enter the Serial No. you want to remove:", "for delete")
If a = blank Then
    MsgBox "Please enter serial no."
Else
    id.RemoveItem (a - 1)
    rate.RemoveItem (a - 1)
    sr.Clear
    For i = 1 To id.ListCount
        sr.AddItem i
    Next i
End If

End Sub
```

```
Private Sub delete_Click()
Set R = New ADODB.Recordset
SQL = "update supplier set status='Inactive' where sup_id=" + 
Combo1.Text + """"
```

```
'SQL = "delete from supplier where sup_id="" + Combo1.Text + """  
Set R = C.Execute(SQL)  
MsgBox "record deleted"  
Adodc1.Refresh  
Combo1.Clear  
auto_sup_id  
Label21.Caption = " "  
TX2.Text = " "  
Tx3.Text = " "  
tx4.Text = " "  
tx5.Text = " "  
tx6.Text = " "  
tx7.Text = " "  
Tx8.Text = " "  
Tx9.Text = " "  
Tx10.Text = " "  
End Sub
```

```
Private Sub Form_Load()  
CONN  
Adodc1.Visible = False  
searchby.Visible = False
```

```
auto_sup_id  
auto_prd_id  
Adodc1.Refresh  
Label27.Visible = False  
Text2.Visible = False  
End Sub
```

```
Private Sub save_Click()
```

```
If Label21.Caption = " " Or TX2.Text = " " Or Tx3.Text = " " Or tx4.Text =  
" " Or tx5.Text = " " Or tx6.Text = " " Or tx7.Text = " " Or Tx8.Text = " "  
Or Tx9.Text = " " Or Tx10.Text = " " Then
```

```
MsgBox "Please click on ADD NEW button first and fill details!!"
```

```
Else
```

```
Set R = New ADODB.Recordset
```

```
SQL = "select *from Supplier"
```

```
Set R = C.Execute(SQL)
```

```
While R.EOF = False
```

```
If (R.Fields(0) = Label21.Caption) Then
```

```
    Text2.Text = Label21.Caption
```

```
End If
```

```
R.MoveNext
```

Wend

```
If (Text2.Text = Label21.Caption) Then  
    Set R = New ADODB.Recordset  
    SQL = "insert into supplierprd values('" + Label21.Caption + "','" +  
id.List(i) + "','" + rate.List(i) + "')"  
    Set R = C.Execute(SQL)  
    MsgBox "record saved"  
  
Else  
    Set R = New ADODB.Recordset  
    SQL = "insert into supplier values('" + Label21.Caption + "','" +  
TX2.Text + "','" + Tx3.Text + "','" + tx4.Text + "','" + tx5.Text + "','" +  
tx6.Text + "','" + tx7.Text + "','" + Tx8.Text + "','" + Tx9.Text + "','" +  
Tx10.Text + "','" + Label27.Caption + "')"  
    Set R = C.Execute(SQL)  
  
    MsgBox "record saved"  
  
Adodc1.Refresh  
Combo1.Clear
```

```
auto_sup_id  
Dim p As Long  
For p = 0 To sr.ListCount - 1  
    SQL = "insert into supplierprd values('" + Label21.Caption + "','" +  
id.List(p) + "','" + rate.List(p) + "')"  
  
    Set R = C.Execute(SQL)  
    Next p  
    MsgBox "data saved"  
End If  
  
End If  
  
Label21.Caption = " "  
TX2.Text = " "  
Tx3.Text = " "  
tx4.Text = " "  
tx5.Text = " "  
tx6.Text = " "  
tx7.Text = " "  
Tx8.Text = " "
```

```
Tx9.Text = " "
Tx10.Text = " "
Combo2.Text = " "
Text1.Text = " "
sr.Clear
id.Clear
rate.Clear
End Sub
```

```
Private Sub search_Click()
searchby.Visible = True
End Sub
```

```
Private Sub Tx2_KeyPress(KeyAscii As Integer)
If KeyAscii >= 65 And KeyAscii <= 90 Or KeyAscii >= 97 And KeyAscii <=
122 Or KeyAscii = 8 Or KeyAscii = 32 Then
    TX2.Locked = False
Else
    TX2.Locked = True
End If
If KeyAscii = 13 Or KeyAscii = 9 Then Tx3.SetFocus
```

```
End Sub
```

```
Private Sub tx2_LostFocus()  
    TX2.Text = UCASE(Mid(TX2.Text, 1, 1)) & Mid(TX2.Text, 2,  
    Len(TX2.Text))
```

```
End Sub
```

```
Private Sub Tx3_KeyPress(KeyAscii As Integer)  
    If KeyAscii >= 48 And KeyAscii <= 57 Or KeyAscii = 8 Then  
        Tx3.Locked = False  
    Else
```

```
        Tx3.Locked = True  
    End If  
    If KeyAscii = 13 Then tx4.SetFocus  
End Sub
```

```
Private Sub Tx4_KeyPress(KeyAscii As Integer)  
    If Len(tx4.Text) = 50 And KeyAscii <> 8 Then  
        tx4.Locked = True  
    ElseIf KeyAscii >= 48 And KeyAscii <= 57 Or KeyAscii >= 65 And KeyAscii  
    <= 90 Or KeyAscii >= 97 And KeyAscii <= 122 Or KeyAscii = 8 Or KeyAscii  
    = 32 Or KeyAscii = 44 Then  
        tx4.Locked = False
```

```
ElseIf KeyAscii = 13 Then
```

```
    tx5.SetFocus
```

```
Else
```

```
    tx4.Locked = True
```

```
End If
```

```
End Sub
```

```
Private Sub tx4_LostFocus()
```

```
    tx4.Text = UCASE(Mid(tx4.Text, 1, 1)) & Mid(tx4.Text, 2, Len(tx4.Text))
```

```
End Sub
```

```
Private Sub Tx5_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii >= 65 And KeyAscii <= 90 Or KeyAscii >= 97 And KeyAscii <= 122 Or KeyAscii = 8 Or KeyAscii = 32 Then
```

```
    tx5.Locked = False
```

```
Else
```

```
    tx5.Locked = True
```

```
End If
```

```
If KeyAscii = 13 Then tx6.SetFocus
```

```
End Sub
```

```
Private Sub tx5_LostFocus()
```

```
tx5.Text = UCASE(MID(tx5.Text, 1, 1)) & MID(tx5.Text, 2, LEN(tx5.Text))
End Sub
```

```
Private Sub Tx6_KeyPress(KeyAscii As Integer)
If KeyAscii >= 65 And KeyAscii <= 90 Or KeyAscii >= 97 And KeyAscii <=
122 Or KeyAscii = 8 Or KeyAscii = 32 Then
    tx6.Locked = False
Else
    tx6.Locked = True
End If
If KeyAscii = 13 Then tx7.SetFocus
End Sub
```

```
Private Sub tx6_LostFocus()
tx6.Text = UCASE(MID(tx6.Text, 1, 1)) & MID(tx6.Text, 2, LEN(tx6.Text))
End Sub
```

```
Private Sub Tx7_KeyPress(KeyAscii As Integer)
If KeyAscii >= 48 And KeyAscii <= 57 Or KeyAscii = 8 Then
    tx7.Locked = False
Else
    tx7.Locked = True
End If
```

```
End If

If KeyAscii = 13 Then Tx8.SetFocus

End Sub

Private Sub Tx8_KeyPress(KeyAscii As Integer)
If Len(Tx8.Text) = 30 And KeyAscii <> 8 Then
    Tx8.Locked = True
ElseIf KeyAscii >= 48 And KeyAscii <= 57 Or KeyAscii >= 65 And KeyAscii
<= 90 Or KeyAscii >= 97 And KeyAscii <= 122 Or KeyAscii = 8 Or KeyAscii
= 32 Or KeyAscii = 44 Then
    Tx8.Locked = False
ElseIf KeyAscii = 13 Then
    Tx9.SetFocus
Else
    Tx8.Locked = True
End If

End Sub
```

```
Private Sub tx8_LostFocus()
Tx8.Text = UCase(Mid(Tx8.Text, 1, 1)) & Mid(Tx8.Text, 2, Len(Tx8.Text))
End Sub
```

```
Private Sub tx9_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then Tx10.SetFocus
```

End Sub

Private Sub update_Click()

Set R = New ADODB.Recordset

SQL = "update supplier set SUP_NM=""" + TX2.Text + """,SUP_MOB=""" +
Tx3.Text + "", SUP_LOCATION=""" + tx4.Text + "", SUP_STATE=""" +
tx5.Text + "", SUP_CITY=""" + tx6.Text + "", SUP_PINCODE=""" + tx7.Text +
",COM=""" + Tx8.Text + "", SUP_EMAIL=""" + Tx9.Text + "", SUP_GSTNO=""" +
Tx10.Text + "" "where SUP_ID=""" + Combo1.Text + """"

Set R = C.Execute(SQL)

MsgBox "record updated"

Adodc1.Refresh

Label21.Caption = " "

TX2.Text = " "

Tx3.Text = " "

tx4.Text = " "

tx5.Text = " "

tx6.Text = " "

tx7.Text = " "

Tx8.Text = " "

Tx9.Text = " "

Tx10.Text = " "

End Sub

```
Public Function auto_sup_id()
Set R = New ADODB.Recordset
SQL = "select *from supplier"
Set R = C.Execute(SQL)
While R.EOF = False
Combo1.AddItem R.Fields(0)
R.MoveNext
Wend
End Function

Public Function auto_prd_id()
Set R = New ADODB.Recordset
SQL = "select *from product"
Set R = C.Execute(SQL)
While R.EOF = False
Combo2.AddItem R.Fields(0)
R.MoveNext
Wend
End Function
```

```
Private Sub Command1_Click()
Set R = New ADODB.Recordset
```

```
SQL = "SELECT * FROM LOGIN"
Set R = C.Execute(SQL)
If (Text1.Text = R.Fields(0) And Text2.Text = R.Fields(1)) Then
    MsgBox "USER VERIFIED"
    Unload Me
    ADDUSER.Show
    Text1.Text = ""
    Text2.Text = ""
    Elseif Text1.Text = "" And Text2.Text = "" Then
        MsgBox ("PLEASE ENTER USER ID AND PASSWORD")
        Text1.SetFocus
        Elseif Text2.Text = "" And Text1.Text <> R.Fields(0) Then
            MsgBox ("PLEASE ENTER PASSWORD AND INPUTED USER ID IS
WRONG")
            Text1.SetFocus
            Elseif Text1.Text = "" And Text2.Text <> R.Fields(1) Then
                MsgBox ("PLEASE ENTER USER ID AND INPUTED PASSWORD IS
WORNG")
                Text1.SetFocus
                Elseif Text2.Text = "" Then
                    MsgBox ("PLEASE ENTER PASSWORD")
                    Text2.SetFocus
                    Elseif Text1.Text <> R.Fields(0) Then
```

```
MsgBox ("INVALID USER ID ENTER CORRECT USER ID")
Text1.SetFocus
ElseIf Text2.Text <> R.Fields(1) Then
    MsgBox ("INVALID PASSWORD ENTER CORRECT PASSWORD")
    Text2.Text = ""
    Text2.SetFocus
Else
    MsgBox "PLEASE ENTER CORRECT USER ID AND PHONE NUMBER"
    Text1.SetFocus
    Text1.Text = ""
    Text2.Text = ""
End If
End Sub

Private Sub Form_Load()
    CONN
End Sub

Private Sub Image2_Click()
    If (Text2.PasswordChar = "*") Then
        Text2.PasswordChar = ""
    Else
        Text2.PasswordChar = "*"
    End If
End Sub
```

End If

End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)

If KeyAscii = 13 Then Text2.SetFocus

End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)

If KeyAscii = 13 Then Command1.SetFocus

End Sub

SQL CODING :-

```
create user PRJ2133F1 identified by abhiram;  
grant resource,connect to PRJ2133F1;  
grant dba to PRJ2133F1;  
conn PRJ2133F1/abhiram;
```

```
create table login(USER_ID varchar(20), PASS varchar(20),  
PHONE NUMBER varchar(10));  
insert into login values('ADMIN','ADMIN','1234567890');
```

```
create table product(  
    p_id varchar(10) constraint pkprd primary key,  
    p_nm varchar(30) not null,  
    p_type char(10) not null,  
    p_comp char(20) not null,  
    p_wt decimal(5,2) not null,  
    p_gst decimal(4,2) not null,  
    p_rate decimal(6,2) not null,  
    p_unit varchar (5) not null,  
    p_hsn varchar(8) not null);
```

```
create table supplier(  
    sup_id varchar(10) constraint pksupp primary key,  
    sup_nm varchar(20) not null,  
    sup_mob number(13) not null,  
    sup_location varchar(50) not null,  
    sup_state varchar(10) not null,  
    sup_city varchar(10) not null,  
    sup_pincode number(6) not null,  
    com varchar(30) not null,  
    sup_email varchar(30) not null,  
    sup_gstno varchar(15) not null,  
    status varchar(15));
```

```
create table supplierprd(  
    Sup_id varchar(10) references supplier (sup_id),  
    p_id varchar(10) references product(p_id),  
    rate number(6,2) );
```

```
create table purordetail(  
    pur_orderno varchar(10) constraint pkprodet primary key,  
    pur_orderdate date not null,
```

```
sup_id varchar(10) references supplier(sup_id),  
noofproduct number (2) not null,  
modeofpayment varchar (8) not null,  
chqno varchar(16),  
totalamount number (8,2) not null,  
totalwithtax number (9,2) not null,  
advamount number (9,2) not null,  
duesamount number (8,2) not null,  
postatus varchar(25));
```

```
create table p_det(  
pur_orderno varchar(10)not null,  
sno number (2) not null,  
p_id varchar(10)not null,  
p_nm varchar(50)not null,  
p_typ varchar(30) not null,  
p_rate number (6,2) not null,  
p_unit varchar(10) not null,  
qty number (4) not null,  
price number(9,2) not null,  
cgstper number (4,2) not null,  
cgstamt number (6,2) not null,
```

```
sgstper number (4,2) not null,  
sgstamt number (6,2) not null,  
total number (9,2) not null);
```

```
create table ordetails(  
    invoiceno varchar(10)not null,  
    invdate date not null,  
    orderno varchar(10)not null,  
    challanno varchar(10)not null,  
    noofprd number(2) not null,  
    pymtby varchar(10) not null,  
    chqno varchar(20) not null,  
    withtaxamt number (8,2) not null,  
    amtpaidinadv number(8,2) not null,  
    netamt number (8,2) not null);
```

```
create table recvd_p_det(  
    pur_orderno varchar(10)not null,  
    sno number (2) not null,  
    p_id varchar(10)not null,  
    p_nm varchar(40)not null,  
    p_typ varchar(30) not null,
```

```
p_rate number (6,2) not null,  
qty number (4) not null,  
price number(9,2) not null,  
cgstper number (4,2) not null,  
cgstamt number (6,2) not null,  
sgstper number (4,2) not null,  
sgstamt number (6,2) not null,  
total number (9,2) not null);
```

```
create table customer(  
    c_id varchar(10) constraint pkcust primary key,  
    c_nm varchar(30)not null,  
    c_mob varchar(10)not null,  
    c_add varchar(50)null,  
    c_gender varchar(15)not null,  
    c_email varchar(30)null,  
    dues number(8,2));
```

```
create table stock(  
    rno varchar(5) constraint pkstk primary key,  
    p_id varchar(10) references product(p_id),  
    avl_qty number(5) not null,
```

```
    min_qty number(5) not null,  
    max_qty number(5) not null);
```

```
create table sell_details(  
    s_ono varchar(10) constraint pkselldet primary key,  
    s_date date not null,  
    c_id varchar2(10) references customer (C_ID),  
    nop number(3) not null,  
    s_pm varchar(10) not null,  
    chqno varchar(15) not null,  
    s_total number(8,2) not null,  
    s_twt number (8,2) not null,  
    c_prevdues number (8,2) ,  
    s_amtpd number(8,2) not null);
```

```
create table sold_pdet(  
    s_ono varchar(10) references sell_details(s_ono),  
    p_id varchar(10) references product (p_id),  
    quant number(6,2) not null,  
    amount number(8,2) not null);
```

```
create table sell_inv(
```

```
s_ono varchar(10) references sell_details(s_ono),  
inv_no varchar(10) not null,  
inv_date date not null,  
amount number(8,2) not null);
```

SOFTWARE TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specialization, design and code generation. Once source code has been generated software must be tested to uncover as many errors as possible before delivery.

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 presents an interesting anomaly for the software engineer. During earlier software engineering activities, the engineer attempts to build software from an abstract concept to a tangible product. Now comes testing. The engineer creates a series of test cases that are intended to demolish the software that has been built. In fact, testing is one step in the software process that could be viewed as destructive rather than constructive.

Software engineer are by nature constructive people. Testing requires that the developer discard preconceived notions of the correctness of software just develop and overcome a conflict of interest that occurs when errors are uncovered.

If testing is conducted successfully, it will uncover errors in the software, as a secondary benefit, testing demonstrate that software functions appear to be benefit, testing demonstrate that software functions appear to be working according to specifications, that are behavioral and performance requirements to have been met.

OBJECTIVE:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an uncovered error.
- A successful test is one that uncovers a yet an uncovered error.

PRINCIPLES:

- All tests should be traceable to customer requirements.
- Tests should be planned long before testing begins.
- The pareto principles apply to software testing. It states that 80% of all errors uncovered during testing that are traceable to 20% of all program components.
- Testing should begin “in the small” and progress toward testing “in the large”. Exhausting testing is not possible.
- To be most effective, testing should be conducted by an independent team.
- A good test has a high probability of finding an error.
- A good test is not redundant.
- A good test should be neither too simple nor too complex.

COST ESTIMATION OF THEPROJECT

Total Module: 10

Total number of function point (fp): $10 \times 6 = 60$

Avg LOC/fp: 50 LOC

Total LOC: $60 \times 50 = 3000$ LOC

LOC/hr: 300 LOC

Total hr: $3000/30 = 100$ hr

Effort (cost)/hr: 100 Rs

Total cost: $100 \times 50 = \text{Rs } 5000$

Testing cost(15%): 15% of 5000 = Rs 750

Total cost: $5000 + 750 = \text{Rs } 5750$

Extra expense: Rs 1000

Total cost: $5750 + 1000 = \text{Rs } 6750$

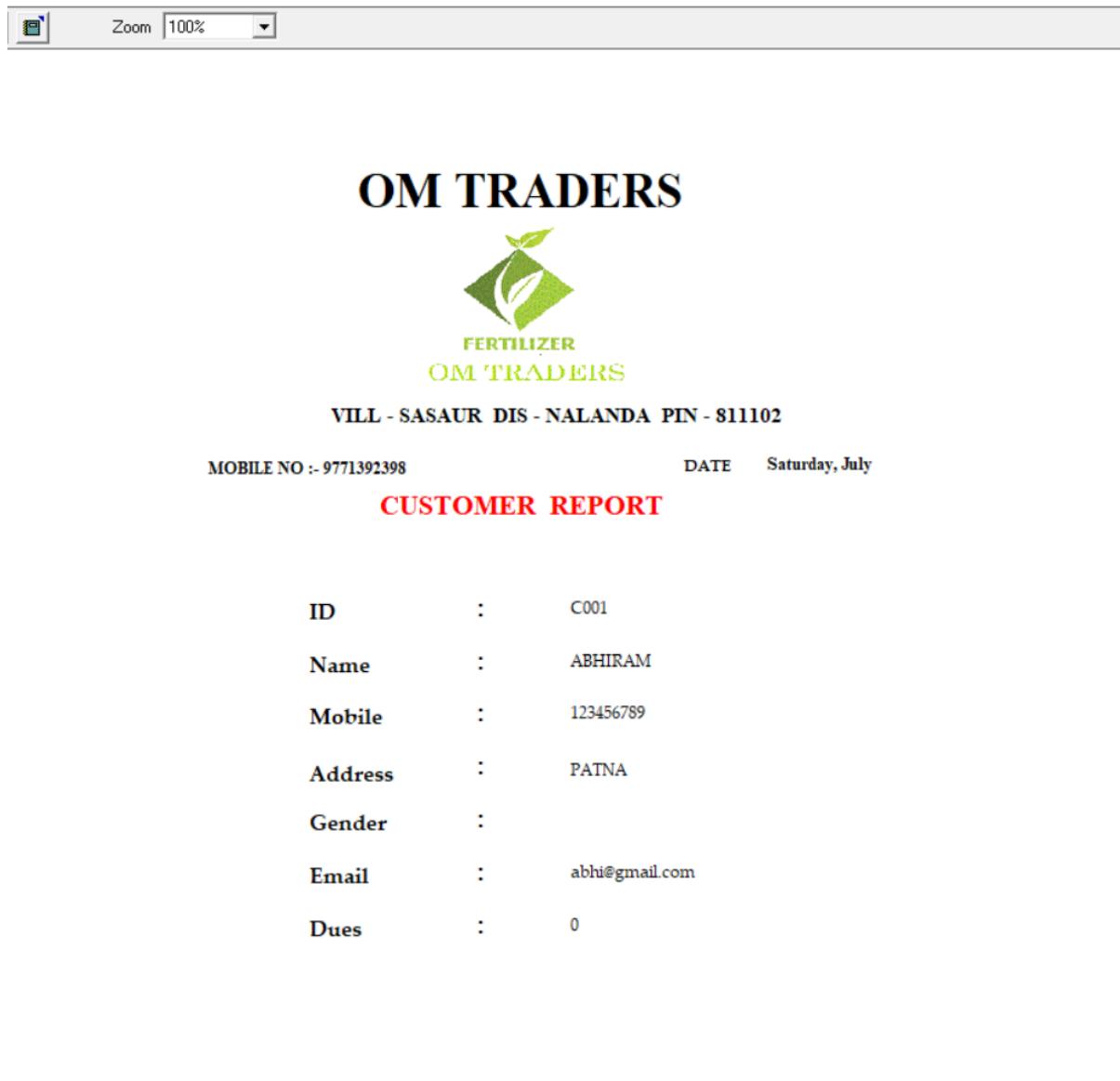
Profit(10%): 10% of 6750 = Rs 675

Final cost: $6750 + 675 = \text{Rs } 7425$

REPORT SNAPS:-

Customer Report :-

CUS_nm_REPORT



STOCK REPORT :-

OM TRADERS



FERTILIZER

OM TRADERS

VILL - SASAUR DIS - NALANDA PIN - 811102

MOBILE NO :- 9771392398

DATE : Saturday, July

STOCK REPORT

Rack No	:	1
Product Id	:	P001
Available Quantity	:	15
Minimum Quantity	:	10
Maximum Quantity	:	100

SALE REPORTS:-

OM TRADERS



FERTILIZER

OM TRADERS

VILL - SASAUR DIS - NALANDA PIN - 811102

MOBILE NO :- 9771392398

DATE : Saturday, July

SELL REPORT

Sell Order No	:	SO001
Sell Date	:	7/2/2022
Customer Id	:	C001
No Of Product	:	1
Payment Mode	:	Cash
Cheque No	:	CASH
Total	:	7980
Total With Tax	:	9416
Previous Dues	:	16
Amount Paid	:	9400

SUPPLIER REPORT :-

SUPPLIER REPORT NAME		
	Zoom	100%

OM TRADERS



FERTILIZER

OM TRADERS

VILL - SASAUR DIS - NALANDA PIN - 811102

MOBILE NO :- 9771392398

DATE : Saturday, July

SUPPLIER REPORT

ID	:	S001
Name	:	ABHIRAM
Mobile	:	9572979465
Location	:	PATNA
State	:	BIHAR
City	:	PATNA
Pincode	:	800026
Company	:	KISAAN
Email	:	ak@gmail.
GST No	:	555555555

FUTURE SCOPE AND FUTURE ENHANCEMENT OF THE PROJECT

If we consider an organization, it seems to be an integration of many functions. As we see there is a lot of work performed by an organization that produces some product but here we are only concerned with the product services and maintenance.

This software, in its current scope, basically enables to automate stock records and ensures smooth management of product service and maintenance department .the organization has so many departments and its time is too costly ,particularly when there is long queue of customer, and waiting for there time. Payment for the product service is quit difficult for the employee of the organization to manage and to provide information about the product status, a little mistake can be a loss of thousands or lakhs so removing all these types of mistake and anomalies this software is developed, this project is very useful when the old customer came with receipt number and give some product for service. This product will show all information of the customer and product within a second on which further operation can be done. It manages the details of Fertilizer ,details of invoices, orders, sales, purchase, and generate report which can be quit helpful for the organization to ease there work.

Future scope of the project circles around maintain information regarding.

- We can add printer in future.
- We can give more advance software for FERTILIZER SHOP ManagementSystem including more facilities.
- Create the master and slave database structure to reduce the overloadof the database queries.

The above mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of Fertilizer Shop Management System. Enhancements can be done to maintain all the Fertilizer shop, stocks, company, and inventory.

We have left the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them. In the last we would like to thank all those persons involved in the development of the system directly or indirectly. We hope that the project will serve its purpose for which it is developed thereby underlining success.

It may help in collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible.

The system generates types of information that can be used for various purposes.

- It satisfies the user requirement.
- Be easy to understand by the user and operator.
- Be easy to operate.
- Have a good user interface
- Be expandable
- Delivered on schedule within the budget.

BIBLIOGRAPHY

Book Reference

**Book Name: - Visual Basic 6 Programming
(Black Book) Author: -Steven Holzner**

- **Book Name: - Oracle Database 10G Dba Handbook 1st EditionAuthor: - Kevin Loney, Bob Bryla**
- **Book Name: - Software Engineering Author: - Pressman**
- **Book Name: - System Analysis & Design Author: - Elias M.Award**

Website Reference

- www.google.com
- www.vbforums.com