Chapter – 1

INTRODUCTION

INTRODUCTION TO PROJECT

1.1Introduction of system
1.2Existing System
1.3Need and Scope of computer system
1.4Organization Profile

1.1 INTRODUCTION OF SYSTEM-

The title of the system is “Sales and Purchase System For Laptop accessories, Desktop sales & Repairs” which gives the idea of project. Select this system for “Sakshi Enterprises, Islampur”.

It sales the all companies Sales and Purchase System For Laptop accessories, Desktop sales & Repairs for etc. to the customers.

At present all work is done manually in shop. Different forms are used for different task. The billing details are maintained in the register. They have to face many problems. In manual system we have to maintain all information about Product Book Detail, Customer Bill Information, Customer Details, Product Bill Information, Product Detail & Service , Supplier Details, Supplier Bill Information , etc manually.

The records are of transaction are easily stored in the computerization system.

1.2 EXISTING SYSTEM-

The existing system of “Sales and Purchase System For Laptop accessories, Desktop sales & Repairs” is totally manual. The existing system gives the structure and function of present system. This gives ideas of requirement and understands the performance of the system. It also gives the idea for the new system.

If any transaction does in the absence of owner that is through the sale it creates more complication to handle the system. And it is difficult to maintain or identify the stock information.

If any customer returns the remaining material which need to owner and the material in any quantity returns to owner if need not be used, so to update the stock record is difficult.

If the material is not according to owner then it supply back to the supplier, otherwise this material is passing to the quantity control for checking quality of the materials and gives the decision as to select or reject the material. If it is rejected it is back to the supplier otherwise material is accepted and then purchase order is prepared for the payment.

The record of purchase material is kept as well as the record of material delivery also has to keep, so the working load is increased. The presenty of each employee and their salary record is also additional work for the system.

1.2 NEED AND SCOPE OF COMPUTER SYSTEM-

NEED-

In an organization the "Sales and Purchase for System For Laptop Accessories, Desktop sales & Repairs" is one of the important systems. In this system the work is also growing fast. The job becoming very tedious and complicated to difficult types of drawbacks and error made manually in daily organization.

To overcome their problem the computerization is most important. The system can reduce the time required for any job that can take long period of completion, also the maintenance of data and storage become easier than old once. Due to possibility of making mistake is reduced, also some other reasons are given below that indicates the proposed system must be computerized.

- The computerization generates number of reports at any time in number of copies in very clean and neat form.
- To get various report in very short form.
- Save the time required to calculate payment of order can be prepared easily.

To save extra time and manpower involved in the same activity. There is a number of loss of date is kept up to date and systematic.

Scope:

- It reduces paper work, which help to decrease complication in maintenance and time.
- Large calculations can be done correctly and in less time.

Any records can be searched in a few seconds and reports can be obtained in any format, we want and when we want.

1.4 ORGANIZATION PROFILE-

- **Owner Name** : Mr. Jayant Babanrao Patil
- **Name of organization** : Sales And Purchase System For "Sakshi Enterprises, Islampur"
- **Address** : Near Gosavi Hospital, At post- Islampur Tehsil- Walwa, District- Sangli
- **Phone Number** : 9970042737
- **Year of Establishment** : 11 August 2011
- **Email ID** : jayant.patil.22@gmail.com

Sakshi Enterprises is a well known in Islampur.

Chapter – 2

PROPOSED SYSTEM

C# is pronounced "C-Sharp".

It is an object-oriented programming language created by Microsoft that runs on the .NET Framework.

C# has roots from the C family, and the language is close to other popular languages like C++ and Java.

The first version was released in year 2002. The latest version, **C# 8**, was released in September 2019.

C# is used for:

- Mobile applications
- Desktop applications
- Web applications
- Web services
- Web sites
- Games
- VR
- Database applications
- And much, much more!

Why Use C#?

- It is one of the most popular programming language in the world
- It is easy to learn and simple to use
- It has a huge community support
- C# is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
- As C# is close to C, C++ and Java, it makes it easy for programmers to switch to C# or vice versa

C# IDE

The easiest way to get started with C#, is to use an IDE.

An IDE (Integrated Development Environment) is used to edit and compile code.

In our tutorial, we will use Visual Studio Community, which is free to download from https://visualstudio.microsoft.com/vs/community/.

Applications written in C# use the .NET Framework, so it makes sense to use Visual Studio, as the program, the framework, and the language, are all created by Microsoft.

C# Install

Once the Visual Studio Installer is downloaded and installed, choose the .NET workload and click on the **Modify/Install** button:

What is SQL Server?

SQL Server is a relational database management system (RDBMS) developed by Microsoft. It is primarily designed and developed to compete with MySQL and Oracle database. SQL Server supports ANSI SQL, which is the standard SQL (Structured Query Language) language. However, SQL Server comes with its own implementation of the SQL language, T-SQL (Transact-SQL).

T-SQL is a Microsoft propriety Language known as **Transact-SQL.** It provides further capabilities of declaring variable, exception handling,

stored procedure, etc.

Version History of SQL Server

- Microsoft and Sybase released version 1.0 in 1989.
- However, the partnership between these two ended in the early 1990s.
- Microsoft maintained ownership rights to the name SQL Server.
- Since the 1990s, subsequent versions of SQL Server have been released including SQL Server 2000, 2005, 2008, 2012, 2014, 2016, 2017, and 2019.

SQL Server Editions

Following are the popular editions/types of SQL server:

SQL Server Enterprise: It is used in the high end, large scale and mission Critical business. It provides High-end security, Advanced Analytics, Machine Learning, etc.

SQL Server Standard: Itis suitable for Mid-Tier Application and data marts. It includes basic reporting and analytics.

Software Requirement Specifications

The production of the requirements stage of the software development process is **Software Requirements Specifications (SRS)** (also called a **requirements document**). This report lays a foundation for software engineering activities and is constructing when entire requirements are elicited and analyzed. **SRS** is a formal report, which acts as a representation of software that enables the customers to review whether it (SRS) is

according to their requirements. Also, it comprises user requirements for a system as well as detailed specifications of the system requirements.

The SRS is a specification for a specific software product, program, or set of applications that perform particular functions in a specific environment. It serves several goals depending on who is writing it. First, the SRS could be written by the client of a system. Second, the SRS could be written by a developer of the system. The two methods create entirely various situations and establish different purposes for the document altogether. The first case, SRS, is used to define the needs and expectation of the users. The second case, SRS, is written for various purposes and serves as a contract document between customer and developer.

Following are the features of a good SRS document:

1. Correctness: User review is used to provide the accuracy of requirements stated in the SRS. SRS is said to be perfect if it covers all the needs that are truly expected from the system.

2. Completeness: The SRS is complete if, and only if, it includes the following elements:

(1). All essential requirements, whether relating to functionality, performance, design, constraints, attributes, or external interfaces.

(2). Definition of their responses of the software to all realizable classes of input data in all available categories of situations.

(3). Full labels and references to all figures, tables, and diagrams in the SRS and definitions of all terms and units of measure.

3. Consistency: The SRS is consistent if, and only if, no subset of individual requirements described in its conflict. There are three types of possible conflict in the SRS:

(1). The specified characteristics of real-world objects may conflicts. For example,

(a) The format of an output report may be described in one requirement as tabular but in another as textual.

(b) One condition may state that all lights shall be green while another states that all lights shall be blue.

(2). There may be a reasonable or temporal conflict between the two specified actions. For example,

(a) One requirement may determine that the program will add two inputs, and another may determine that the program will multiply them.

(b) One condition may state that "A" must always follow "B," while other requires that "A and B" co-occurs.

(3). Two or more requirements may define the same real-world object but use different terms for that object. For example, a program's request for user input may be called a "prompt" in one requirement's and a "cue" in another. The use of standard terminology and descriptions promotes consistency.

4. Unambiguousness: SRS is unambiguous when every fixed requirement has only one interpretation. This suggests that each element is uniquely interpreted. In case there is a method used with multiple definitions, the requirements report should determine the implications in the SRS so that it is clear and simple to understand.

5. Ranking for importance and stability: The SRS is ranked for importance and stability if each requirement in it has an identifier to indicate either the significance or stability of that particular requirement.

Typically, all requirements are not equally important. Some prerequisites may be essential, especially for life-critical applications, while others may be desirable. Each element should be identified to make these differences clear and explicit. Another way to rank requirements is to distinguish classes of items as essential, conditional, and optional.

6. Modifiability: SRS should be made as modifiable as likely and should be capable of quickly obtain changes to the system to some extent. Modifications should be perfectly indexed and cross-referenced.

7. Verifiability: SRS is correct when the specified requirements can be verified with a cost-effective system to check whether the final software meets those requirements. The requirements are verified with the help of reviews.

8. Traceability: The SRS is traceable if the origin of each of the requirements is clear and if it facilitates the referencing of each condition in future development or enhancement documentation.

9. Design Independence: There should be an option to select from multiple design alternatives for the final system. More specifically, the SRS should not contain any implementation details.

10. Testability: An SRS should be written in such a method that it is simple to generate test cases and test plans from the report.

11. Understandable by the customer: An end user may be an expert in his/her explicit domain but might not be trained in computer science. Hence, the purpose of formal notations and symbols should be avoided too as much extent as possible. The language should be kept simple and clear.

12. The right level of abstraction: If the SRS is written for the requirements stage, the details should be explained explicitly. Whereas,for

a feasibility study, fewer analysis can be used. Hence, the level of abstraction modifies according to the objective of the SRS.

There are two types of Traceability:

1. Backward Traceability: This depends upon each requirement explicitly referencing its source in earlier documents.

2. Forward Traceability: This depends upon each element in the SRS having a unique name or reference number.

The forward traceability of the SRS is especially crucial when the software product enters the operation and maintenance phase. As code and design document is modified, it is necessary to be able to ascertain the complete set of requirements that may be concerned by those modifications.

Properties of a good SRS document

The essential properties of a good SRS document are the following:

Concise: The SRS report should be concise and at the same time, unambiguous, consistent, and complete. Verbose and irrelevant descriptions decrease readability and also increase error possibilities.

Structured: It should be well-structured. A well-structured document is simple to understand and modify. In practice, the SRS document undergoes several revisions to cope up with the user requirements. Often, user requirements evolve over a period of time. Therefore, to make the modifications to the SRS document easy, it is vital to make the report well-structured.

Black-box view: It should only define what the system should do and refrain from stating how to do these. This means that the SRS document should define the external behavior of the system and not discuss the implementation issues. The SRS report should view the system to be

developed as a black box and should define the externally visible behavior of the system. For this reason, the SRS report is also known as the black-box specification of a system.

Conceptual integrity: It should show conceptual integrity so that the reader can merely understand it. Response to undesired events: It should characterize acceptable responses to unwanted events. These are called system response to exceptional conditions.

Verifiable: All requirements of the system, as documented in the SRS document, should be correct. This means that it should be possible to decide whether or not requirements have been met in an implementation.

2.1 OBJECTIVES OF PROPOSED SYSTEM-

Every system is designed to achieve one or more objectives. The main objective of this system is to overcome problem that come during existing system. The main objective of system is to store the large amount of data within a less time. The system also aims to maintain correct and quick processing and to get some printed output like Customer Detail, Customer Bill Information, Supplier Bill Information, Product Book Detail, Product Detail, Supplier Detail, using computer calculation correction and addition are possible quickly.

By doing computerization of this system many goals can be attained. Computer can achieve tremendous speed. It becomes possible to store large amount of information and previous records in small space and they can be accessed immediately when wants. At a time one can have more than one copy immediately per requirement, which is not possible in manual writing.

The most important objectives of system are as follows:

1. To provide security, authority and future privacy.
2. To provide different conclusion.
3. To provide information most quickly as per requirement.
4. Any unauthorized person can’t get or can’t change information.
5. There is no loss of data.

2.2 REQUIREMENT OF ENGINEERING-

In this step what the requirements of the user are as identified. The success of any system will depend upon how needs of the user are identified.

These requirements will lead to accurate definition of problem. According to these user requirements through investigation of a system and proper solution can be defined. This step is intended to help the user and the analyst understand the real problem rather than its symptoms. The user or the analyst may identify the need for a candidate system or for enhancement in the existing system.

- **INITIAL INVESTIGATION**

The first step in the system developed life cycle is the identification of requirement. In this first step the two questions asked if one is What is the problem? That is why system has to change, or improve or to enhance? Then second one is who has requested for this change or improve or enhance.

To get the answer of this question initial investigation is done. In initial investigation it is checked that whether the request is valid or feasible. During this initial investigation a user's request from is given to user. As per user's request the need for the change is identified. Once the request is approved, activities like background investigation, fact-finding and analysis and presentation of result such activities are carried out.

Hence this request is turned into how will be a project proposal. The proposal when approved, initiates a detailed user-oriented specification of system performance and analysis of the feasibility of the candidate system.

2.2.1 REQUIREMENT GATHERING-

In initial investigation and requirement identification is gathered from the user. This information is primarily used to understand the problem of the user and nature of operation and how each activity takes place.

Information gathering is an art of science. It requires special skills as well as training, experience and preparation. Following are some tools used for information gathering.

- **INTERVIEW-**

The interview is a face-to-face inter personal role situation in which person called the inter-viewer ask a person being inter-viewed question, designed to gather information about a problem area.

The interview is the oldest and most often used device for gathering information. It can be used for two main purposes.

1. As an exploratory device to identify relations or verify information.
2. To capture information as it exists.
3. The main advantages of this tool are –
    I. It's flexibility.
    II. Valid information can be gathered.

Regarding complex subject's maximum information can be gathered. Co-operation of people those who are interred viewed is quite good. Compare to other techniques like questionnaires.

The major drawback of the interview is:

1. Long preparation time.
2. It takes lot of time conduct. This will involve time and money.

- **QUESTIONNRIES-**

Here there are number of questions which individual respond. It is usually associated with self-administrated tools with items of the closed or fixed alternative type.

It has following advantages:

It is economical and requires less skill to administer than the interview.

Unlike the interview, which generally questions one subject at a time, a questionnaire can be administered to large number of individuals simultaneously.

The standardized wording and other of the questions and the standardized, instructions for reporting response ensure uniformity of questions. In contrast the interview.

Situation is rarely uniform from one interview to the next. The respondent feels greater confidence in the anonymity of questionnaire than in that of an interview. In an interview, the analyst usually knows the user's staff by name, job function or other

identification, with a questionnaire respondent gives opinions without fear that the answer will not be connected to their names. The questionnaire places less pressure on subject for immediate response. Respondent have time to thank the questions over and do calculation to provide more accurate data.

The major disadvantage of this are:

1. Low percentage of returns.
2. Many people have difficulty expressing themselves in writing epically when responding to open questions.
3. Many dislikes writing.

2.2.2.SRS (SYSTEM REQUIREMENT SPECIFICATION)-

This document seeks to provide software requirement specification for Pure It Management System. Software requirement specification will provide a broader understanding of requirement specification of this system and features of this system along with requirements. This document will guide the developer's requirements provided by the users. This document will help to narrow the gap between the requirements of the user and the perspective of the developer. Finally, it will lead assists as criteria for a quality final product.

In system under consideration during the development phase following methods are adapted.

1. Firstly, permission was taken from owner of the furnishing store for collecting the information.
2. The existing system was carefully studied with its drawbacks.
3. Requirements of billing system are noted after interview the concern employee.
4. The various files are they studied and relevant data taken from files.
5. While developing the system various suggestions they are taken from the files.
6. The necessary documents they are prepared.
7. The necessary programs they are developed.
8. Debugging and testing of program there performer with test data.
9. Integration of whole system to achieve the co-ordination is done.

Chapter – 3

SYSTEM ANALYSIS

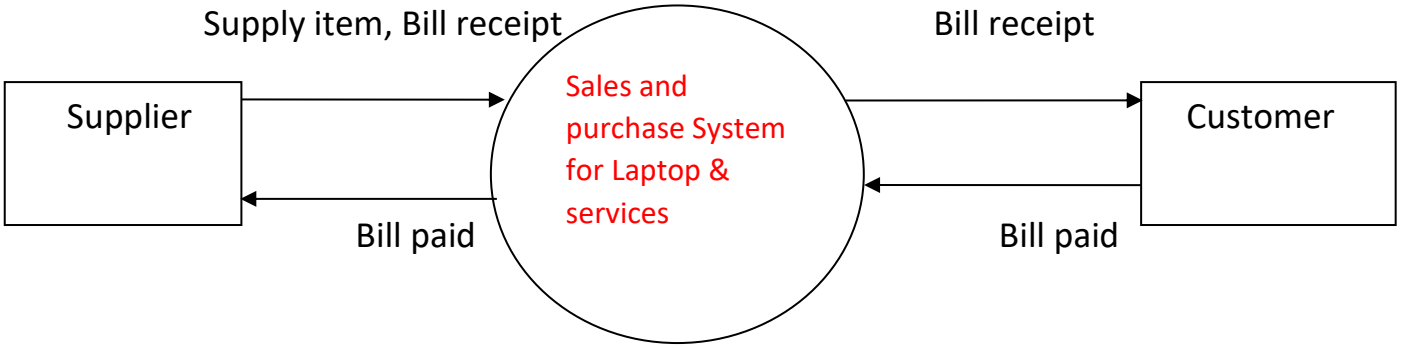
SYSTEM ANYLYSIS

3.1 System Diagram

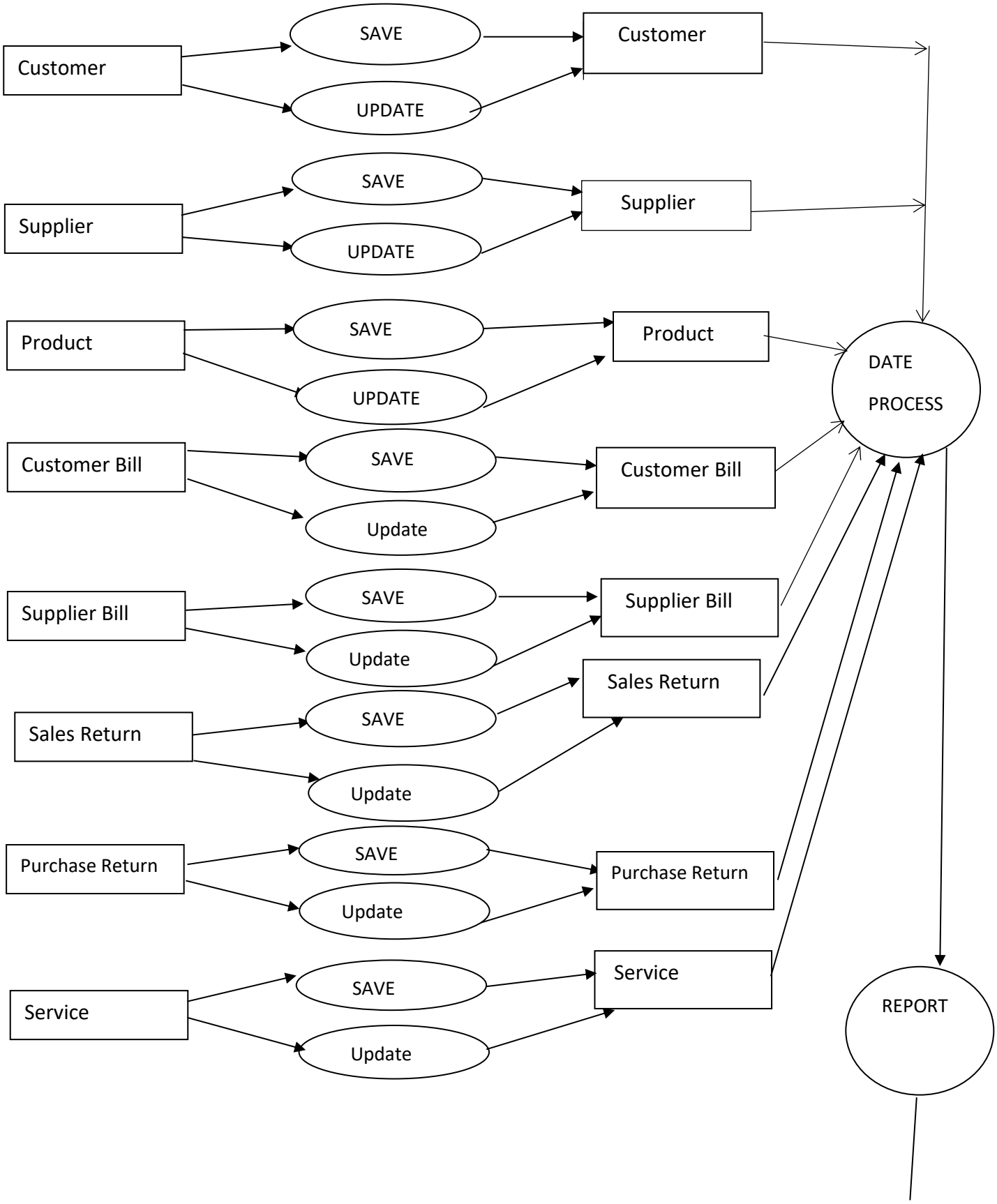
3.1.1 Data flow Diagram (DFD)

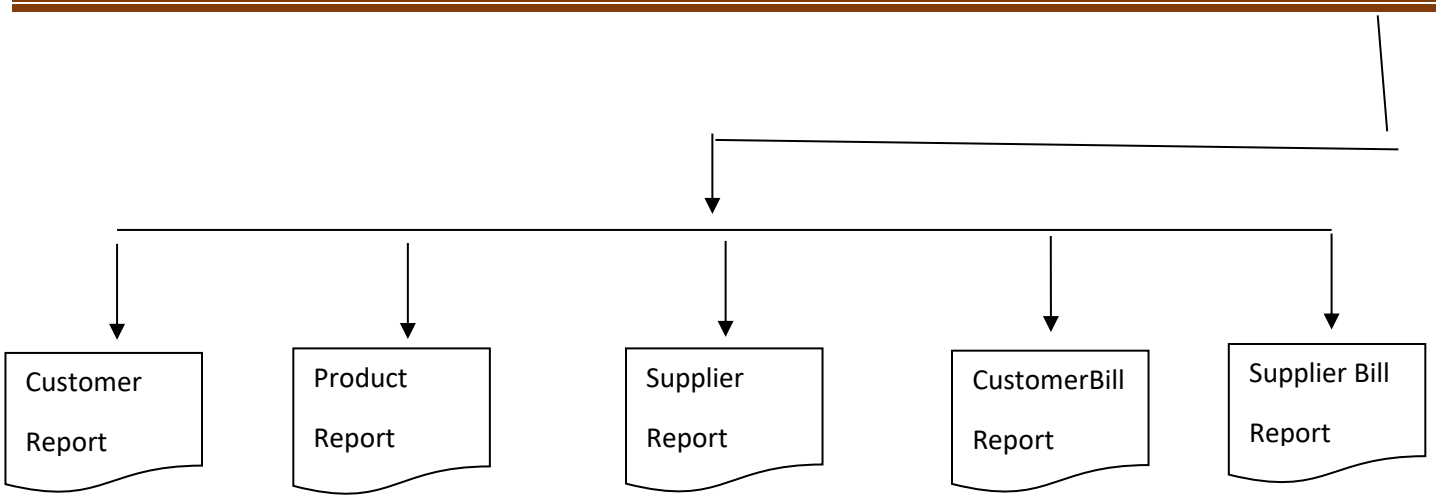
3.1.2 Entity Relationship

3.1 CONTEXT LEVEL DIAGRAM-

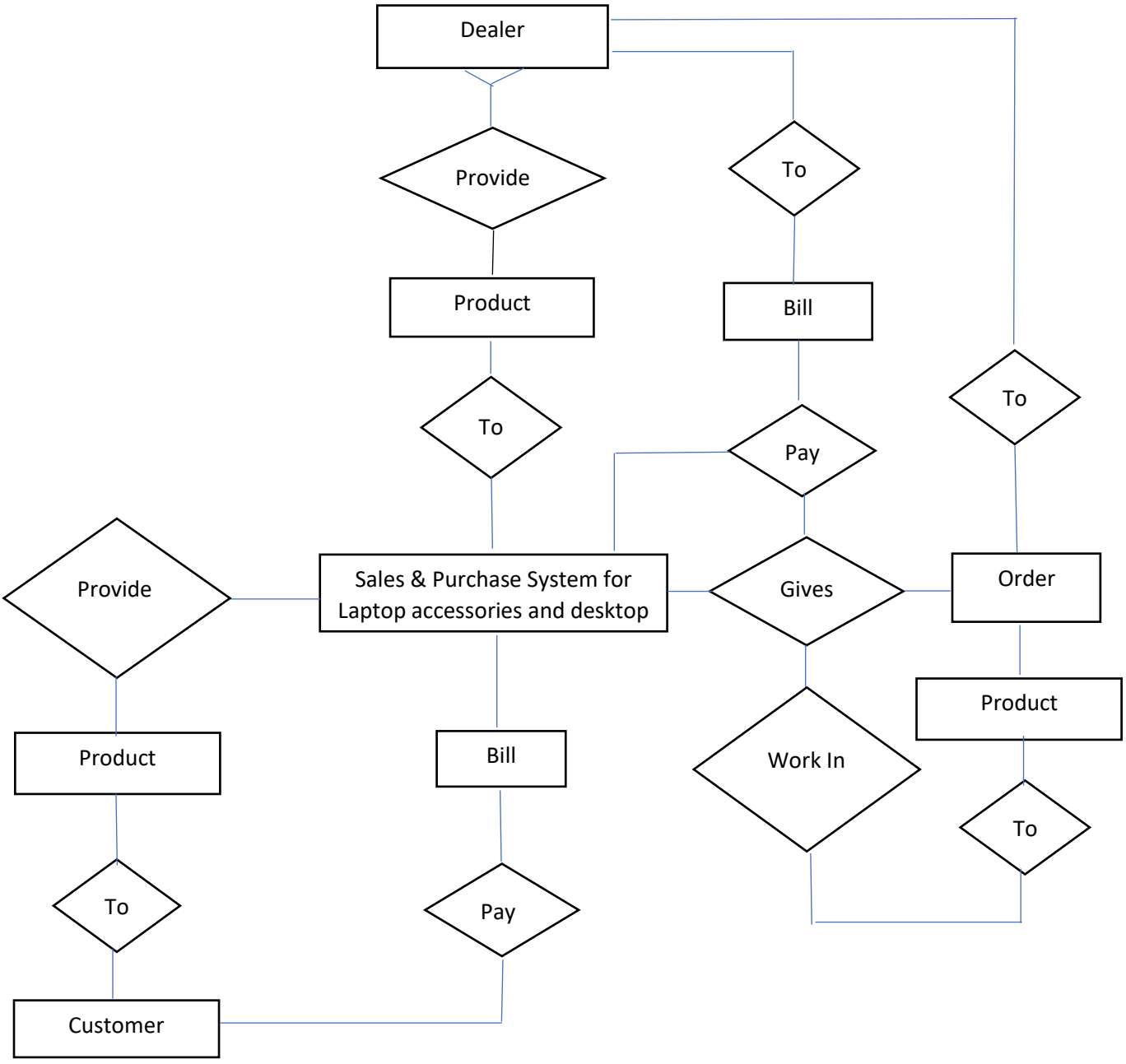


3.1.1 DATA FLOW DIAGRAM-




Customer Report
Product Report
Supplier Report
CustomerBill Report
Supplier Bill Report

3.1.2. ENTITY RELATIONSHIP DIAGRAM



Chapter – 4

SYSTEM DESIGN

SYSTEM DESIGN

4.1 Feasibility Study

4.2 Fact Finding Technique

4.3 Introduction To VB.NET

4.4 Introduction to MS-Access

4.5 Database Design

4.1 FEASIBILITY STUDY-

The most important part of every system is the feasibility study. The feasibility study is useful to evaluate the cost and benefits of the system requested. This includes technical, operational and economical feasibility. It is as follow:

1. TECHNICAL FEASIBILITY-

The technical feasibility always focuses on existing computer hardware and software. The technical feasibility pointed towards the hardware and software and software requirement of this system.

2. OPERATIONAL FEASIBILITY-

It considers the acceptability of the system, It checks whether system will be used if it is developed and implemented are the users of the system able to handle the system, whether the proposed system cause any trouble, etc.

3.ECONOMICAL FEASIBILITY-

Economical feasibility mostly includes the profile of proposed system. When we are going to decide the economical feasibility, we must consider different point in account and they are cost of hardware and software system analysis and design, programming, training, installing, operating cost etc.

4.2 FACT FINDING TECTNIQUE-

In the System development following methodology is adopted.

After deciding the subject, you must be permitted for designing of that system by man in charge of that organization. With the plus points, you should take the drawback in account. You should cover all requirement of the organization. For this it is necessary for you to collect the detail information by intern wing with as follows.

The most important part of the system design is to consider the various file to data and suggestion files.

Prepare the necessary document and programs in clean and neat formatted.

When you completed your program writing you should test the program as the test data, the data if necessary.

Integration of whole system was done to achieve the co-ordination.

If you have satisfactory output of all above study then and then prepare final coding and create required output result.

4.3 INTRODUCTION TO VISUAL BASIC .NET-

- **INTRODUCTION-**

Visual Basic.NET not just a language, it is an integrated development environment in which you can develop, run, test and debug your applications. Visual Basic.NET helps you create solution on that run on Microsoft windows operating system.

Visual Basic.NET is a language in which you can tell your computer how it does things. But like a child, the computer will only understand if you have never programmed before this sound like an arduous task sometimes it is. However Visual Basic.NET gives you a simple language to explain some complex things. Although it never hurts you to have an understanding of what is happening at the lowest level.

The .NET is more than just one thing; it is a collection of software the concept that work together to enable the creation of business solution. A major goal of the .NET concept is to decrease the building is distributed system in which work is several different locations.

The most important advantage of Visual Basic.NET is that it has been designed and to make even more productive in your development work specially. If you need to use information in database or create solution for the internet-built and important benefits is that once you become comfortable with the development environment in Visual Basic.

- **NEED FOR VISUAL BASIC .NET PROGRAMMING-**

There are several or many programming tools such as C, C++ etc that allow building programs. These tools or languages allow us to design interfaces user friendly by window. Menus and buttons etc but the disadvantage of such tools is that the interface in design using code. The program has the code for user interface elements. Such a program spends

a lot of time in writing code for the user interface and for the purpose the need of house was an environment that would allow easy design of the user interface.

- **USER INTERFACE-**

The User interface is what appears in the applications window when it runs. It consists of various elements with which the user can interface and control the application. The first element of the user interface is the Form. This is a window display at runtime, and it acts as a container for all the elements of the interface. The element in the user interface is common to all window application, and they all shown as icon in the toolbox.

- **EVENT DRIVEN PROGRAMMING-**

The most important feature of Visual Basic.NET is Event Driven Programming. Programming the application isn't simple, but there's a methodology you can't escape. A Visual Basic.NET application isn't a monolithic program such as application you may have developed with other languages.

In procedural application the application itself control's which portion of code execution and in what sequence. Execution starts with first line of code and follows a predefined path through the application. When you program in Visual Basic.NET; you must first decide how the application interacts with the user. In other words, you must first decide how each control reacts with the user actions, such as the click of the mouse, keystrokes, and so on, and you must program these reactions. You program the application to react to various external conditions (events), and the user's action determines the application flow.

FEATURES OF VB.NET PROGRAMMING-

1. THE PROGRAMMING LANGUAGE-

The Visual Basic is window basic programming language that provided a graphical interface.

I. Compiler based.
II. Has online help.
III. Provides sophisticated error handling.

2. THE DEVELOPMENT INTERFACE-

Visual Basic has an Integrated Development Environment (IDE). An Integrated Development Environment integrates all the aspects of programming required by the programmer. The IDE of VB.NET comprises of the menu, the form window, code window, project explorer window, property window and toolbox.

3. TOOLBOX-

The one of the most important features of Visual Basic.NET revolves around the toolbox that contains array essential controls and we also add our controls to VB.NET toolbox.

OBJECT ORIENTED CONCEPTS-

1. INHERITANCE-

Inheritance is the ability of a class to derive its characteristics from an existing system. Using Visual Basic.NET, you can create a class that provides basic functionality so that other classes can inherit its members.

2. CONSTRUCTORS AND DESTRUCTORS-

Visual Basic.NET supports constructors and destructors. A constructor is a special type of method that is invoked when you create a new instance of a class.

3. OVERLOADING-

Overloading enables methods to be defined with the same name but different parameters. In other words, it allows you to have multiple implementations of a method.

4. OVERRIDING-

Overriding is the capability of a derived class to override the characteristic of the parent class. The methods that can be overridden by the derived classes need to be marked as overridden in the base class.

5. STRUCTURED EXCEPTION HANDLING-

Exception is the error that is generated at run time as a result of an erroneous statement or condition or because of some unexpected behavior of the application.

6. MULTIPLICATION-

Visual Basic.NET provides full support for creating multithreading application.

ADVANTAGES OF VB.NET PROGRAMMING-

1. CONSISTENT PROGRAMMING MODEL-

The .NET framework provides common object-oriented programming model across languages. This object model can be used in the code to perform several tasks such as reading from and write to files, connecting to database, and retrieving data.

2. MULTIPLATFORM APPLICATION-

There are several versions of windows most of which run on x86 CPU's. Some versions, such as windows CE and 64-bit Windows, run on non-x86 CPU's as well.

3. MULTILANGUAGE INTEGATION-

VB.NET allows multiple languages to be integrated. For example it is possible to create a class in C# that derives from a class implemented in VB.

4. AUTOMATIC RESOURCE MANAGEMENT-

While creating an application, a application may be required to write code for managing resources such as files, memory, network connections, and database resources.

5. EASE OF DEVELOPMENT-

One the goal of the .NET framework is to simplify application development .NET application can be deployed simply by copying files.

The largest computer development of components has also been simplified.

SOME NEW FEATURES OF VB.NET-

1. Full support for object oriented programming.
2. Structured error handling capability.
3. Access to .NET Framework.
4. Powerful unified Integrated Development Environment (IDE).
5. Inherent support for XML and Web Services.
6. New console capability of VB.NET.
7. New console capability with web forms.
8. Immense power of tools and controls (including server controls).
9. Inter portability with other .NET complied languages.
10. Better database programming approach with ADO.NET

4. INTRODUCTION OF MICROSOFT ACCESS-

➢ WHAT IS DATABASE?

A database is a collection of information relates to a particular subject of purpose such as customer order or maintaining billing details. If your database is not stored on a computer, or only part if it is stored, you may be tracking information, from a variety of sources that are required to co-ordinate and organize yourself.

Access is easy to use data management tool. Access is a use to stored track and report information. You can use access to create database that track membership lists, clients data and student registration. Access also works seamless with word when you need to merge data for mass mailings and works with power point when you need to show data that substantiates the point being made in presentation. Access can display a forms or point catalogs and directories, embed a power point slide in a fields, and take information from outlook.

Using Microsoft Access, you can manage all your information from a single database file. Within the file divide you data into separate storage containers called tables, view, add and update table data using online forms; find the retrieve just the data you want using queries, and analyze or print data in specific layout reports. To store your data, create one table for each type of information. Bring data from multiple tables together in query. From or report, you define relationship between the tables.

To find and retrieve just the data the meets conditions you specify, include data from multiple tables, and create a query. A query can also update or delete multiple records at the sometimes and perform build in or custom calculation on your data.

To easily view, enter and change data directly in tables, create a form. When you open a form, Microsoft access retrieve the data from

one or more tables, and display it on screen using the layout that you can created from scratch.

To analyze your data or present it a certain way in print, way in print, create a report, for example, you might print one report that group data and calculation, it is design to create tables, forms tables, forms and report based on record that you create access helps to organize this information, and so use it is the basic of queries or searches, which filter the data in specified ways. Access helps you to manage large amount of information and to show relationship among records. Like all other Microsoft Office application. Access allows you to arrange your data in visually attractive formats.

STEPS IN DESIGNING DATABASE-

- **TO CREATE A DATABASE FOLLOWING STEPS ARE USED-**
  1. Determine the purpose of your database.
  2. Determine the tables you need in the database.
  3. Determine the fields.
  4. Identify fields with unique valued.
  5. Determine the relationship between tables.
  6. Define your design.
  7. Add data and create other between objects.

- **ADVANTAGES OF ACCESS-**
  1. Data quality enhanced.
  2. Data independent allows dynamic changes and growth.
  3. Integrity can be maintained.
  4. Data duplication is eliminated with controlled redundancy.
  5. Standard can be enforced.

6. Redundancy enhancement possible.
7. Security enhancement possible.
8. Inconsistency can be avoided.

- **QUERIES-**

There are three types of Queries-

1. Select Queries:

   Select queries are essential question that you want to access about entries in fields

2. Action Queries:

   Action queries are allows user to change an entire group of records in one field.

3. Parameter Queries:

   Parameter queries are follows user to change criteria with each use of queries.

4.5 DATABASE DESIGN-

1. Table: Customer

Field Name	Data Type	Size
Customer ID	Int	Integer
Customer Name	nvarchar(50)	50
Email	nvarchar(50)	50
Mobile No.	nvarchar(20)	20
Address	nvarchar(MAX)	Max

2. Table: Customer Bill

Field Name	Data Type	Size
Sales ID	Int	Integer
Bill No.	Int	Integer
Customer ID	Int	Integer
Customer Name	nvarchar(50)	50
Product ID	Int	Integer
Product Name	nvarchar(50)	50
Quality	nvarchar(50)	50
Quantity	Int	Integer
Price	Int	Integer
Total	Int	Integer
Total GST	nvarchar(50)	50
Grand Total	Int	Integer
Sales Date	nvarchar(50)	50
Warranty Date	nvarchar(50)	50

3. Table: Product

Field Name	Data Type	Size
Product ID	Int	Integer
Product Name	nvarchar(50)	50
Product Type	nvarchar(50)	50
Company	nvarchar(50)	50
Price	Int	Integer
Quantity	Int	Integer

4. Table: Service

Field Name	Data Type	Size
Service ID	Int	Integer
Customer ID	Int	Integer
Price	Int	Integer
Product Name	nvarchar(50)	50
Reason	nvarchar(50)	50

5. Table: Supplier

Field Name	Data Type	Size
Supplier ID	Int	Integer
Supplier Name	nvarchar(50)	50
Email	nvarchar(50)	50
Mobile No.	nvarchar(50)	20
Address	nvarchar(MAX)	MAX

6. Table: Supplier Bill

Field Name	Data Type	Size
Purchase ID	Int	Integer
Product Name	nvarchar(50)	50
Bill No.	Int	Integer
Product ID	Int	Integer
Supplier ID	Int	Integer
Price	Int	Integer
Quantity	Int	Integer
GST	nvarchar(50)	50
Grand Total	Int	Integer
Manufacture Date	nvarchar(50)	50
Purchase Date	nvarchar(50)	50

7. Table: Sales Return

Field Name	Data Type	Size
Reason	nvarchar(50)	50
Return ID	Int	Integer
Purchase ID	Int	Integer
Bill No.	Int	Integer
Product Name	nvarchar(50)	50

8.Table: Purchase Return

Field Name	Data Type	Size
Return ID	Int	Integer
Purchase ID	Int	Integer
Bill No.	Int	Integer
Product ID	Int	Integer
Product Name	nvarchar(50)	50
Reason	nvarchar(50)	50

Chapter – 5

IMPLEMENTATION

IMPLEENTATION

5.1 System Requirement

5.1.1 Hardware And Software Selection

5.2 User Guideline

5.1 SYSTEM REQUIREMENTS-

5.1.1 HARDWARE AND SOFTWARE SELECTIONS-

System requirements divided into two categories. Hardware and software requirements are as follows

- Hardware Requirement:
  1. Processor- Intel(R) Core (TM) i3-5005U CPU @ 2.00GHz 2.00GHz
  2. Monitor 17" color
  3. CPU-Intel P4, 2.66 GHz HT .2MB cache.
  4. 4.00 GB RAM.
  5. 40 GB Hard disk (7200 RPM).
  6. Optical Drive CD-RW 48X.
  7. Multimedia keyboard, Optical Mouse.

- Software Requirements
  1. Operating system- Windows 8.1 Pro
  2. Front End Tool- Microsoft Visual Basic .NET 2017
  3. Back End Tool- MS Access 2007
  4. Crystal Report
  5. Anti-virus software- Net Protector

5.2 USER GUIDELINE-

The system entitled "Sales and Purchase System For Laptop accessories, desktop sales & repairs" designed as per requirement. The software runs under the operating system with Visual Basic.Net as frontend tool and Microsoft Access 2007 as backend tool.

When user runs the system then system firstly show password screen, when correct password is entered in the textbox then click on log in button or press enter key then next screen is appeared on the screen, Without entering the correct password you can't enter in the system. If you want to exit the system then click on cancel button.

The main menu contains following items-

1. Master
2. Transaction
3. Quick Report
4. Report
5. Utility
6. About us
7. Help
8. Exit

1) MASTER-

When user click on master menu then it shows the following sub menus-

1. SUPPLIER -

When you click supplier Detail it displays screen of information. The screen contains 6 buttons as SEARCH, NEW, SAVE, UPDATE, DELETE, EXIT when you enter new record to save in database then click save button. Delete button is used to delete the record. Update button is used to update record. Search button is used to find the record. Exit button is used to close form.

2. CUSTOMER DETAIL-

If you click on customer Detail menu, then it will display the data screen. When you click customer Detail it displays screen of information this click contains 6 buttons as SEARCH, NEW, SAVE, UPDATE, DELETE, EXIT when you enter new record to save in database then click save button. Delete button is used to delete the record. Update button is used to update record. Search button is used to find the record. Exit button is used to close form.

3. PRODUCT DETAIL-

If you click on Product Detail menu, then it will display the data screen. When you click Product Detail it displays screen of information this click contains 6 buttons as SEARCH, NEW, SAVE, UPDATE, DELETE, EXIT when you enter new record to save in database then click save button. Delete button is used to delete the record. Update button is used to update record. Search button is used to find the record. Exit button is used to close form.

4. SERVICE -

If you click on Service Detail menu, then it will display the data screen. When you click Product Book Detail it displays screen of information this click contains 6 buttons as NEW, SAVE, SEARCH, UPDATE, DELETE, EXIT when you enter new record to save in database then click save button. Delete button is used to delete the record. Update button is used to update record. Search button is used to find the record. Exit button is used to close form.

5. SALES RETURN -

If you click on Service Detail menu, then it will display the data screen. When you click Product Book Detail it displays screen of information this click contains 7 buttons as NEW, SAVE, SEARCH, UPDATE, DELETE, EXIT and Pick. when you enter new record to save in database then click save button. Delete button is used to delete the record. Update button is used to update record. Search button is used to find the record. Exit button is used to close form. Pick is used to find the record.

6. PURCHASE RETURN -

If you click on Service Detail menu, then it will display the data screen. When you click Product Book Detail it displays screen of information this click contains 7 buttons as NEW, SAVE, SEARCH, UPDATE, DELETE, EXIT

and Pick. when you enter new record to save in database then click save button. Delete button is used to delete the record. Update button is used to update record. Search button is used to find the record. Exit button is used to close form. Pick is used find the record.

2) TRANSACTION-

1. SUPPLIER BILL -

If you click billing system menu, then it will display the data screen. When you billing system information it displays screen of information this click contains 7 buttons as PICK, SEARCH, NEW, SAVE, UPDATE, DELETE, EXIT when you enter new record to save in database then click save button. Pick button is used to find the record. Clear button is used to clear record. Exit button is used to close form.

2. CUSTOMER BILL -

If you click on purchase information menu, then it will display the data screen. When you click purchase information it displays screen of information this click contains 7 buttons as PICK, NEW, SAVE, UPDATE, DELETE, EXIT when you enter new record to save in database then click save button. Pick button is used to find the record. Find button is used to search the record. Clear button is used to clear the record. Exit button is used to close form.

3) QUICK REPORT -

When you click Quick Report it displays a list of report. In this option we can choose any report and we can get report from selected IDwise, Datewise & Datebetween.

1) IDwise Customer

2) IDwiseCustomer Bill

3) IDwiseSupplier Bill

4) Date Between Customer Bill

5) Datewise Supplier Bill

4) UTILITY-

1. Calculator
2. Notepad

5) ABOUT US-

Information about "Sakshi Enterprises", Islampur.

6) HELP-

If you select help, you will get help if you have any queries.

7) EXIT-

If you select exit menu, then you will exit from the system.

Chapter – 6

OUTPUTS

OUTPUTS

Source Code

SPLASH FORM -

```
namespace SakshiEnterprises
{
    public partial class Splash : Form
    {
        public Splash()
        {
            InitializeComponent();
        }

        private void Splash_Load(object sender, EventArgs e)
        {
            timer1.Enabled = true;
            timer1.Start();
            timer1.Interval = 1000;
            progressBar1.Maximum = 90;
            timer1.Tick += new EventHandler(timer1_Tick);
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            if(progressBar1.Value==10)
            {
                label1.Text = "WELCOME";
            }
            else if(progressBar1.Value==20)
            {
                label2.Text = "Sakshi Enterprises ,Islampur";
            }
            else if(progressBar1.Value==30)
            {
                label3.Text = "Project Developed By";
```

```
}
else if(progressBar1.Value==40)
{
    label4.Text = "Miss. Patil Aditi Sanjay";
}
else if(progressBar1.Value==50)
{
    label5.Text = "Miss. Patil Manasi Sudhir";
}
else if(progressBar1.Value==60)
{
    label6.Text = "Project Guide";
}
else if(progressBar1.Value==70)
{
    label7.Text = "Mrs.Shekhar S.V.";
}
else if(progressBar1.Value==80)
{
    label8.Text = "Please Wait";
}
else if(progressBar1.Value==90)
{
    timer1.Stop();
    this.Hide();
    login  l= new login();
    l.Show();

}
else
{
    progressBar1.Enabled = false;
```

```
            }
            try
            {
                progressBar1.Value++;
            }
            catch
            {

            }
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {

        }
    }
}
```

MDI -

```
namespace SakshiEnterprises
{
    public partial class MDI : Form
    {
        public MDI()
        {
            InitializeComponent();
        }

        private void menuStrip1_ItemClicked(object sender,
ToolStripItemClickedEventArgs e)
        {

        }

        private void customerToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            customer cs = new customer();
            cs.Show();
        }

        private void supplierToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            supplier sc = new supplier();
            sc.Show();
        }
```

```
        private void productToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            product ps = new product();
            ps.Show();
        }

        private void purchasereturnToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            purchasereturn pr = new purchasereturn();
            pr.Show();
        }

        private void salesreturnToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            salesreturn sr = new salesreturn();
            sr.Show();
        }

        private void serviceToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            service sc = new service();
            sc.Show();
        }

        private void salesToolStripMenuItem_Click(object sender, EventArgs e)
        {
            sales ss = new sales();
            ss.Show();
```

```
        }

        private void purchaseToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            purchase ps = new purchase();
            ps.Show();
        }

        private void customerToolStripMenuItem1_Click(object sender,
EventArgs e)
        {
            Report.rptcustomer cr1 = new Report.rptcustomer();
            cr1.Show();
        }

        private void supplierToolStripMenuItem1_Click(object sender,
EventArgs e)
        {
            Report.rptsupplier cr1 = new Report.rptsupplier();
            cr1.Show();
        }

        private void MDI_Load(object sender, EventArgs e)
        {

        }

        private void productToolStripMenuItem1_Click(object sender,
EventArgs e)
        {
            Report.rptproduct cr1 = new Report.rptproduct();
```

```
            cr1.Show();
        }

        private void purchaseToolStripMenuItem1_Click(object sender,
EventArgs e)
        {
            Report.rptpurchase cr1 = new Report.rptpurchase();
            cr1.Show();
        }

        private void salesToolStripMenuItem1_Click(object sender, EventArgs
e)
        {
            Report.rptsales cr1 = new Report.rptsales();
            cr1.Show();
        }

        private void purchasereturnToolStripMenuItem1_Click(object sender,
EventArgs e)
        {
            Report.rptpurchasereturn cr1 = new Report.rptpurchasereturn();
            cr1.Show();
        }

        private void salesreturnToolStripMenuItem1_Click(object sender,
EventArgs e)
        {
            Report.rptsalesreturn cr1 = new Report.rptsalesreturn();
            cr1.Show();
        }
```

```
    private void serviceToolStripMenuItem1_Click(object sender,
EventArgs e)
    {
        Report.rptservice cr1 = new Report.rptservice();
        cr1.Show();
    }

    private void customerInfoToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        Report.rptcustomer1 cr2 = new Report.rptcustomer1();
        cr2.Show();
    }

    private void supplierIfoToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        Report.rptsupplier1 cr2 = new Report.rptsupplier1();
        cr2.Show();
    }

    private void customerBillToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        Report.rptsales1 cr2 = new Report.rptsales1();
        cr2.Show();
    }

    private void supplierBillToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        Report.rptpurchase1 cr2 = new Report.rptpurchase1();
```

```
            cr2.Show();
        }

        private void productInfoToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            Report.rptproduct1 cr2 = new Report.rptproduct1();
            cr2.Show();
        }

        private void customerbillToolStripMenuItem1_Click(object sender,
EventArgs e)
        {
            Report.rptsales2 cr3 = new Report.rptsales2();
            cr3.Show();
        }

        private void supplierbillToolStripMenuItem1_Click(object sender,
EventArgs e)
        {
            Report.rptpurchase2 cr3 = new Report.rptpurchase2();
            cr3.Show();
        }

        private void notepadToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            System.Diagnostics.Process.Start("notepad.exe");
        }

        private void calculatorToolStripMenuItem_Click(object sender,
EventArgs e)
```

```
        {
            System.Diagnostics.Process.Start("calc.exe");
        }

        private void aboutusToolStripMenuItem1_Click(object sender,
EventArgs e)
        {
            aboutus cs = new aboutus();
            cs.Show();
        }

        private void exitToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            DialogResult d = MessageBox.Show("Are you sure you want to exit
", "Question", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        }

        private void helpToolStripMenuItem1_Click(object sender, EventArgs
e)
        {
            help cs = new help();
            cs.Show();
        }

        private void customerbillToolStripMenuItem2_Click(object sender,
EventArgs e)
        {
            Report.rptsales3 cr3 = new Report.rptsales3();
            cr3.Show();
        }
```

```
        private void supplierbillToolStripMenuItem2_Click(object sender,
EventArgs e)
        {
            Report.rptpurchase3 cr3 = new Report.rptpurchase3();
            cr3.Show();
        }
    }
}
```

Master-

1.Customer -

```
namespace SakshiEnterprises
{
    public partial class customer : Form
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataReader dr;
        SqlDataAdapter da;
        DataSet ds;
        int idno;
        string str1,str2,str3,str4,str5;
        public customer()
        {
            InitializeComponent();
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
        }
        private void customer_Load(object sender, EventArgs e)
```

```
        {
            con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=F:\\Bsc.cs[E]project\
\SakshiEnterprises\\SakshiEnterprises\\shop.mdf;Integrated
Security=True");
        }
        private void txtcustomeremail_TextChanged(object sender, EventArgs
e)
        {
        }
        private void btnserch_Click(object sender, EventArgs e)
        {
            this.Hide();
            customersearch cs = new customersearch();
            cs.Show();
        }
        private void btnexit_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        private void txtcustomerid_Leave(object sender, EventArgs e)
        {
```

```
    }
    private void txtcustomername_Leave(object sender, EventArgs e)
    {
        if(!Regex.Match(txtcustomername.Text,"^[A-Z][a-zA-
Z\\.\\s]+$").Success)
        {
            MessageBox.Show("Invalid name", "Message",
MessageBoxButtons.OKCancel, MessageBoxIcon.Error);
            txtcustomername.Focus();
            return;
        }
    }
    private void txtcustomermob_Leave(object sender, EventArgs e)
    {
        if(!Regex.Match(txtcustomermob.Text,"^\\d{10}$").Success)
        {
             MessageBox.Show("Invalid mobile no", "Message",
MessageBoxButtons.OKCancel, MessageBoxIcon.Error);
            txtcustomermob.Focus();
            return;
        }
    }
    private void txtcustomeremail_Leave(object sender, EventArgs e)
```

```
        {
            if(!Regex.Match(txtcustomeremail.Text,"^([\\w\\.\\-]+)@([\\w\\-
]+)((\\.(\\w){2,3})+)$").Success)
            {
                MessageBox.Show("Invalid email", "Message",
MessageBoxButtons.OKCancel, MessageBoxIcon.Error);
                txtcustomeremail.Focus();
                return;
            }
        }
        private void txtcustomerid_TextChanged(object sender, EventArgs e)
        {

        }
        private void txtcustomername_TextChanged(object sender, EventArgs
e)
        {

        }
        private void btndelete_Click(object sender, EventArgs e)
        {
```

```
        DialogResult d=MessageBox.Show("Are you sure you want to
delete this Record", "Question", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
        if(d==DialogResult.Yes)
        {
            con.Open();
            cmd = new SqlCommand();
            cmd.Connection = con;
            cmd.CommandText = "delete from customer where
customerid=" + txtcustomerid.Text + "";
            cmd.ExecuteNonQuery();
            MessageBox.Show("Record Deleted");
            con.Close();
        }
    }
    private void btnupdate_Click(object sender, EventArgs e)
    {
        try
        {
            con.Open();
            str4 = "update customer set customername='" +
txtcustomername.Text  + "',email='" + txtcustomeremail.Text +
"',mobileno='" + txtcustomermob.Text + "',address='" +
txtcustomeraddress.Text + "'where customerid=" + txtcustomerid.Text + "";
```

```
            cmd = new SqlCommand(str4, con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("Record Updated");
            con.Close();
        }
       catch(Exception ex)
        {
            MessageBox.Show("Error is coding");
        }
    }
    private void btnsave_Click(object sender, EventArgs e)
    {
        if(txtcustomername.Text=="")
        {
            MessageBox.Show("Please Enter customer name");
            txtcustomername.Focus();
        }
        else  if(txtcustomeremail.Text=="")
        {
            MessageBox.Show("Please Enter customer email");
            txtcustomeremail.Focus();
        }
```

```
        else if(txtcustomermob.Text=="")
        {
            MessageBox.Show("Please Enter customer mobileno");
            txtcustomermob.Focus();
        }
        else if(txtcustomeraddress.Text=="")
        {
            MessageBox.Show("Please Enter customer address");
            txtcustomeraddress.Focus();
        }
        else
        {
            con.Open();
            str3 = "insert into customer([customerid]," +
                "[customername]," +
                "[email]," +
                "[mobileno]," +
                "[address])" +

"values(@customerid,@customername,@email,@mobileno,@address)";
            cmd.CommandText = str3;
            cmd.Parameters.AddWithValue("@customerid",
txtcustomerid.Text);
```

```
            cmd.Parameters.AddWithValue("@customername",
txtcustomername.Text);
            cmd.Parameters.AddWithValue("@email", txtcustomeremail.Text);
            cmd.Parameters.AddWithValue("@mobileno",
txtcustomermob.Text);
            cmd.Parameters.AddWithValue("@address",
txtcustomeraddress.Text);
            cmd.ExecuteNonQuery();
            showgrid();
            MessageBox.Show("Record Saved");
            con.Close();
        }
    }
    private void btnnew_Click(object sender, EventArgs e)
    {
        con.Open();
        str1 = "select max(customerid) from customer";
        cmd = new SqlCommand(str1, con);
        dr = cmd.ExecuteReader();
        while(dr.Read())
        {
            str2 = dr[0].ToString();
            if (str2 == "")
```

```
            {
                txtcustomerid.Text = "1";


            }
            else
            {
                idno = Convert.ToInt32(str2) + 1;
                txtcustomerid.Text = idno.ToString();
            }
            txtcustomerid.Enabled = false;
        }
        dr.Close();
        con.Close();
    }
  }
}
```

2.Product -

```
namespace SakshiEnterprises
{
    public partial class product : Form
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataReader dr;
        SqlDataAdapter da;
        DataSet ds;
        int idno;
        string str1, str2,str3,str4,str5;
        public product()
        {
            InitializeComponent();
        }

        private void btndelete_Click(object sender, EventArgs e)
        {
            DialogResult d = MessageBox.Show("Are you sure you want to
delete this record", "Question", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
            if(d==DialogResult.Yes)
```

```
            {
                con.Open();
                cmd = new SqlCommand();
                cmd.Connection = con;
                cmd.CommandText = "delete from product where productid=" +
txtproductid.Text + "";
                cmd.ExecuteNonQuery();
                //showgrid();

                MessageBox.Show("Record Deleted");
                con.Close();
            }
        }
        private void txtproducttype_TextChanged(object sender, EventArgs e)
        {
        }
        private void product_Load(object sender, EventArgs e)
        {
            con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=F:\\Bsc.cs[E]project\
\SakshiEnterprises\\SakshiEnterprises\\shop.mdf;Integrated
Security=True");
        }
```

```
private void pictureBox1_Click(object sender, EventArgs e)
{


}
private void btnexit_Click(object sender, EventArgs e)
{
    this.Close();
}
private void txtproductname_Leave(object sender, EventArgs e)
{
}
private void btnsearch_Click(object sender, EventArgs e)
{
    this.Hide();
    productsearch ps = new productsearch();
    ps.Show();
}

private void btnupdate_Click(object sender, EventArgs e)
{
   try
    {
```

```
            con.Open();
            str4 = "update product set productname='" +
txtproductname.Text + "',producttype='" + txtproducttype.Text +
"',company='" + txtcompany.Text + "',price='" + txtprice.Text + "' where
productid=" + txtproductid.Text + "";
            cmd = new SqlCommand(str4, con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("Record Updated");
            con.Close();
        }
         catch(Exception ex)
         {
            MessageBox.Show("error in coding");
         }
      }
      private void btnsave_Click(object sender, EventArgs e)
      {
        if(txtproductname.Text=="")
         {
            MessageBox.Show("Please Enter Product Name ");
            txtproductname.Focus();
         }
        else if(txtproducttype.Text=="")
```

```
{
    MessageBox.Show("Please Enetr Product Type");
    txtproducttype.Focus();
}
else if(txtcompany.Text=="")
{
    MessageBox.Show("Please Enter Company");
    txtcompany.Focus();
}
else if(txtprice.Text=="")
{
    MessageBox.Show("Please Enter Price");
    txtprice.Focus();
}
else
{
    con.Open();
    str3 = "insert into product([productid]," +
        "[productname]," +
        "[producttype]," +
        "[company]," +
```

```
                "[price])" +
"values(@productid,@productname,@producttype,@company,@price)";
            cmd.CommandText = str3;
            cmd.Parameters.AddWithValue("@productid", txtproductid.Text);
            cmd.Parameters.AddWithValue("@productname",
txtproductname.Text);
            cmd.Parameters.AddWithValue("@producttype",
txtproducttype.Text);
            cmd.Parameters.AddWithValue("@company", txtcompany.Text);
            cmd.Parameters.AddWithValue("@price", txtprice.Text);
            cmd.ExecuteNonQuery();
            //showgrid();
            MessageBox.Show("Record Saved");
            con.Close();
        }
    }
    private void btnnew_Click(object sender, EventArgs e)
    {
        con.Open();
        str1 = "select max(productid) from product";
        cmd = new SqlCommand(str1, con);
        dr = cmd.ExecuteReader();
        while(dr.Read())
```

```
        {
            str2 = dr[0].ToString();
            if(str2=="")
            {
                txtproductid.Text = "1";
            }
            else
            {
                idno = Convert.ToInt32(str2) + 1;
                txtproductid.Text = idno.ToString();

            }
            txtproductid.Enabled = false;
        }
        dr.Close();
        con.Close();
    }
  }
}
```

3. Service -

```
namespace SakshiEnterprises
{
    public partial class service : Form
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataReader dr;
        SqlDataAdapter da;
        DataSet ds;
        int idno;
        string str1, str2, str3, str4, str5, str6;
        private void btnupdate_Click(object sender, EventArgs e)
        {
            try
            {
                con.Open();
                str4 = "update service set customerid='" + txtcustomerid.Text +
"',price='" + txtprice.Text + "',productname='" + txtproductname.Text + "',
reason'" + txtreason.Text  + "'where serviceid=" + txtserviceid.Text + "";
                cmd = new SqlCommand(str4, con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Record Updated");
```

```
            con.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error in coding");
        }
    }
    private void btnexit_Click(object sender, EventArgs e)
    {
        this.Close();
    }
    private void btnsearch_Click(object sender, EventArgs e)
    {
        this.Hide();
        servicesearch ss = new servicesearch();
        ss.Show();
    }
    private void btndelete_Click(object sender, EventArgs e)
    {
        DialogResult d = MessageBox.Show("Are you sure you want to
delete this record", "Question", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
        if (d == DialogResult.Yes)
```

```
        {
            con.Open();
            cmd = new SqlCommand();
            cmd.Connection = con;
            cmd.CommandText = "delete from  service where serviceid=" +
txtserviceid.Text + "";
            cmd.ExecuteNonQuery();
            //showgrid();
            MessageBox.Show("Record Deleted");
            con.Close();
        }
    }
    private void btnsave_Click(object sender, EventArgs e)
    {
        if(txtcustomerid.Text=="")
        {
            MessageBox.Show("PLease Enter Customer ID");
            txtcustomerid.Focus();
        }
        else if(txtprice.Text=="")
        {
            MessageBox.Show("Please Enter productname");
```

```
            txtprice.Focus();
        }
        else if(txtproductname.Text=="")
        {
            MessageBox.Show("Please Enter Price");
            txtproductname.Focus();
        }
        else if(txtreason.Text=="")
        {
            MessageBox.Show("Please Enter Reason");
            txtreason.Focus();
        }
        else
        {
            con.Open();
            str3 = "Insert into service([serviceid]," +
                "[customerid]," +
                "[price]," +
                "[productname]," +
                "[reason])"+
"Values(@serviceid,@customerid,@price,@productname,@reason)";
            cmd.CommandText = str3;
```

```
            cmd.Parameters.AddWithValue("@serviceid", txtserviceid.Text);
            cmd.Parameters.AddWithValue("@customerid",
txtcustomerid.Text);
            cmd.Parameters.AddWithValue("@price", txtprice.Text);
            cmd.Parameters.AddWithValue("@productname",
txtproductname.Text);
            cmd.Parameters.AddWithValue("@reason", txtreason.Text);
            cmd.ExecuteNonQuery();
            MessageBox.Show("Record saved");
            con.Close();
        }
    }
    public service()
    {
        InitializeComponent();
    }
    private void service_Load(object sender, EventArgs e)
    {
        con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=F:\\Bsc.cs[E]project\
\SakshiEnterprises\\SakshiEnterprises\\shop.mdf;Integrated
Security=True");
    }
```

```
private void btnnew_Click(object sender, EventArgs e)
{
    con.Open();
    str1 = "select max(serviceid) from service";
    cmd = new SqlCommand(str1, con);
    dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        str2 = dr[0].ToString();
        if (str2 == "")
        {
            txtserviceid.Text = "1";
        }
        else
        {
            idno = Convert.ToInt32(str2) + 1;
            txtserviceid.Text = idno.ToString();
        }
        txtserviceid.Enabled = false;
    }
    dr.Close();
    con.Close();
```

```
        }
    }
}
```

4. Supplier -

```
namespace SakshiEnterprises
{
    public partial class supplier : Form
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataReader dr;
        SqlDataAdapter da;
        DataSet ds;
        int idno;
        string str1, str2,str3,str4,str5;
        public supplier()
        {
            InitializeComponent();
        }
        private void supplier_Load(object sender, EventArgs e)
        {
```

```
        con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=F:\\Bsc.cs[E]project\
\SakshiEnterprises\\SakshiEnterprises\\shop.mdf;Integrated
Security=True");
    }
    private void label7_Click(object sender, EventArgs e)
    {
    }
    private void btndelete_Click(object sender, EventArgs e)
    {
        DialogResult d = MessageBox.Show("Are you sure you want to
delete the record", "Question", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
        if(d==DialogResult.Yes)
        {
            con.Open();
            cmd = new SqlCommand();
            cmd.Connection = con;
            cmd.CommandText = "delete from supplier where supplierid=" +
txtsupplierid.Text + "";
            cmd.ExecuteNonQuery();
            MessageBox.Show("Record Deleted");
            con.Close();
        }
```

```
        }
        private void btnexit_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        private void txtsuppliername_Leave(object sender, EventArgs e)
        {
           if(!Regex.Match(txtsuppliername.Text,"^[A-Z][a-zA-
Z\\.\\s]+$").Success)
            {
                MessageBox.Show("Invalid name", "Message",
MessageBoxButtons.OKCancel, MessageBoxIcon.Error);
                txtsuppliername.Focus();
               return;
            }
        }
        private void txtsuppliermobile_TextChanged(object sender, EventArgs
e)
        {
        }
        private void txtsupplieremail_Leave(object sender, EventArgs e)
        {
```

```
        if(!Regex.Match(txtsupplieremail.Text,"^([\\w\\.\\-]+)@([\\w\\-
]+)((\\.(\\w){2,3})+)$").Success)
        {
  MessageBox.Show("Invalid email", "Message",
MessageBoxButtons.OKCancel, MessageBoxIcon.Error);
            txtsupplieremail.Focus();
            return;
        }
    }
    private void txtsuppliermobile_Leave(object sender, EventArgs e)
    {
        if (!Regex.Match(txtsuppliermobile.Text, "^\\d{10}$").Success)
        {
            MessageBox.Show("Invalid mobileno", "Message",
MessageBoxButtons.OKCancel, MessageBoxIcon.Error);
            txtsuppliermobile.Focus();
            return;
        }
    }
    private void btnsearch_Click(object sender, EventArgs e)
    {
        this.Hide();
        suppliersearch ss = new suppliersearch();
```

```
            ss.Show();
        }
        private void btnupdate_Click(object sender, EventArgs e)
        {
            try
            {
                con.Open();
                str4 = "update supplier set suppliername='" +
txtsuppliername.Text + "',email='" + txtsupplieremail.Text + "',mobileno='"
+ txtsuppliermobile.Text + "',address='" + txtsupplieraddress.Text +
"'where supplierid=" + txtsupplierid.Text + "";
                cmd = new SqlCommand(str4, con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Record Updated");
                con.Close();
            }
            catch(Exception ex)
            {
                MessageBox.Show("Error in coding");
            }
        }
        private void btnsave_Click(object sender, EventArgs e)
        {
```

```
if(txtsuppliername.Text=="")
{
    MessageBox.Show("please Enter supplier name");
    txtsuppliername.Focus();
}
else if(txtsupplieremail.Text=="")
{
    MessageBox.Show("please enter supplier email");
    txtsupplieremail.Focus();
}
else if(txtsuppliermobile.Text=="")
{
    MessageBox.Show("please enter supplier mobile no");
    txtsuppliermobile.Focus();
}
else if(txtsupplieraddress.Text=="")
{
    MessageBox.Show("please enter supplier address");
    txtsupplieraddress.Focus();
}
else
{
```

```
            con.Open();
            str3 = "insert into supplier([supplierid]," +
                "[suppliername]," +
                "[email]," +
                "[mobileno]," +
                "[address])" +
"values(@supplierid,@suppliername,@email,@mobileno,@address)";
            cmd.CommandText = str3;
            cmd.Parameters.AddWithValue("@supplierid", txtsupplierid.Text);
            cmd.Parameters.AddWithValue("@suppliername",
txtsuppliername.Text);
            cmd.Parameters.AddWithValue("@email", txtsupplieremail.Text);
            cmd.Parameters.AddWithValue("@mobileno",
txtsuppliermobile.Text);
            cmd.Parameters.AddWithValue("@address",
txtsupplieraddress.Text);
            cmd.ExecuteNonQuery();
            showgrid();
            MessageBox.Show("Record Saved");
            con.Close();
        }
    }
    private void btnnew_Click(object sender, EventArgs e)
```

```
{
    con.Open();
    str1 = "select max(supplierid)from supplier";
    cmd = new SqlCommand(str1, con);
    dr = cmd.ExecuteReader();
    while(dr.Read())
    {
        str2 = dr[0].ToString();
        if(str2=="")
        {
            txtsupplierid.Text = "1";
        }
        else
        {
            idno = Convert.ToInt32(str2) + 1;
            txtsupplierid.Text = idno.ToString();
        }
        txtsupplierid.Enabled = false;
    }
    dr.Close();
    con.Close();
}
```

```
    }
}
```

5. Sales Return-

```
namespace SakshiEnterprises
{
    public partial class salesreturn : Form

    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataReader dr;
        SqlDataAdapter da;
        DataSet ds;
        int idno;
        string str1, str2,str3,str4,str5;
        public salesreturn()
        {
            InitializeComponent();
        }
        private void getdata()
        {
            try
```

```
        {
            cmd = new SqlCommand("select * from sales", con);
            cmd.CommandType = CommandType.Text;
            da = new SqlDataAdapter(cmd);
            DataSet ds = new DataSet();
            da.Fill(ds, "sales");
            dataGridView1.DataSource = ds.Tables["sales"];
        }
        catch
        {
            MessageBox.Show("No record found");
        }
    }
        private void btnnew_Click(object sender, EventArgs e)
    {
        con.Open();
        str1 = "select max(returnid)from salesreturn";
        cmd = new SqlCommand(str1, con);
        dr = cmd.ExecuteReader();
        while(dr.Read())
        {
            str2 = dr[0].ToString();
```

```
            if(str2=="")
            {
                txtreturnid.Text = "1";
            }
            else
            {
                idno = Convert.ToInt32(str2) + 1;
                txtreturnid.Text = idno.ToString();
            }
            txtreturnid.Enabled = false;
        }
        dr.Close();
        con.Close();
    }
    private void btnsearch_Click(object sender, EventArgs e)
    {
        this.Hide();
        salesreturnsearch srs = new salesreturnsearch();
        srs.Show();
    }
    private void btnexit_Click(object sender, EventArgs e)
    {
```

```
            this.Close();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            this.Hide();
            salessearch1 ss = new salessearch1();
            ss.Show();
        }
        private void txtproductid_TextChanged(object sender, EventArgs e)
        {
        }
        private void btndelete_Click(object sender, EventArgs e)
        {
            DialogResult d = MessageBox.Show("Are you sure you want to
delete this record", "Question", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
            if(d==DialogResult.Yes)
            {
                con.Open();
                cmd = new SqlCommand();
                cmd.Connection = con;
                cmd.CommandText = "delete from salesreturn where
returnid=" + txtreturnid.Text + "";
```

```
            cmd.ExecuteNonQuery();
            MessageBox.Show("Record Deleted");
            con.Close();
        }
    }
    private void btnupdate_Click(object sender, EventArgs e)
    {
        try
        {
            con.Open();
            str4 = "update salesreturn set reason='" + txtreason.Text +
"',billno='" + txtbillno.Text + "',salesid='" + txtsalesid.Text +
"',productname='" + txtproductname.Text + "' where returnid=" +
txtreturnid.Text + "";
            cmd = new SqlCommand(str4, con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("Record Updated");
            con.Close();
        }
        catch(Exception ex)
        {
            MessageBox.Show("Error in coding");
        }
```

```
}
private void btnsave_Click(object sender, EventArgs e)
{
    if (txtreason.Text == "")
    {
        MessageBox.Show("Please Enter reason");
        txtreason.Focus();
    }
    else if (txtbillno.Text == "")
    {
        MessageBox.Show("Please Enter Returnid");
        txtbillno.Focus();
    }
    else if (txtsalesid.Text == "")
    {
        MessageBox.Show("Please Enter Salesid");
        txtsalesid.Focus();
    }
    else if (txtproductname.Text == "")
    {
        MessageBox.Show("Please Enter Productname");
        txtproductname.Focus();
```

```
            }
            else
            {
                con.Open();
                str3 = "insert into salesreturn([returnid]," +
                    "[reason]," +
                    "[billno]," +
                    "[salesid]," +

                    "[productname])" +
"values(@returnid,@reason,@billno,@salesid,@productname)";
                cmd.CommandText = str3;
                cmd.Parameters.AddWithValue("@returnid", txtreturnid.Text);
                cmd.Parameters.AddWithValue("@reason", txtreason.Text);
                cmd.Parameters.AddWithValue("@billno", txtbillno.Text);
                cmd.Parameters.AddWithValue("@salesid", txtsalesid.Text);
                cmd.Parameters.AddWithValue("@productname",
txtproductname.Text);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Record saved");
                con.Close();
            }
```

```
        }
        private void salesreturn_Load(object sender, EventArgs e)
        {
            con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=F:\\Bsc.cs[E]proje
ct\\SakshiEnterprises\\SakshiEnterprises\\shop.mdf;Integrated
Security=True");
            getdata();
        }
    }
}
```

6. Purchase Return-

```
namespace SakshiEnterprises
{
    public partial class purchasereturn : Form
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataReader dr;
        SqlDataAdapter da;
        DataSet ds;
        int idno;
```

```
string str1, str2,str3,str4;
public purchasereturn()
{
    InitializeComponent();
}
private void getdata()
{
    try
    {
        cmd = new SqlCommand("select * from purchase", con);
        cmd.CommandType = CommandType.Text;
        da = new SqlDataAdapter(cmd);
        DataSet ds = new DataSet();
        da.Fill(ds, "purchase");
        dataGridView1.DataSource = ds.Tables["purchase"];
    }
    catch
    {
        MessageBox.Show("No record found");
    }
}
    private void purchasereturn_Load(object sender, EventArgs e)
```

```
        {
            con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=F:\\Bsc.cs[E]proje
ct\\SakshiEnterprises\\SakshiEnterprises\\shop.mdf;Integrated
Security=True");
            getdata();
        }
        private void btnsearch_Click(object sender, EventArgs e)
        {
            this.Hide();
            purchasereturnsearch prs = new purchasereturnsearch();
            prs.Show();
        }
        private void btndelete_Click(object sender, EventArgs e)
        {
            DialogResult d = MessageBox.Show("Are you sure you want to
delete this record", "Question", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
            if (d == DialogResult.Yes)
            {
                con.Open();
                cmd = new SqlCommand();
                cmd.Connection = con;
```

```
            cmd.CommandText = "delete from purchasereturn where
returnid=" + txtreturnid.Text + "";
            cmd.ExecuteNonQuery();


            MessageBox.Show("Record Deleted");
            con.Close();
        }
    }
    private void btnupdate_Click(object sender, EventArgs e)
    {
        try
        {
            con.Open();
            str4 = "update purchasereturn set purchaseid='" +
txtpurchaseid.Text + "',billno='" + txtbillno.Text + "',productid='" +
txtproductid.Text + "',productname='" + txtprodutname.Text +
"',reason='" + txtreason.Text + "'where returnid=" + txtreturnid.Text +
"";
            cmd = new SqlCommand(str4, con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("Recod Updated");
            con.Close();
        }
```

```
    catch(Exception ex)
    {
        MessageBox.Show("Error in coding");
    }
}
private void button1_Click(object sender, EventArgs e)
{
    this.Hide();
    purchasesearch1 ps = new purchasesearch1();
    ps.Show();
}
private void btnexit_Click(object sender, EventArgs e)
{
    this.Close();
}
private void btnsave_Click(object sender, EventArgs e)
{
    if(txtpurchaseid.Text=="")
    {
        MessageBox.Show("Please Enter Purchase ID ");
        txtpurchaseid.Focus();
    }
```

```
else if(txtbillno.Text=="")
{
    MessageBox.Show("Please Enter Bill NO");
    txtbillno.Focus();
}
else if(txtproductid.Text=="")
{
    MessageBox.Show("Please Enter Product ID");
    txtproductid.Focus();
}
else if(txtprodutname.Text=="")
{
    MessageBox.Show("Please Enter Product Name");
    txtprodutname.Focus();
}
else if(txtreason.Text=="")
{
    MessageBox.Show("Please Enter Reason");
    txtreason.Focus();
}
else
{
```

```
            con.Open();
            str3 = "insert into purchasereturn([returnid]," +
                "[purchaseid]," +
                "[billno]," +
                "[productid]," +
                "[productname]," +
                "[reason])" +
"values(@returnid,@purchaseid,@billno,@productid,@productname,@r
eason)";
            cmd.CommandText = str3;
            cmd.Parameters.AddWithValue("@returnid", txtreturnid.Text);
            cmd.Parameters.AddWithValue("@purchaseid",
txtpurchaseid.Text);
            cmd.Parameters.AddWithValue("@billno", txtbillno.Text);
            cmd.Parameters.AddWithValue("@productid",
txtproductid.Text);
            cmd.Parameters.AddWithValue("@productname",
txtprodutname.Text);
            cmd.Parameters.AddWithValue("@reason", txtreason.Text);
            cmd.ExecuteNonQuery();
            MessageBox.Show("Record Saved");
            con.Close();
        }
```

```
}
private void btnnew_Click(object sender, EventArgs e)
{
    con.Open();
    str1 = "select max(returnid) from purchasereturn";
    cmd = new SqlCommand(str1, con);
    dr = cmd.ExecuteReader();
    while(dr.Read())
    {
        str2 = dr[0].ToString();
        if(str2=="")
        {
            txtreturnid.Text = "1";
        }
        else
        {
            idno = Convert.ToInt32(str2) + 1;
            txtreturnid.Text = idno.ToString();
        }
        txtreturnid.Enabled = false;
    }
    dr.Close();
```

```
            con.Close();
        }
    }
}
```

Transaction-

1. Customer Bill –

```
namespace SakshiEnterprises
{
    public partial class sales : Form
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataReader dr;
        SqlDataAdapter da;
        DataSet ds;
        int idno;
        string str1, str2,str3,str4,str5,str6,str7;
        public sales()
        {
            InitializeComponent();
        }
        private void getdata()
        {
            try
            {
```

```
                cmd = new SqlCommand("select * from customer",
con);
                cmd.CommandType = CommandType.Text;
                da = new SqlDataAdapter(cmd);
                DataSet ds = new DataSet();
                da.Fill(ds, "customer");
                dataGridView1.DataSource = ds.Tables["customer"];
            }
            catch
            {
                MessageBox.Show("No record found");
            }
        }
        private void sales_Load(object sender, EventArgs e)
        {
            con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=F:\\Bsc.c
s[E]project\\SakshiEnterprises\\SakshiEnterprises\\shop.mdf;Inte
grated Security=True");
            getdata();
        }
        private void btnsearch_Click(object sender, EventArgs e)
        {
            this.Hide();
            salessearch ss = new salessearch();
            ss.Show();
        }
        private void btnexit_Click(object sender, EventArgs e)
        {
            this.Close();
```

```
        }
        private void txtcustomername_Leave(object sender,
EventArgs e)
        {
            if (!Regex.Match(txtcustomername.Text, "^[A-z][a-zA-
Z\\.\\s]+$").Success)
            {
                MessageBox.Show("Invalid name", "Message",
MessageBoxButtons.OKCancel, MessageBoxIcon.Error);
                txtcustomername.Focus();
                return;
            }
        }
        private void txttotalgst_TextChanged(object sender,
EventArgs e)
        {
            int t, gsr, ga, gt;
            t = Convert.ToInt32(txttotal.Text);
            gsr = Convert.ToInt32(txttotalgst.Text);
            ga = t * gsr / 100;
            gt = t + ga;
            txtgrandtotal.Text = gt.ToString();
        }
        private void txtquantity_TextChanged(object sender,
EventArgs e)
        {
            int q, p, t,gsa,gt;
            q = Convert.ToInt32(txtquantity.Text);
            p = txtprice?.Text!="" ? Convert.ToInt32(txtprice?.Text):0;
            t = p * q;
```

```
            txttotal.Text = t.ToString();
        }
        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
            foreach (DataGridViewRow dr in
dataGridView1.SelectedRows)
            {
                txtcustomerid.Text = dr.Cells[0].Value.ToString();
                txtcustomername.Text = dr.Cells[1].Value.ToString();

                txtprice.Text = dr.Cells[8].Value.ToString();
            }
        }
        private void btnpick_Click(object sender, EventArgs e)
        {
            this.Hide();
            customersearch1 cs = new customersearch1();
            cs.Show();
        }
        private void btndelete_Click(object sender, EventArgs e)
        {
            DialogResult d = MessageBox.Show("Are you sure you
want to delete this record", "Question",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (d == DialogResult.Yes)
            {
                con.Open();
                cmd = new SqlCommand();
                cmd.Connection = con;
```

```
                cmd.CommandText = "delete from sales where
salesid=" + txtsalesid.Text + "";
                cmd.ExecuteNonQuery();
                MessageBox.Show("Record Deleted");
                con.Close();
            }
        }
        private void btnupdate_Click(object sender, EventArgs e)
        {
            try
            {
             con.Open();
             str4 = "update sales set billno='"+txtbillno.Text+"',
customerid='" + txtcustomerid.Text + "',customername='"+
txtcustomername.Text +"',productid='" + txtproductid.Text +
"',productname='" + txtproductname.Text + "',quality='" +
txtquality.Text + "', quantity='"+txtquantity.Text +"', price='" +
txtprice.Text + "',total='"+txttotal.Text+ "', totalgst='" +
txttotalgst.Text + "',grandtotal='" + txtgrandtotal.Text +
"',salesdate='" + txtsalesdate.Text + "',warrantydate='" +
txtwarrantyupto.Text + "'where salesid=" + txtsalesid.Text + "";
             cmd = new SqlCommand(str4, con);
             cmd.ExecuteNonQuery();
             MessageBox.Show("Record Updated");
             con.Close();
            }
           catch (Exception ex)
            {
                MessageBox.Show("Error in coding");
            }
```

```
}
private void btnsave_Click(object sender, EventArgs e)
{
    if(txtsalesid.Text=="")
    {
        MessageBox.Show("Please enter billno");
        txtbillno.Focus();
    }
    else if(txtcustomerid.Text=="")
    {
        MessageBox.Show("Please Enter Customerid");
        txtcustomerid.Focus();
    }
    else if(txtcustomername.Text=="")
    {
        MessageBox.Show("Please enter customername");
        txtcustomername.Focus();
    }
    else if(txtproductid.Text=="")
    {
        MessageBox.Show("Please Enter Productid");
        txtproductid.Focus();
    }
    else if(txtproductname.Text=="")
    {
        MessageBox.Show("Please Enter Productname");
        txtproductname.Focus();
    }
    else if(txtquality.Text=="")
    {
```

```
        MessageBox.Show("Please Enter quality");
        txtquality.Focus();
    }
    else if(txtquantity.Text=="")
    {
        MessageBox.Show("please Enter quantity");
        txtquantity.Focus();
    }
    else if(txtprice.Text=="")
    {
        MessageBox.Show("Please Enter Price");
        txtprice.Focus();
    }
    else if(txttotal.Text=="")
    {
        MessageBox.Show("Please enter total");
        txttotal.Focus();
    }
    else if(txttotalgst.Text=="")
    {
        MessageBox.Show("Please Enter Totalgst");
        txttotalgst.Focus();
    }
    else if(txtgrandtotal.Text=="")
    {
        MessageBox.Show("Please Enter Grandtotal");
        txtgrandtotal.Focus();
    }
    else if(txtsalesdate.Text=="")
    {
```

```
            MessageBox.Show("Please Enter salesdate");
            txtsalesdate.Focus();
        }
        else if(txtwarrantyupto.Text=="")
        {
            MessageBox.Show("Please Enter Warrantyupto");
            txtwarrantyupto.Focus();
        }
        else
        {
            con.Open();
            str3 = "insert into sales([salesid],"+
                "[billno],"+
                "[customerid]," +
                "[customername],"+
                "[productid]," +
                "[productname]," +
                "[quality]," +
                "[quantity],"+
                "[price]," +
                "[total],"+
                "[totalgst]," +
                "[grandtotal]," +
                "[salesdate]," +
                "[warrantydate])" +
"values(@salesid,@billno,@customerid,@customername,@prod
uctid,@productname,@quality,@quantity,@price,@total,@total
gst,@grandtotal,@salesdate,@warrantydate)";
            cmd.CommandText = str3;
```

```
            cmd.Parameters.AddWithValue("@salesid",
txtsalesid.Text);
            cmd.Parameters.AddWithValue("@billno",
txtbillno.Text);
            cmd.Parameters.AddWithValue("@customerid",
txtcustomerid.Text);
            cmd.Parameters.AddWithValue("@customername",
txtcustomername.Text);
            cmd.Parameters.AddWithValue("@productid",
txtproductid.Text);
            cmd.Parameters.AddWithValue("@productname",
txtproductname.Text);
            cmd.Parameters.AddWithValue("@quality",
txtquality.Text);
            cmd.Parameters.AddWithValue("@quantity",
txtquantity.Text);
            cmd.Parameters.AddWithValue("@price",
txtprice.Text);
            cmd.Parameters.AddWithValue("@total",
txttotal.Text);
            cmd.Parameters.AddWithValue("@totalgst",
txttotalgst.Text);

 cmd.Parameters.AddWithValue("@grandtotal",
txtgrandtotal.Text);

cmd.Parameters.AddWithValue("@salesdate",Convert.ToDateTi
me(txtsalesdate.Text));
```

```
 cmd.Parameters.AddWithValue("@warrantydate",
Convert.ToDateTime(txtwarrantyupto.Text));
                    cmd.ExecuteNonQuery();

                    MessageBox.Show("Record Saved");
                    con.Close();
                }
            }
            private void btnnew_Click(object sender, EventArgs e)
            {
                con.Open();
                str1 = "select max(salesid)from sales";
                cmd = new SqlCommand(str1, con);
                dr = cmd.ExecuteReader();
                while(dr.Read())
                {
                    str2 = dr[0].ToString();
                    if(str2=="")
                    {
                        txtsalesid.Text = "1";
                    }
                    else
                    {
                        idno = Convert.ToInt32(str2) + 1;
                        txtsalesid.Text = idno.ToString();
                    }
                    txtsalesid.Enabled = false;
                }
```

```
                dr.Close();
                con.Close();
            }
        }
    }
```

2. Supplier Bill -

```
namespace SakshiEnterprises
{
    public partial class purchase : Form
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataReader dr;
        SqlDataAdapter da;
        DataSet ds;
        int idno;
        string str1, str2,str3,str4,str5,str6;
        public purchase()
        {
            InitializeComponent();
        }
        private void getdata()
        {
            try
            {
                cmd = new SqlCommand("select * from supplier", con);
                cmd.CommandType = CommandType.Text;
                da = new SqlDataAdapter(cmd);
                DataSet ds = new DataSet();
                da.Fill(ds, "supplier");
                dataGridView1.DataSource = ds.Tables["supplier"];
            }
            catch
            {
                MessageBox.Show("no Record Found");
            }
```

```
        }
        private void btndelete_Click(object sender, EventArgs e)
        {
            DialogResult d = MessageBox.Show("Are you sure you want to
delete this record","Question", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
            if(d==DialogResult.Yes)
            {
                con.Open();
                cmd = new SqlCommand();
                cmd.Connection = con;
                cmd.CommandText = "delete from purchase where purchaseid="
+ txtpurchaseid.Text + "";
                cmd.ExecuteNonQuery();

                MessageBox.Show("Record Deleted");
                con.Close();
            }
        }
        private void purchase_Load(object sender, EventArgs e)
        {
            con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=F:\\Bsc.cs[E]project\
\SakshiEnterprises\\SakshiEnterprises\\shop.mdf;Integrated
Security=True");

            getdata();
        }
        private void btnexit_Click(object sender, EventArgs e)
        {
            this.Close();
        }
```

```
private void txtgst_TextChanged(object sender, EventArgs e)
{
    int t, gsr, ga, gt;
    t = Convert.ToInt32(txtprice.Text);
    gsr = Convert.ToInt32(txtgst.Text);
    ga = t * gsr / 100;
    gt = t + ga;
    txtgrandtotal.Text = gt.ToString();
}

private void txtquntity_TextChanged(object sender, EventArgs e)
{
        int r, q, t;
        r = txtprice?.Text!=""? Convert.ToInt32(txtprice?.Text):0;
        q = txtquntity?.Text!="" ? Convert.ToInt32(txtquntity?.Text):0;
        t = r * q;
        txtprice.Text = t.ToString();
}

private void btnpick_Click(object sender, EventArgs e)
{
    this.Hide();
    suppliersearch1 cs = new suppliersearch1();
    cs.Show();
}
private void btnsearch_Click(object sender, EventArgs e)
{
    this.Hide();
    purchasesearch phs = new purchasesearch();
    phs.Show();
}
private void btnupdate_Click(object sender, EventArgs e)
```

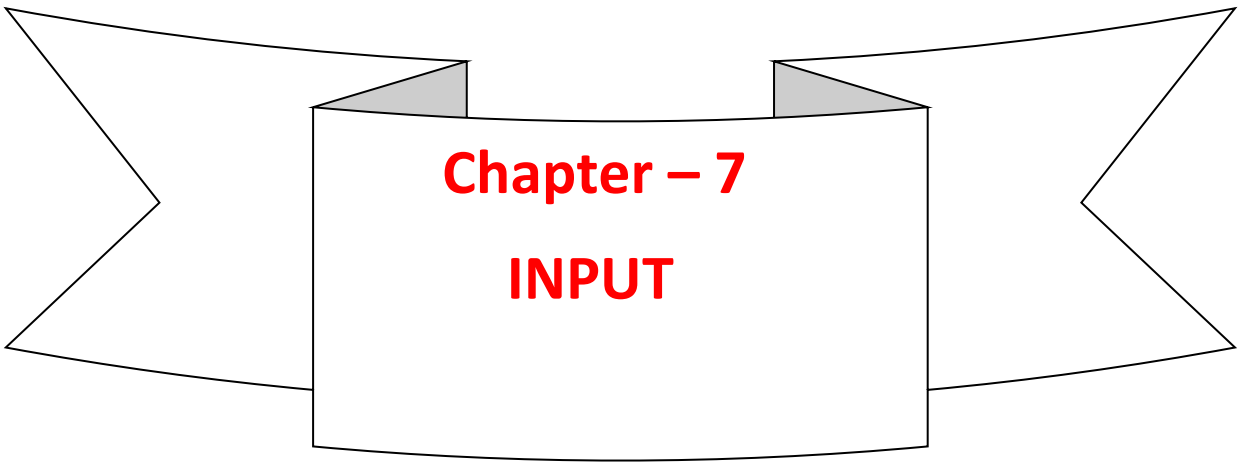
```
        {
            try
            {
                con.Open();
                str4 = "update purchase set productname='" +
txtproductname.Text + "',billno='" + txtbillno.Text + "',productid='" +
txtproductid.Text + "',
supplierid='"+txtsuppliername.Text+"',suppliername='"
+txtsuppliername.Text+"', price='" + txtprice.Text + "',quantity='" +
txtquntity.Text + "',gst='" + txtgst.Text + ",grandtotal='" +
txtgrandtotal.Text + "',manufacturdate='" + txtmanufacturdate.Text +
"',purchasedate='" + txtpurchasedate.Text + "'where purchaseid=" +
txtpurchaseid.Text + "";
                cmd = new SqlCommand(str4, con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Record Updated");
                con.Close();
            }
            catch(Exception ex)
            {
                MessageBox.Show("Error in coding");
            }
        }
        private void btnsave_Click(object sender, EventArgs e)
        {
         if(txtproductname.Text=="")
            {
                MessageBox.Show("Please Eneter Product Name");
                txtproductname.Focus();
            }
         else if(txtbillno.Text=="")
            {
```

```
            MessageBox.Show("Please Enter Bill No");
            txtbillno.Focus();
        }
    else if(txtproductid.Text=="")
        {
            MessageBox.Show("Please Enter Product Id");
            txtproductid.Focus();
        }
        else if (txtsupplierid.Text == "")
        {
            MessageBox.Show("PLease Enter Supplier Id");
            txtsupplierid.Focus();
        }
        else if (txtsuppliername.Text == "")
        {
            MessageBox.Show("please enter supplier name");
            txtsuppliername.Focus();
        }
        else if(txtprice.Text=="")
        {
            MessageBox.Show("Please Enter Price");
            txtprice.Focus();
        }
        else if(txtquntity.Text=="")
        {
            MessageBox.Show("Please Enter Quantity");
            txtquntity.Focus();
        }
        else if(txtgst.Text=="")
        {
            MessageBox.Show("Please Enter Gst");
            txtgst.Focus();
```

```
}
else if(txtgrandtotal.Text=="")
{
    MessageBox.Show("Please Enter Grandtoatl");
    txtgrandtotal.Focus();
}
else if(txtmanufacturdate.Text=="")
{
    MessageBox.Show("Please Enter Manufacturdate");
    txtmanufacturdate.Focus();
}
else if(txtpurchasedate.Text=="")
{
    MessageBox.Show("Please Enter Purchasedate");
    txtpurchasedate.Focus();
}
else
{
    con.Open();
    str3 = "insert into purchase([purchaseid]," +
        "[productname]," +
        "[billno]," +
        "[productid]," +
        "[supplierid]," +
        "[suppliername] ," +
        "[price]," +
        "[quantity]," +
        "[gst]," +
        "[grandtotal]," +
        "[manufacturdate]," +
        "[purchasedate])" +
```

```
"values(@purchaseid,@productname,@billno,@productid,@supplierid,@su
ppliername,@price,@quantity,@gst,@grandtotal,@manufacturdate,@purc
hasedate)";
                cmd.CommandText = str3;
                cmd.Parameters.AddWithValue("@purchaseid",
txtpurchaseid.Text);
                cmd.Parameters.AddWithValue("@productname",
txtproductname.Text);
                cmd.Parameters.AddWithValue("@billno", txtbillno.Text);
                cmd.Parameters.AddWithValue("@productid", txtproductid.Text);
                cmd.Parameters.AddWithValue("@supplierid", txtsupplierid.Text);
                cmd.Parameters.AddWithValue("@suppliername",
txtsuppliername.Text);
                cmd.Parameters.AddWithValue("@price", txtprice.Text);
                cmd.Parameters.AddWithValue("@quantity", txtquntity.Text);
                cmd.Parameters.AddWithValue("@gst", txtgst.Text);
                cmd.Parameters.AddWithValue("@grandtotal",
txtgrandtotal.Text);

cmd.Parameters.AddWithValue("@manufacturdate",Convert.ToDateTime(tx
tmanufacturdate.Text));

cmd.Parameters.AddWithValue("@purchasedate",Convert.ToDateTime(txtp
urchasedate.Text));
                cmd.ExecuteNonQuery();
                MessageBox.Show("Record Saved");
                con.Close();
            }
        }

        private void btnnew_Click(object sender, EventArgs e)
```

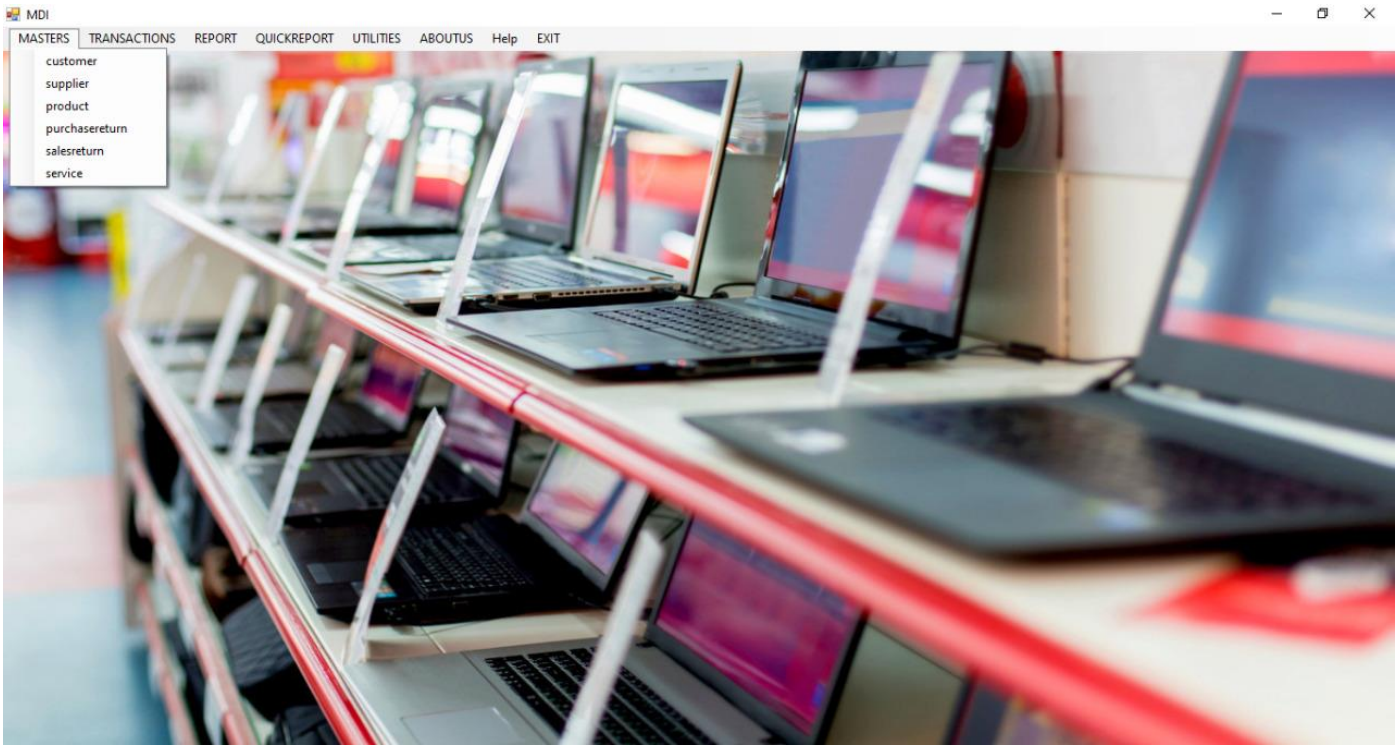
```
        {
            con.Open();
            str1 = "select max(purchaseid) from purchase";
            cmd = new SqlCommand(str1, con);
            dr = cmd.ExecuteReader();
            while(dr.Read())
            {

                str2 = dr[0].ToString();
                if(str2=="")
                {
                    txtpurchaseid.Text = "1";

                }
                else
                {
                    idno = Convert.ToInt32(str2) + 1;
                    txtpurchaseid.Text = idno.ToString();
                }
                txtpurchaseid.Enabled = false;

            }

            dr.Close();
            con.Close();
        }
    }
}
```

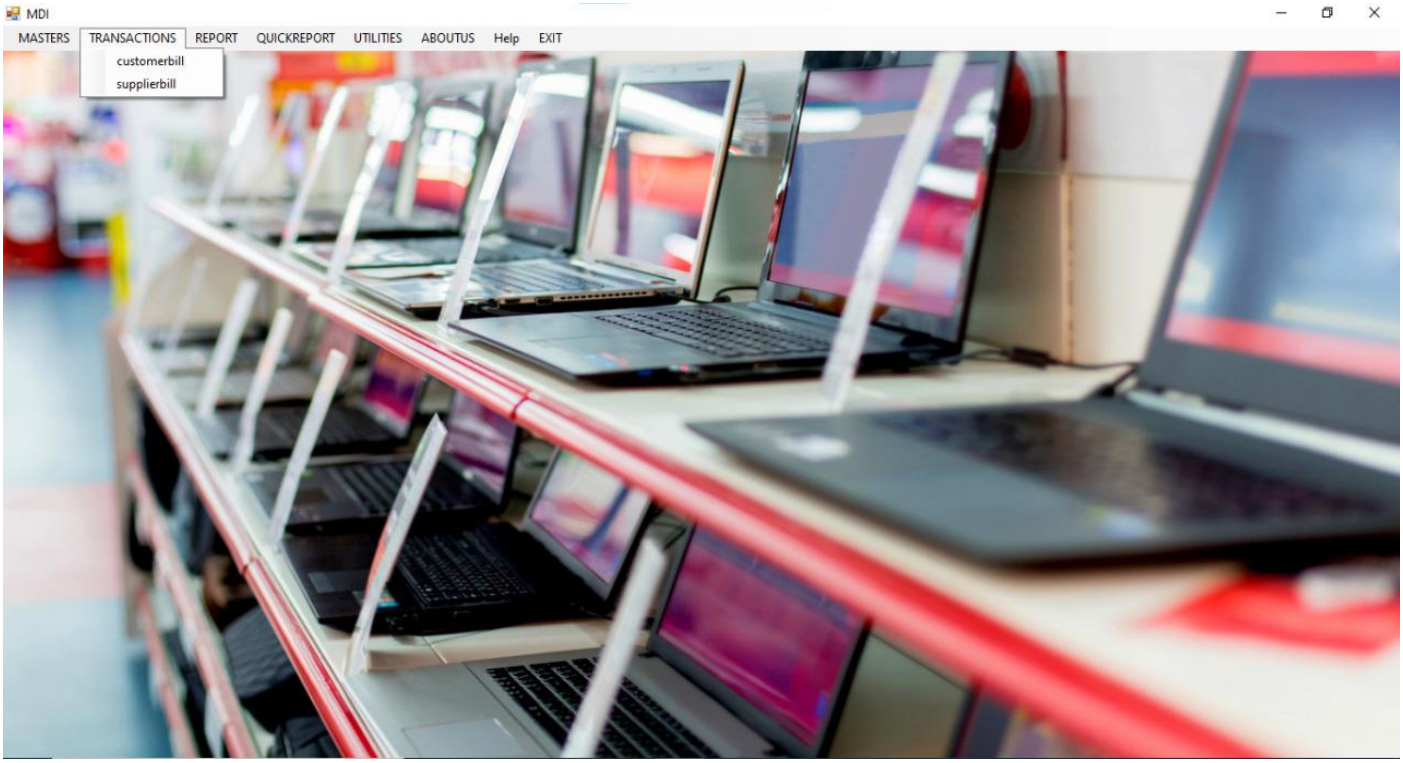

Chapter – 7
INPUT

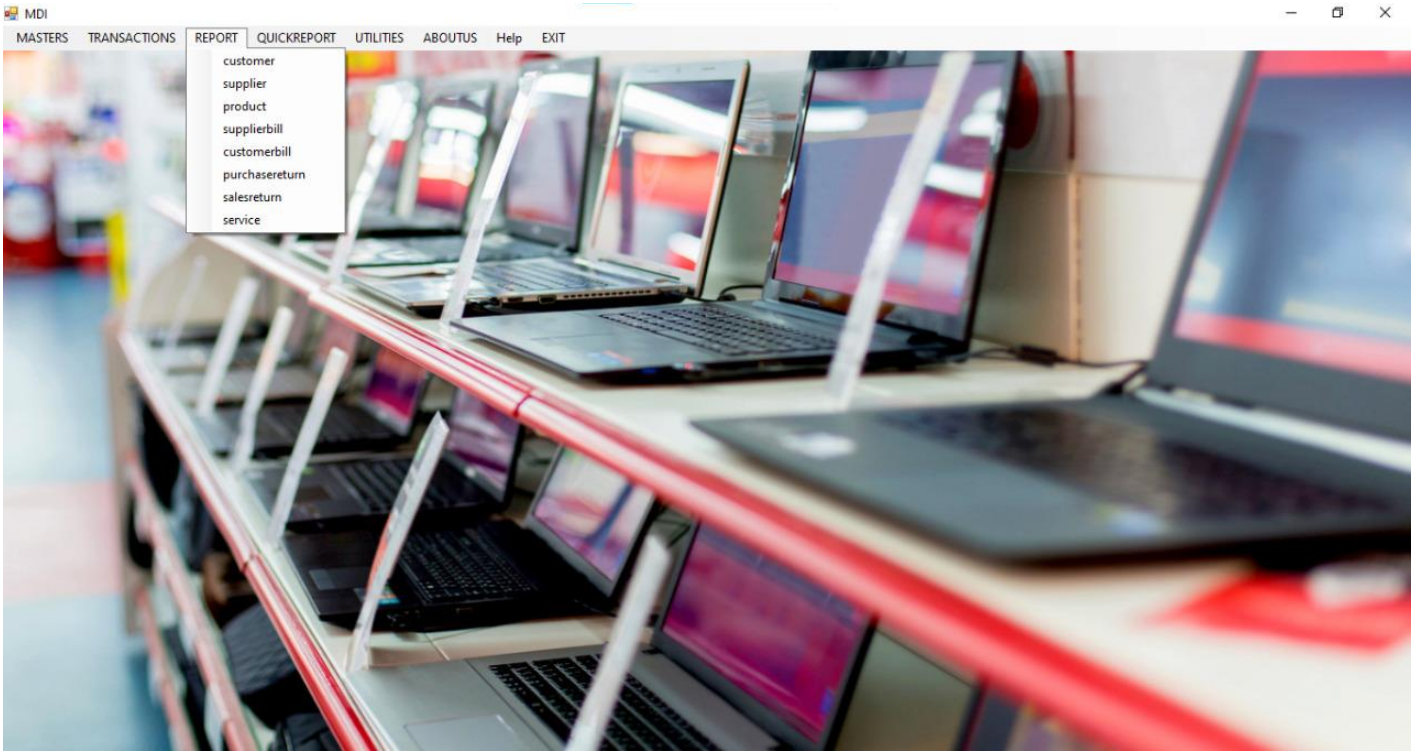
.

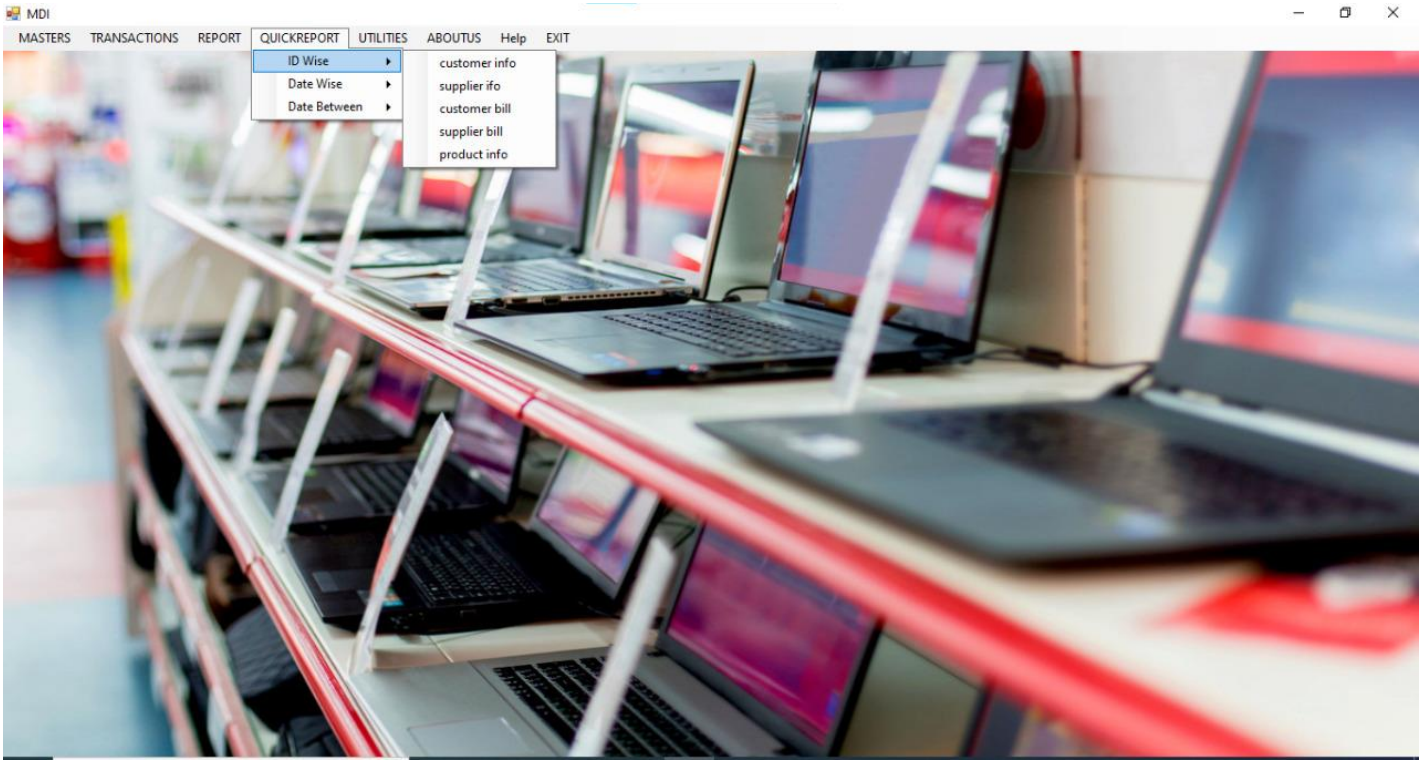
Splash Screen

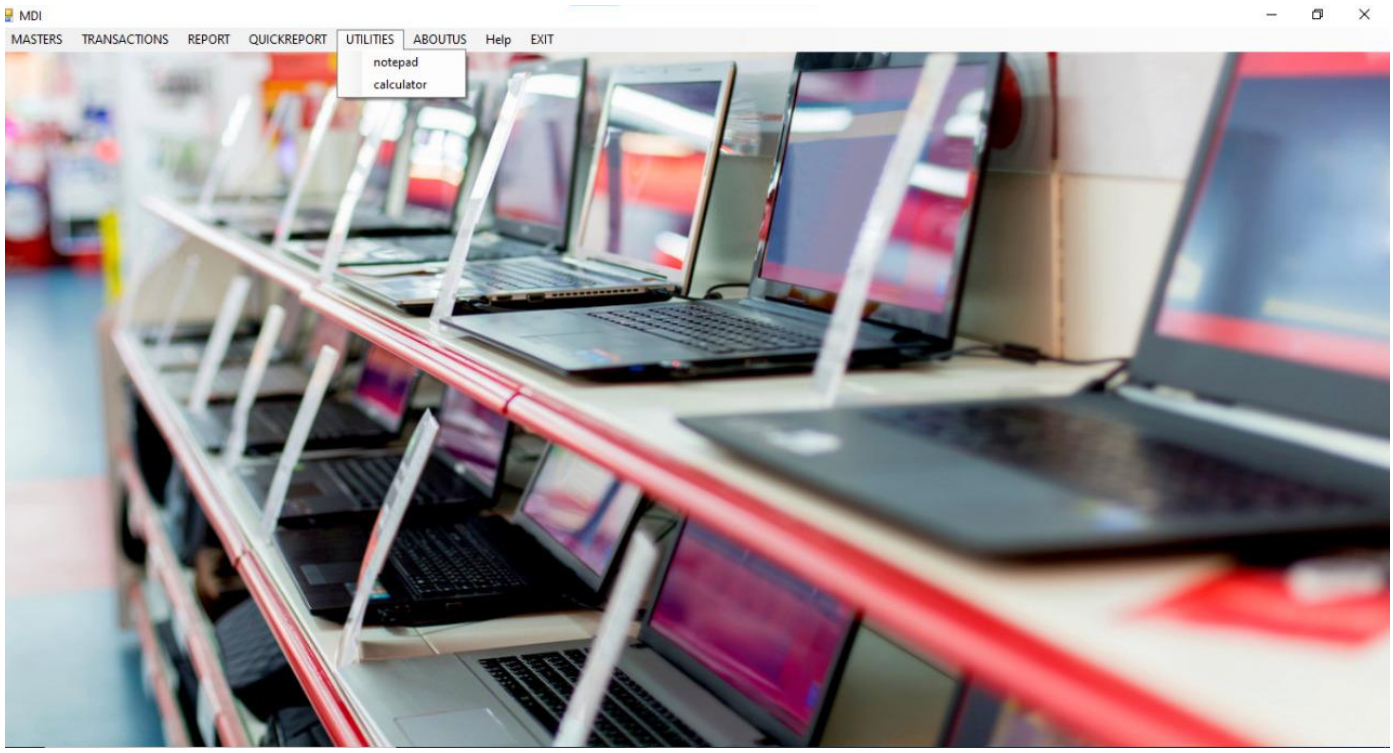
MDI

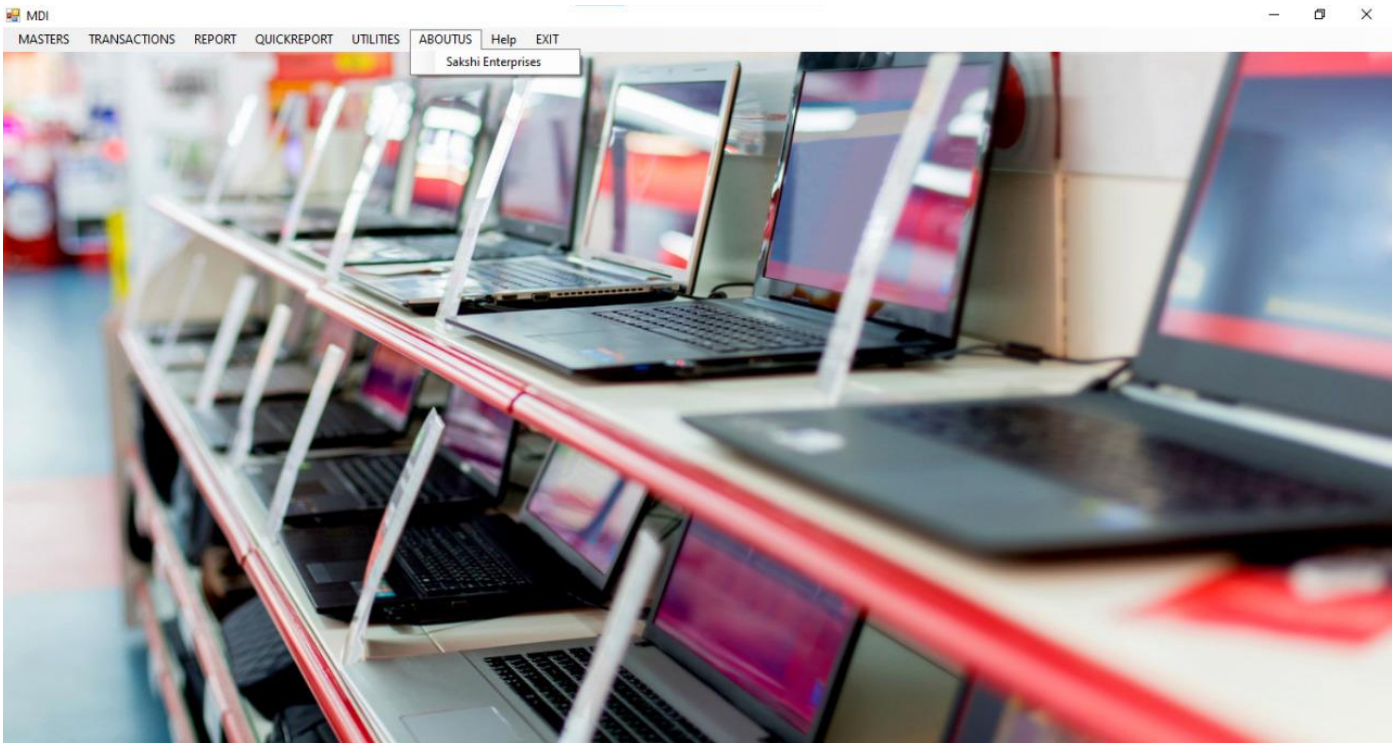

MDI
MASTERS
TRANSACTIONS
REPORT
QUICKREPORT
UTILITIES
ABOUTUS
Help
EXIT
customer
supplier
product
purchasereturn
salesreturn
service

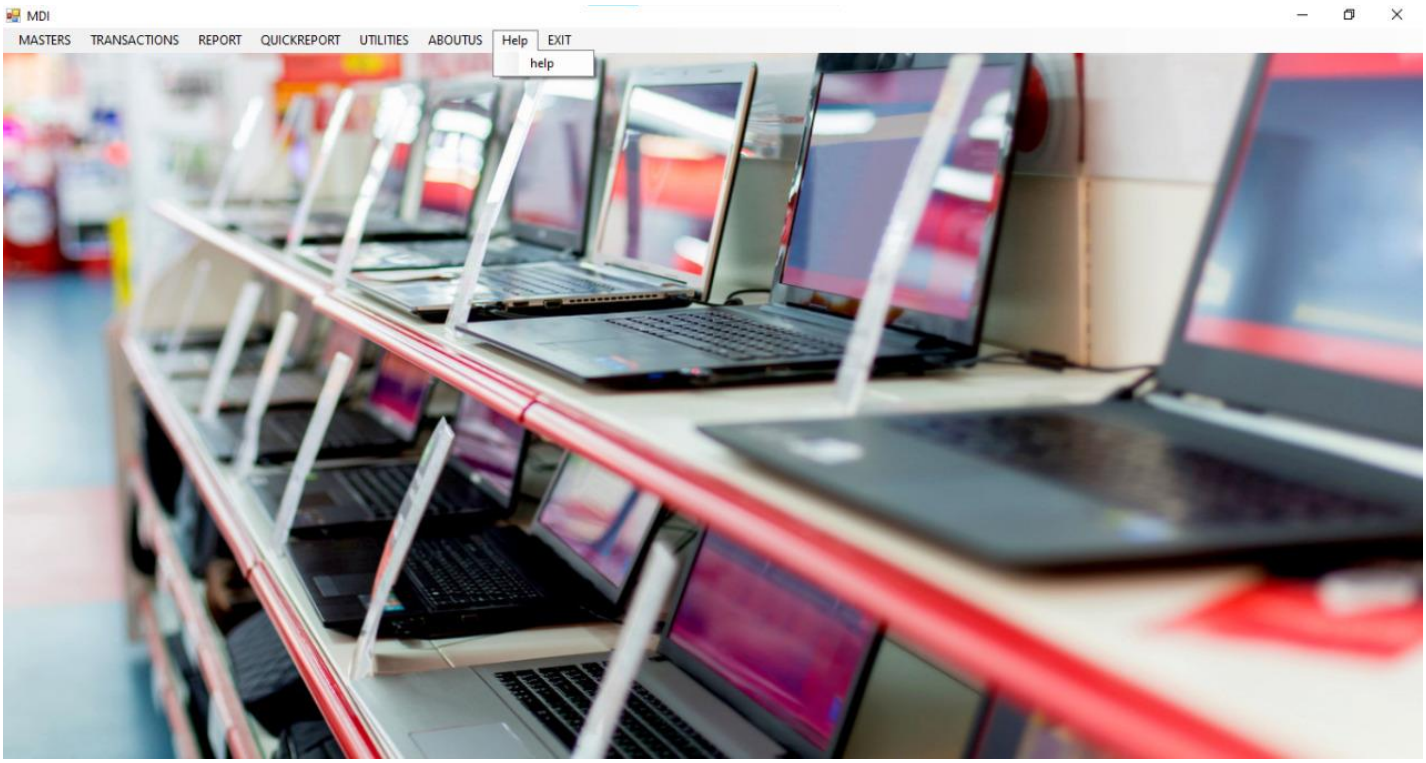

MDI
MASTERS
TRANSACTIONS
REPORT
QUICKREPORT
UTILITIES
ABOUTUS
Help
EXIT
customerbill
supplierbill

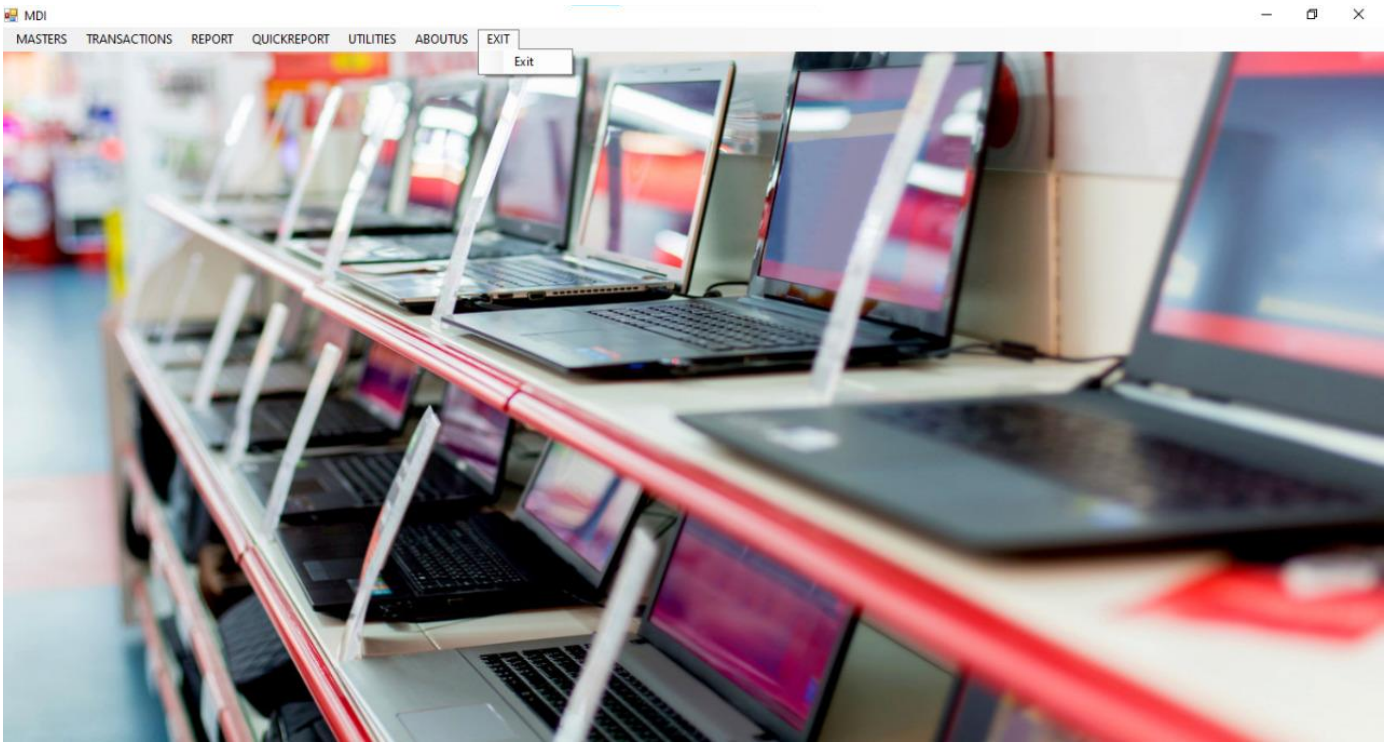

MDI
MASTERS
TRANSACTIONS
REPORT
QUICKREPORT
UTILITIES
ABOUTUS
Help
EXIT
customer
supplier
product
supplierbill
customerbill
purchasereturn
salesreturn
service


MDI
MASTERS
TRANSACTIONS
REPORT
QUICKREPORT
UTILITIES
ABOUTUS
Help
EXIT
ID Wise
Date Wise
Date Between
customer info
supplier ifo
customer bill
supplier bill
product info

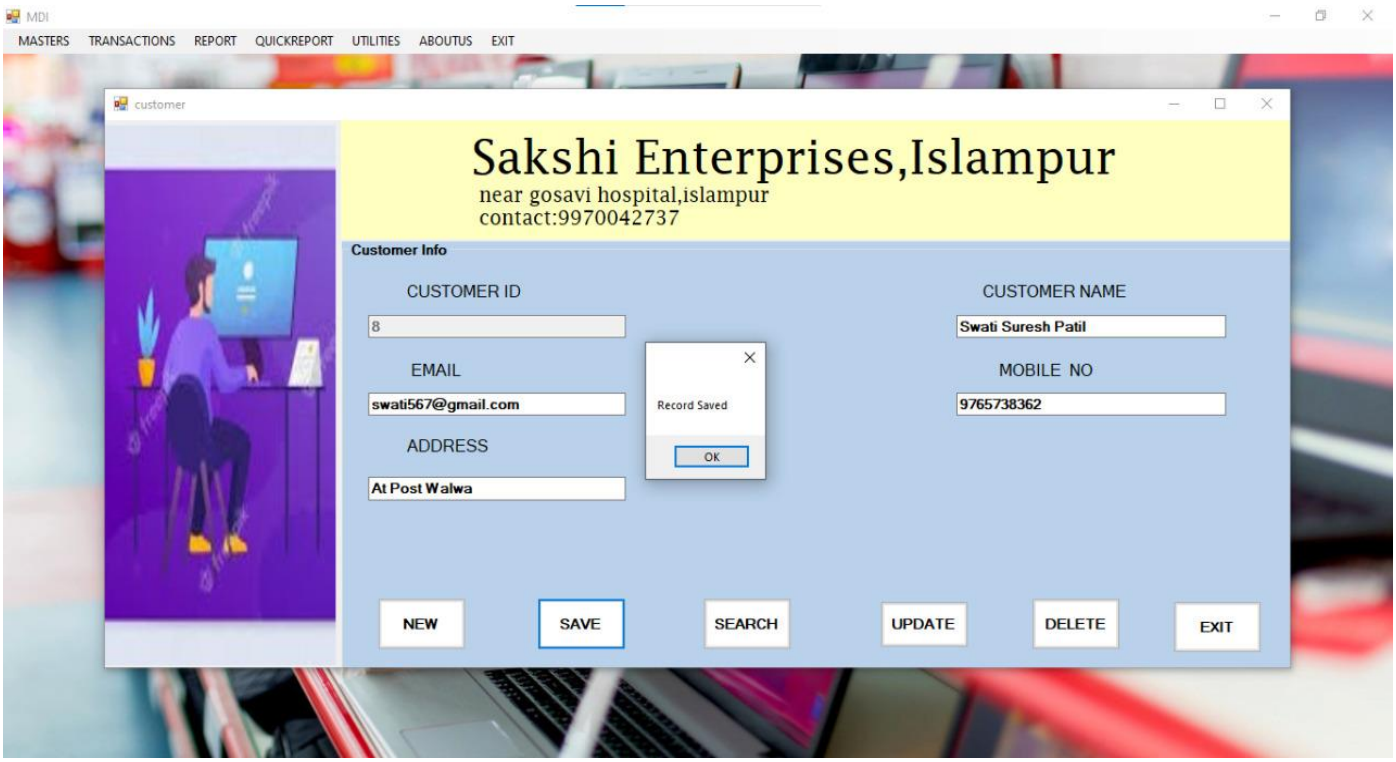

MDI
MASTERS
TRANSACTIONS
REPORT
QUICKREPORT
UTILITIES
ABOUTUS
Help
EXIT
notepad
calculator


MDI
MASTERS
TRANSACTIONS
REPORT
QUICKREPORT
UTILITIES
ABOUTUS
Help
EXIT
Sakshi Enterprises

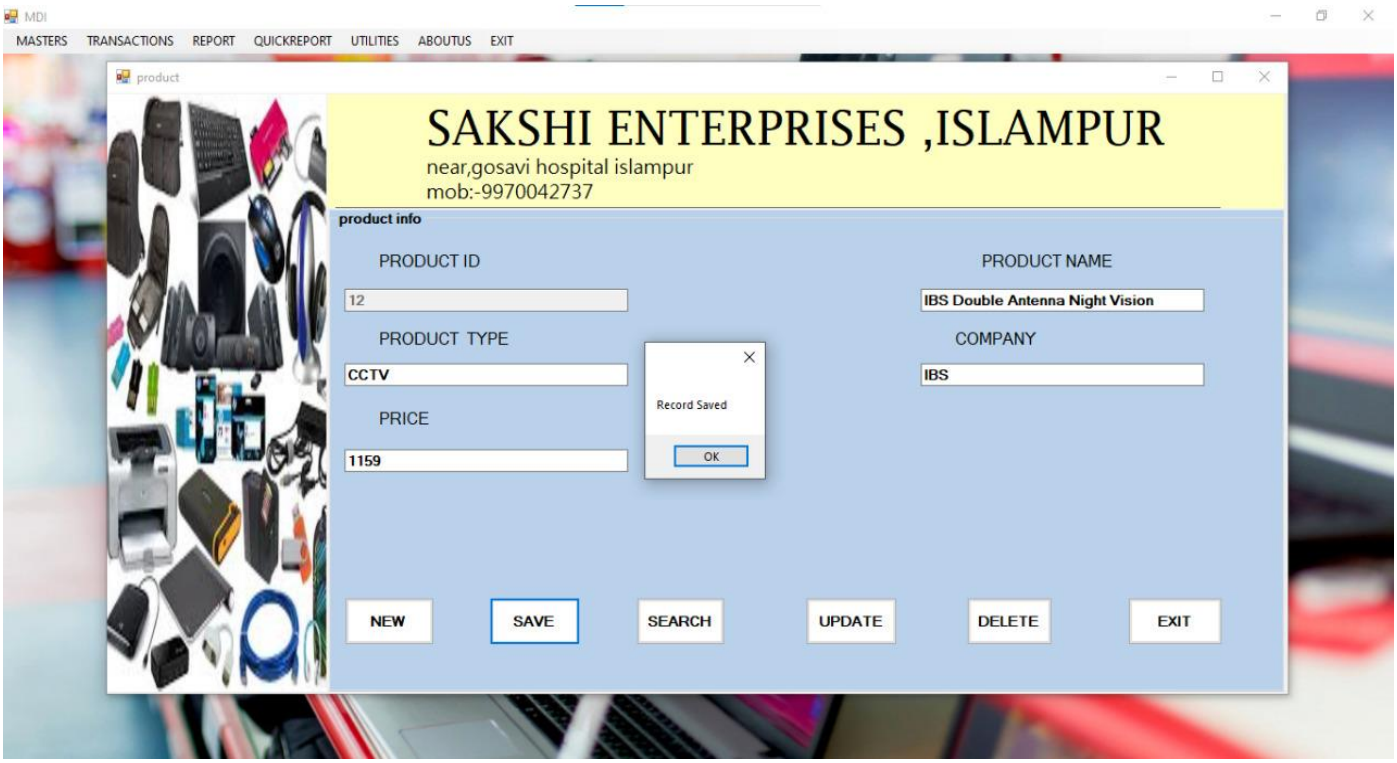

MDI
MASTERS
TRANSACTIONS
REPORT
QUICKREPORT
UTILITIES
ABOUTUS
Help
EXIT
help


MDI
MASTERS
TRANSACTIONS
REPORT
QUICKREPORT
UTILITIES
ABOUTUS
EXIT
Exit

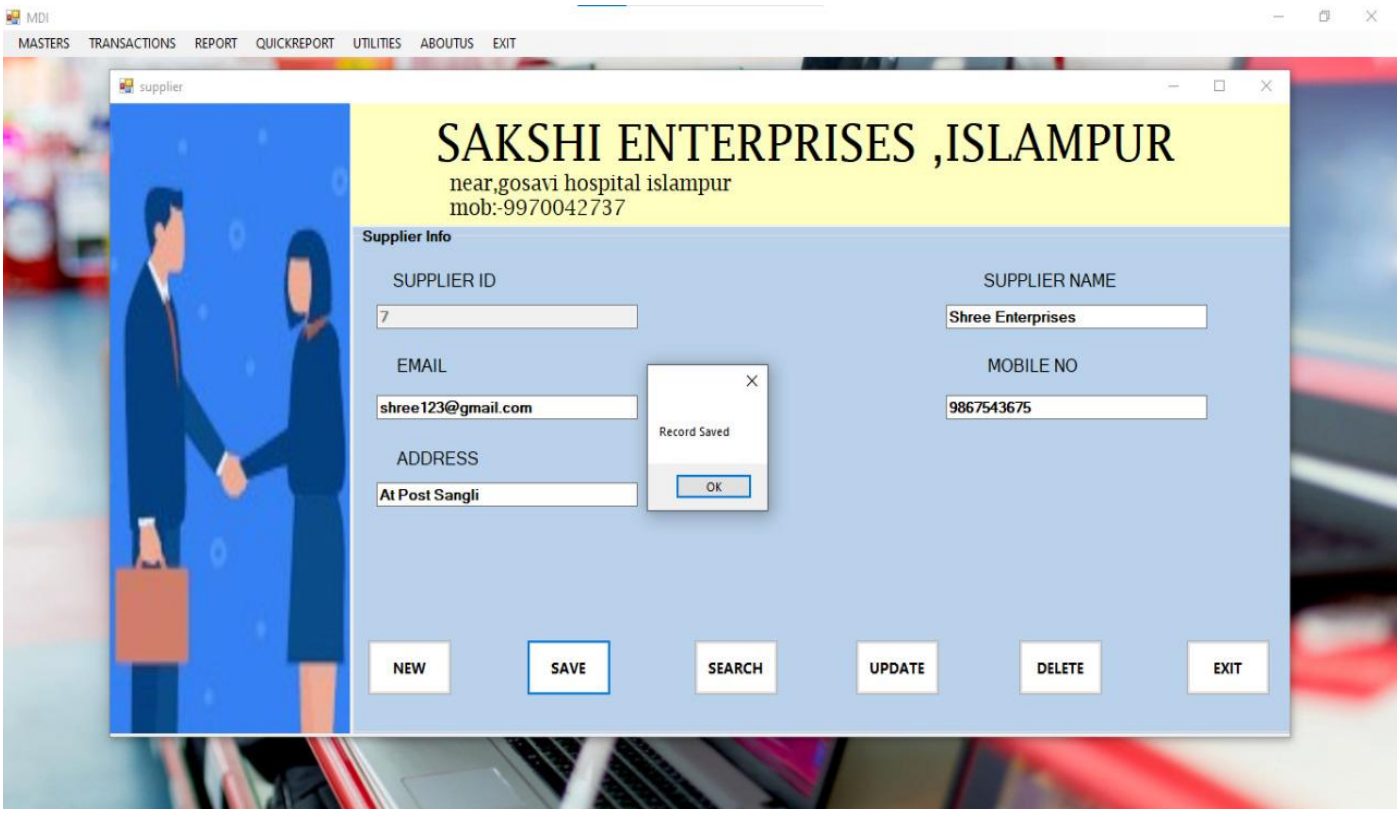
Customer


MDI
MASTERS
TRANSACTIONS
REPORT
QUICKREPORT
UTILITIES
ABOUTUS
EXIT
customer
Sakshi Enterprises,Islampur
near gosavi hospital,islampur
contact:9970042737
Customer Info
CUSTOMER ID
8
CUSTOMER NAME
Swati Suresh Patil
EMAIL
swati567@gmail.com
MOBILE NO
9765738362
ADDRESS
At Post Walwa
Record Saved
OK
NEW
SAVE
SEARCH
UPDATE
DELETE
EXIT

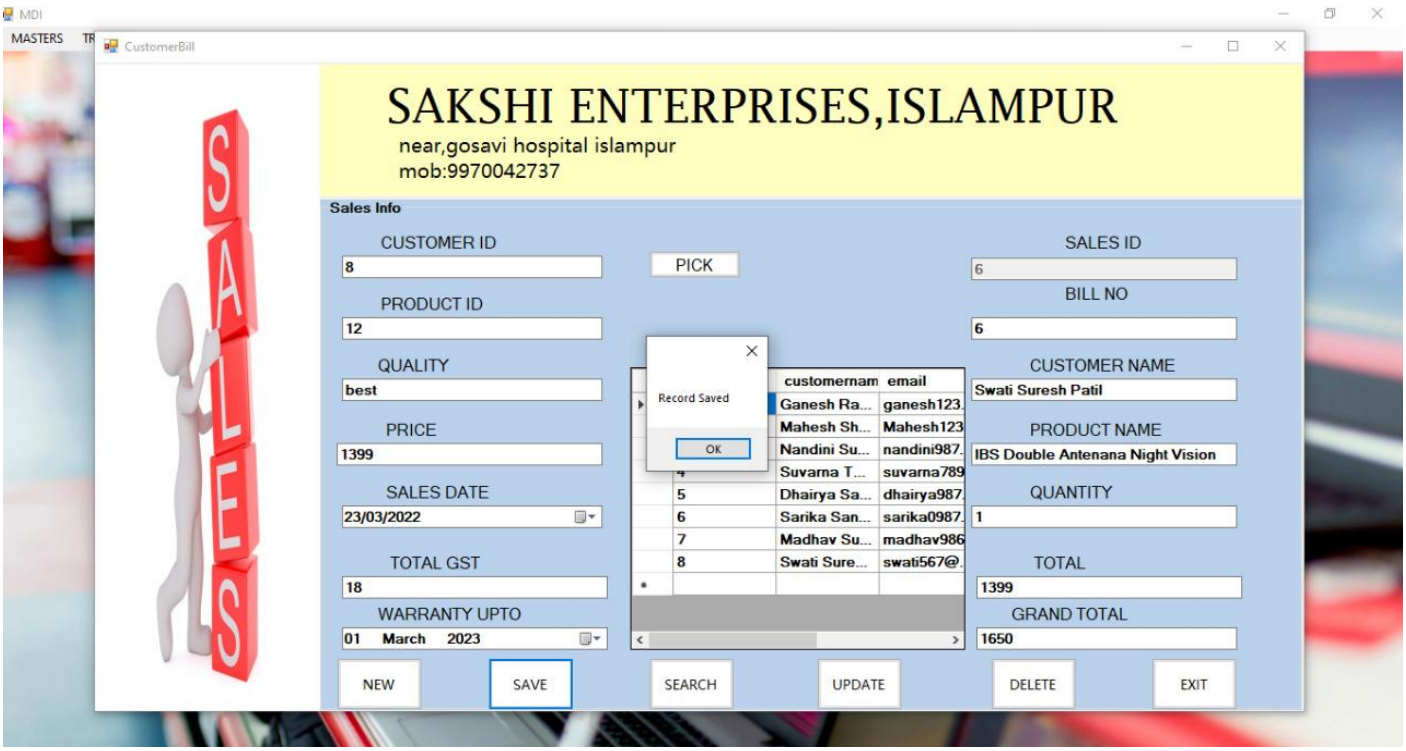
Product


MDI
MASTERS
TRANSACTIONS
REPORT
QUICKREPORT
UTILITIES
ABOUTUS
EXIT
product
SAKSHI ENTERPRISES ,ISLAMPUR
near,gosavi hospital islampur
mob:-9970042737
product info
PRODUCT ID
12
PRODUCT NAME
IBS Double Antenna Night Vision
PRODUCT TYPE
CCTV
COMPANY
IBS
PRICE
1159
Record Saved
OK
NEW
SAVE
SEARCH
UPDATE
DELETE
EXIT

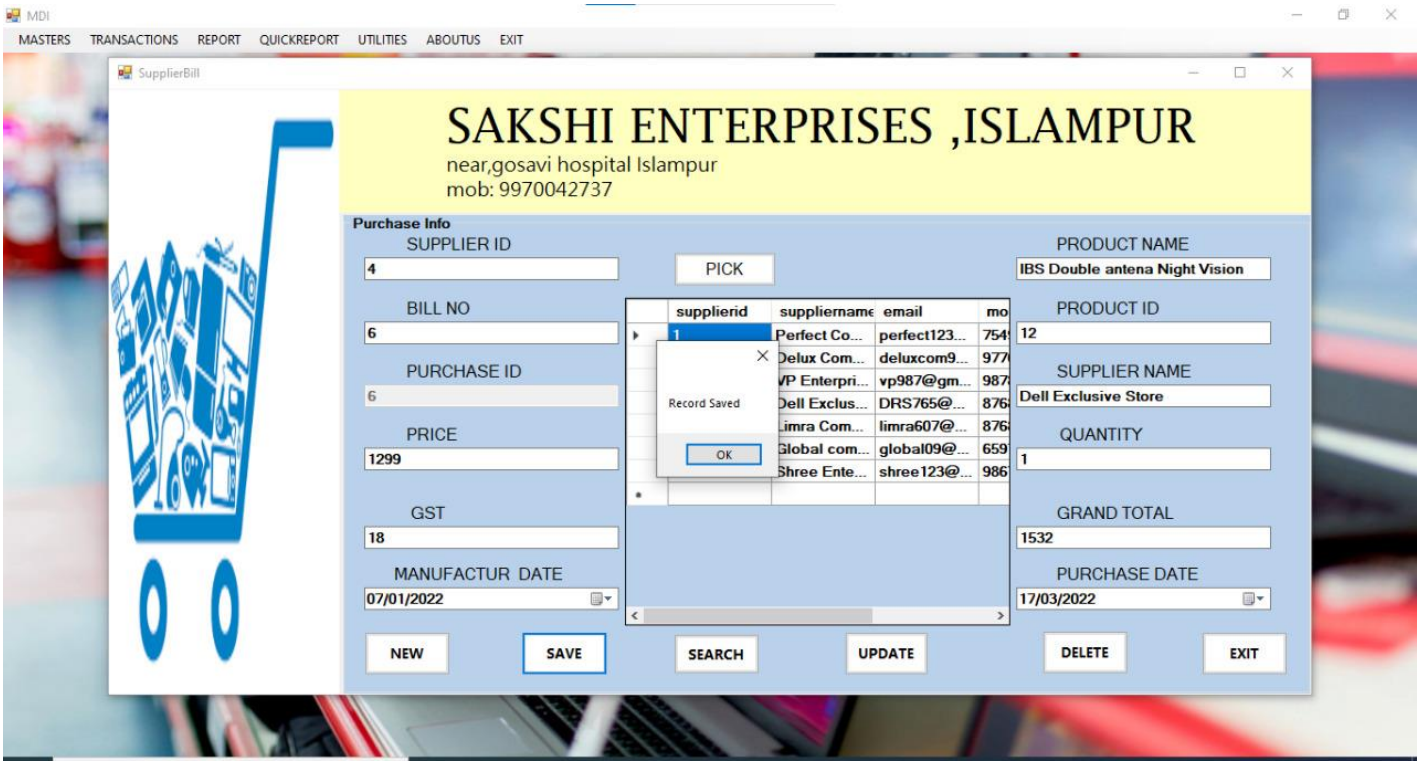
Supplier


MDI
MASTERS
TRANSACTIONS
REPORT
QUICKREPORT
UTILITIES
ABOUTUS
EXIT
supplier
SAKSHI ENTERPRISES ,ISLAMPUR
near,gosavi hospital islampur
mob:-9970042737
Supplier Info
SUPPLIER ID
7
SUPPLIER NAME
Shree Enterprises
EMAIL
shree123@gmail.com
MOBILE NO
9867543675
ADDRESS
At Post Sangli
Record Saved
OK
NEW
SAVE
SEARCH
UPDATE
DELETE
EXIT

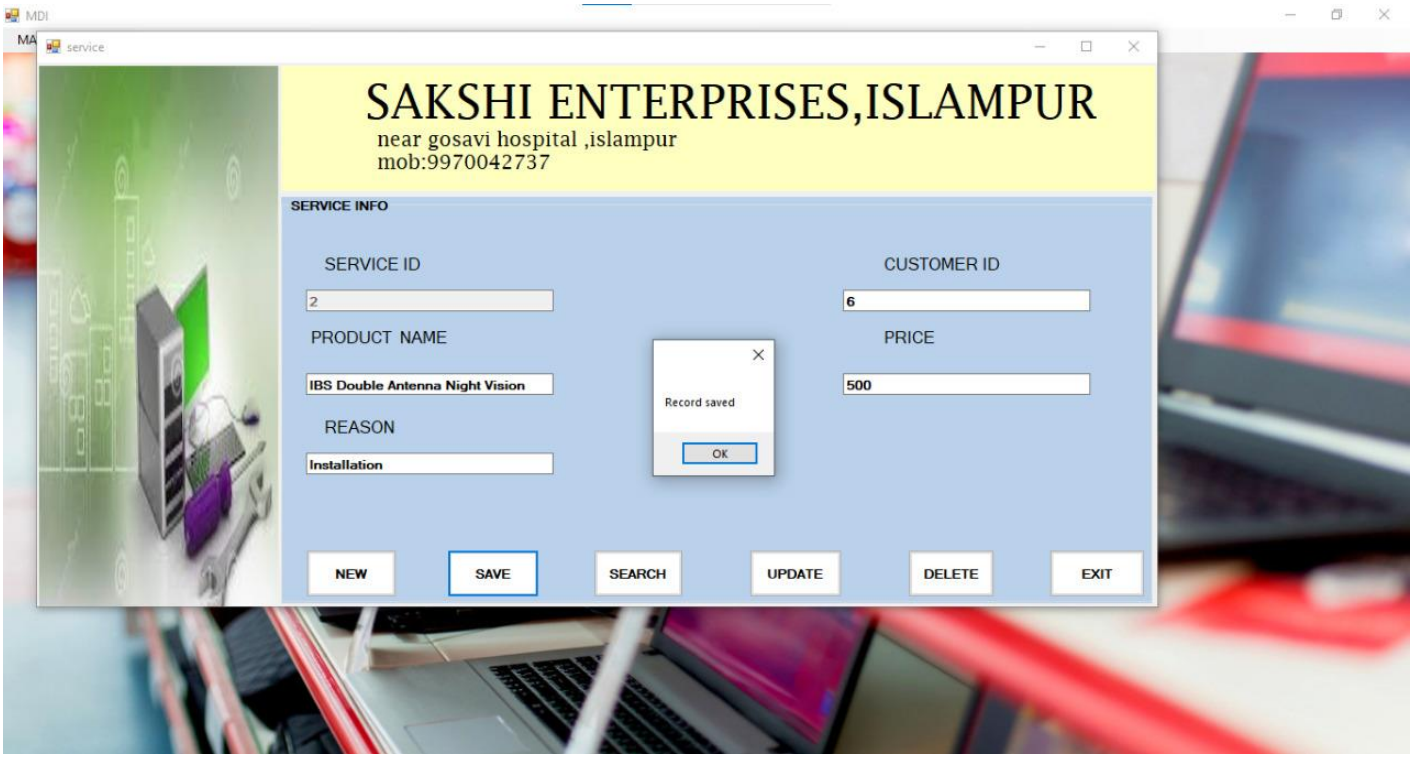
Customer Bill



Supplier Bill



Service


MDI
service
SAKSHI ENTERPRISES,ISLAMPUR
near gosavi hospital ,islampur
mob:9970042737
SERVICE INFO
SERVICE ID
2
CUSTOMER ID
6
PRODUCT NAME
IBS Double Antenna Night Vision
PRICE
500
REASON
Installation
Record saved
OK
NEW
SAVE
SEARCH
UPDATE
DELETE
EXIT

Sales Return

Purchase Return

About us


MDI
MASTERS
TRANSACTIONS
REPORT
QUICKREPORT
UTILITIES
ABOUTUS
EXIT
aboutus
Owner
Jayant Babanrao Patil
Shop Name
Sakshi Enterprises
Address
Near gosavi hospital Islampur
Contact No
9970042737

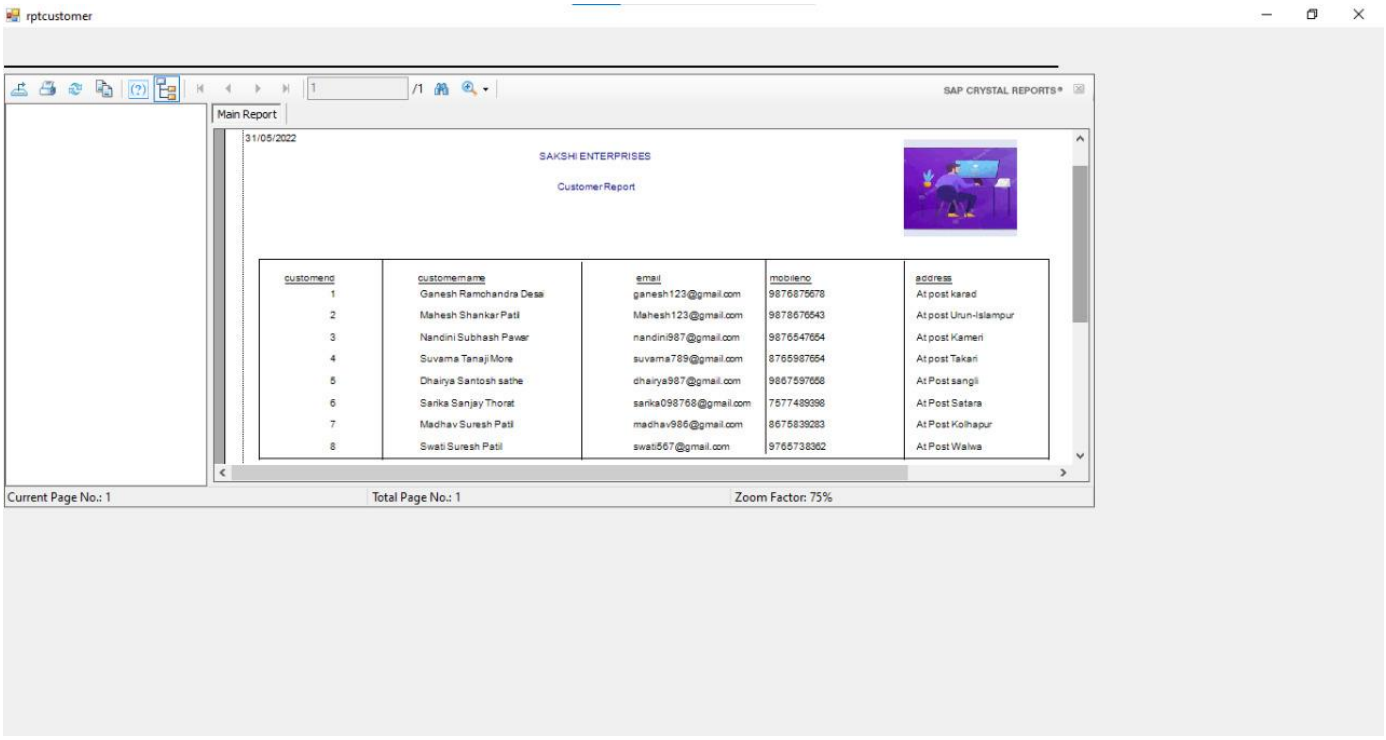
Chapter – 8

OUTPUT

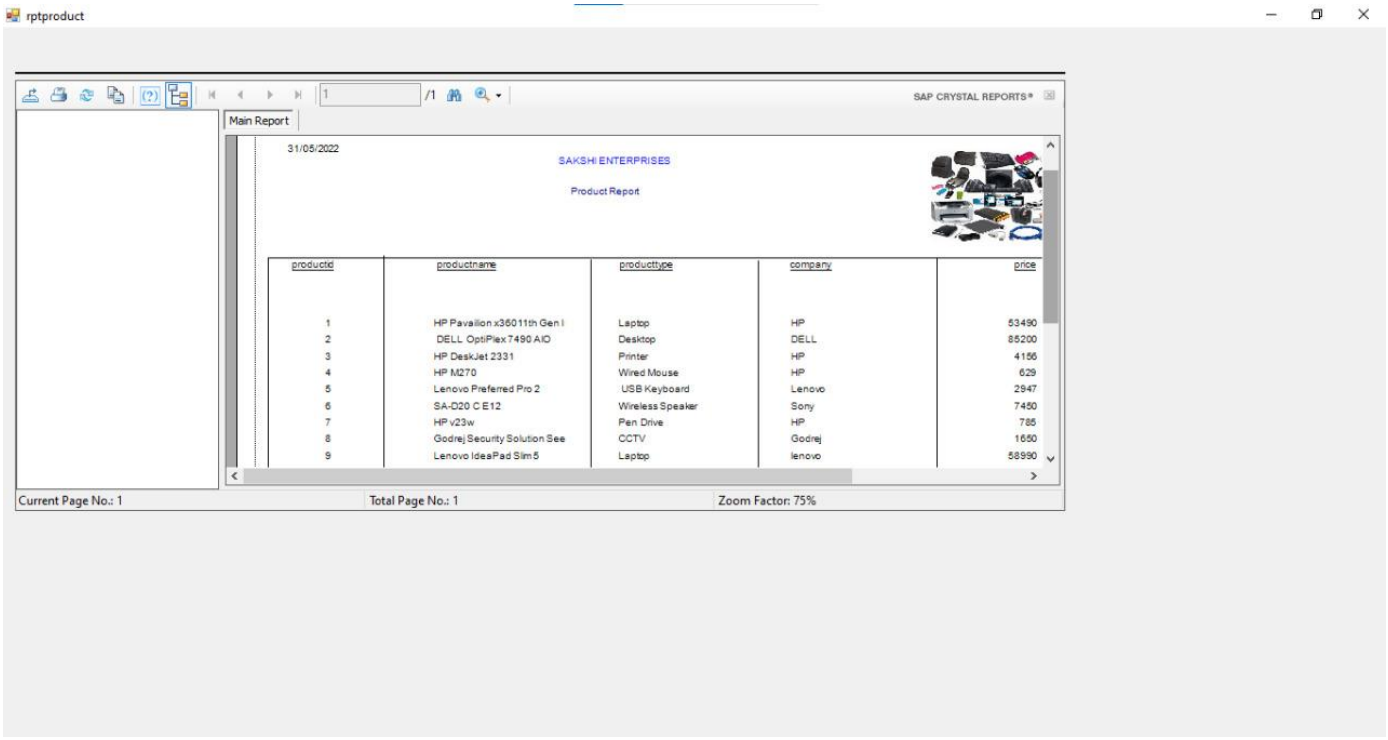
OUTPUT SCREEN

8.1 Reports

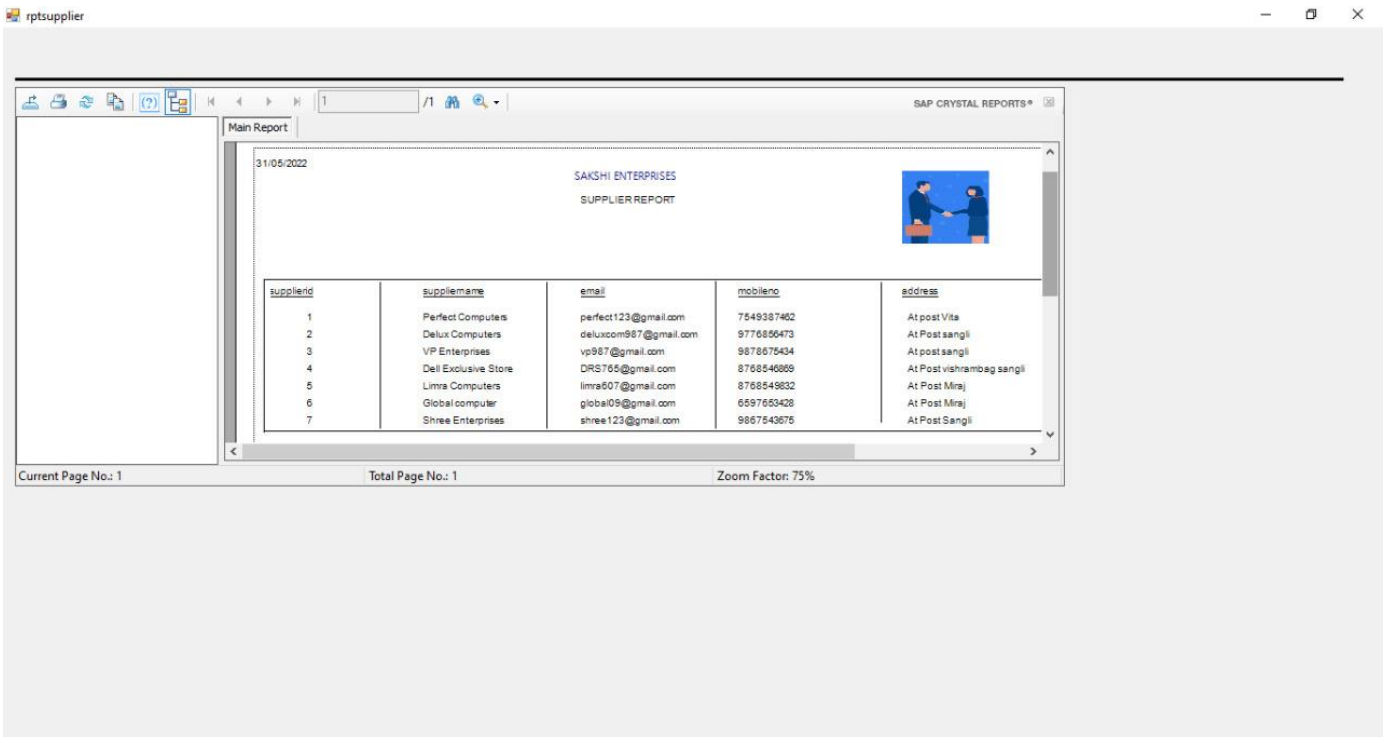
Customer


rptcustomer
SAP CRYSTAL REPORTS®
Main Report
31/05/2022
SAKSHI ENTERPRISES
Customer Report
customerid customername email mobileno address
1 Ganesh Ramchandra Desai ganesh123@gmail.com 9876875678 At post karad
2 Mahesh Shankar Patil Mahesh123@gmail.com 9878676543 At post Urun-Islampur
3 Nandini Subhash Pawar nandini987@gmail.com 9876547654 At post Kameri
4 Suvarna Tanaji More suvarna789@gmail.com 8765987654 At post Takari
5 Dhairya Santosh sathe dhairya987@gmail.com 9867597658 At Post sangli
6 Sarika Sanjay Thorat sarika098768@gmail.com 7577489398 At Post Satara
7 Madhav Suresh Patil madhav986@gmail.com 8675839283 At Post Kolhapur
8 Swati Suresh Patil swati567@gmail.com 9765738362 At Post Walwa
Current Page No.: 1
Total Page No.: 1
Zoom Factor: 75%

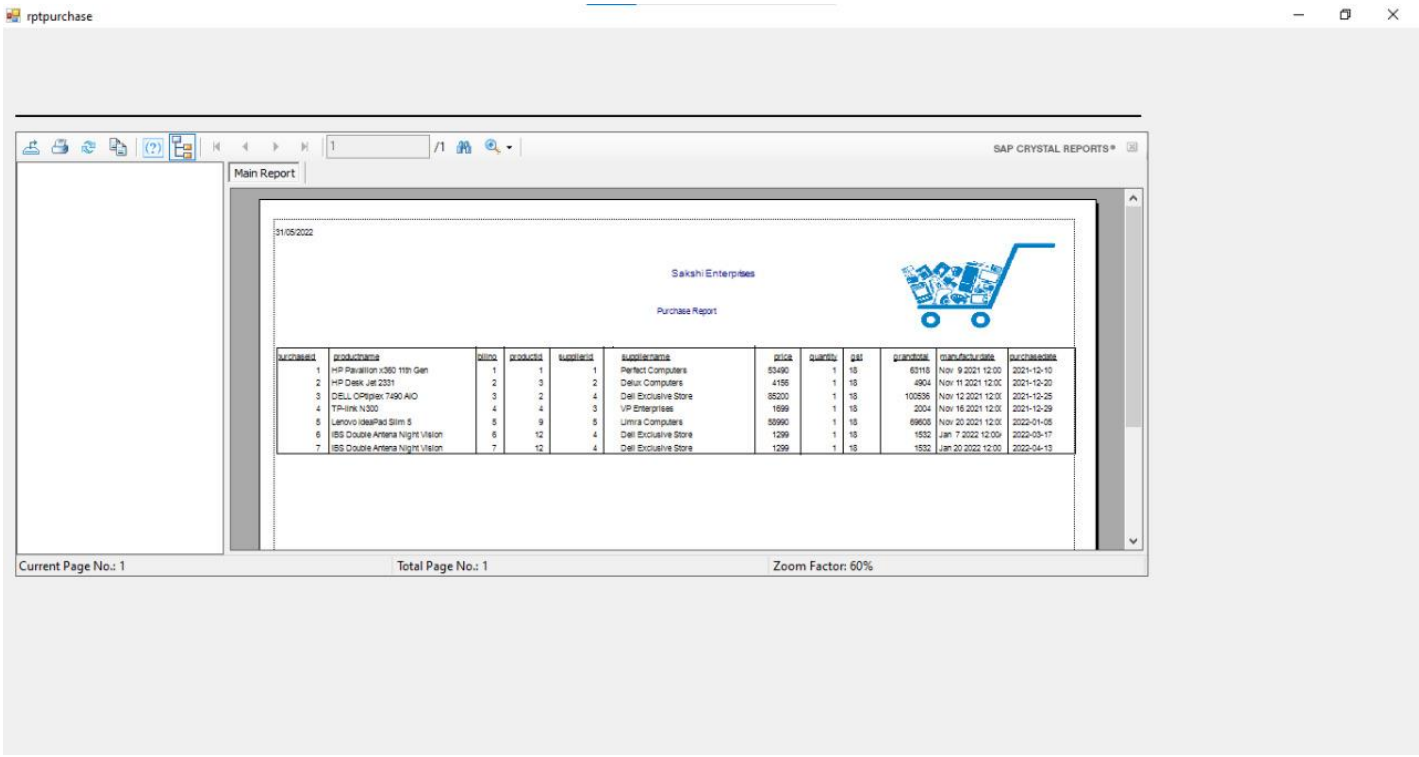
Product


rptproduct
SAP CRYSTAL REPORTS®
Main Report
31/05/2022
SAKSHI ENTERPRISES
Product Report
productid | productname | producttype | company | price
1 | HP Pavalion x36011th Gen I | Laptop | HP | 53490
2 | DELL OptiPlex 7490 AIO | Desktop | DELL | 85200
3 | HP DeskJet 2331 | Printer | HP | 4156
4 | HP M270 | Wired Mouse | HP | 629
5 | Lenovo Preferred Pro 2 | USB Keyboard | Lenovo | 2947
6 | SA-D20 C E12 | Wireless Speaker | Sony | 7450
7 | HP v23w | Pen Drive | HP | 785
8 | Godrej Security Solution See | CCTV | Godrej | 1650
9 | Lenovo IdeaPad Slim 5 | Laptop | lenovo | 58990
Current Page No.: 1
Total Page No.: 1
Zoom Factor: 75%

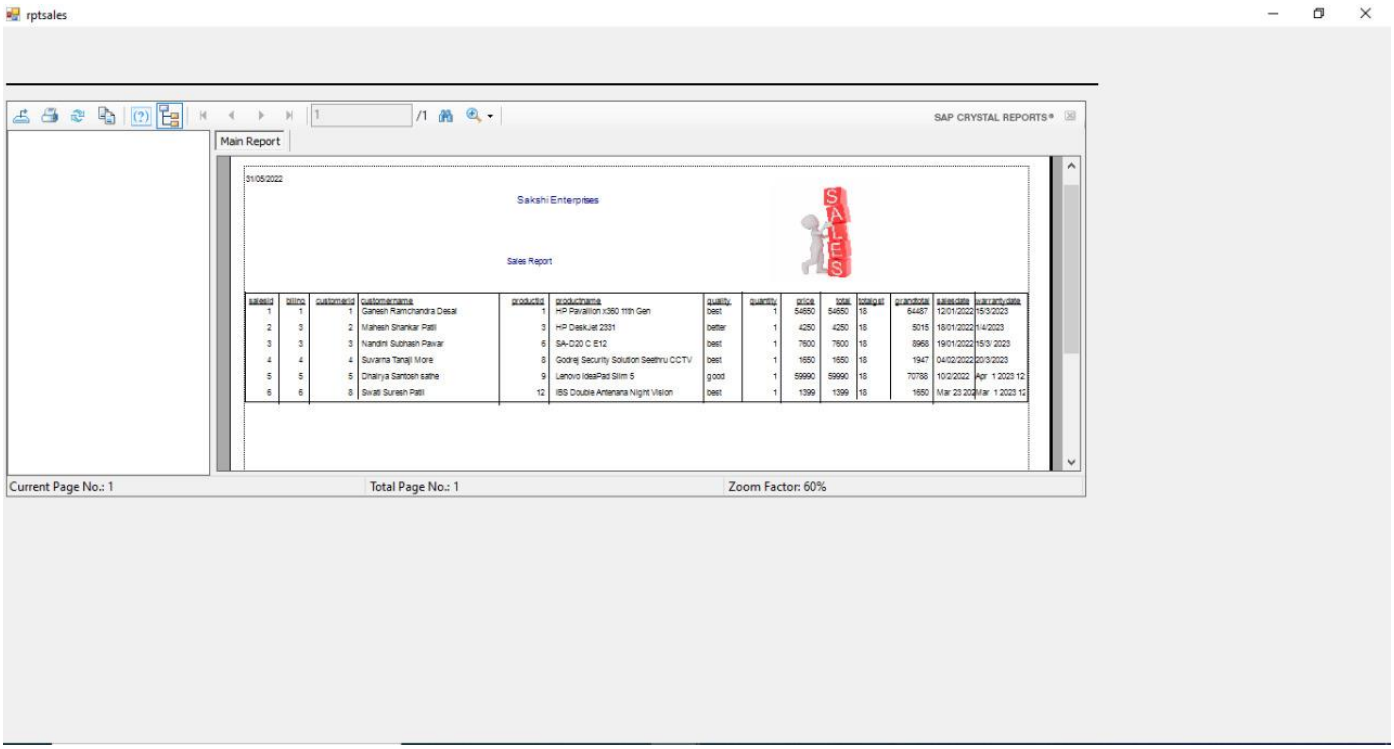
Supplier


rptsupplier
SAP CRYSTAL REPORTS®
Main Report
31/05/2022
SAKSHI ENTERPRISES
SUPPLIER REPORT
supplierid | suppliername | email | mobileno | address
1 | Perfect Computers | perfect123@gmail.com | 7549387462 | At post Vita
2 | Delux Computers | deluxcom987@gmail.com | 9776856473 | At Post sangli
3 | VP Enterprises | vp987@gmail.com | 9878675434 | At post sangli
4 | Dell Exclusive Store | DRS765@gmail.com | 8768546869 | At Post vishrambag sangli
5 | Limra Computers | limra607@gmail.com | 8768549832 | At Post Miraj
6 | Global computer | global09@gmail.com | 6597653428 | At Post Miraj
7 | Shree Enterprises | shree123@gmail.com | 9867543675 | At Post Sangli
Current Page No.: 1
Total Page No.: 1
Zoom Factor: 75%

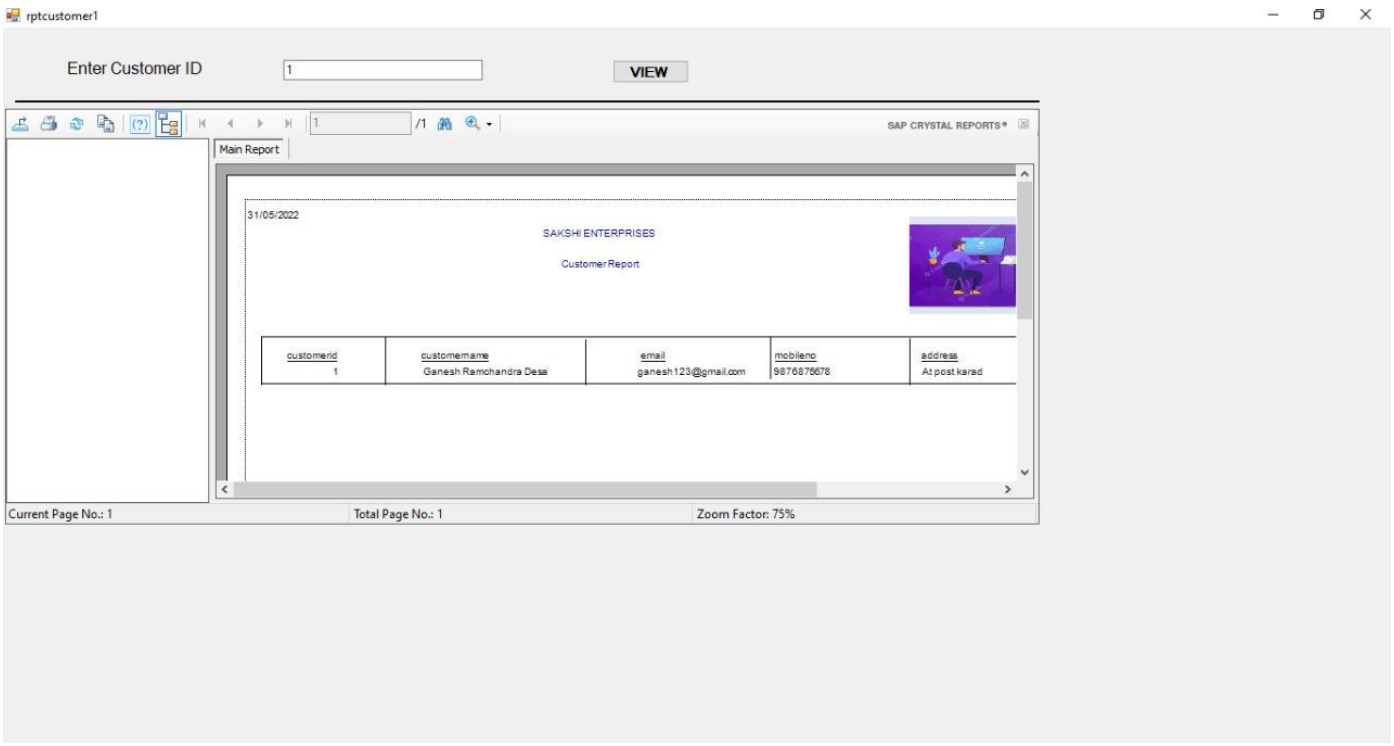
Purchase


rptpurchase
SAP CRYSTAL REPORTS®
Main Report
31/05/2022
Sakshi Enterprises
Purchase Report
Current Page No.: 1
Total Page No.: 1
Zoom Factor: 60%

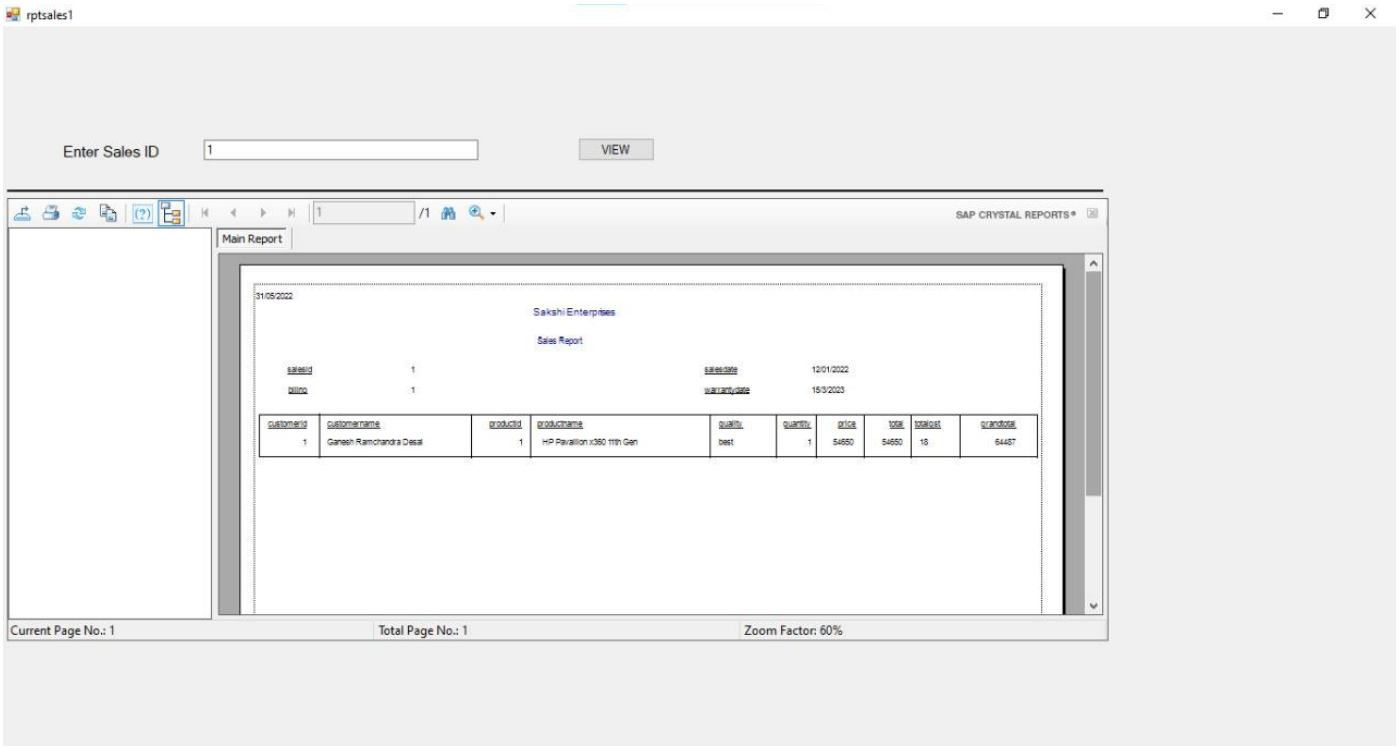
Sales


rptsales
SAP CRYSTAL REPORTS®
Main Report
31/05/2022
Sakshi Enterprises
Sales Report
Current Page No.: 1
Total Page No.: 1
Zoom Factor: 60%

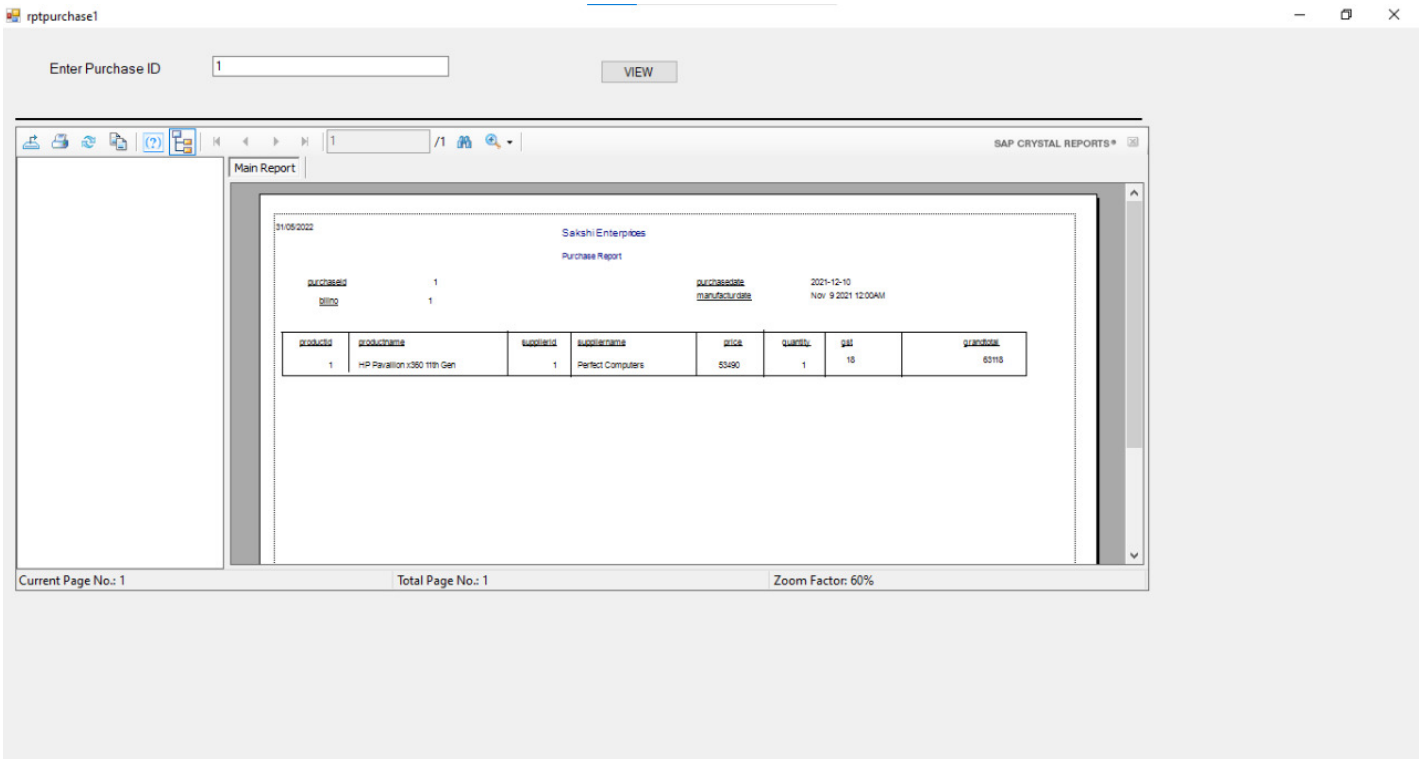
IDwise Customer



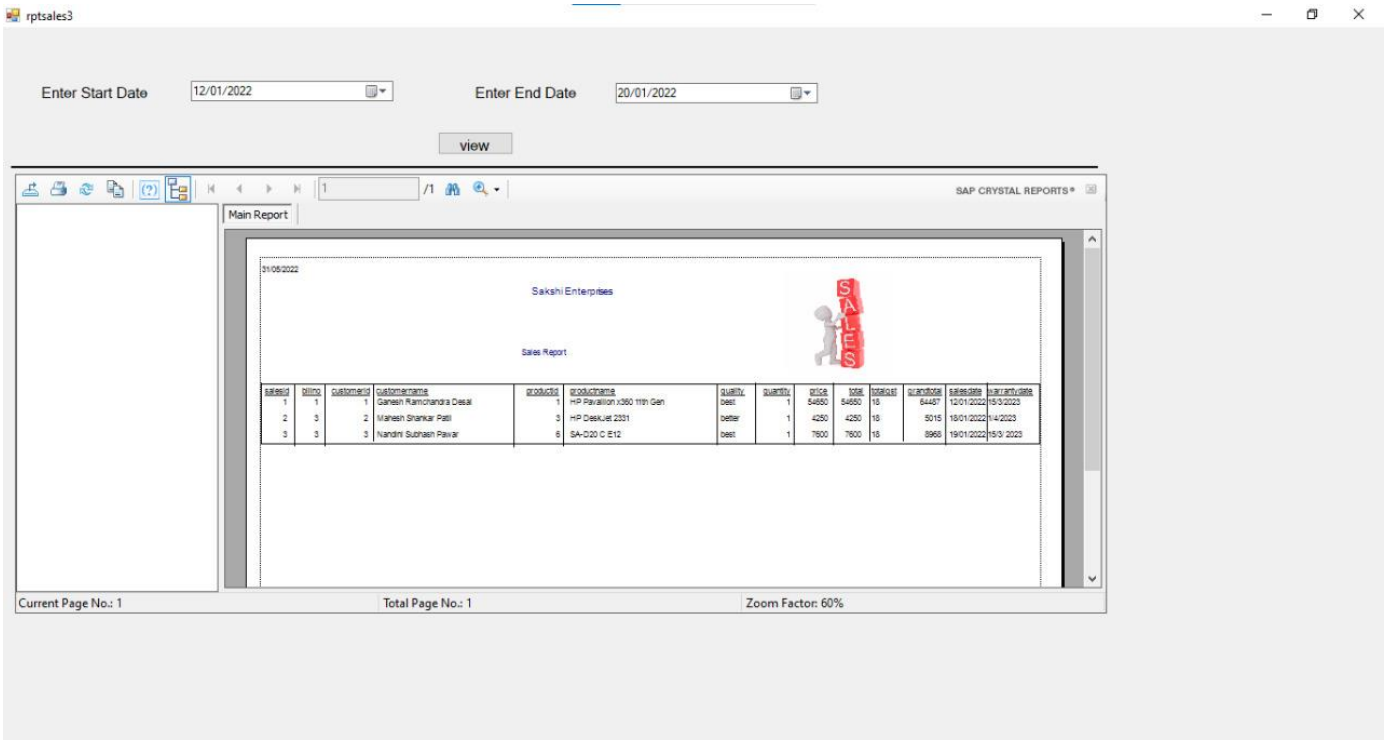
IDwiseCustomer Bill



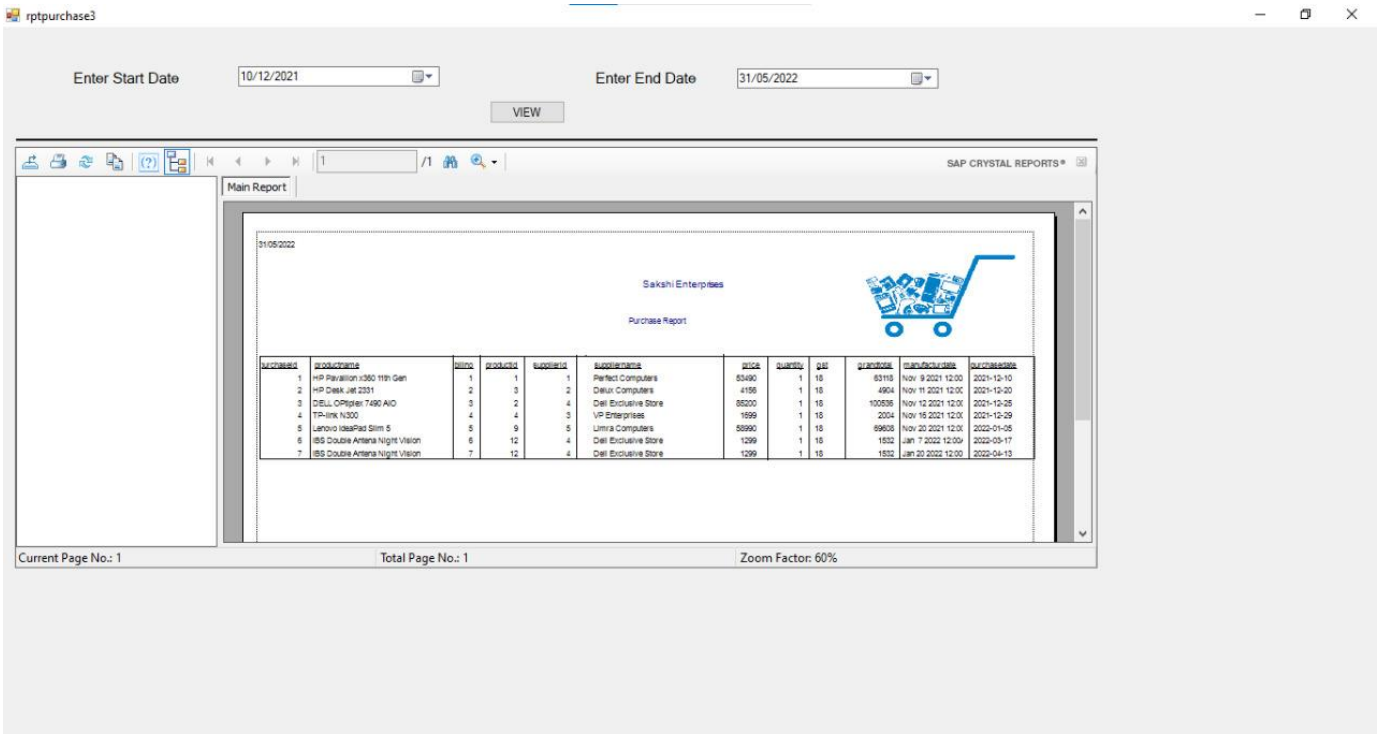
IDwise Supplier Bill



Date Between Customer Bill



Datewise Supplier Bill



Chapter – 9

CONCLUSION AND SUGGESTION

Conclusion-

The system is developed by using VB.NET. It is menu driven so very easy to handle any user can operate the system by using menus and sub-menus.

Most of work in calculation is reduced. This is very useful for the total management of Sales and purchase system for laptop accessories, desktop sales & repairs.

1. System is mainly established for storage of data and processing of that data.

2. The less space utilization and time saving is main target.

3. This system is very quick in action and require less time for processing.

Suggestions-

1. In this system you have to give correct password for the system.

2. The user should enter data correctly.

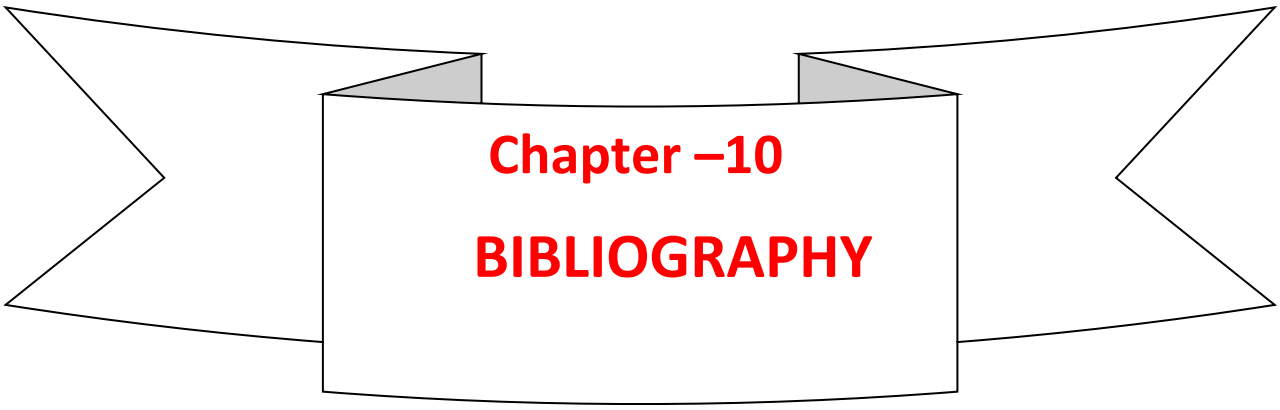
3. For future development it is necessary to change some forms and reports style.

Limitations-

1. This system is a single user.

2. Once record is deleted it can- not be recovered.

3. This system is only supported by Microsoft Visual studio software.



Chapter –10

BIBLIOGRAPHY

BOOK-

1. Mastering In Visual Basic .NET: By Evangelos Petroulos.

2. System Analysis and Design: By Elis M. Award.

3.Visual Basic Programming: By Stwen holzener.

4. Analysis and Design of Information System: By James A. Senn.

5. Database Management System: Raghu ramakrishnan and Johanehs Gehrke.

WEBSITES -

www.google.com

www.pureitdesigns.com