

**Project Report on**

**STOCK MANAGEMENT SYSTEM** By:

**Surendra Kumar**

**(Roll No: 12604039)**

**submitted for partial fulfillment of the requirements**

**for the degree of**

**BCA**

**MSITM DEGREE COLLEGE**



**Affiliated to: Dr. Ram Manohar Lohia Awadh University, Faizabad(U.P.)**

***Preface***

“Almost a century ago a spate of invention ushered into the world first through the industrial revolution within a short span of time many countries became industrialized.”

Now the world is in another industrial revolution .The major causes of first industrial Revolution were the invention of many mechanical devices and machines. The cause of second industrial revolution is electronic devices. Man invented many electronic devices and the computer is one of these but the computer has made a greater impact on society than any other single device and became major cause of second industrial revolution .They have made sufficient significant contribution to the society during the last three decades .Now no country can assume these developments without the use of the computers .They are versatile and are almost indispensable to engineers, scientists, business, executives, managers, administrators, accountants, teachers, and students.

A computer has some powerful features and characteristics as following---

* Perform complex and respective calculations at a very high speed and accuracy.
* Store large amount of data, information and subsequent manipulation.
* Make decisions under predefined instructions and conditions.
* Provide information to the user.
* Automatically correct or modify certain parameters of any system under control.
* Draw and print graphs.
* Print information and record effectively.
* Converse the user through terminal.

Though the computers are used in many areas but now a days 80% use of computer and automation of industries and business system is prevalent.So keeping the above facts in mind we prepare our project report.

***Acknowledgement***

I feel very privileged that through these words I be able to flaunt my thought and show my gratitude and a sense deep contentment for the people who have been part of this project in numerous ways. From those people I have got relentless support right from the stage the project idea was conceived to its completion .In Particular, I would like to shower special praise on my guide Mr. Ajay Tondon Sir and I am very thankful for his endless help during the development of entire project.

There are times in such projects when the clock beats you time and you run out of energy and you just want to finish it once and forever. My near & dear one’s induces self-belief in me to endure adverse times and make me mentally strong to cope with any situation instead of succumbing to pressure.

With great pleasure, I would like to express my sincere gratitude to **Mr. Ajay Tandon, Centre Manager, MSITM Gonda** for guidance during the period.

I want to give my heartiest thanks to my parents whose principle and morale helped me to rise to this position. I would also like to thank all the friends who provided useful suggestions time to time. I also like to thank almighty god for what he has given till today.

**(Surendra kumar) (Roll Number: 12604039)**

# *TABLE OF CONTENTS*

**Index:**

# Objective and Scope

* **Definition of the Problem**
* **Introduction to SQL Database**
* **Introduction to C#**
* **Software Engineering paradigm**
* **Software Engineering methodology and process model**
* **Estimation**
* **Pert Chart Explanation**
* **PERT CHART**
* **GANTT CHART**
* **Cost Benefit Analysis**
* **Entity Relationship Model**
* **Data Flow Diagram**
* **Hardware and software requirement**
* **PROGRAM FLOW CHART**
* **EVOLUTION**
* **Testing technique & Strategies**
* **Implementation**
* **MAINTENANCE**
* **Code Efficiency**
* **DATA DICTIONARY**
* **DATABASE STRUCTURE**
* **OUTPUT SCREENS AND CODING**
* **CONCLUSION**
* **BIBLOGRAPHY AND REFERENCE**

# *Objective and Scope*

**Objective:**

The chief objective of “Inventory Management System”-that it provides an easy wayto automate all the functionalities of Inventory just like is to store the entire database at one place. From this the record retrieval, storing and manipulation are fast and auditing can be handled quite easily.

Inventory Management System provides an attractive environment where you can manipulate data & information about Product and Customer easily. So we can say that the core purpose ofdesigning “Inventory management System” is to manage the task related to the Warehouse customer/employees and to reduce time to searching of appropriate product and views.

**Scope:**

The scope of an inventory system defines which needs it addresses, including valuing the inventory, measuring the change in inventory and planning for future inventory levels. The value of the inventory at the end of each period provides a basis for financial reporting on the balance sheet. Measuring the change in inventory allows the company to determine the cost of inventory sold during the period. Together, inventory values and level changes allow the company to plan for future inventory needs.

Regardless of which type of inventory system a company uses, the scope of the inventory may change based on the strategic targets of the business. Scope may refer to different aspects of how inventory counts are conducted or to the way inventory information is used. A manufacturer of durable goods may use inventory information to control the costs of raw materials and to maximize margins, while a food manufacturer may prioritize fast turnover of inventory to ensure freshness. The importance of inventory counts in those examples may require staff dedicated to inventory management, while a small cleaning business may not need more than an occasional scan of its cleaning supplies inventory to function efficiently.

***Definition of the Problem***

Generally in now days, all the banks take a step to computerized their working environment because traditional working that is manual is tough and time consuming. Generally the record searching or verification of a customers is a cumbersome process, for the entire working separate registers, ledgers etc are to be maintained.

Inventories that are counted weekly or monthly, compared with counts from the previous period, and adjusted to include purchases within the period can only generate data at or after the point of counting. This results in a lack of detail in the information about how inventory moves. The nature of a small business may not require such detail, or minimum order quantities of raw materials at aggressive discounts may cover several periods, making such detail unnecessary, so these limitations may be minimal for certain businesses.

***Definition:***

The software system “Stock Management System” provides facilities according to the requirement of Inventory/stocks.

***Responsibilities overview:***

My prime responsibility is that the performance of software never ecodes in any environment; each and every query will respond in an optimal time and satiate the prospective seeker. All the responses are carried out in a minimum time in each particular situation.

***Introduction to SQL Database***

SQL (pronounced "ess-que-el") stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database. This tutorial will provide you with the instruction on the basics of each of these commands as well as allow you to put them to practice using the SQL Interpreter.

Structure Query Language(SQL) is a programming language used for storing and managing data in RDBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model. Today almost all RDBMS(MySql, Oracle, Infomix, Sybase, MS Access) uses **SQL** as the standard database language. SQL is used to perform all type of data operations in RDBMS.

SQL was one of the first commercial languages for [Edgar F. Codd](https://en.wikipedia.org/wiki/Edgar_F._Codd)'s [relational model](https://en.wikipedia.org/wiki/Relational_model), as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks."[[10]](https://en.wikipedia.org/wiki/SQL#cite_note-codd-relational-model-10) Despite not entirely adhering to [the relational model as described by Codd](https://en.wikipedia.org/wiki/Codd%27s_12_rules), it became the most widely used database language.[[11]](https://en.wikipedia.org/wiki/SQL#cite_note-SQL-Fundamentals-11)[[12]](https://en.wikipedia.org/wiki/SQL#cite_note-IBM-sql-12)

Originally based upon [relational algebra](https://en.wikipedia.org/wiki/Relational_algebra) and [tuple relational calculus](https://en.wikipedia.org/wiki/Tuple_relational_calculus), SQL consists of a [data definition language](https://en.wikipedia.org/wiki/Data_definition_language), [data manipulation language](https://en.wikipedia.org/wiki/Data_manipulation_language), and [Data Control Language](https://en.wikipedia.org/wiki/Data_Control_Language). The scope of SQL includes data insert, query, update and delete, [schema](https://en.wikipedia.org/wiki/Database_schema) creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a [declarative language](https://en.wikipedia.org/wiki/Declarative_programming) ([4GL](https://en.wikipedia.org/wiki/4GL)), it also includes [procedural](https://en.wikipedia.org/wiki/Procedural_programming) elements.

***Introduction to C#***

C# (pronounced "C sharp") is a programming language that is designed for building a variety of applications that run on the .NET Framework. C# is simple, powerful, type-safe, and object-oriented. The many innovations in C# enable rapid application development while retaining the expressiveness and elegance of C-style languages.

If you don't already have C#, you can acquire a version of Visual Studio that includes C# for free from the [Visual Studio](https://www.visualstudio.com/products/free-developer-offers-vs) site.

C# is a simple, modern, general-purpose, object-oriented programming language developed by Microsoft within its .NET initiative led by Anders Hejlsberg. This tutorial will teach you basic C# programming and will also take you through various advanced concepts related to C# programming language.

C# programming is very much based on C and C++ programming languages, so if you have a basic understanding of C or C++ programming, then it will be fun to learn C#.

C# is designed for Common Language Infrastructure (CLI), which consists of the executable code and runtime environment that allows use of various high-level languages on different computer platforms and architectures.

Although C# constructs closely follow traditional high-level languages, C and C++ and being an object-oriented programming language. It has strong resemblance with Java, it has numerous strong programming features that make it endearing to a number of programmers worldwide.

The .Net framework applications are multi-platform applications. The framework has been designed in such a way that it can be used from any of the following languages: C#, C++, Visual Basic, Jscript, COBOL, etc. All these languages can access the framework as well as communicate with each other.

***Software Engineering paradigm***

Software engineering occurs as a consequence of a process called “System Engineering” instead of concentrating solely on software, system engineering focuses on a variety of elements, analysing, designing and organizing those elements into a system that can be a product, service or a technology for the transformation of information or control. The system engineering process takes on different forms depending on the application domain in which it is applied.

A software system exists for one reason: to provide value to its users. All decision should be made with this in mind.

**Keep It Simple**,

Software design is not a haphazard process. There are many factors to consider in any design effort. All design should be as simple as possible but no simpler. These facilities having a more easily understood and easily maintained system. This is not to say that features, even internal features should be discarded in the name of simplicity.Indeed, the more elegant designs are usually the simple ones. Simple also does not mean “quick and dirty” . In fact it often takes a lot of thought and work over multiple iterations to simplify. The pay off is software that is more maintainable and less error prone.

**Maintain the vision:**

A clear vision is essential to the success of a software project. Without one, a project almost unfailingly ends up being “of two minds” about itself. Without conceptual integrity, a system threatens to become a patchwork of incompatible designs, held together by the wrong kind of screws. Compromising the architectural vision of a software system weakens and will eventually break down the well-designed system.

**What You Produce, Others will consume:**

The audience for any project of software development is potentially large. So always specify, design, and implement knowing someone else will have to understand what you are doing. Design keeping the implementers in mind. Code with concern for those who must maintain and extend the system. Someone may have to debug the code you write, and that makes them a user of code. Making their job easier adds value to the system.

**Be Open to the Future:**

A system with a long lifeline has more value. In today’s computing environment, were specifications change on a moment’s notice and hardware platform are obsolete after just a few months, software lifetimes are typically measured in months instead of years. However true “industrial-strength” software systems must endure for longer. To do this successfully, these systems must be ready to adapt to these and other changes. Systems that do this successfully are those that have been designed this way from the start. Never design yourself into a corner. Always ask “what-if” and prepare for all possible answers by creating systems that solve the general problem, not just the specific one.

**Plan Ahead for Reuse:**

Reuse saves time and effort .Achieving a high level of reuse is arguably the hardest goal to accomplish in developing a software system. The reuse of code designs has been proclaimed as a major benefit of using object-oriented technologies. However, the return on this investment is not automatic. To leverage the reuse possibilities that object-oriented programming provides require fore thought and planning. There are many techniques to realize reuse at every level of the system development process. Those at the detailed designs and code level are well known and documented. New literature is addressing the reuse of design in the form of software patterns. However this is just part of the battle. Communicating opportunities for reuse to others in the organization is paramount. How can you reuse something that you don’t know exists? Planning ahead for reuse reduces the cost and increases the value of both the reusable components and the systems into which they are incorporated.

**Think:**

Placing clear, complete thought before action almost always produces better results. When you think about something, you are more likely to do it right. You also gain knowledge about to do it right again. If you do think about something and still do it wrong, it becomes valuable experience. Aside effect of thinking is learning to recognize when you don’t know something, at which point you can research the answer. When clear thought has gone into a system, value comes out.

Software has become the key element in the evolution of computer based system and products and one of the most important technologies on the world stage. Over the past 50 years. Software has evolved from a specialized problem solving and information analysis tool to an industry in itself. Yet we still have trouble developing high quality software on time and within budget. Software-programs, data and documents-addresses a wide array of technology and application areas, yet all software evolves according to a set of laws that remained the same for over 0 years. The intent of software engineering is to provide a framework for building higher quality software.

Software engineering is a discipline that integrates process, methods, and tools for the development of computer software. All the different process models for software engineering define a set of umbrella activities that span the entire process. Personal and team models for the software process have been proposed, both emphasize measurement, planning, and self direction as key ingredients for successful software process.

An agile philosophy for software engineering stresses four key issues:

* The importance of self-organizing teams that have control over the work they perform.
* Communication and collaboration between team members and between practitioners and their customers.
* Recognition that change represents an opportunity
* An emphasis on rapid delivery of software that satisfies the customer.

***Software Engineering methodology***

***and process model***

A production process of software is a sequence of steps. Each step performs a well-defined activity leading toward the satisfaction of the project goals with the output of one step forming the input of the next one. How to perform the activity of the particular step or phrase is an issue addressed by methodologies for that activity.

The process model I have used is Waterfall Model .In these phases are organized in a linear order.

Project begins with feasibility analysis and project planning begins .The design starts after analysis is complete and coding begins after the design is complete .Then coding is integrated and testing is done .After this regular operation and maintenance of the system takes place.

Feasibility Study

System Analysis

System Design Coding

Development of software

Requirement Analysis

System Testing

**Waterfall Model**

Implementation

***Estimation***

Software is the most expensive elements of virtually all computer-based systems .For example, custom systems; a large cost estimation error can make the difference between profit and loss. Cost overrun can be disastrous for the developer. Although software engineering effort is a dominant elements of project cost. It is important to remember that others costs (e.g. development environment and tools, travel, training, office space, hardware) must also be considered.

Software cost and effort estimation will never be an exact science, but a combination of good historical data and systematic technique can improve estimation accuracy. Too many variable –human, technical, environmental, political-can effect the ultimate cost of software and effort applied to develop it.However, software project estimation can be transformed from a black art to series of systematic steps that provide estimates with acceptable risk. To achieve reliable cost and effort estimates, a number of options arise.

* Delay estimation until late in the project (obviously we can achieve 100% accurate estimates after the project is completed).
* Base estimates on similar projects that have already been completed.
* Use relatively simple decomposition techniques to generate project cost and effort estimates.
* Use one or more empirical method for software cost and effort estimation.

Unfortunately the first option however attractive is not practical. Cost estimates must be provided “up front”. However we should recognize that longer we wait, the more we know, the less likely we are to make serious errors in our estimates.

The second option can work reasonably well, if the current project is quite similar to past efforts and other project influences are roughly equivalent. Unfortunately, past experience has not always been a good indicator of future results.

The remaining options are viable approaches to software project estimation Ideally, the technique noted for each option should be applied in tandem; each used a cross check for the other. Decomposition techniques take a “divide and conquer” approaches to software project estimation. By decomposing a project into major functions and related software engineering’s activities, cost and effort estimation can be performed in a stepwise fashion.

Empirical estimation models can be used to complement decomposition techniques and offer a potentially valuable estimation approach in their own right.

Each of the viable software cost estimation options is only as good as the historical data used to seed the estimate. If no historical data exists, costing rests on a very shaky foundation.

***Pert Chart Explanation***

The Program Evaluation and review Technique commonly abbreviated as PERT is a model for project management invented by United States Department of Defense’s US Navy special projects Office in 1958 as part of the Polaris mobile submarine-launched ballistic missile project. The project was a direct response to the sputnik crisis.

PERT is basically a project scheduling method for analysing the tasks involved in completing a given project, especially the time needed to complete each task, and identifying the minimum time needed to complete the total project. It is driven by the information aid on already developed in earlier project planning activities:

* Estimates of effort
* A decomposition of the appropriate process model and task set
* Decomposition of tasks

Interdependencies among tasks may be defined using a task network. tasks sometimes called the project work breakdown structure(WBS),are defined for the product as a whole or for individual functions.

PERT provide quantitative tools that allow the software planner to

* Determines that the critical path-the chain of tasks that determines the duration of the project
* Establish ‘most likely” time estimates for individual tasks by applying statistical models.
* Calculate “boundary times” that define a time “window’ for a particular task

The objective of project scheduling tools is to enable a project manager to define work tasks, establish their dependencies, assign human resources to tasks, and develop a variety of graphs,charts,and tables that aid in tracking and control of the software project .It was able to incorporate uncertainty in the sense that

It was possible to schedule a project not knowing precisely the details and durations of all the activities .It is more of an event-oriented technique rather than start and completion oriented.

**TERMINOLOGY USED IN PERT CHART**

The various terms and terminology used in PERT CHART and their meaning are as follows:

**ANALYSIS:**

In analysis phase detailed study of user’s requirements has been done and then gets the idea of the proposed system to build.

**DESIGN**:

There are various type designs:

1. External design

2. Internal design

3. Architectural design

In external designing phase externally observable characteristics of software was conceived, planned out and specified. These characteristics include user displays, report format; external data source etc.External design was started during analysis phase.

**INTERNAL DESIGN**:

In internal designing phase following things has been done. Specification of internal structure and details, elaboration of test plans and development of a blueprint.

**ARCHITECTURAL DESIGN**:

In architectural designing phase following things has been done .Identification of internal processing functions, decomposition high level functions into publications, definition of internal datastreams and datastores, establishment of relationships among functions, datastreams and datastores.

**CODING:**

In the coding phase proper are written which performs the required tasks.

**TESTING**:

1. **UNIT TESTING**: In unit testing individual modules are tested. It means internal logic has been checked and required changes have been made in the code.

2. **INTEGRATED TESTING**: In this phase modules have been integrated and it is checked that after integration these modules are working effectively or not .

3. **ACCEPTANCE TESTING**: In acceptance testing it is checked that whether the software is fulfilling the user’s requirement or not.

***PERT CHART***

**Activities Predecessor Activity Duration(Days)**

A. Preparation for designing - 5

B. Designing DFD & ERD A 3

C. Designing Structure Chart A 2

D. Designing Input forms Layouts B, C 15

E. Designing Report Layouts B, C 15

F. Testing and Verification of forms D, E 7

G. Designing Database files D, E 10

H. Designing Flow Chart G, H 15

I. Coding H 60

J. Testing I 5

K. Implementation J 5

A (5)

B (3)

**Total Development Time=5+3=2+10+15+7+15+15+60+5+5=142 Days**

***GANTT CHART***

A Gantt chart is a popular type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project .Some Gantt charts also show the dependency (i.e. Precedence network) relationships between activities .Basically it is an activity scheduling method as a rudimentary aid to plot individual tasks against time.

The initial format of the chart was developed by henry L.Gantt (1861-1919)in 1910.The Advent of personal computers in the 1980s resulted in the ability to produce and edit elaborate Gantt charts. These desktop applications were primarily intended for project managers and project schedulers.In the late 1990s and 2000s, Gantt charts became a common feature of web-based applications, including collaborative groupware. Although now considered a common charting technique,Gnatt charts were considered quite revolutionary at the time they were introduced .In recognition of Henry Gantt’s contributions, the Henry Laurence Gantt medal is awarded for distinguished achievement in management and services to the community.

Gantt charts have become a common technique for representing the phases and activities of a project work breakdown structure, so a wide audience can understand them. A Gantt chart is easily comprehend for small projects that fit on a single sheet or screen, but they can become quite unwidely for projects that fit on a single sheet or screen, but they can become quite unwidely for projects with more than about 30 activities. Larger Gantt charts may not be suitable for most computer displays. A related criticism is that Gantt charts communicate relatively little information per unit area of display .That is, projects are often considerably more complex than can be communicated effectively with a Gantt chart .They only represent part of the triple constraints of projects because they focus primarily on schedule management.

Although project management software can show schedule dependencies as lines between activities, displaying a large number odd dependencies may result in a cluttered or unreadable chart.

***Cost Benefit Analysis***

With the help of this project a lot of time and money can be saved. Because when the task of producing reports will be accomplished through this project then it will require less amount of money.

While making this project cost benefit analysis has been done and following conclusions have been drawn.

Through this project via one time investment a specific task can be done several times .If we will perform same task manually then it will require a lot of cost and time and obviously user will have to do same sort of task again and again .

Cost of Hardware and software required for this project have to be paid only once but if we use manpower or some other option for this purpose then it will be very expensive and this is obviously not required.

Cost Benefit analysis is a tool for evaluating the effectiveness of the project. It may be used by the management .The objective of cost Benefit analysis is to find out whether it is economically worthwhile to invest in the proposed project .If the return on the investment is high then the project is considered economically worthwhile.

The three types of cost benefits are-

1. Tangible or Intangible cost and benefits.

2. Fixed or Variable cost and benefits.

3. Direct or Indirect cost and benefits

***Entity Relationship Model***

In computer science, an entity-relationship model (ERM) is a model providing a high level description of a conceptual data model .Data modelling provides a graphical notation for representing such data models in the form of entity-relationship diagrams (ERD).The first stage of information system design uses these models to describe information needs or the type of information that is to be stored in a database during the requirement analysis. The data modelling technique can be used to describe any ontology (i.e. an overview and classifications of used terms and their relationships) for a certain universe of discourse (i.e. area of interest).In the case of the design of an information system that is based on the database, the conceptual data model is at a later stage(usually called logical design), mapped to a logical model, such as the relational model, this in turn is mapped to a physical model ical during “physical design.”.

There are a number of conventions for entry-relationship diagrams (ERDs).The classical notation is described in the remainder of this article, and mainly relates to conceptual modelling. There are a range of notations more typically employed in logical and physical database design, including information engineering, IDEF1x (ICAM Definition Language) and dimensional modelling.

An entity represents a discrete object. Entities can be thought of (roughly) as nouns. Example: a computer, an employee, a song, a mathematical theorem. A relationship captures how two or more entities are related to one another .Relationships can be thought of (again, roughly) as verbs.Examples: an owns relation between a company and a computer, a supervises relation between an employee and a departmental performs relation between an artist and a song, a proved relation between a mathematician and a theorem. Entities are drawn as rectangles, relationships as diamonds.

Entities and relationships can both have attributes.Examples: an employee entity might have a social security number (in the US); the proved relation may have a date attribute .Attribute are drawn as ovals connected to their owning entity sets by a line. Every entity (unless it is a weak entity) must have a minimal set of uniquely identifying attributes. This set is called the entity’s primary key.

Entity-Relationship diagrams don’t show single entities or single instances of relations. Rather, they show entity sets and relationship sets displayed as rectangles and diamonds respectively).Example: a particular song is an entity. The collection of all songs in a database is an entity set .The proved relationship between Andrew Wiles and Fermat’s last theorem is a single relationship .The set of all such mathematicians-theorem relationships in a database is a relationship set.

Lines are drawn between entity sets and the relationship sets they involved in. If all entities set must participate in a relation in the relationship set, a thick line is drawn. This is called a participation constraint. If each entity of the entity set can participate in at most one relationship in the relationship set, an arrow is drawn from the entity set to the relationship set. This is called a key constraint. To indicate that each entity set is involved in exactly one relationship, a thick arrow is drawn. Associative entity is used to solve the problem of two entities with a many-to-many relationship.

**Unary relationship** – A unary relationship is a relationship between the rows of a single table.

A weak entity is an entity that can’t be uniquely identified by its own attributes alone, and therefore must use as its primary key both its own attributes and the primary key of an entity it is related to. A weak entity set is indicated by a bold rectangle (the entity) connected by a bold arrow to a bold diamond (the relationship).Double lines can be used instead of bold ones.

Attribute in an ER model may be further described as multi-valued, composite or derived. A multi-valued attribute, illustrated with a double-line ellipse , may have more than one value for at least one instance of its entity. For example , a piece of software (entity=application) may have the multivalued attribute “platform” because at least one instance of that entity runs on more than one operating system. A composite attribute may itself contain two or more attributes and its indicated as having at least contributing attributes of its own. For example, addresses usually are composite attributes, composed of attributes such as street address, city and so forth. Derived attributes are attributes whose value is entirely dependent on another attributes and are indicated by dashed ellipses. For example, if we have an employee database with an employee entity with an age attribute, the age attribute would be derived from a birth date attribute.

Sometimes two entities are more specific subtype of a more general type of entity. For example, programmers and marketers might both be types of employees at a software company. To indicate this, a triangle with “ISA” on the inside is drawn. The super class is connected to the point on top and two (or more) subclasses are connected to the base.

A relation and all its participating entity sets can be treated as a single entity set for the purpose of taking part in another relation through aggregation, indicated by drawing a dotted rectangle around all aggregated entities and relationships.

**Classification:**

Entity relationship models can be classified in BERMs (Binary Entity Relationship Model) and GERMs (General Entity Relationship Model) according to if only binary relationships are allowed. A binary relationship is a relationship between two entities .Thus, in a GERM, relationships between three or more entities are also allowed.

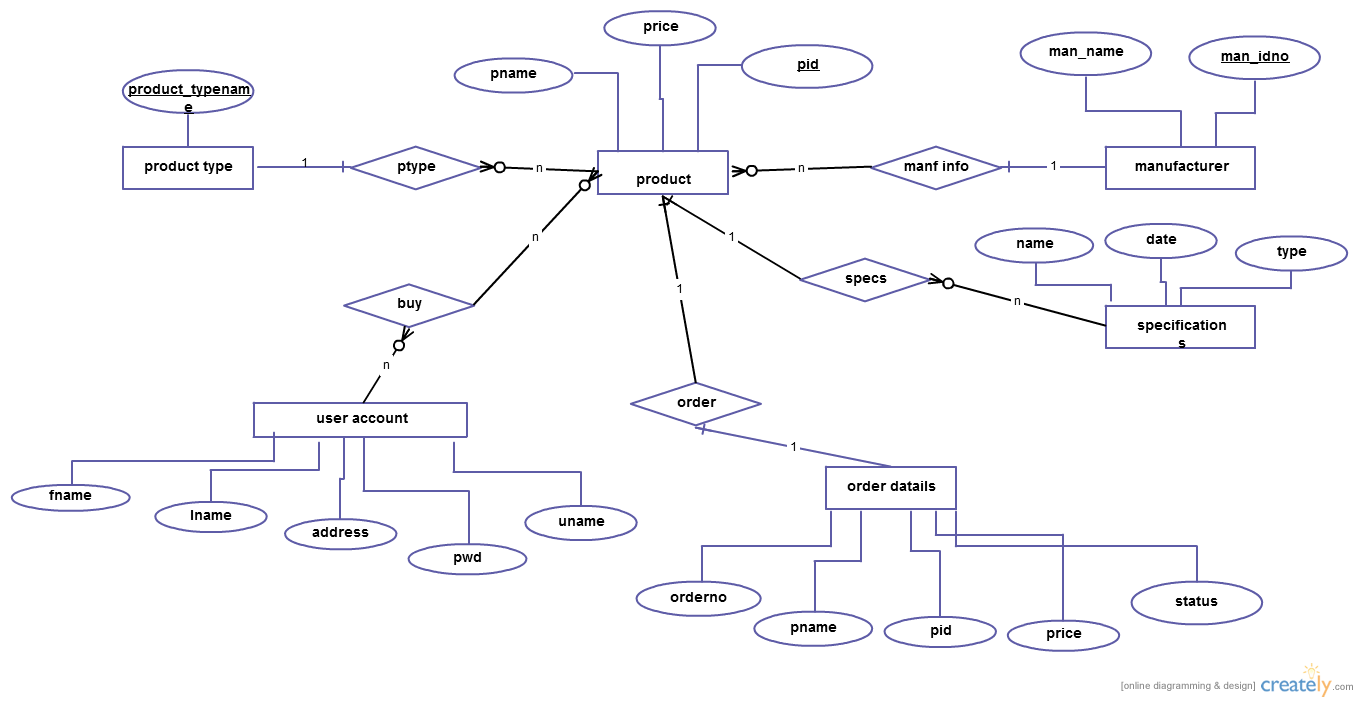
**ERD Symbols:**

Rectangl it indicates Entities

Oval defi it indicates Attributes

Diamond defines relationship

between entities and ……………………………….relationship between entity and attributes



***ER Diagram of Stock Management System***

***Data Flow Diagram***

A data flow diagram (DFD) is a graphical representation of the **“flow”** of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design) .It is common practice for a designer to draw a context –level DFD first, which shows the interaction between the system and outside entities. This context-level DFD is then “exploded” to show more detail of the system being modelled.

Data flow diagram were invented by Larry Constantine, the original developer of structured design, based on Martin and Estrin’s data flow graph” data flow graph” model of computation .Data flow diagram (DFD) are one of the three essential perspectives of SSADM.The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system evolution. With a dataflow diagram, users are able to visualize how the system will operate, what the system will accomplish and how the system will be implemented. Old system dataflow diagrams can be drawn up and compared with the new systems dataflow diagrams to draw comparisons to implement a more efficient system. Dataflow diagrams can be used to provide the end user with a physical idea of where the data they input, ultimately has an effect upon the structure of the whole system from order to dispatch to restock how any system is developed can be determined through a dataflow diagram.

In analyzing a business, several sets of DFDs are drawn. Initial DFDs might model the existing system (flaws and all), while later DFDs may model a solution to the problem being analyzed. For these solutions DFDs a logical and physical DFD is drawn .Physical DFDs represents physical files and transactions, while logical or conceptual DFDs can be used to represents business functions or processes.

A dataflow diagram illustrated the processes, data stores, and external entities in a business or other system and the connecting data flows.

The four components of a data flow diagram (DFD) are:

* External Entities/ Terminators/Sources/Sinks(represented by a square or oval)
* Processes(represented by a circle or rounded rectangle)
* Data Flows(represented by a arrow)
* Data Stores(represented by two parallel lines, sometimes connected by a vertical line)

External Entities/Terminators are outside of the system being modelled Terminators represent where information comes from and where it goes. In designing a system, we have no idea about what these terminators do or how they do it. Processes modify the inputs in the process of generating the outputs. Data stores represent a place in the processes, so a data store might be a place to accumulate data over a year for the annual accounting process. Data flows are how data moves between terminators, processes and data stores.

Every page in DFD should contain fewer than 10 components. If a process has more than 10 components, then one or more components (typically a process) should be combined into one and another DFD be generated that describes that components in more detail. Each component should be numbered, as should each subcomponent, and so on.

**Data Process:**

A data process represents the transformation of data in the system. Generally, this represents something that happens in the system. Such as “Student enrollment” .Data that flows into a process should be different from the data that flows out of the process.

**Data Store:**

A data store is a repository for data. Data stores can be manual, digital or temporary.

**External entities:**

An external entity represents the source or sink of data external to the system. When modelling a DFD, the designer is not interested in the inner workings of the external entity, but only what data is produced/needed by the entity. It flows Duplication External entities and data stores can be duplicated in the system for more clarity, while processes cannot. External entities that have been replicated are marked by an asterisk (\*) in the lower right part of the oval that represents that entity. Data stores a double line of the left side of their box.

**Developing a DFD:**

**Top-Down Approach**

The system designer makes a context level DFD, which shows the interaction (data flows) between the system (represented by one process) and the system environment(represented by terminators).The system is decomposed in lower level DFD into a set of processes, data stores, and the data flows between these processes and data stores.

Each process is then decomposed into an even lower diagram containing its sub processes. This approach then continues on the subsequent sub processes, until a necessary and sufficient level of detail is reached.

**Construct detail DFD:**

The list of all events is made. For each events a process is constructed. Each process is linked (with incoming data flows) directly with other processes or via data stores, so that it has enough information to given event. The reaction of each process to a given event is modelled by an outgoing data flow.

Develop higher-level DFDs for clarity. A group of related processes is aggregated into a process of higher level. Related aggregated processes are aggregated into a process of even higher level. This process then continues until a contextual level DFD is reached.

***Context Level DFD***

**Sale**

Avail services

**ADMINISTRATOR/ Manager**

**Customer**

Manages

**REPORTS**

**DATA STORE**

Entry

Generate

Stores

**REPORTS**

**DATA STORE**

Entry

Generate

Stores

***Hardware and software requirement***

**HARDWARE:**

* Pentium IV Processor
* HDD 20 GB
* RAM 128 MB
* VGA color Monitor
* Scroll mouse
* Multimedia Keyboard

**SOFTWARE:**

Front end C#.Net, Visual Basic

Back end SQL Server 2008

O.S. Microsoft windows xp

***PROGRAM FLOW CHART***

Anything represented in the pictorial form is easier to understand than when presented in a descriptive form .The solution to any problem is expressed as an algorithm .The Flow Chart is a graphical tool for pictorial representation of an algorithm.

A Flowchart uses different symbols for different operations. A flowchart is not a computer program .It is just an aid to represent a program in a pictorial form .In flowchart it is read from top to bottom.

There are symbols that are used in a flow chart .Each is designed for a particular function .Flow lines, showing the order in which functions are to be carried out connect these symbols .These symbols have been recognized by ANSI for information processing.

1. Terminal Symbol

2. Input/Output Symbol

3. Processing Symbol

4. Decision Symbol

5. Connectors

6. Flow Lines

Start

Welcome form

Login form

Input login password

If login

Password is correct

Message “wrong user name

/password”

On selection

login

A

Yes

No

NO

On selection

Cancel

B

Stop

NO

YES

***EVOLUTION***

Regardless of its application domain, size, or complexity, computer software will evolve over time. Change often referred to as software maintenance drives this process and occurs when errors are corrected, when the software is adapted to a new environment, when the customer requests new features or functions, and when the application is reengineered to provide benefit in a modern context.

* E-types systems are software that has been implemented in a real world-computing context and will therefore evolve over time.
* E-type must be continually adapted, or else they become progressively less satisfactory.
* As E-type system evolves its complexity increases unless work is done to maintain or reduce it.
* The E-type system evolution process is self-regulating with distribution of product and process measures close to normal.
* The average effective global act rate in an evolving E-type system is invariant over product lifetime.
* As an E-type system evolves all associated with it developers, sales, personnel, and users , for example must maintain mastery of its context and behaviour to achieve satisfactory evolution. Excessive growth diminishes that mastery. Hence the average incremental growth remains invariants as the system evolves.
* The functional context of E-type systems must be continually increased to maintain user satisfaction over the system’s lifetime.
* The quality of E-type systems will appear to be declining unless they are rigorously maintained and adapted to operational environmental changes.
* E-type evolution processes continue multilevel, multiloop, multiagent systems and must be treated as such to achieve significant improvement over any reasonable base.

***Testing technique & Strategies***

Software testing accounts for the largest percentage of technical effort in the software development process. The objective of software testing is to uncover errors. To fulfill this objective, a series of test steps concentrate on functional verification of a component and incorporation of components in to the software architecture. Validation testing demonstrates trace ability to software requirements, and system testing validates software once it has been incorporated into a large system.

Each test step is accomplished through a series of systematic test techniques that assist in the design of test cases. With each testing step, the level of abstraction with which software is considered is broadened.

Unlike testing (a systematic, planned activity), debugging must be viewed as an art beginning with a symptomatic indication of a problem, the debugging activity must track down the cause of an error. Of the many resources available during

Debugging, the most valuable is caused of other members of the software engineering staff.

The requirement for higher-quality software demands a more systematic approach to testing .What is required is an overall strategy, spanning the strategic test space, quite as deliberate in its methodology as was the systematic development on which analysis, design and code were based.

Testing is vital to the success of the any system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. Inadequate testing or non-testing leads to errors that may not appear until months later. This creates two problems:

* The time lag between the cause and the appearance of the problem (the longer the time interval, the more complicated the problem has become)
* The effect of system errors on files and records within the system.

A small system error on conceivably explode into a much larger problem. Effective testing early in the process translates directly into long term cost savings from a reduced number of errors.

Another reason for system testing is it’s utility as a user oriented vehicle before implementation. The best program is worthless if it does not meet the user needs. Unfortunately the user demands are often compromised by efforts to facilitate program or design efficiency in terms of processing time or memory utilization. Often the computer technician and the user have communication barriers due to different backgrounds, interest, priorities and perhaps languages. The system tester (designer, programmer or user)who has developed some computer mastery can bridge this barrier.

The planned test of a system should include a thorough auditing technique and introduce control elements unique to the system. The auditor’s role is to judge the controls of the system and maintain an audit trail to ensure the integrity, reliability, and confidence of the user at all levels. Being included in the system development team makes it easier for the auditor to monitor testing procedures and consider the acceptance of new controls to replace those changed by the new design.

The primary objective for test case design is to derive a set of tests that have the highest likelihood of uncovering errors in software. To accomplish this objective, two different categories of test case design techniques-applicable top conventional and object-oriented systems are used

* White-box Testing
* Black-Box Testing

White-box tests focus on the program control structure. Test cases derived to ensure that all statements in the program have been executed at least once testing and that all logical conditions have been exercised. basic path testing, a white box technique, makes use of program graphs to deriver a set of linearly independent tests that will ensure coverage. Condition and data flow testing further exercise program logic, and loop testing compliments other white-box techniques by providing a procedure for exercising loops of varying degrees of complexity.

Black-Box tests are designed to validate functional requirements without regard to the internal workings of a program. Black-Box testing techniques focus on the information domain of a program in a manner that provides thorough test coverage Equivalence partitioning divides the input domain into classes of data that are likely to exercise specific software function. Boundary value analysis probes the program’s ability to handle data at limits of acceptability.

Orthogonal array testing provides an efficient systematic method foe testing systems with small number of input parameters.

Integration testing can be accomplished using a use-based strategy. Use-based testing constructs the system in layers, beginning with those classes that do not use server classes .Integration test case design methods can also random and partition tests.

Specialized testing methods encompass a broad array of software capabilities and application areas. Testing for graphical user interfaces, client/server architectures, documentation and help facilities, and real-time systems each require specialized guidelines and techniques.

Experienced software developers often say, “Testing never ends, it just gets transferred from you to your customer. Every time your customer uses the programs, a test is being conducted.” By optimally applying the test case design, the software engineer can achieve more complete testing and thereby uncover and correct the highest number of errors before the “customer’s tests” begin.

***Implementation***

Implementation is nothing but coding, testing and installing a developed software package on the computer system. The successful implementation of the new software package is the most important part of the system Development Life Cycle. System implementation comprises the following activity.

* Creating computer compatible files
* Training the operating staff
* Installing the necessary hardware ,other necessary devices
* Installing the application package and utilities
* Testing and performing audit trail

Implementing a system is just like putting the pieces of jigsaw puzzle together. We know what we want and must put it together piece by piece. We have to buy the equipment plan individual sub-systems and hire and allocate people to implement the whole system.

Traditional approach to Implementation:

* Each module or small group of modules is coded tested and debugged individually.
* After all the modules of the whole system are debugged, these modules are grouped into subsystems.
* Subsequently each subsystem is tested and debugged.
* Next the subsystems are combined to from the whole system, this is known as system integration.
* Finally the whole system is tested and debugged.

Before the proposed new system is up and running .A major concern in updating to new system is to let there be no disruption in the current functioning of the organization.

The implementation phase of software development is also concerned with translating design specification into source code. It is necessary to write source code and internal and documentation so that conformance of the code to its specification can be easily verified, and so that debugging ,testing and modification can be performed easily .This can be achieved by making the source code as crystal clear and unambiguous as possible. The software implementation team should therefore be provided with detailed description of software requirements, an optimum architectural design specification and a detailed design representation.

Human behavioural factors should not be overlooked in the system implementation phase. System changes can provide unnecessary psychological stir among the potential users of the new system because they may show the signs of resistance to the new system. User training is another vital area, which is responsible for minimizing resistance to change which is inevitable and giving the new system a chance to proof its worth. Training aids, such as user –friendly manuals, data dictionary, and jobs performance enhancing aids that communicate information about the new system provides the user with a good start on the new system.

Conversion:

* Conversion begins with the review of the project plan, implementation plan and system test documentation. The project team and programmers along with the users do review of these documents.
* The conversion portion of the implementation plan is ultimately approved
* The conversion method to be followed is decided and applied.
* Required files are converted.
* Output generated and operations by the new system are recorded on a special form. These are documented for future reference.
* If no difficulties are encountered with the new system, it is allowed to continue its operations in the organizations.
* With this the conversion is completed and plans for the post –implementation review are prepared.
* After the review, if any modifications are suggested, these are implemented.

Following this, the new system is officially declared as operational.

**Conversion Control:**

One method for gaining control of the conversion is to use well-planned test files for testing the new system. Test file is nothing but a program that will test the new system using some test data , before the system is tested with live data. Test files must have the ability to generate initially a small number of simple records. Progressively records, which are challenging to the new system, should be generated.

The test file should provide previously determined output results to compare with those produced by the new system.

**Data entry and audit control**:

Many system fail due to inadequate attention given to data entry control or protective features such as audit control. These must be part of overall plan for conversion. Before a data entry operator starts working, a data entry validation program can be written to keep track of wrong data entered. In order to exercise data entry control, number of records that can be keyed in must be conveyed to the operator. The data entry program should possess the provision for log. In order to exercise audit control trail, the user should have a copy of additions or deletions to any file. A good audit control trail is the key to detecting fraud and errors in data entry.

At the time installation of the software it is required that SQL server must be installed on the server of the company and it must contain the data files related to the software before implementing the developed system on all machines machine connected to the server. For this the exported data is imported to the client machine and the database is checked that all tables are properly imported to the client machine or not as the software is developed in SQL server data base and also going to be implemented in oracle there is no conversion is required.

**User training:**

User manual is the most potent training aid to impart information of the proposed new system to the prospective users. The user manual illustrates in easy manner, the different steps required to accomplish the task the user is intended to perform. A user manual provides essential information required to operate a system in a short and concise format. It may include the following contents:

This project is having three module:

* Stock Management Module
* Customer Module
* Reports Module

***MAINTENANCE***

Maintenance is the enigma of system development. It holds the software industry captive, trying up programming resources. Analyst and programmers spend far more time maintaining programs than they do writing them.

Maintenance covers a wide range of activities, including correcting coding and designs errors, updating documentation and test data, and upgrading user support. More often than not most activities classified as maintenance are actually enhancements. Unlike hardware, software does not wear out; it is corrected .Although software does not war out like a piece of hardware; it ages and eventually fails to perform because of cumulative maintenance. Over time the integrity of the program, test data, and documentation degenerates as a result of modifications. Eventually, it takes more effort to maintain the application than to rewrite it.

A major problem with software maintenance is the labor-intensive nature and therefore likely hood of errors. It is error prone process that is still perceived by many as more cost effective than writing programs from scratch.

Maintenance activities begin where conversion leaves off and it handled by the same planning and control used in a formal system project. Documentation is as much a part of maintenance as it is of system development. Briefly, the maintenance staff receives a request for service from an authorized user, followed by a definition of the required modifications. The source program and written procedures for the system are acquired from the programming library.Program changes are then tested and submitted to the user for approval .Once approved , the modification documentation is filled with the library and a project complication notice is sent to the user, signaling the termination of the project.

Once a system has been installed and is fully operational, the system’s maintenance task begins.It involves keeping the software operational, that is undertaking preventive measures to keep computer running, monitoring the system’s operation, fixing problems, if any, and modifying programs in response to new user requirements if any

System maintenance is of the following three types:

* Corrective Maintenance means repairing processing or performance failures or making changes because of previously uncovered problems or false assumptions.
* Adaptive Maintenance means changing the program function.
* Perfective Maintenance enhancing the performance or modifying the program to respond to the user’s additional or changing needs.

Of these three more time and money are spent on perfective than on corrective and adaptive maintenance together.

In summary the work of maintenance is certainly not for neophyte programmers.To put maintenance in its proper perspective requires considerable skill and experience and is an important and ongoing aspect of system development.The ability of the maintenance programmer to make sound judgement depends on his/her technical expertise and ability to identified user needs.This means understanding the operational environment and what the user is trying to accomplish with the software.Maintenance demands more orientation and training than any other programming activities.

Maintenance is expensive .One way to reduce maintenance costs is through maintenance management and software modification audits.Software modifications consists of program rewrites system level updates, and re-audits of low ranking programs to verify and correct the soft spots.The outcome should be more reliable software, a reduced maintenance backlog, and higher satisfaction and morale among the maintenance staff.

***Code Efficiency***

It has been realized upteen times that good style can overcome many of the deficiencies of a selected programming language, while poor styles ca wreck havoc on the intent of the same language .The goal of a good coding style is to provide easily understood, straightforward and elegant code. writing code with efficiency requires a combination of experience ,an eye for detail, and a basic understanding of the architecture of the language and how processors work .the guidelines for coding include:

* Constraint your algorithms by following structured programming practice
* Minimum Use of unconditional branching statement i.e. “goto”
* Keep conditional logic as simple as possible
* Understand the software architecture and create interfaces that are consistent with it.

In computer science, code efficiency is used to describe several desirable properties of an algorithm or other construct. Besides clean design, functionality, etc.Code efficiency is generally contained in two properties; speed ,(the time it takes for an operation to complete),and space,(the memory or non-volatile storage used up by the construct).Optimization is the process of making code as efficient as possible ,sometimes focusing on space at the cost speed or vice versa.

The speed of an algorithm is measured in various ways .The most common method uses time complexity to determine the Big-O of an algorithm: often, it is possible to make an algorithm faster at the expense of space .This is the case whenever you cache the result of an expensive calculation rather than recalculating it on demand .This is a very common method of improving speed, so much so that language often add special features to support it, such as cc++ mutable keyword.

The space of an algorithm is actually two separate but related things. The first part is the space taken up by the compiled executable on disk (or equivalent, depending on the hardware and language)by the algorithm. This can often be reduced by preferring run-time decision making mechanisms (such as virtual functions and run-time type information)over certain compile-time decision making mechanism

(Such as macro substitution and templates).This, however ,comes at the cost of speed.

The other part of algorithm space measurement is the amount of temporary memory taken up during processing. For example, pre-caching results, as mentioned earlier, improves speed at the cost of this attribute.

Optimization of algorithms frequently depends on the properties of the machine the algorithm will be executed on. For example, one might optimize code for time efficiency in applications for home computers with sizable amounts of memory, while code to be placed in small, memory-tight devices may have to be made to run slower to conserve space.

One simple way to determine whether an optimization is worthwhile is as follows: Let the original time and space requirements (generally in Big-O notation) of the algorithm be O1 and O2.Let the new code require N1 and N2 time and space respectively. If N1N2<O1O2, the optimization should be carried out. However, as mentioned above, this may not always be true.

One must be careful, in the pursuit of good coding style, not to over-emphasize efficiency. Nearly all of the time, a clean and usable design is much more important than a fast, small design. There are exceptions to this rule (such as embedded systems, where space is tight and processing power minimal) but these are rarer than one might expect.

***DATA DICTIONARY***

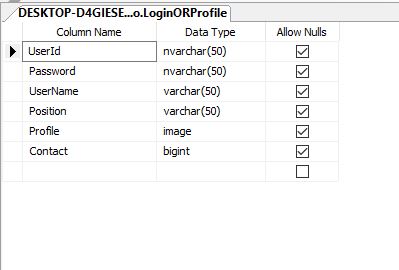
The logical characteristics of current systems data stores, including name, description, aliases, contents, and organization, identifies processes where the data are used and where immediate access to information required, Serves as the basis for identifying database requirements during system design.

**Uses of Data Dictionary:**

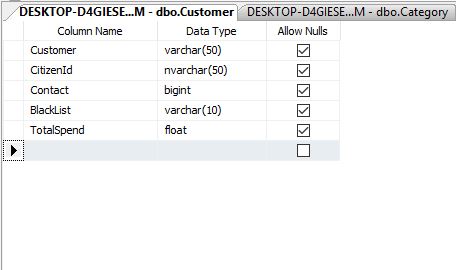
1. To manage the details in large systems.
2. To communicate a common meaning for all system elements.
3. To Document the features of the system.
4. To facilitate analysis of the details in order to evaluate characteristics and determine where system changes should be made.
5. To locate errors and omissions in the system.

***DATABASE STRUCTURE***

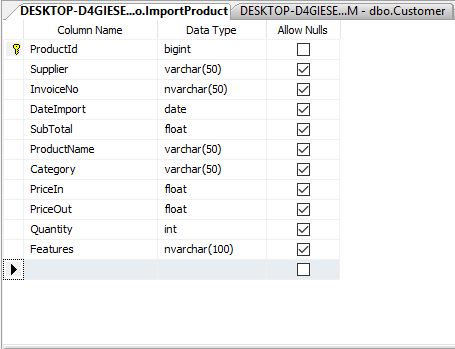
***login:***



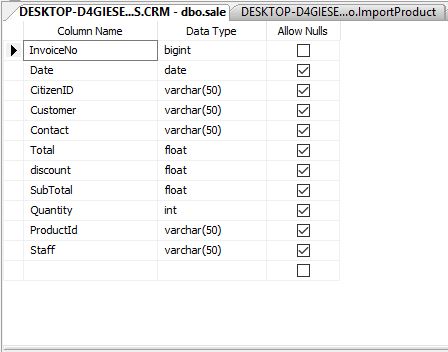
***Add Customer:***

******

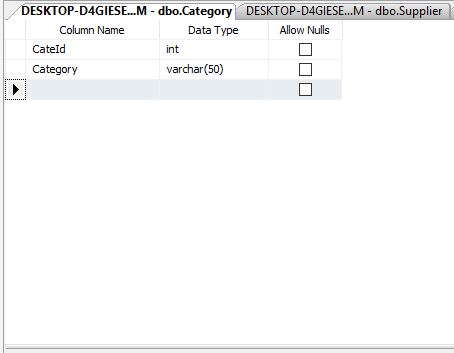
***Product:***



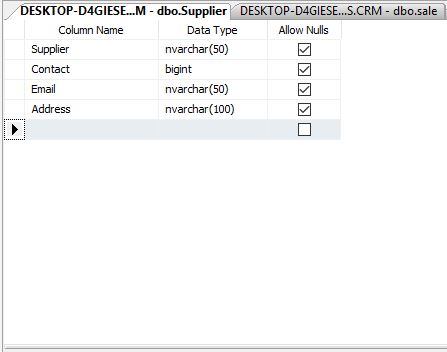
***Sale:***



***Category:***

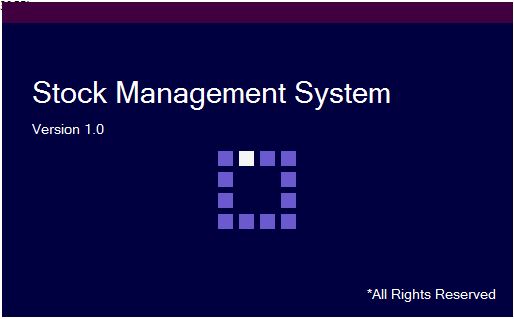


***Supplier:***



***OUTPUT SCREENS AND CODING***

***Welcome Form:-***



***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace StartPageScreen

{

public partial class Form1 : Form

{

int i = 2,count=1;

bool flag = true;

public Form1()

{

InitializeComponent();

}

void ColorChange()

{

p2.BackColor = Color.SlateBlue;

p3.BackColor = Color.SlateBlue;

p4.BackColor = Color.SlateBlue;

p5.BackColor = Color.SlateBlue;

p6.BackColor = Color.SlateBlue;

p7.BackColor = Color.SlateBlue;

p8.BackColor = Color.SlateBlue;

p9.BackColor = Color.SlateBlue;

p10.BackColor = Color.SlateBlue;

p11.BackColor = Color.SlateBlue;

p12.BackColor = Color.SlateBlue;

p13.BackColor = Color.SlateBlue;

}

private void timer1\_Tick(object sender, EventArgs e)

{

//progressBar1.Increment(1);

if (i == 14)

i = 2;

ColorChange();

switch (i)

{

case 2: p2.BackColor = Color.WhiteSmoke;

break;

case 3: p3.BackColor = Color.WhiteSmoke;

break;

case 4: p4.BackColor = Color.WhiteSmoke;

break;

case 5: p5.BackColor = Color.WhiteSmoke;

break;

case 6: p6.BackColor = Color.WhiteSmoke;

break;

case 7: p7.BackColor = Color.WhiteSmoke;

break;

case 8: p8.BackColor = Color.WhiteSmoke;

break;

case 9: p9.BackColor = Color.WhiteSmoke;

break;

case 10: p10.BackColor = Color.WhiteSmoke;

break;

case 11: p11.BackColor = Color.WhiteSmoke;

break;

case 12: p12.BackColor = Color.WhiteSmoke;

break;

case 13: p13.BackColor = Color.WhiteSmoke;

break;

}

i++;

count++;

if(count==27)

{

timer1.Stop();

MessageBox.Show("Timer Stopped");

/\*\*\*

\*

\* next page code here

\*

\*

\* \*/

}

}

private void Form1\_Load(object sender, EventArgs e)

{

timer1.Start();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void p2\_Paint(object sender, PaintEventArgs e)

{

}

private void p3\_Paint(object sender, PaintEventArgs e)

{

}

private void p4\_Paint(object sender, PaintEventArgs e)

{

}

private void p5\_Paint(object sender, PaintEventArgs e)

{

}

private void p6\_Paint(object sender, PaintEventArgs e)

{

}

private void p7\_Paint(object sender, PaintEventArgs e)

{

}

private void p8\_Paint(object sender, PaintEventArgs e)

{

}

private void p13\_Paint(object sender, PaintEventArgs e)

{ }

private void p12\_Paint(object sender, PaintEventArgs e)

{

}

private void p11\_Paint(object sender, PaintEventArgs e)

{

}

private void p10\_Paint(object sender, PaintEventArgs e)

{ }

private void p9\_Paint(object sender, PaintEventArgs e)

{ }

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

}

}

***Login Form:***





***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.IO;

namespace CRM

{

public partial class loing : Form

{

MemoryStream ms;

SqlCommand cm; string str1;

SqlConnection cn;

public static string user,pass;

public loing()

{

InitializeComponent();

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("select \* from LoginORProfile where UserId='"+textBox1.Text+"' and Password='"+ textBox2.Text+"' ", cn);

SqlDataReader dr = cm.ExecuteReader();

dr.Read();

if(dr.HasRows)

{

user=textBox1.Text;

pass = textBox2.Text;

MessageBox.Show("Login Suuccess");

cn.Close();

Home ob = new Home();

ob.Show();

}

else

{

MessageBox.Show("Login Failure");

}

}catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void label3\_Click(object sender, EventArgs e)

{

this.Close();

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

public void circle()

{

System.Drawing.Drawing2D.GraphicsPath gp = new System.Drawing.Drawing2D.GraphicsPath();

gp.AddEllipse(0, 0, pictureBox1.Width - 3, pictureBox1.Height - 3);

Region rg = new Region(gp);

pictureBox1.Region = rg;

}

private void loing\_Load(object sender, EventArgs e)

{

label10.Hide();

circle(); timer1.Enabled = true;

}

private void timer1\_Tick(object sender, EventArgs e)

{

label9.Text = DateTime.Now.ToString("MM/dd/yyyy HH:mm:ss");

}

private void textBox2\_Click(object sender, EventArgs e)

{

string str = textBox1.Text;

try

{

cn.Open();

cm = new SqlCommand("Select \* from LoginORProfile Where UserId='" + str + "' ", cn);

SqlDataReader dr = cm.ExecuteReader();

dr.Read();

if (dr.HasRows)

{

string str2 = dr[0].ToString();

str1 = dr[2].ToString();

byte[] data = (byte[])dr[4];

ms = new MemoryStream(data);

pictureBox1.Image = Image.FromStream(ms);

if (str.Equals(str2))

{

label1.Hide();

textBox1.Hide();

label10.Text = str1;

label10.Show();

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

forget f = new forget();

f.ShowDialog();

}

private void button2\_Click(object sender, EventArgs e)

{

textBox1.Text = "";

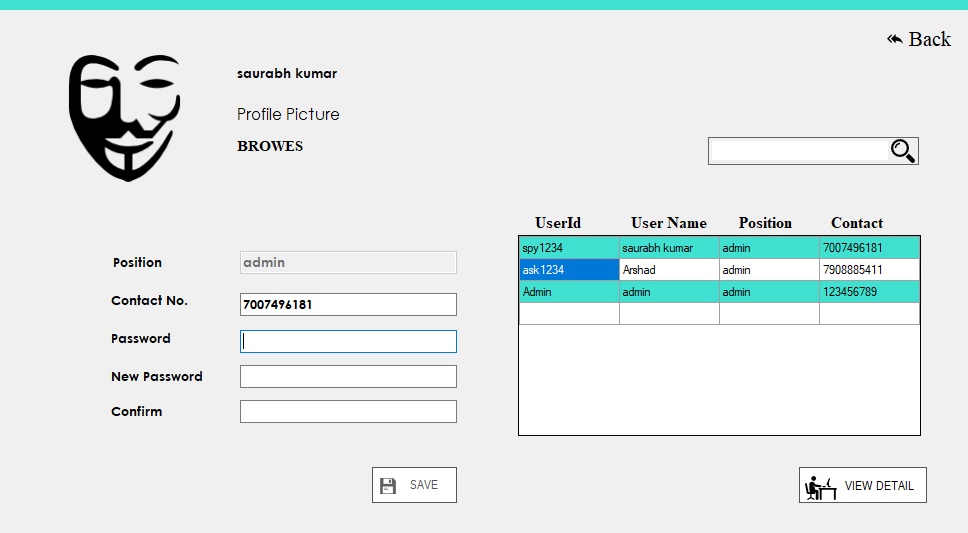
textBox2.Text = "";

}

}

}

***Account/Signup:***

******

***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.IO;

namespace CRM

{

public partial class Account : Form

{

SqlCommand cm;

SqlConnection cn;

private PictureBox pictureBox1;

private Label label2;

private Label label3;

private Label label4;

private TextBox textBox2;

private Label label6;

private TextBox textBox4;

private Label label7;

private TextBox textBox5;

private Button button2;

private Panel panel4;

private PictureBox pictureBox3;

private TextBox textBox1;

private OpenFileDialog openFileDialog1;

private Label label9;

private Label label10;

private Label label11;

private Label label12;

private Button button5;

private Label label8;

private TextBox textBox6;

private Label label5;

private TextBox textBox3;

private Panel panel1;

private TextBox textBox7;

private CRMDataSet cRMDataSet;

private BindingSource loginORProfileBindingSource;

private IContainer components;

private CRMDataSetTableAdapters.LoginORProfileTableAdapter loginORProfileTableAdapter;

private Label label1;

private DataGridView dataGridView1;

private DataGridViewTextBoxColumn userIdDataGridViewTextBoxColumn;

private DataGridViewTextBoxColumn userNameDataGridViewTextBoxColumn;

private DataGridViewTextBoxColumn positionDataGridViewTextBoxColumn;

private DataGridViewTextBoxColumn contactDataGridViewTextBoxColumn;

private bool lo;

public Account()

{

InitializeComponent();

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void label10\_Click(object sender, EventArgs e)

{

}

private void InitializeComponent()

{

this.components = new System.ComponentModel.Container();

System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Account));

this.panel1 = new System.Windows.Forms.Panel();

this.pictureBox1 = new System.Windows.Forms.PictureBox();

this.label2 = new System.Windows.Forms.Label();

this.label3 = new System.Windows.Forms.Label();

this.label4 = new System.Windows.Forms.Label();

this.textBox2 = new System.Windows.Forms.TextBox();

this.label6 = new System.Windows.Forms.Label();

this.textBox4 = new System.Windows.Forms.TextBox();

this.label7 = new System.Windows.Forms.Label();

this.textBox5 = new System.Windows.Forms.TextBox();

this.button2 = new System.Windows.Forms.Button();

this.panel4 = new System.Windows.Forms.Panel();

this.pictureBox3 = new System.Windows.Forms.PictureBox();

this.textBox1 = new System.Windows.Forms.TextBox();

this.openFileDialog1 = new System.Windows.Forms.OpenFileDialog();

this.label9 = new System.Windows.Forms.Label();

this.label10 = new System.Windows.Forms.Label();

this.label11 = new System.Windows.Forms.Label();

this.label12 = new System.Windows.Forms.Label();

this.button5 = new System.Windows.Forms.Button();

this.label8 = new System.Windows.Forms.Label();

this.textBox6 = new System.Windows.Forms.TextBox();

this.label5 = new System.Windows.Forms.Label();

this.textBox3 = new System.Windows.Forms.TextBox();

this.textBox7 = new System.Windows.Forms.TextBox();

this.cRMDataSet = new CRM.CRMDataSet();

this.loginORProfileBindingSource = new System.Windows.Forms.BindingSource(this.components);

this.loginORProfileTableAdapter = new CRM.CRMDataSetTableAdapters.LoginORProfileTableAdapter();

this.label1 = new System.Windows.Forms.Label();

this.dataGridView1 = new System.Windows.Forms.DataGridView();

this.userIdDataGridViewTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();

this.userNameDataGridViewTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();

this.positionDataGridViewTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();

this.contactDataGridViewTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();

this.panel4.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.pictureBox3)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.cRMDataSet)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.loginORProfileBindingSource)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).BeginInit();

this.SuspendLayout();

//

// panel1

//

this.panel1.BackColor = System.Drawing.Color.Turquoise;

this.panel1.Location = new System.Drawing.Point(0, 0);

this.panel1.Name = "panel1";

this.panel1.Size = new System.Drawing.Size(972, 10);

this.panel1.TabIndex = 0;

//

// pictureBox1

//

this.pictureBox1.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox1.Image")));

this.pictureBox1.Location = new System.Drawing.Point(64, 55);

this.pictureBox1.Name = "pictureBox1";

this.pictureBox1.Size = new System.Drawing.Size(127, 127);

this.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;

this.pictureBox1.TabIndex = 1;

this.pictureBox1.TabStop = false;

this.pictureBox1.Click += new System.EventHandler(this.pictureBox1\_Click);

//

// label2

//

this.label2.AutoSize = true;

this.label2.Font = new System.Drawing.Font("Century Gothic", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label2.Location = new System.Drawing.Point(236, 103);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(113, 21);

this.label2.TabIndex = 3;

this.label2.Text = "Profile Picture";

//

// label3

//

this.label3.AutoSize = true;

this.label3.Font = new System.Drawing.Font("Times New Roman", 11.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label3.Location = new System.Drawing.Point(237, 137);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(74, 17);

this.label3.TabIndex = 4;

this.label3.Text = "BROWES";

this.label3.Click += new System.EventHandler(this.label3\_Click);

//

// label4

//

this.label4.AutoSize = true;

this.label4.Font = new System.Drawing.Font("Century Gothic", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label4.Location = new System.Drawing.Point(113, 254);

this.label4.Name = "label4";

this.label4.Size = new System.Drawing.Size(57, 16);

this.label4.TabIndex = 6;

this.label4.Text = "Position";

//

// textBox2

//

this.textBox2.Enabled = false;

this.textBox2.Font = new System.Drawing.Font("Century Gothic", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.textBox2.Location = new System.Drawing.Point(243, 251);

this.textBox2.Name = "textBox2";

this.textBox2.Size = new System.Drawing.Size(217, 23);

this.textBox2.TabIndex = 7;

//

// label6

//

this.label6.AutoSize = true;

this.label6.Font = new System.Drawing.Font("Century Gothic", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label6.Location = new System.Drawing.Point(111, 330);

this.label6.Name = "label6";

this.label6.Size = new System.Drawing.Size(68, 16);

this.label6.TabIndex = 6;

this.label6.Text = "Password";

//

// textBox4

//

this.textBox4.Font = new System.Drawing.Font("Century Gothic", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.textBox4.Location = new System.Drawing.Point(243, 330);

this.textBox4.Name = "textBox4";

this.textBox4.Size = new System.Drawing.Size(217, 23);

this.textBox4.TabIndex = 7;

this.textBox4.Leave += new System.EventHandler(this.textBox4\_Leave);

//

// label7

//

this.label7.AutoSize = true;

this.label7.Font = new System.Drawing.Font("Century Gothic", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label7.Location = new System.Drawing.Point(111, 368);

this.label7.Name = "label7";

this.label7.Size = new System.Drawing.Size(100, 16);

this.label7.TabIndex = 6;

this.label7.Text = "New Password";

//

// textBox5

//

this.textBox5.Font = new System.Drawing.Font("Century Gothic", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.textBox5.Location = new System.Drawing.Point(243, 365);

this.textBox5.Name = "textBox5";

this.textBox5.Size = new System.Drawing.Size(217, 23);

this.textBox5.TabIndex = 7;

//

// button2

//

this.button2.BackColor = System.Drawing.Color.White;

this.button2.FlatAppearance.BorderColor = System.Drawing.Color.Aqua;

this.button2.FlatAppearance.BorderSize = 0;

this.button2.FlatAppearance.MouseDownBackColor = System.Drawing.Color.Aqua;

this.button2.FlatAppearance.MouseOverBackColor = System.Drawing.SystemColors.ActiveCaption;

this.button2.FlatStyle = System.Windows.Forms.FlatStyle.Popup;

this.button2.Image = ((System.Drawing.Image)(resources.GetObject("button2.Image")));

this.button2.Location = new System.Drawing.Point(802, 467);

this.button2.Name = "button2";

this.button2.Size = new System.Drawing.Size(128, 36);

this.button2.TabIndex = 9;

this.button2.Text = " VIEW DETAIL";

this.button2.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageBeforeText;

this.button2.UseVisualStyleBackColor = false;

this.button2.Click += new System.EventHandler(this.button2\_Click);

//

// panel4

//

this.panel4.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

this.panel4.Controls.Add(this.pictureBox3);

this.panel4.Controls.Add(this.textBox1);

this.panel4.Location = new System.Drawing.Point(711, 137);

this.panel4.Name = "panel4";

this.panel4.Size = new System.Drawing.Size(211, 28);

this.panel4.TabIndex = 13;

//

// pictureBox3

//

this.pictureBox3.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox3.Image")));

this.pictureBox3.Location = new System.Drawing.Point(182, 1);

this.pictureBox3.Name = "pictureBox3";

this.pictureBox3.Size = new System.Drawing.Size(24, 24);

this.pictureBox3.SizeMode = System.Windows.Forms.PictureBoxSizeMode.AutoSize;

this.pictureBox3.TabIndex = 10;

this.pictureBox3.TabStop = false;

//

// textBox1

//

this.textBox1.BorderStyle = System.Windows.Forms.BorderStyle.None;

this.textBox1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F);

this.textBox1.Location = new System.Drawing.Point(3, 3);

this.textBox1.Name = "textBox1";

this.textBox1.Size = new System.Drawing.Size(176, 19);

this.textBox1.TabIndex = 9;

//

// openFileDialog1

//

this.openFileDialog1.FileName = "openFileDialog1";

//

// label9

//

this.label9.AutoSize = true;

this.label9.Font = new System.Drawing.Font("Times New Roman", 15.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label9.Image = ((System.Drawing.Image)(resources.GetObject("label9.Image")));

this.label9.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.label9.Location = new System.Drawing.Point(888, 28);

this.label9.Name = "label9";

this.label9.Size = new System.Drawing.Size(72, 23);

this.label9.TabIndex = 14;

this.label9.Text = " Back";

this.label9.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

this.label9.Click += new System.EventHandler(this.label9\_Click\_1);

//

// label10

//

this.label10.AutoSize = true;

this.label10.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label10.Location = new System.Drawing.Point(630, 213);

this.label10.Name = "label10";

this.label10.Size = new System.Drawing.Size(85, 19);

this.label10.TabIndex = 16;

this.label10.Text = "User Name";

//

// label11

//

this.label11.AutoSize = true;

this.label11.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label11.Location = new System.Drawing.Point(738, 213);

this.label11.Name = "label11";

this.label11.Size = new System.Drawing.Size(62, 19);

this.label11.TabIndex = 16;

this.label11.Text = "Position";

//

// label12

//

this.label12.AutoSize = true;

this.label12.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label12.Location = new System.Drawing.Point(830, 213);

this.label12.Name = "label12";

this.label12.Size = new System.Drawing.Size(61, 19);

this.label12.TabIndex = 16;

this.label12.Text = "Contact";

//

// button5

//

this.button5.BackColor = System.Drawing.Color.White;

this.button5.FlatAppearance.BorderColor = System.Drawing.Color.Aqua;

this.button5.FlatAppearance.BorderSize = 0;

this.button5.FlatAppearance.MouseDownBackColor = System.Drawing.Color.Aqua;

this.button5.FlatAppearance.MouseOverBackColor = System.Drawing.SystemColors.ActiveCaption;

this.button5.FlatStyle = System.Windows.Forms.FlatStyle.Popup;

this.button5.Image = ((System.Drawing.Image)(resources.GetObject("button5.Image")));

this.button5.Location = new System.Drawing.Point(375, 467);

this.button5.Name = "button5";

this.button5.Size = new System.Drawing.Size(85, 36);

this.button5.TabIndex = 28;

this.button5.Tag = " ";

this.button5.Text = " SAVE";

this.button5.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageBeforeText;

this.button5.UseVisualStyleBackColor = false;

this.button5.Click += new System.EventHandler(this.button5\_Click);

//

// label8

//

this.label8.AutoSize = true;

this.label8.Font = new System.Drawing.Font("Century Gothic", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label8.Location = new System.Drawing.Point(111, 403);

this.label8.Name = "label8";

this.label8.Size = new System.Drawing.Size(59, 16);

this.label8.TabIndex = 6;

this.label8.Text = "Confirm";

//

// textBox6

//

this.textBox6.Font = new System.Drawing.Font("Century Gothic", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.textBox6.Location = new System.Drawing.Point(243, 400);

this.textBox6.Name = "textBox6";

this.textBox6.Size = new System.Drawing.Size(217, 23);

this.textBox6.TabIndex = 7;

//

// label5

//

this.label5.AutoSize = true;

this.label5.Font = new System.Drawing.Font("Century Gothic", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label5.Location = new System.Drawing.Point(111, 292);

this.label5.Name = "label5";

this.label5.Size = new System.Drawing.Size(85, 16);

this.label5.TabIndex = 6;

this.label5.Text = "Contact No.";

//

// textBox3

//

this.textBox3.Font = new System.Drawing.Font("Century Gothic", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.textBox3.Location = new System.Drawing.Point(243, 293);

this.textBox3.Name = "textBox3";

this.textBox3.Size = new System.Drawing.Size(217, 23);

this.textBox3.TabIndex = 7;

//

// textBox7

//

this.textBox7.BackColor = System.Drawing.SystemColors.Control;

this.textBox7.BorderStyle = System.Windows.Forms.BorderStyle.None;

this.textBox7.Font = new System.Drawing.Font("Century Gothic", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.textBox7.Location = new System.Drawing.Point(240, 65);

this.textBox7.Name = "textBox7";

this.textBox7.Size = new System.Drawing.Size(191, 16);

this.textBox7.TabIndex = 7;

//

// cRMDataSet

//

this.cRMDataSet.DataSetName = "CRMDataSet";

this.cRMDataSet.SchemaSerializationMode = System.Data.SchemaSerializationMode.IncludeSchema;

//

// loginORProfileBindingSource

//

this.loginORProfileBindingSource.DataMember = "LoginORProfile";

this.loginORProfileBindingSource.DataSource = this.cRMDataSet;

//

// loginORProfileTableAdapter

//

this.loginORProfileTableAdapter.ClearBeforeFill = true;

//

// label1

//

this.label1.AutoSize = true;

this.label1.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label1.Location = new System.Drawing.Point(534, 213);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(55, 19);

this.label1.TabIndex = 16;

this.label1.Text = "UserId";

//

// dataGridView1

//

this.dataGridView1.AutoGenerateColumns = false;

this.dataGridView1.BackgroundColor = System.Drawing.Color.White;

this.dataGridView1.ColumnHeadersHeightSizeMode = System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;

this.dataGridView1.ColumnHeadersVisible = false;

this.dataGridView1.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {

this.userIdDataGridViewTextBoxColumn,

this.userNameDataGridViewTextBoxColumn,

this.positionDataGridViewTextBoxColumn,

this.contactDataGridViewTextBoxColumn});

this.dataGridView1.DataSource = this.loginORProfileBindingSource;

this.dataGridView1.Location = new System.Drawing.Point(521, 235);

this.dataGridView1.Name = "dataGridView1";

this.dataGridView1.RowHeadersVisible = false;

this.dataGridView1.Size = new System.Drawing.Size(403, 201);

this.dataGridView1.TabIndex = 29;

//

// userIdDataGridViewTextBoxColumn

//

this.userIdDataGridViewTextBoxColumn.DataPropertyName = "UserId";

this.userIdDataGridViewTextBoxColumn.HeaderText = "UserId";

this.userIdDataGridViewTextBoxColumn.Name = "userIdDataGridViewTextBoxColumn";

//

// userNameDataGridViewTextBoxColumn

//

this.userNameDataGridViewTextBoxColumn.DataPropertyName = "UserName";

this.userNameDataGridViewTextBoxColumn.HeaderText = "User Name";

this.userNameDataGridViewTextBoxColumn.Name = "userNameDataGridViewTextBoxColumn";

//

// positionDataGridViewTextBoxColumn

//

this.positionDataGridViewTextBoxColumn.DataPropertyName = "Position";

this.positionDataGridViewTextBoxColumn.HeaderText = "Position";

this.positionDataGridViewTextBoxColumn.Name = "positionDataGridViewTextBoxColumn";

//

// contactDataGridViewTextBoxColumn

//

this.contactDataGridViewTextBoxColumn.DataPropertyName = "Contact";

this.contactDataGridViewTextBoxColumn.HeaderText = "Contact";

this.contactDataGridViewTextBoxColumn.Name = "contactDataGridViewTextBoxColumn";

//

// Account

//

this.ClientSize = new System.Drawing.Size(972, 533);

this.Controls.Add(this.dataGridView1);

this.Controls.Add(this.button5);

this.Controls.Add(this.label12);

this.Controls.Add(this.label11);

this.Controls.Add(this.label1);

this.Controls.Add(this.label10);

this.Controls.Add(this.label9);

this.Controls.Add(this.panel4);

this.Controls.Add(this.button2);

this.Controls.Add(this.textBox6);

this.Controls.Add(this.label8);

this.Controls.Add(this.textBox5);

this.Controls.Add(this.label7);

this.Controls.Add(this.textBox4);

this.Controls.Add(this.label6);

this.Controls.Add(this.textBox7);

this.Controls.Add(this.textBox3);

this.Controls.Add(this.label5);

this.Controls.Add(this.textBox2);

this.Controls.Add(this.label4);

this.Controls.Add(this.label3);

this.Controls.Add(this.label2);

this.Controls.Add(this.pictureBox1);

this.Controls.Add(this.panel1);

this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;

this.Location = new System.Drawing.Point(328, 152);

this.Name = "Account";

this.Load += new System.EventHandler(this.Form1\_Load);

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();

this.panel4.ResumeLayout(false);

this.panel4.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.pictureBox3)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.cRMDataSet)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.loginORProfileBindingSource)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).EndInit();

this.ResumeLayout(false);

this.PerformLayout();

}

private void Form1\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'cRMDataSet.LoginORProfile' table. You can move, or remove it, as needed.

button5.Enabled = false;

try

{

cn.Open();

//..MessageBox.Show(loing.user);/.

cm = new SqlCommand("Select \* from LoginORProfile where UserId='" + loing.user + "'", cn);

SqlDataReader dr = cm.ExecuteReader();

while (dr.Read())

{

textBox2.Text = dr[3].ToString();

textBox3.Text = dr[5].ToString();

textBox7.Text = dr[2].ToString();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void label9\_Click(object sender, EventArgs e)

{

}

private void label9\_Click\_1(object sender, EventArgs e)

{

this.Close();

}

private void button5\_Click(object sender, EventArgs e)

{

if (textBox5.Text == textBox6.Text)

{

try

{

cn.Open();

cm = new SqlCommand("update LoginORProfile set UserName=@a,Profile=@pic,Contact=@c,Password=@d where UserId='" + loing.user + "'", cn);

cm.Parameters.AddWithValue("@a", textBox7.Text);

cm.Parameters.AddWithValue("@c", textBox3.Text);

cm.Parameters.AddWithValue("@d", textBox5.Text);

MemoryStream ms = new MemoryStream();

pictureBox1.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg);

byte[] p = ms.ToArray();

cm.Parameters.AddWithValue("@pic", p);

int re = cm.ExecuteNonQuery();

if (re > 0)

{

MessageBox.Show("Updation successfull");

}

else

{

MessageBox.Show("Data is not updated");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

else

{

MessageBox.Show("Password is not same...checkOut ");

}

}

private void label1\_Click(object sender, EventArgs e)

{

textBox7.Show();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

openFileDialog1.ShowDialog();

openFileDialog1.Filter = "\*jpeg|\*.jpg|\*.bmp|\*.gif|all files|\*.\*";

pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);

}

private void textBox4\_Leave(object sender, EventArgs e)

{

if (loing.pass == textBox4.Text)

{

MessageBox.Show("Password is match.");

button5.Enabled = true;

}

else

{

MessageBox.Show("Password is not match.");

button5.Enabled = false;

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.loginORProfileTableAdapter.Fill(this.cRMDataSet.LoginORProfile);

for(int i=0;i<dataGridView1.Rows.Count;i=i+2)

{

dataGridView1.Rows[i].DefaultCellStyle.BackColor = Color.Turquoise;

}

}

private void label3\_Click(object sender, EventArgs e)

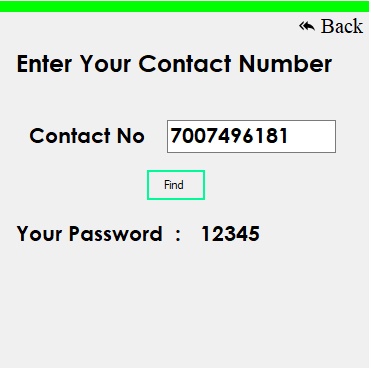
{

}

}

}

***Forgot Password:***

******

***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient ;

namespace CRM

{

public partial class forget : Form

{ SqlConnection cn;

SqlCommand cm;

public forget()

{

InitializeComponent();

cn=new SqlConnection (@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("Select \* from LoginORProfile Where Contact=@c", cn);

cm.Parameters.AddWithValue("@c", textBox1.Text);

SqlDataReader dr = cm.ExecuteReader();

dr.Read();

if (dr.HasRows)

{

label2.Show();

label3.Show();

label2.Text = dr[1].ToString();

}

else

{

MessageBox.Show("Contact Number Is Not Right ");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void forget\_Load(object sender, EventArgs e)

{

label2.Hide();

label3.Hide();

}

private void label9\_Click(object sender, EventArgs e)

{

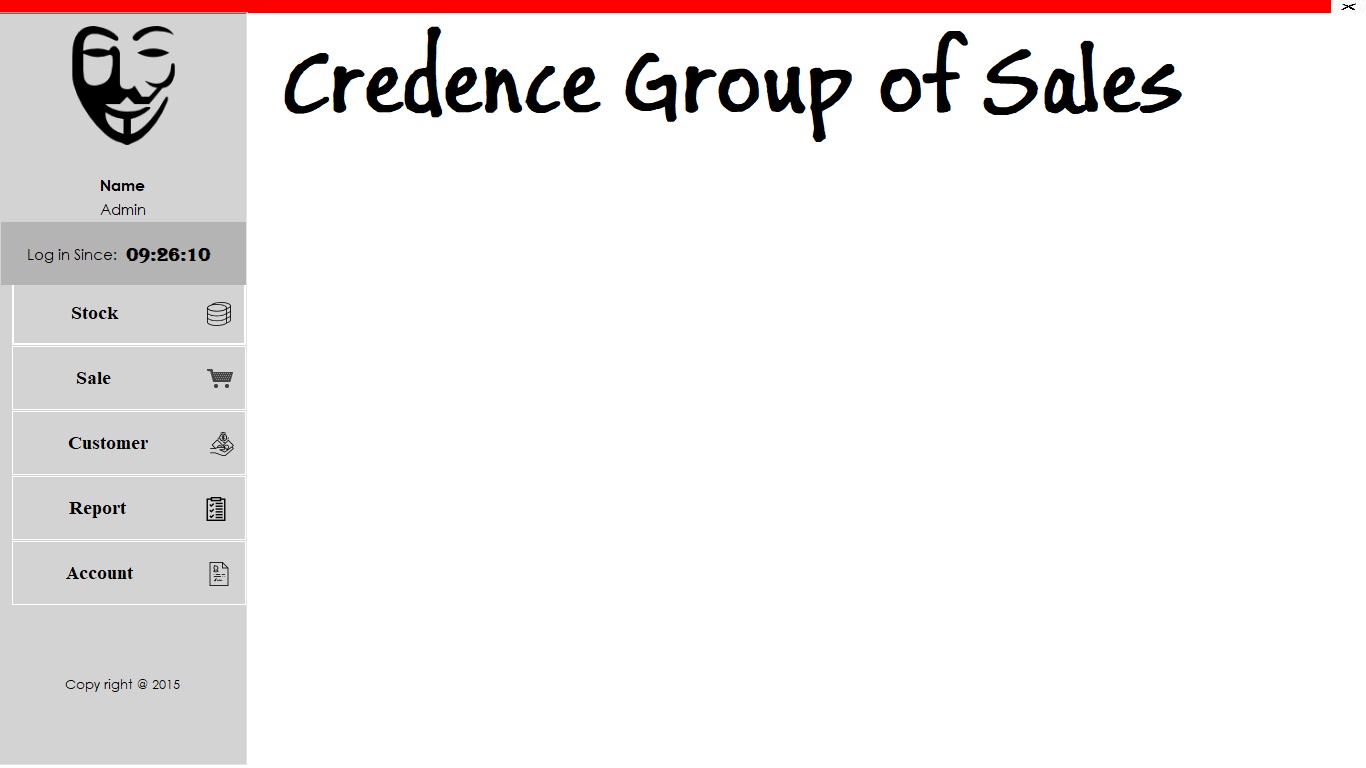
this.Close();

}

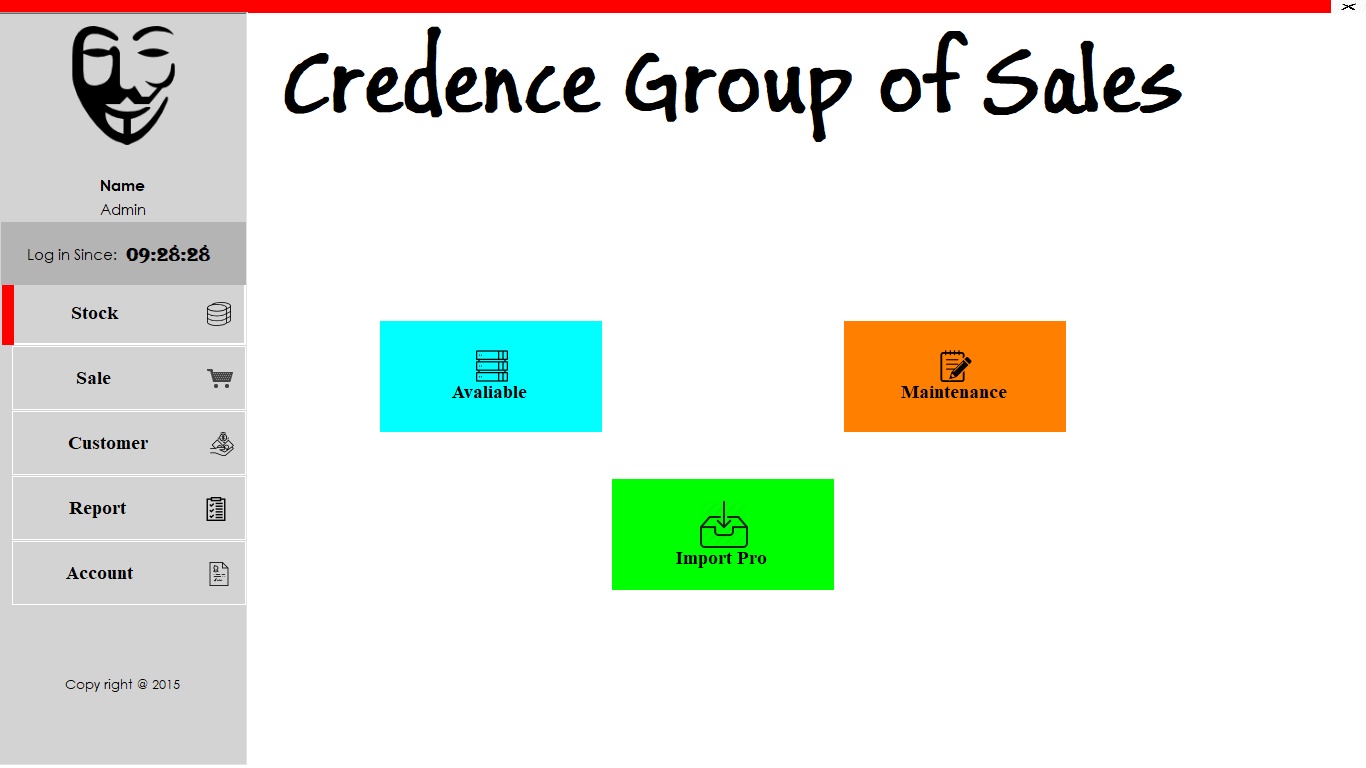
}

}

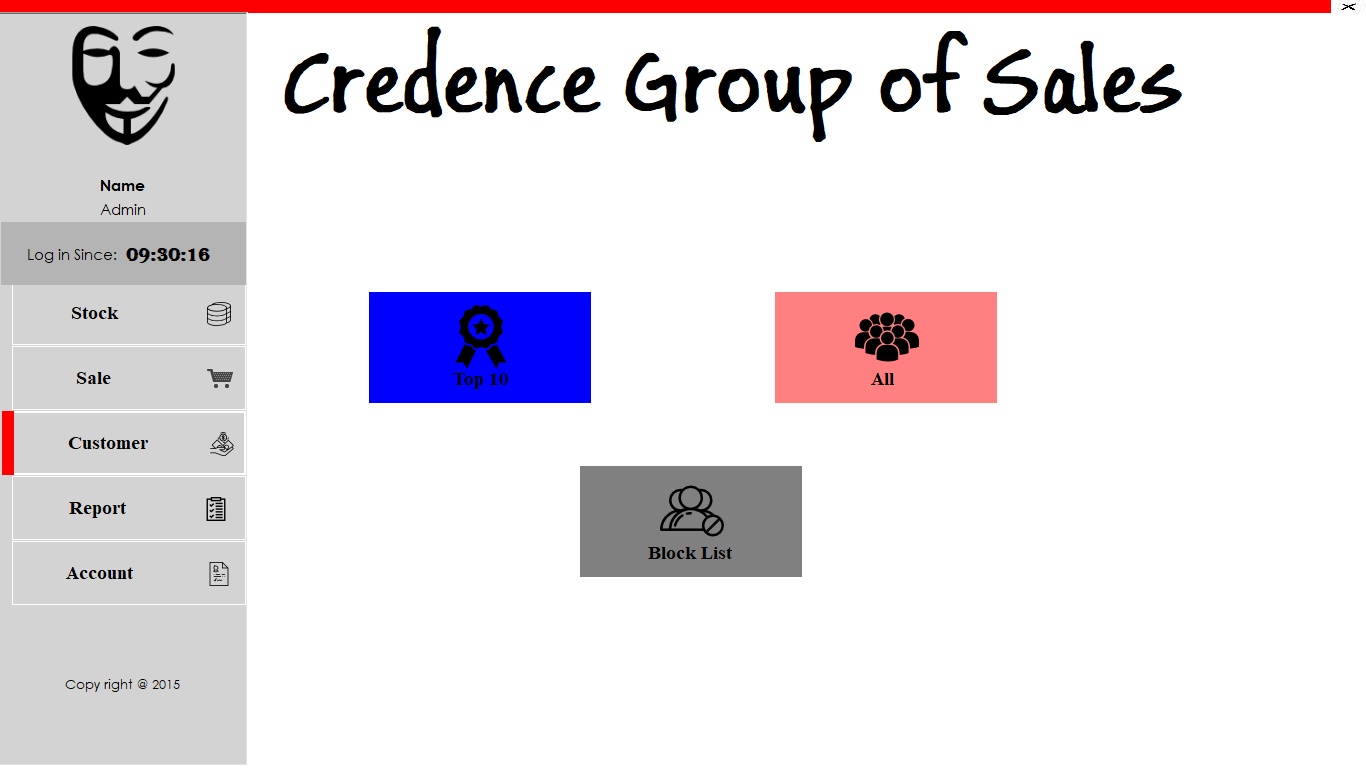
***Home Screen :***

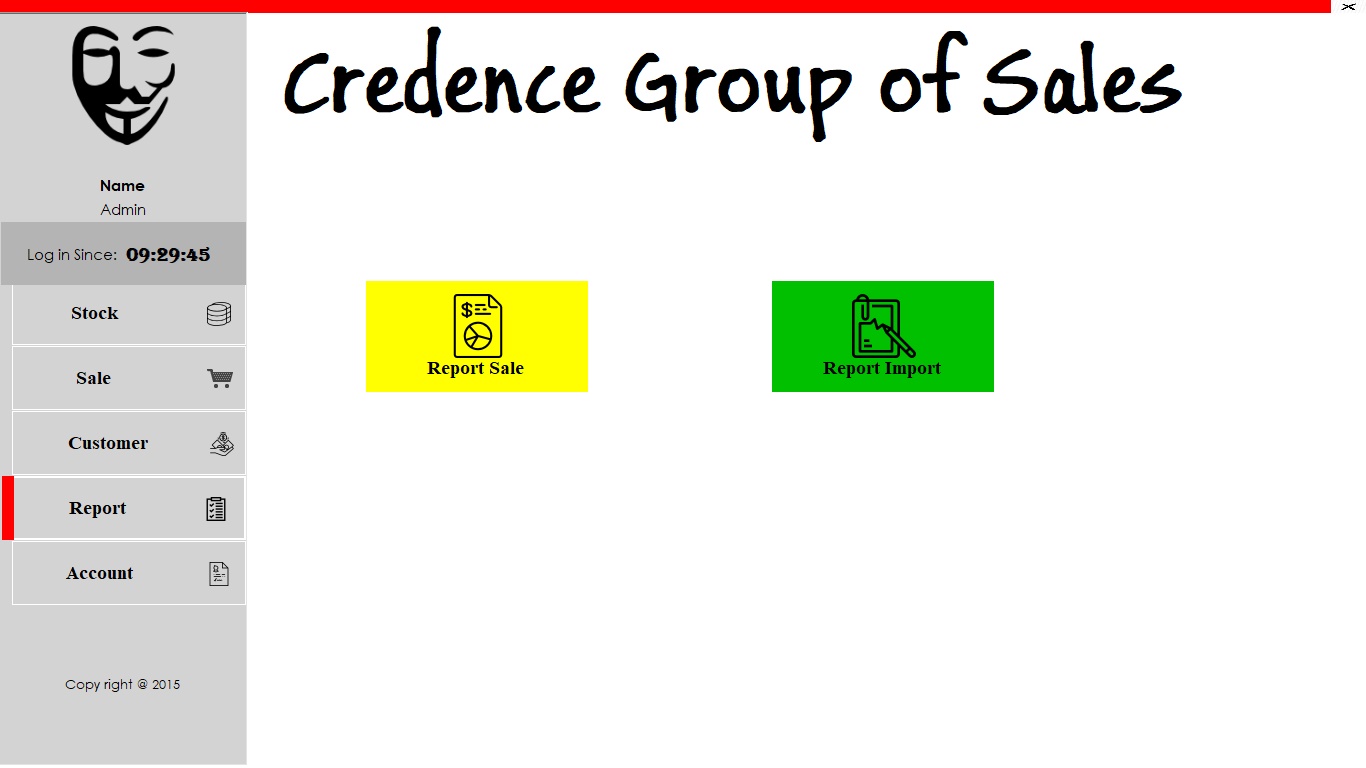
******

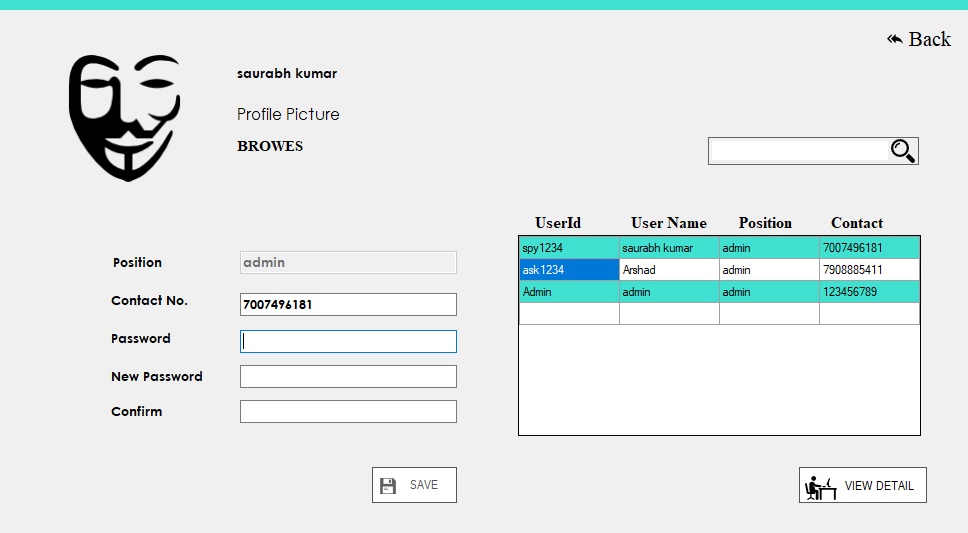
***Home Menu:***

******

******

******

******

******

***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CRM

{

public partial class Home : Form

{

public Home()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

panel8.Hide();

panel4.Hide();

panel5.Hide();

panel6.Hide();

panel7.Hide();

//groupBox1.Hide();

//groupBox2.Hide();

//groupBox3.Hide();

//groupBox4.Hide();

timer1.Start();

}

private void panel2\_Paint(object sender, PaintEventArgs e)

{

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

panel8.Show();

panel8.Height = button1.Height;

panel8.Top = button1.Top;

panel4.Hide();

panel5.Hide();

panel6.Hide();

panel7.Show();

}

private void timer1\_Tick(object sender, EventArgs e)

{

label5.Text = DateTime.Now.ToString("hh:mm:ss");

}

private void TopB\_Click(object sender, EventArgs e)

{

top10 tp = new top10();

tp.ShowDialog();

}

private void BlockB\_Click(object sender, EventArgs e)

{

BlackList bl = new BlackList();

bl.ShowDialog();

}

private void AllB\_Click(object sender, EventArgs e)

{

All ab = new All();

ab.ShowDialog();

}

private void button2\_Click(object sender, EventArgs e)

{

panel4.Show();

panel5.Hide();

panel6.Hide();

panel7.Hide();

panel8.Show();

panel8.Height = button2.Height;

panel8.Top = button2.Top;

}

private void button3\_Click(object sender, EventArgs e)

{

panel8.Show();

panel4.Hide();

panel5.Hide();

panel6.Show();

panel7.Hide();

panel8.Height = button3.Height;

panel8.Top = button3.Top;

}

private void button4\_Click(object sender, EventArgs e)

{

panel8.Show();

panel4.Hide();

panel5.Show();

panel6.Hide();

panel7.Hide();

panel8.Height = button4.Height;

panel8.Top = button4.Top;

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

panel4.Hide();

panel5.Hide();

panel6.Hide();

panel7.Hide();

panel8.Hide();

}

private void button5\_Click(object sender, EventArgs e)

{

panel8.Show();

panel4.Hide();

panel5.Hide();

panel6.Hide();

panel7.Hide();

panel8.Height = button5.Height;

panel8.Top = button5.Top;

Account ob = new Account();

ob.ShowDialog();

}

private void AvaliableB\_Click(object sender, EventArgs e)

{

AvailableProuct ap = new AvailableProuct();

ap.ShowDialog();

}

private void MaintenanceB\_Click(object sender, EventArgs e)

{

AvailableProduct ma = new AvailableProduct();

ma.ShowDialog();

}

private void ImportB\_Click(object sender, EventArgs e)

{

ImportPro ip = new ImportPro();

ip.ShowDialog();

}

private void SaleB\_Click(object sender, EventArgs e)

{

Sale sa = new Sale();

sa.ShowDialog();

}

private void ReportIB\_Click(object sender, EventArgs e)

{

ReportImport rp = new ReportImport();

rp.ShowDialog();

}

private void ReportSB\_Click(object sender, EventArgs e)

{

reportSale rs = new reportSale();

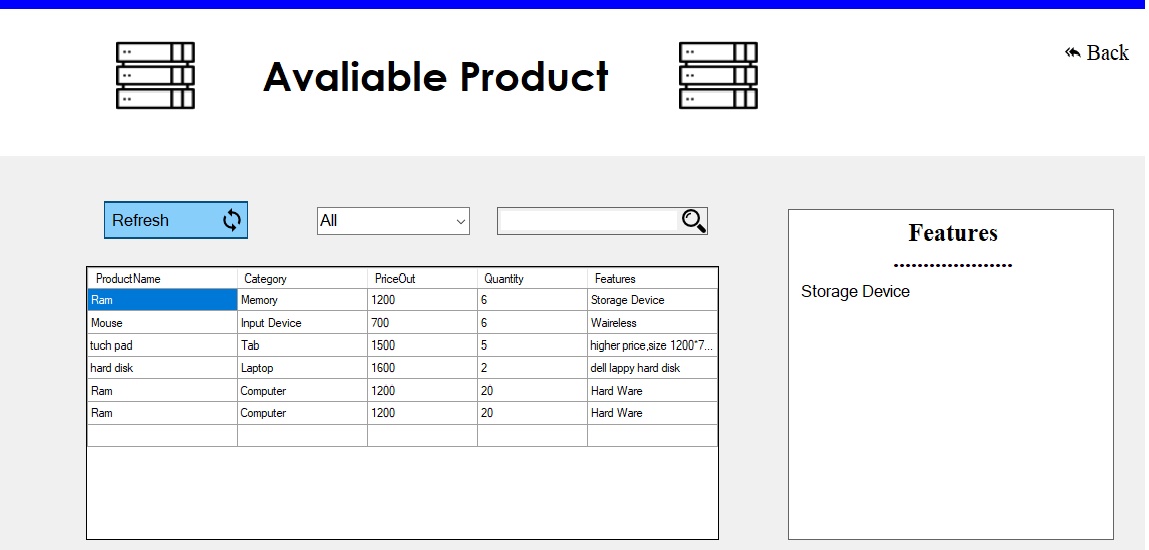
rs.ShowDialog();

}

}

}

***Available Stock:***

******

***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CRM

{

public partial class AvailableProuct : Form

{

SqlConnection cn;

SqlCommand cm;

public AvailableProuct()

{

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

InitializeComponent();

}

private void AvailableProuct\_Load(object sender, EventArgs e)

{

grid();

panel2.Show();

comboBox1.SelectedIndex=0;

}

public void grid()

{

try

{

cn.Open();

cm = new SqlCommand("select \* from ImportProduct ", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "temp");

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

comboBox1.SelectedIndex = 0;

textBox1.Text = "";

grid();

}

public void select(string str)

{

if (comboBox1.Text == "All")

{

grid();

}

else

{

Try

{

cn.Open();

cm = new SqlCommand("select \* from ImportProduct where " + comboBox1.Text + "='" + str + "' ", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "temp");

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

string str = textBox1.Text;

select(str);

}

private void dataGridView1\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("Select \* from importProduct where ProductName='" + dataGridView1.CurrentRow.Cells[0].Value + "'", cn);

SqlDataReader dr = cm.ExecuteReader();

dr.Read();

textBox2.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

string str = textBox1.Text;

select(str);

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

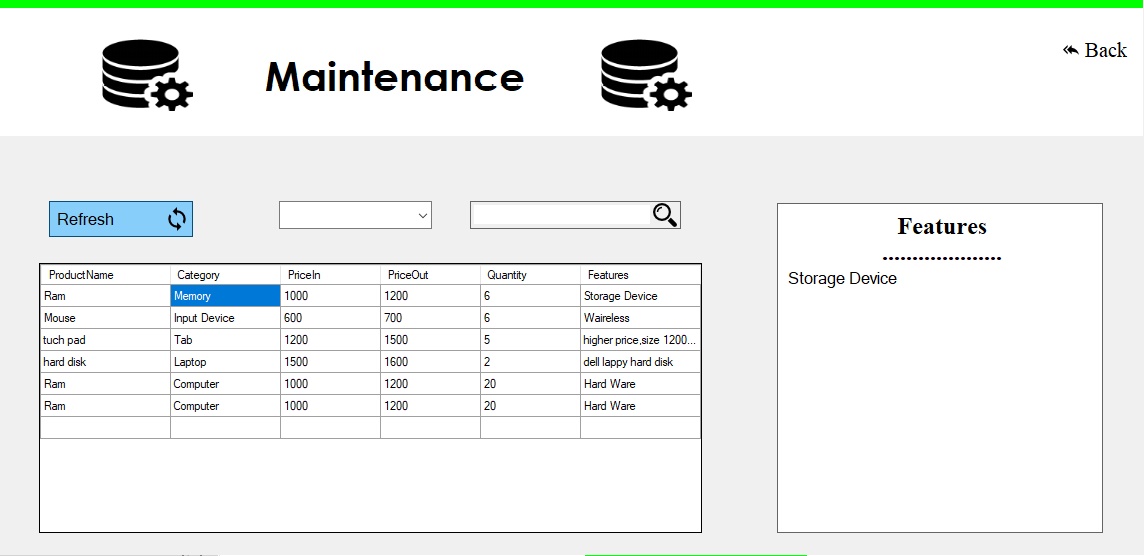
{

}

}

}

***Edit Stock Details:***



***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.IO;

namespace CRM

{

public partial class AvailableProduct : Form

{

public static string pn, cat, pi,po, ft;

SqlCommand cm;

SqlConnection cn;

public AvailableProduct()

{

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

InitializeComponent();

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void AvailableProduct\_Load(object sender, EventArgs e)

{grid();

}

public void grid()

{

try

{

cn.Open();

cm = new SqlCommand("select \* from ImportProduct", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "temp");

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

string str = textBox1.Text;

select(str);

}

public void select(string str)

{

if (comboBox1.Text == "All")

{

grid();

}

else

{

try

{

cn.Open();

cm = new SqlCommand("select \* from ImportProduct where " + comboBox1.Text + "='" + str + "' ", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "temp");

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

string str = textBox1.Text;

select(str);

}

private void dataGridView1\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("Select \* from importProduct where ProductName='" + dataGridView1.CurrentRow.Cells[0].Value + "'", cn);

SqlDataReader dr = cm.ExecuteReader();

dr.Read();

pn = dataGridView1.CurrentRow.Cells[0].Value.ToString();

cat = dataGridView1.CurrentRow.Cells[1].Value.ToString();

pi = dataGridView1.CurrentRow.Cells[2].Value.ToString();

po = dataGridView1.CurrentRow.Cells[3].Value.ToString();

ft = dataGridView1.CurrentRow.Cells[5].Value.ToString();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

EditProduct ed = new EditProduct();

ed.Show();

}

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("Select \* from importProduct where ProductName='" + dataGridView1.CurrentRow.Cells[0].Value + "'", cn);

SqlDataReader dr = cm.ExecuteReader();

dr.Read();

textBox2.Text = dataGridView1.CurrentRow.Cells[5].Value.ToString();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void button1\_Click(object sender, EventArgs e)

{

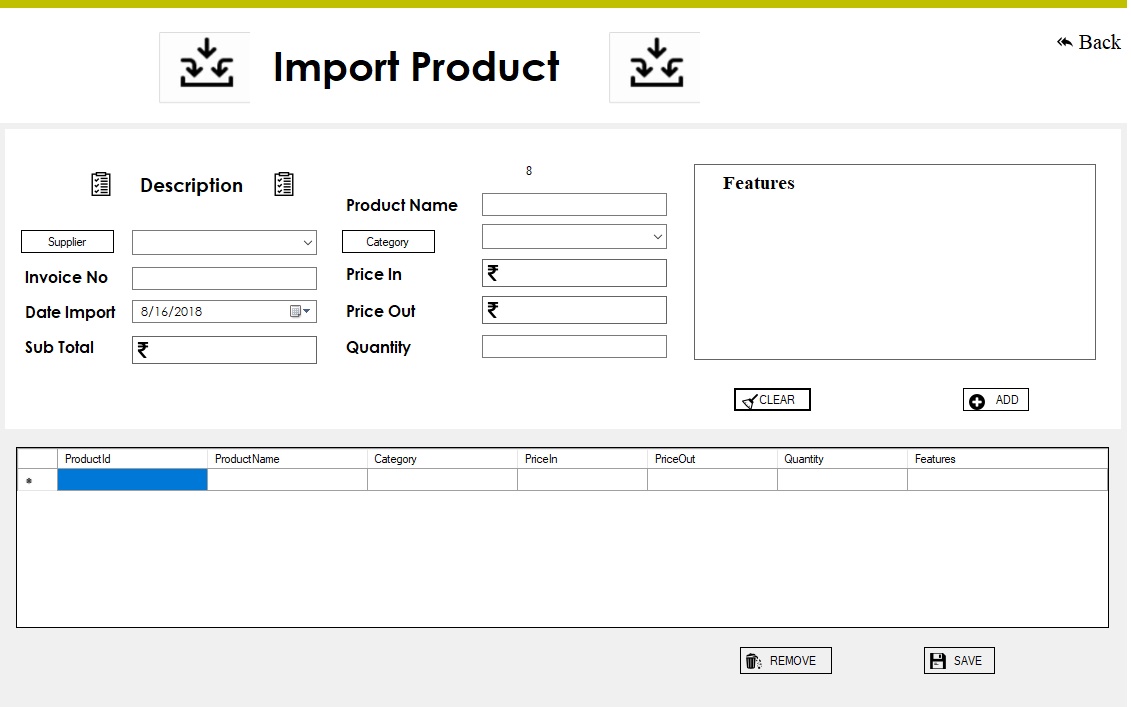
grid();

}

}

}

***Import Product:***



***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.IO;

namespace CRM

{

public partial class ImportPro : Form

{ int i = 1;

SqlConnection cn;

SqlCommand cm;

public ImportPro()

{

InitializeComponent();

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void label12\_Click(object sender, EventArgs e)

{

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void ImportPro\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'cRMDataSet4.ImportProduct' table. You can move, or remove it, as needed.

this.importProductTableAdapter.Fill(this.cRMDataSet4.ImportProduct);

ComboLoad();

}

public void ComboLoad()

{

try

{

cn.Open();

cm= new SqlCommand("Select \* from Supplier ",cn);

SqlDataReader dr = cm.ExecuteReader();

while(dr.Read())

{

comboBox1.Items.Add(dr[0].ToString());

}

}catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

try

{

cn.Open();

cm = new SqlCommand("Select \* from Category ", cn);

SqlDataReader dr = cm.ExecuteReader();

while (dr.Read())

{

comboBox3.Items.Add(dr[1].ToString());

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

try

{

cn.Open();

cm = new SqlCommand("Select Max(ProductId) from ImportProduct ", cn);

SqlDataReader dr = cm.ExecuteReader();

while (dr.Read())

{

int i =Convert.ToInt32( dr[0].ToString());

i++;

label13.Text = i.ToString();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void button4\_Click(object sender, EventArgs e)

{

Citizen\_Customer sp = new Citizen\_Customer();

sp.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

}

private void pictureBox5\_Click(object sender, EventArgs e)

{

}

private void panel2\_Paint(object sender, PaintEventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

category ct = new category();

ct.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("insert into ImportProduct(Supplier,InvoiceNo,DateImport,SubTotal,ProductName,category,PriceIn,PriceOut,Quantity,Features) values(@sup,@ino,@dimp,@st,@pn,@ctg,@pin,@pout,@qty,@feat)", cn);

cm.Parameters.AddWithValue("@sup",comboBox1.Text);

cm.Parameters.AddWithValue("@ino",textBox2.Text);

cm.Parameters.AddWithValue("@dimp",dateTimePicker1.Value.Date);

cm.Parameters.AddWithValue("@pn", textBox8.Text);

cm.Parameters.AddWithValue("@ctg", comboBox3.Text);

// cm.Parameters.AddWithValue("@pid", label13.Text);

cm.Parameters.AddWithValue("@st",textBox1.Text);

cm.Parameters.AddWithValue("@pin",textBox3.Text);

cm.Parameters.AddWithValue("@pout",textBox5.Text);

cm.Parameters.AddWithValue("@qty",textBox6.Text);

cm.Parameters.AddWithValue("@feat",textBox7.Text);

int re= cm.ExecuteNonQuery();

if(re>0)

{

MessageBox.Show("Record Added");

cn.Close();

GridCall();

}else

{

MessageBox.Show("Record Not Added");

}

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}finally

{

cn.Close();

}

}

public void GridCall()

{

try

{

cn.Open();

cm = new SqlCommand("select \* from ImportProduct where ProductId='" + label13.Text + "'", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "Temp");

dataGridView1.DataSource = ds.Tables[0];

}catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void comboBox1\_MouseDoubleClick(object sender, MouseEventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("Select \* from Supplier ", cn);

SqlDataReader dr = cm.ExecuteReader();

while (dr.Read())

{

comboBox1.Items.Add(dr[0].ToString());

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void comboBox3\_MouseDoubleClick(object sender, MouseEventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("Select \* from Category ", cn);

SqlDataReader dr = cm.ExecuteReader();

while (dr.Read())

{

comboBox3.Items.Add(dr[1].ToString());

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void button6\_Click(object sender, EventArgs e)

{

a = Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value);

try

{

cn.Open();

cm = new SqlCommand("Delete from ImportProduct where ProductId='" + a + "'",cn);

int re = cm.ExecuteNonQuery();

if (re > 0)

{

MessageBox.Show("Product is Deleted.....");

}

else

{

MessageBox.Show("Product is not Deleted.....");

}

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

this.importProductTableAdapter.Fill(this.cRMDataSet4.ImportProduct);

cn.Close();

}

}

private void button2\_Click(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

textBox7.Clear();

textBox8.Clear();

comboBox1.Text = "";

comboBox3.Text = "";

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

public int a;

private void dataGridView1\_CellContentDoubleClick(object sender, DataGridViewCellEventArgs e)

{

a = Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value);

try

{

MessageBox.Show(a.ToString());

cn.Open();

cm = new SqlCommand("select \* from ImportProduct where ProductId='" + a + "'", cn);

SqlDataReader dr = cm.ExecuteReader();

dr.Read();

if(dr.HasRows)

{

label13.Text = dr[0].ToString();

comboBox1.Text = dr[1].ToString();

textBox2.Text = dr[2].ToString();

dateTimePicker1.Value = Convert.ToDateTime(dr[3]);

textBox1.Text = dr[4].ToString();

textBox8.Text = dr[5].ToString();

comboBox3.Text = dr[6].ToString();

textBox3.Text = dr[7].ToString();

textBox5.Text=dr[8].ToString();

textBox6.Text = dr[9].ToString();

textBox7.Text = dr[10].ToString();

}

else

{

MessageBox.Show("data is not found....");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

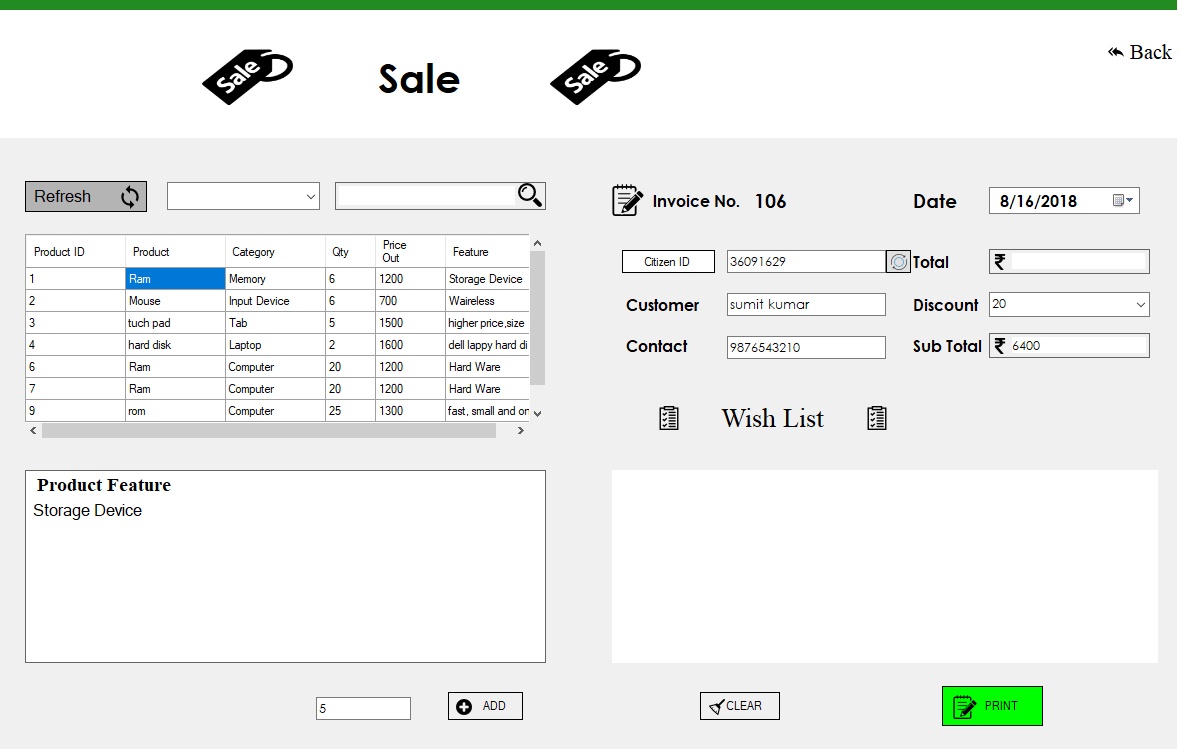
}

}

}

}

***Sale Detail :***

******

***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CRM

{

public partial class Sale : Form

{

public static string inv;

SqlConnection cn;

SqlCommand cm;

int qty;

int pid;

int pout;

int total;

int aqty;

public Sale()

{

InitializeComponent();

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

}

public void invoice()

{

try

{

cn.Open();

cm = new SqlCommand("Select Max(InvoiceNo) from sale ", cn);

SqlDataReader dr = cm.ExecuteReader();

dr.Read();

int i = Convert.ToInt32(dr[0].ToString());

i++;

label4.Text = i.ToString();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void Sale\_Load(object sender, EventArgs e)

{

invoice();

// TODO: This line of code loads data into the 'saleDataSet.ImportProduct' table. You can move, or remove it, as needed.

this.importProductTableAdapter.Fill(this.saleDataSet.ImportProduct);

textBox4.Visible = false;

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button7\_Click(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

button3.Visible = false;

textBox4.Visible = true;

}

private void button4\_Click(object sender, EventArgs e)

{

Customer cc = new Customer();

cc.ShowDialog();

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("select \* from ImportProduct where ProductId='" + dataGridView1.CurrentRow.Cells[0].Value + "'", cn);

SqlDataReader dr = cm.ExecuteReader();

dr.Read();

textBox2.Text = dr[10].ToString();

qty = Convert.ToInt32(dr[9]);

pout = Convert.ToInt32(dr[8]);

pid = Convert.ToInt32(dr[0]);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void button6\_Click(object sender, EventArgs e)

{

try

{

int qty1 = Convert.ToInt32(textBox4.Text);

total = qty1 \* pout;

aqty = qty - qty1;

if (qty1 < qty)

{

cn.Open();

cm = new SqlCommand("Select ProductId,ProductName,Category,PriceOut,Features from ImportProduct where ProductId='" + pid + "'", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "Temp");

dataGridView2.DataSource = ds.Tables[0];

cn.Close();

textBox7.Text = total.ToString();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

int dsc = Convert.ToInt32(comboBox2.Text);

textBox10.Text = (total - (total \* dsc / 100)).ToString();

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

}

private void button5\_Click(object sender, EventArgs e)

{

invoice();

try

{

cn.Open();

cm = new SqlCommand("insert into sale(Date,CitizenID,Customer,Contact,Total,discount,SubTotal,Quantity,ProductId) values(@date,@cid,@customer,@Contact,@total,@dsc,@stotal,@qty,@pid)", cn);

//cm.Parameters.AddWithValue("@ino", label4.Text);

cm.Parameters.AddWithValue("@date", dateTimePicker1.Value.Date);

cm.Parameters.AddWithValue("@cid", textBox3.Text);

cm.Parameters.AddWithValue("@customer", textBox6.Text);

cm.Parameters.AddWithValue("@Contact", textBox8.Text);

cm.Parameters.AddWithValue("@total", textBox7.Text);

cm.Parameters.AddWithValue("@dsc", comboBox2.Text);

cm.Parameters.AddWithValue("@stotal", textBox10.Text);

cm.Parameters.AddWithValue("@qty", textBox4.Text);

string pide = pid.ToString();

cm.Parameters.AddWithValue("@pid", pide);

// cm.Parameters.AddWithValue("@staff", textBox7.Text);

int re = cm.ExecuteNonQuery();

if (re > 0)

{

MessageBox.Show("Record Added");

}

else

{

MessageBox.Show("Record Not Added");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

inv= label4.Text.ToString();

SReport ob = new SReport();

ob.ShowDialog();

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

}

private void button8\_Click(object sender, EventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("select \* from Customer where Citizenid='" + textBox3.Text + "'", cn);

SqlDataReader dr = cm.ExecuteReader();

dr.Read();

string str = dr[3].ToString();

if (str == "yes")

{

MessageBox.Show("Citizen Is BlackListed");

textBox6.Text = dr[0].ToString();

textBox8.Text = dr[2].ToString();

}

else

{

textBox6.Text = dr[0].ToString();

textBox8.Text = dr[2].ToString();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void button2\_Click(object sender, EventArgs e)

{

foreach(Control x in this.Controls)

{

if(x is TextBox)

{

x.Text = "";

}

}

textBox2.Text = "";

textBox7.Text = "";

textBox10.Text = "";

comboBox2.Text = "";

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

string str = textBox1.Text;

select(str);

}

public void select(string str)

{

if (comboBox1.Text == "All")

{

grid();

}

else

{

try

{

cn.Open();

cm = new SqlCommand("Select ProductId,ProductName,Category,PriceOut,Features From ImportProduct where " + comboBox1.Text + "='" + str + "' ", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "temp");

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

} public void grid()

{

try

{

cn.Open();

cm = new SqlCommand("Select ProductId,ProductName,Category,Quantity,PriceOut,Features from ImportProduct", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "temp");

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void button1\_Click(object sender, EventArgs e)

{

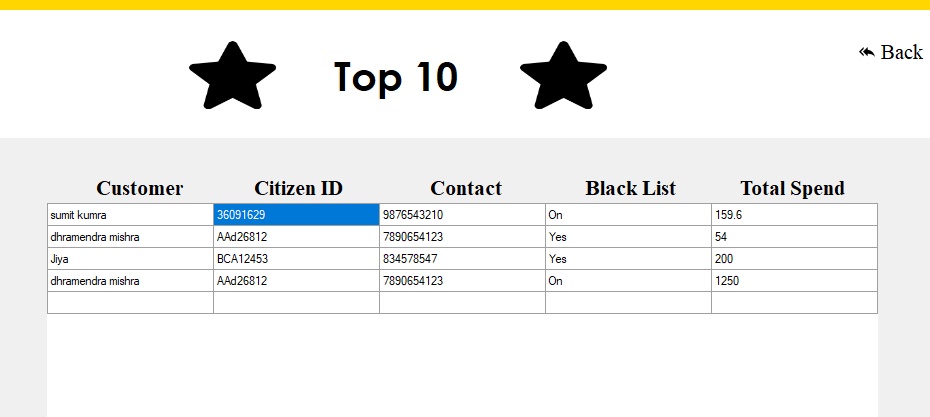
grid();

}

}

}

***Top 10 Customers:***

******

***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CRM

{

public partial class top10 : Form

{

SqlCommand cm;

SqlConnection cn;

public top10()

{

InitializeComponent();

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void top10\_Load(object sender, EventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("Select \* from Customer Where TotalSpend>='50' ", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "Temp");

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{ cn.Close(); }

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void panel1\_Paint(object sender, PaintEventArgs e)

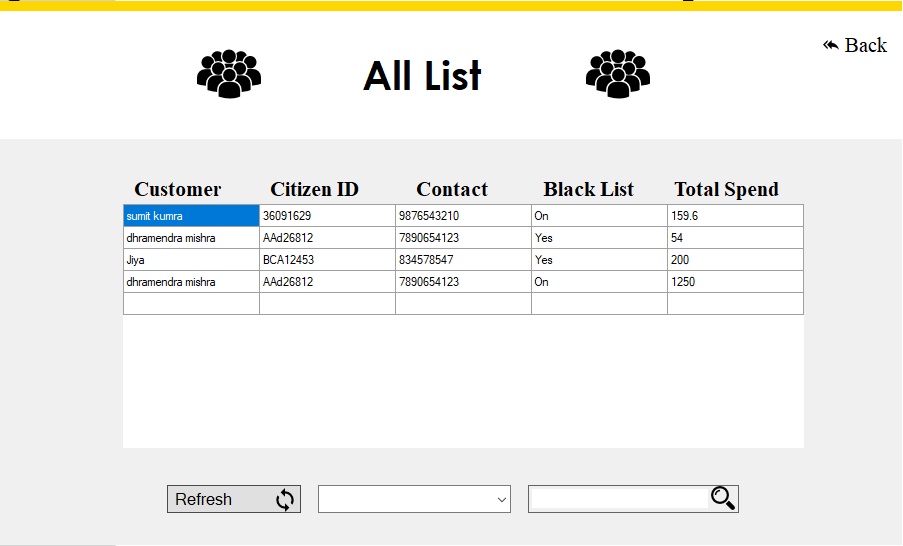
{

}

}

}

***All Customer List :***

******

***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CRM

{

public partial class All : Form

{

SqlConnection cn;

SqlCommand cm;

public All()

{

InitializeComponent();

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

}

private void All\_Load(object sender, EventArgs e)

{

grid();

}

public void grid()

{

try

{

cn.Open();

cm = new SqlCommand("Select \* From Customer ", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "temp");

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

string str = textBox1.Text;

select(str);

}

public void select(string str)

{

if (comboBox1.Text == "All")

{

grid();

}

else

{

try

{

cn.Open();

cm = new SqlCommand("select \* from Customer where " + comboBox1.Text + "='" + str + "' ", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "temp");

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

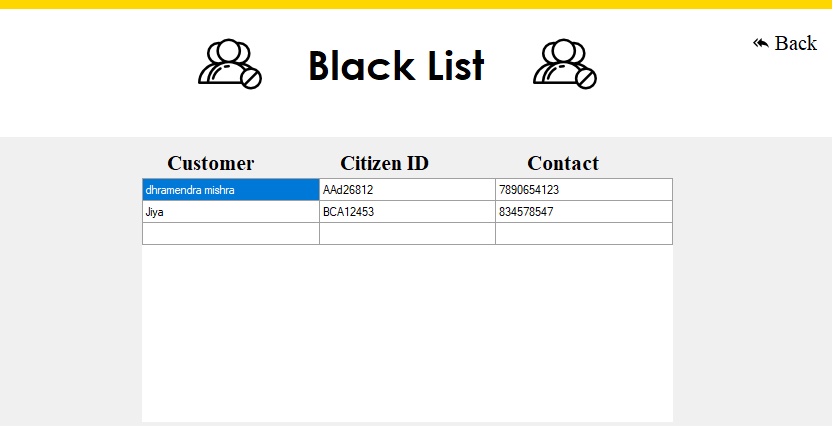
grid();

}

}

}

***Black Listed Customers:***

******

***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CRM

{

public partial class BlackList : Form

{

SqlCommand cm;

SqlConnection cn;

public BlackList()

{

InitializeComponent();

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void BlackList\_Load(object sender, EventArgs e)

{

string str= "Yes";

try

{

cn.Open();

cm = new SqlCommand("Select Customer , CitizenId, Contact from Customer Where BlackList='" + str + "'", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "Temp");

dataGridView1.DataSource = ds.Tables[0];

}catch(Exception ex)

{

MessageBox.Show(ex.Message);

}finally

{

cn.Close();

}

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

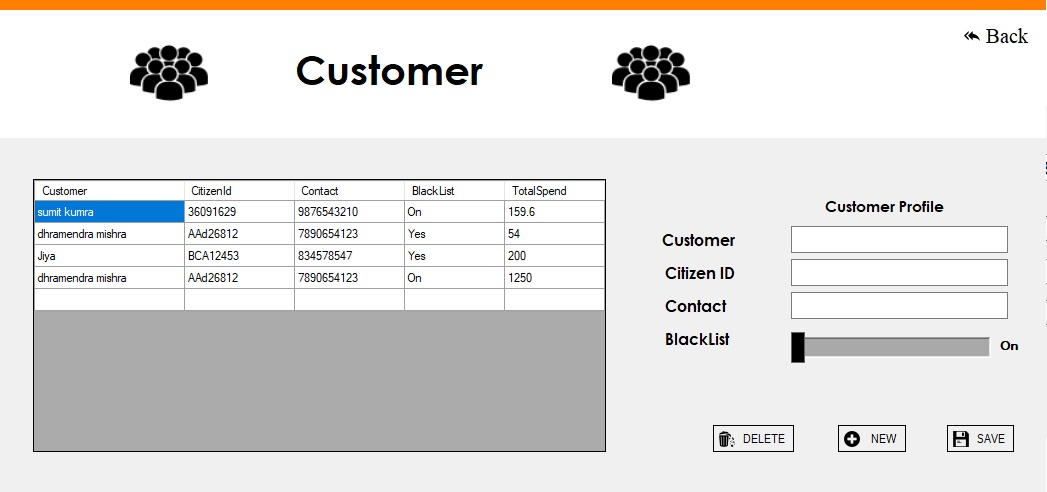
{

}

}

}

***Customers Detail:***

******

***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CRM

{

public partial class Customer : Form

{

SqlCommand cm;

SqlConnection cn;

int i = 1;

string a;

public Customer()

{

InitializeComponent();

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

}

private void Customer\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'cRMDataSet5.Customer' table. You can move, or remove it, as needed.

gridCall();

}

public void gridCall()

{

this.customerTableAdapter.Fill(this.cRMDataSet5.Customer);

}

private void panel2\_Click(object sender, EventArgs e)

{

if (i == 1)

{

panel4.Location = new Point(791,332);

label7.Visible = true;

label8.Visible = false;

panel2.BackColor = Color.DarkGray;

panel4.BackColor = Color.Black;

i = 0;

}

else

{

panel4.Location = new Point(977, 332);

label7.Visible = false;

label8.Visible = true;

panel2.BackColor = Color.Red;

panel4.BackColor = Color.Black;

i = 1;

}

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button5\_Click(object sender, EventArgs e)

{

}

private void button6\_Click(object sender, EventArgs e)

{

try{

cn.Open();

cm=new SqlCommand("delete from customer where CitizenId='"+textBox2.Text+"'",cn);

int re = cm.ExecuteNonQuery();

if (re > 0)

{

MessageBox.Show("Record is Deleted");

cn.Close();

gridCall();

}

else

MessageBox.Show("Record is Not Deleted ");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void button3\_Click(object sender, EventArgs e)

{

try

{

cn.Open();

if (i == 1)

{

a= label8.Text;

}

else

{

a= label7.Text;

}

cm=new SqlCommand("insert into Customer(Customer,CitizenId,Contact,BlackList)values(@cust,@citId,@con,@BList)",cn);

cm.Parameters.AddWithValue("@cust", textBox1.Text);

cm.Parameters.AddWithValue("@citId", textBox2.Text);

cm.Parameters.AddWithValue("@con", textBox3.Text);

cm.Parameters.AddWithValue("@BList", a);

int re = cm.ExecuteNonQuery();

if (re > 0)

{

MessageBox.Show("Record Added");

cn.Close();

gridCall();

}

else

MessageBox.Show("Record Not Added");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void dataGridView1\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

textBox1.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();

textBox2.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();

textBox3.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();

int i = 0;

string str = dataGridView1.CurrentRow.Cells[3].Value.ToString();

if (str == "On")

i = 0;

else

i = 1;

if (i == 1)

{

panel4.Location = new Point(791, 332);

label7.Visible = true;

label8.Visible = false;

panel2.BackColor = Color.DarkGray;

panel4.BackColor = Color.Black;

i = 0;

}

else

{

panel4.Location = new Point(977, 332);

label7.Visible = false;

label8.Visible = true;

panel2.BackColor = Color.Red;

panel4.BackColor = Color.Black;

i = 1;

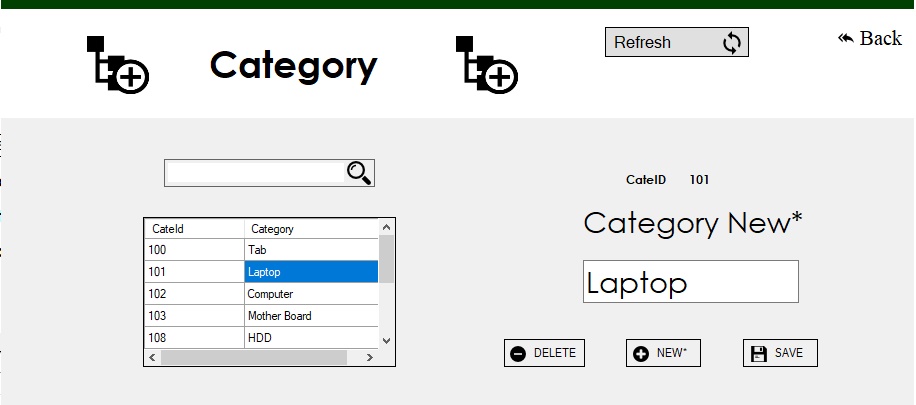
}

}

}

}

***Category :***

******

***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CRM

{

public partial class category : Form

{

SqlCommand cm;

SqlConnection cn;

public category()

{

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

InitializeComponent();

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void category\_Load(object sender, EventArgs e)

{

grid();

cateId();

}

public void cateId()

{

try

{

cn.Open();

cm = new SqlCommand("Select Max(cateId) from category ", cn);

SqlDataReader dr = cm.ExecuteReader();

while (dr.Read())

{

int i = Convert.ToInt32(dr[0].ToString());

i++;

label4.Text = i.ToString();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void button5\_Click(object sender, EventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("insert into category values(@cate)", cn);

cm.Parameters.AddWithValue("@cate",textBox2.Text);

int re = cm.ExecuteNonQuery();

if (re > 0)

{

MessageBox.Show("Record Added");

cn.Close();

cateId();

}

else

{

MessageBox.Show("Record Not Added");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

textBox2.Text = "";

cn.Close();

}

}

public void grid()

{

this.categoryTableAdapter.Fill(this.categoryDataSet6.Category);

}

private void button1\_Click(object sender, EventArgs e)

{

cateId();

grid();

textBox1.Clear();

textBox2.Clear();

}

private void button6\_Click(object sender, EventArgs e)

{

cateId();

grid();

textBox1.Clear();

textBox2.Clear();

}

private void button7\_Click(object sender, EventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("delete from Category where CateId= '" + dataGridView1.CurrentRow.Cells[0].Value.ToString() + "'", cn);

int ree = cm.ExecuteNonQuery();

if (ree > 0)

{

MessageBox.Show("Category is deleted from dataBase......");

}

else

{

MessageBox.Show("Category is NOT deleted......");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

grid();

}

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("Select \* from Supplier where Supplier='" + dataGridView1.CurrentRow.Cells[0].Value + "'", cn);

SqlDataReader dr = cm.ExecuteReader();

dr.Read();

textBox2.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();

label4.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

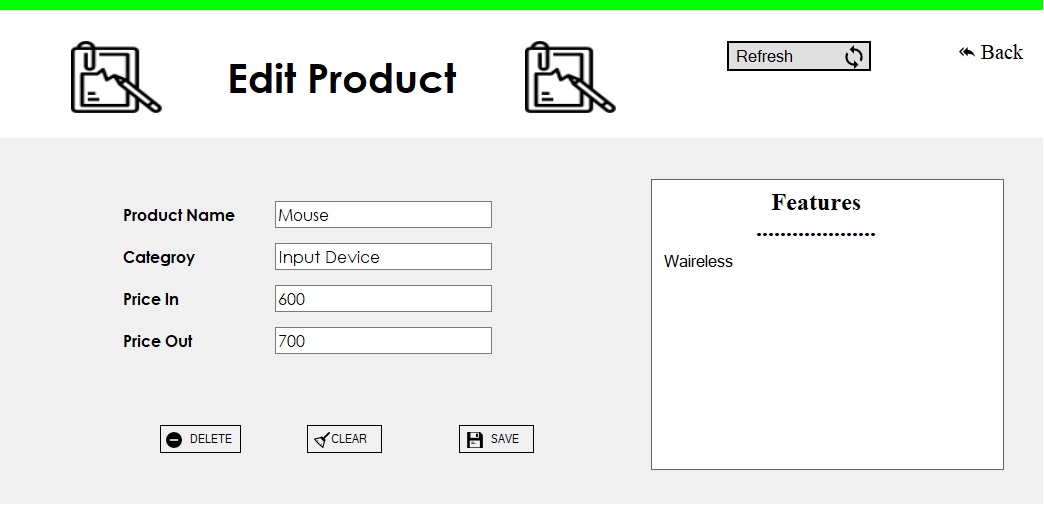
}

}

}

}

***Edit Product Details:***

******

***Code for the above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.IO;

namespace CRM

{

public partial class EditProduct : Form

{

SqlCommand cm;

SqlConnection cn;

public EditProduct()

{

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

textBox2.Text = AvailableProduct.pn;

textBox3.Text = AvailableProduct.cat;

textBox4.Text = AvailableProduct.pi;

textBox5.Text = AvailableProduct.po;

textBox1.Text = AvailableProduct.ft;

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button5\_Click(object sender, EventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("Update ImportProduct set ProductName=@pn,Category=@cat,PriceIn=@pi,PriceOut=@po,Features=@ft where ProductName=@pname and Category=@cate and PriceIn=@pin and PriceOut=@pout and Features=@fet", cn);

cm.Parameters.AddWithValue("@pin",AvailableProduct.pi);

cm.Parameters.AddWithValue("@pout", AvailableProduct.po);

cm.Parameters.AddWithValue("@fet", AvailableProduct.ft);

cm.Parameters.AddWithValue("@pname", AvailableProduct.pn);

cm.Parameters.AddWithValue("@cate", AvailableProduct.cat);

cm.Parameters.AddWithValue("@pn", textBox2.Text);

cm.Parameters.AddWithValue("@cat", textBox3.Text);

cm.Parameters.AddWithValue("@pi", textBox4.Text);

cm.Parameters.AddWithValue("@po", textBox5.Text);

cm.Parameters.AddWithValue("@ft", textBox1.Text);

int re = cm.ExecuteNonQuery();

if(re>0)

{

MessageBox.Show("Update successfull");

}else

{

MessageBox.Show("Update failur");

}

}catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void EditProduct\_Load(object sender, EventArgs e)

{

textBox2.Text = AvailableProduct.pn;

textBox3.Text = AvailableProduct.cat;

textBox4.Text = AvailableProduct.pi;

textBox5.Text = AvailableProduct.po;

textBox1.Text = AvailableProduct.ft;

}

private void button3\_Click(object sender, EventArgs e)

{

foreach (Control x in this.Controls)

{

if (x is TextBox)

{

x.Text = "";

}

textBox1.Text = "";

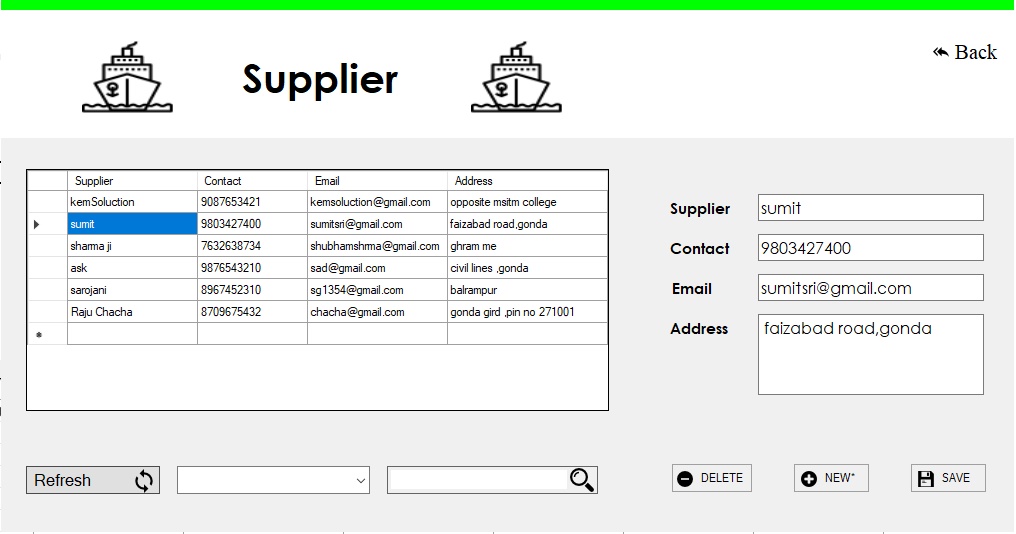
}

}

}

}

***Supplier Detail:***

******

***Code for the above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CRM

{

public partial class Citizen\_Customer : Form

{

SqlCommand cm;

SqlConnection cn;

public Citizen\_Customer()

{

InitializeComponent();

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

}

private void Supplier\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'supplierDataset.Supplier' table. You can move, or remove it, as needed.

grid();

}

public void grid()

{

this.supplierTableAdapter.Fill(this.supplierDataset.Supplier);

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button5\_Click(object sender, EventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("insert into supplier values(@sup,@con,@email,@add)", cn);

cm.Parameters.AddWithValue("@sup", textBox2.Text);

cm.Parameters.AddWithValue("@con", textBox3.Text);

cm.Parameters.AddWithValue("@email", textBox4.Text);

cm.Parameters.AddWithValue("@add", textBox5.Text);

int re = cm.ExecuteNonQuery();

if (re > 0)

{

MessageBox.Show("Record Added");

}

else

{

MessageBox.Show("Record Not Added");

}

}catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

grid();

}

}

private void button1\_Click(object sender, EventArgs e)

{

grid();

textBox1.Clear();

comboBox1.Text = "";

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

}

private void button6\_Click(object sender, EventArgs e)

{

textBox1.Clear();

comboBox1.Text = "";

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

}

private void button7\_Click(object sender, EventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("delete from Supplier where Email='" + dataGridView1.CurrentRow.Cells[2].Value.ToString() + "'", cn);

int ree = cm.ExecuteNonQuery();

if(ree>0)

{

MessageBox.Show("Supplier is deleted from dataBase......");

}

else

{

MessageBox.Show("Supplier is NOT deleted......");

}

}catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

grid();

}

}

public void select(string str)

{

if (comboBox1.Text == "All")

{

grid();

}

else

{

try

{

cn.Open();

cm = new SqlCommand("select \* from Supplier where " + comboBox1.Text + "='" + str + "' ", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "temp");

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

string str = textBox1.Text;

select(str);

}

private void textBox1\_Enter(object sender, EventArgs e)

{

string str = textBox1.Text;

select(str);

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("Select \* from Supplier where Supplier='" + dataGridView1.CurrentRow.Cells[0].Value + "'", cn);

SqlDataReader dr = cm.ExecuteReader();

dr.Read();

textBox2.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();

textBox3.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();

textBox4.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();

textBox5.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

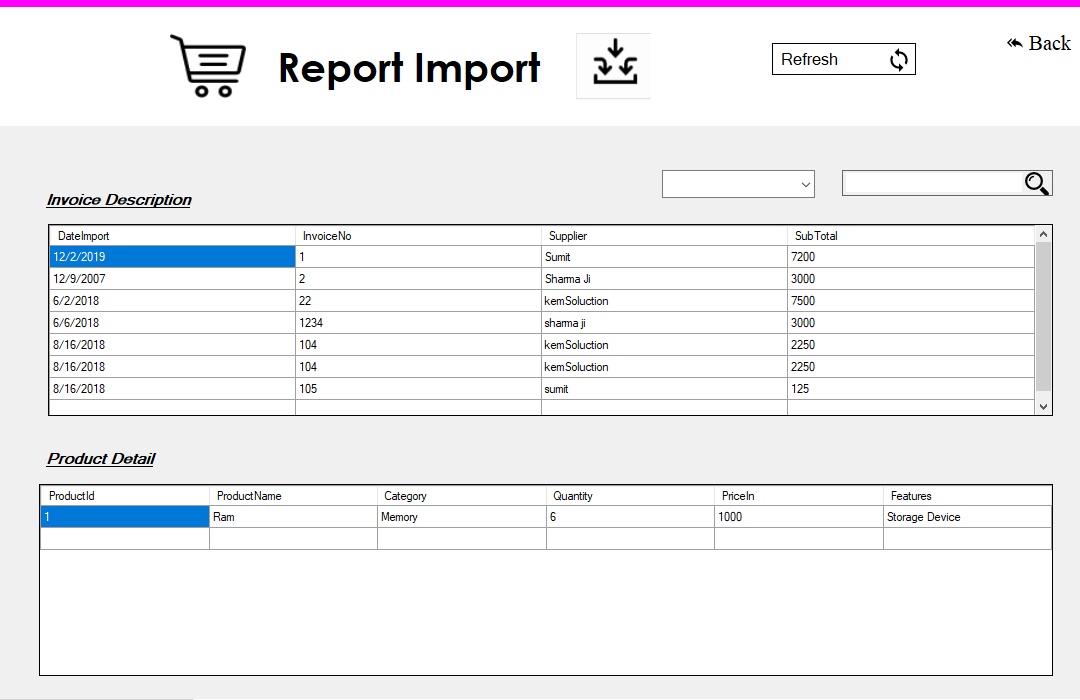
}

}

}

}

***Import Report:***

******

***Code for the above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CRM

{

public partial class ReportImport : Form

{

SqlConnection cn;

SqlCommand cm;

public ReportImport()

{

InitializeComponent();

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

}

public void grid()

{

try

{

cn.Open();

cm = new SqlCommand("select DateImport,InvoiceNo,Supplier,SubTotal from ImportProduct ", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "temp");

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

string str = textBox1.Text;

select(str);

}

public void select(string str)

{

if (comboBox1.Text == "All")

{

grid();

}

else

{

try

{

cn.Open();

cm = new SqlCommand("select DateImport,InvoiceNo,Supplier,SubTotal from ImportProduct where " + comboBox1.Text + "='" + str + "' ", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "temp");

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

string str = textBox1.Text;

select(str);

}

private void dataGridView2\_CellClick(object sender, DataGridViewCellEventArgs e)

{

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

try

{

cn.Open();

cm = new SqlCommand("select ProductId,ProductName,Category,Quantity,PriceIn,Features from ImportProduct where InvoiceNo='" + dataGridView1.CurrentRow.Cells[1].Value.ToString() + "'", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "Fil");

dataGridView2.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void ReportImport\_Load(object sender, EventArgs e)

{

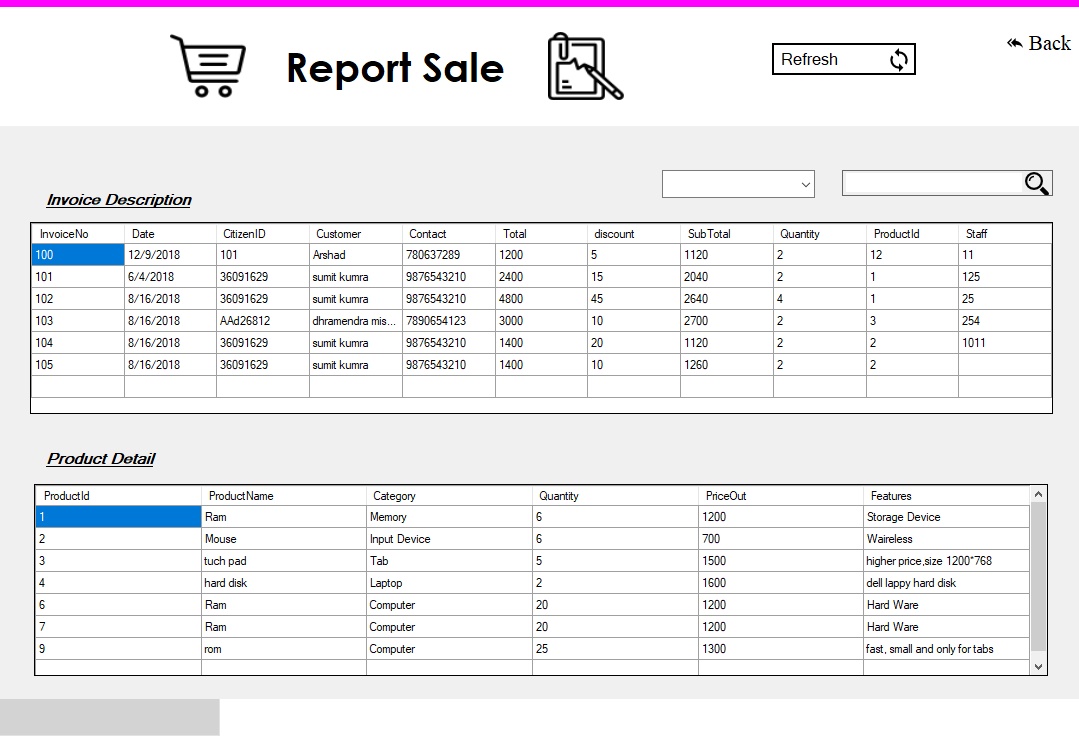
grid();

}

}

}

***Sale Report:***

******

***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace CRM

{

public partial class reportSale : Form

{

SqlConnection cn;

SqlCommand cm;

public reportSale()

{

InitializeComponent();

cn = new SqlConnection(@"Data Source=DESKTOP-D4GIESE\SQLEXPRESS;Initial Catalog=CRM;Integrated Security=True");

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

}

public void grid()

{

try

{

cn.Open();

cm = new SqlCommand("select \* from sale ",cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "Temp");

dataGridView1.DataSource = ds.Tables[0];

}catch(Exception ex)

{

MessageBox.Show(ex.Message);

}finally

{

cn.Close();

}

try

{

cn.Open();

cm = new SqlCommand("select ProductId,ProductName,Category,Quantity,PriceOut,Features from ImportProduct ", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "Temp");

dataGridView2.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

private void reportSale\_Load(object sender, EventArgs e)

{

grid();

}

public void select(string str)

{

if (comboBox1.Text == "All")

{

grid();

}

else

{

try

{

cn.Open();

cm = new SqlCommand("select \* from ImportProduct where " + comboBox1.Text + "='" + str + "' ", cn);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataSet ds = new DataSet();

da.Fill(ds, "temp");

dataGridView2.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

cn.Close();

}

}

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

string str = textBox1.Text;

select(str);

}

private void button1\_Click(object sender, EventArgs e)

{

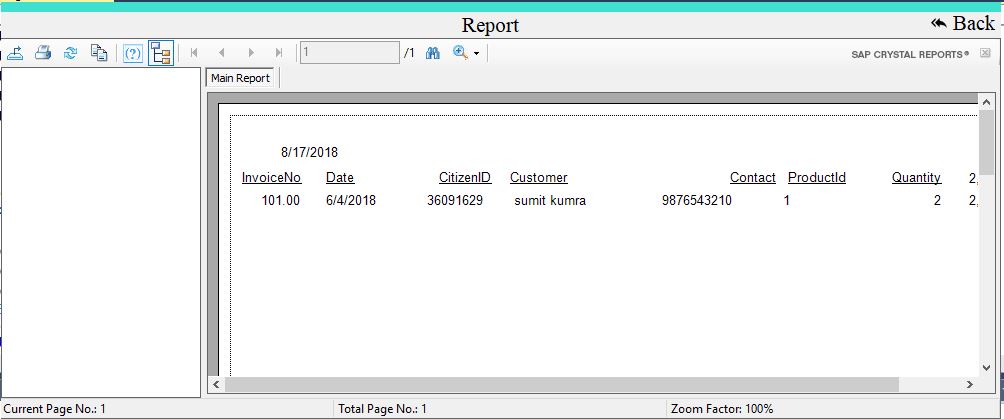
grid();

}

}

}

***Report:***

******

***Code for above Form:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CRM

{

public partial class SReport : Form

{

public SReport()

{

InitializeComponent();

}

private void SReport\_Load(object sender, EventArgs e)

{

MessageBox.Show("your Invoice Number Is = " + Sale.inv);

}

private void label9\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

***CONCLUSION***

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in C#.net and no some extent Windows Application and SQL Server, but also about all handling procedure related with **“Hotel MANAGEMENT SYSTEM”.** It also provides knowledge about the latest technology used in developing applications that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

**BENEFITS:**

The project is identified by the merits of the system offered to the user. The merits of this project are as follows: -

* This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
* The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updating so that the user cannot enter the invalid data, which can create problems at later date.
* Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover there is restriction for his that he cannot change the primary data field. This keeps the validity of the data to longer extent.
* User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him.
* From every part of the project the user is provided with the links through framing so that he can go from one option of the project to other as per the requirement. This is bound to be simple and very friendly as per the user is concerned. That is, we can sat that the project is user friendly which is one of the primary concerns of any good project.
* Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
* Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time then manual system.
* Allocating of sample results becomes much faster because at a time the user can see the records of last years.
* Easier and faster data transfer through latest technology associated with the computer and communication.
* Through these features it will increase the efficiency, accuracy and transparency.

**LIMITATIONS:**

* The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
* Training for simple computer operations is necessary for the users working on the system.

***BIBLOGRAPHY AND REFERENCE***

**Websites**

* www.w3schools.com
* www.webdevelopersnotes.com
* www.tutorialpoint.com
* www.codeproject.com
* [www.codeguru.com](http://www.codeguru.com)
* [www.stackoverflow.com](http://www.stackoverflow.com)
* www.techGIG.com

**Useful Books**

* Black book using C#
* Head First C#
* C# in Depth
* Adaptive Code Via C#
* The C# Player’s Guide
* Murach’s C# 2012