**Stock Management System**

**PROJECT**

SUBMITTED IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE MASTERS OF COMPUTER APPLICATION (MCA)

Submitted By:

Name: Roll No.:

1: Tanveer Ahmad Roll No: - 24

2:ROHIT RollNO:-25

**MAY** 2018



PRESIDENCY COLLEGE BANGLORU

DEPARTMENT OF COMPUTER APLICATIONS (MCA)

Guided By: Guided By : Guided By:

Mr. Harish Naik (professor) Ms.sheetal Ms.Harshini A S

Signature Signature Signature

Supervised By:

Dr. Narayana Swamy

(H.O.D Deptt of Computer Applications)

Signature

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We are grateful to our project guide Mr. Harish Naik for the guidance, inspiration and constructive suggestions that helpful us in the preparation of this project. We also thank our colleagues who have helped in successful completion of the project.

Tanveer Ahmad

Rohit

CERTIFICATE

It is certify that the thesis entitled “**study of stock management**” which is being submitted by students namely Mr. Tanveer Ahmad Khan, Mr. Rohit for partial fulfilment of the requirements for the award of degree of MCA 2nd sem, from Presidency college Bangalore, under the supervision of Mr. Mr. Harish Naik Dept. of computer science.

It is further certified that Mr. Tanveer Ahmad khan, Rohit work under me for the period as per university rule and they have put the required attendance in the department of computer science(Presidency college Bangalore) during this period

Dr. Muddu Vinay

Principal

(Presidency college Bangloru)

Dr. Narayana Swamy

(H.O.D Computer Applications Department)

Table of CONTENTS

|  |  |  |
| --- | --- | --- |
| S.No. | PARTICULARS | Page No. |
| **1.** | ABSTRACT | 05 |
| **2.** | INTRODUCTION | 06 |
| **3.** | PROBLEMS IN CURRENT SYSTEM AND PROPOSED WORK | 07 |
| **4.** | FEASIBILITY STUDY | 08 |
| **5.** | TESTING | 10 |
| **6.** | DATA TABLES | 12 |
| **7.** | SCREEN SHOTS OF FORMS | 17 |
| **8.** | DATA FLOW DIAGRAM | 26 |
| **9.** | ENTITY RELATIONSHIP DIAGRAM | 30 |
| **10.** | CODING | 31 |
| **11.** | FUTURE SCOPE | 89 |
| **12.** | CONCLUSION | 90 |
| **13.** | BIBLIOGRAPHY | 91 |

ABSTRACT

In college we have different types of stock e.g.; maintaining records of library, transport, hostel etc. In the stock management system the purpose beyond this is to maintain the details of stock which is available in the organization. In the present system the stock is maintained in the files, which is the wastage of time, lengthy process, and requires a lot of manual intervention. In this project the working about the stock will be automated as well as computerized. In order to get a meaningful and desired results. The project is being developed in order to create a software which will solve the problems e.g., managing the stock. It is a simple system, which will provide following facilities, add category, add/remove stock, and search stock. This type of management will minimize mismanagements. The software will keep track of all the information about the stock in the organization, their cost, their complete detail and total number of stock available in the organization. The project stock management system will overcome all problems we are facing in today manual system we are using.

INTRODUCTION

The project entitled product stock Management is developed as part of the MCA 2nd sem Project for the fulfilment of the MCA degree. In every organization or in any institution the stock is a very costly. If any problem arises in arranging the stock it will be non acceptable thing. So an application is needed which will arrange the information about stock. Managing stock effectively is important for any business, because without enough stock, production and sales will grind to a halt; it is a software application maintaining the records, related to all the transactions occurring at the counter of desired place. In this stock Management project the main objective is to maintain the records in the system. The main goal of the application is to maintain the records of the stock. Managing stock effectively is important for any organization, because without enough stock all work will be dump. Stock control involves careful planning to ensure that the organization has sufficient stock of right quality available in the right time. Stock can mean different things and depends on the organization. It includes raw materials and components from suppliers, work in progress or part finished goods. In college we have different types of stock e.g.; maintaining records of library, transport, hostel etc. in the stock management system the purpose beyond this is to maintain the details of stock which is available in the organization. In the present system the stock is maintained in the files, which is the wastage of time, lengthy process, and requires a lot of persons. In this project the working about stock will be automated as well as computerized. In order to get a meaningful and desired result, the project is being developed in order to create a software which will solve the problems e.g., managing the stock. It is a simple system, which will provide following facilities, add category, add/remove stock, and search stock. This type of management will minimize mismanagements. The software will keep track of all the information about the stock in the organization, their cost, their complete detail and total number of stock available in the organization. The project stock management system will overcome all problems we are facing in today manual system we are using. The project stock Management System is a project in which we are discussing the problems that occurs in the organization during the working time.

PROBLEMS IN EXISTING SYSTEM:-

In the current system we have a lot of problems in managing the stock. The problems which we are facing in today’s manual system are defined below:-

* The transactions related to goods in, goods out and returns are maintained manually at present along with maintaining the accounts of the customers and the suppliers.
* The application should provide quick access to the records maintained and must reveal the important reviews about the business so that the growth can be easily compared and should provide with the various reports showing the related details so that the important decisions could be taken easily.
* The current system is not secure because the file system is not better system.
* The application is not user friendly.
* Difficulty in report generating**:** There is a problem in generating the reports.
* Manual control: All calculations to generate report are done manually so there is greater chance of errors.

PROPOSED WORK:-

Stock management system is a management information system (MIS) for retails, who want to keep track of their profit and stock levels without the need of complex setups.

Software features:-

* Stock management.
* Payment bill generation.
* Profit wise reports.
* Multi language support

All these are to be automated and an application is required to relate all of them relatively and logically so that the current system can be replaced and accepted without major changes and problems. All these are to be automated and an application is required to relate all of them relatively and logically so that the current system can be replaced and accepted without major changes and problems.

Feasibility Study

One of the main meanings of the feasibility study is possibility. Checking of different criteria for successful system is included in this feasibility study section. These criteria are cost, time, efficiency etc. all these factors play important role in achieving objective of system. That means the system should be such that it gives optimum performance at minimum cost, time and requirements. In feasibility study phase we had undergone through various Steps which are describe as under:

1. Identify the origin of the information at different level.

2. Identify the expectation of user from computerized System.

3. Analyze the drawback of existing system (manual) System.

The six types of feasibility study are,

* Operational Feasibility
* Technical Feasibility
* Schedule Feasibility
* Economical Feasibility
* Management Feasibility
* Social Feasibility

**OPERATIONAL FEASIBILITY**

At this level the designer will focus on the person who will actually operate the system. The system should be easy to operate by the system user. The user should be given operating manual about the system. In this system the receptionist will work at the operational level. He will able to use or operate the system because of easy understanding.

**TECHNICAL FEASIBILITY**

Technically the system configuration should be less complex. Here, for software system technical feasibility means technically it should be comfortable for further maintenance**.** I have used VB and MS-Access. So one person required who have knowledge of VB and MS-Access. The minimum hardware and software requirement for the system is listed below.

Pentium IV

64 MB RAM

10 GB HD

The owner should have license version of VB and MS-Access server.

**SCHEDULE FEASIBILTY**

Schedule feasibility study is a determination of whether a proposed system will be completed within given time. The time period of the system will be given to the system owner. In our system has been developed in three phases:

In First phase we will do analysis and design of the system. At the end of the first phase the design of the system will be given to the system owner. At the end of the second phase the working model of the system is given to the system owner. And at last, end of the third phase the system will be implemented in the organization. The full working system will be given after end of the three phases.

**ECONOMICAL FEASIBILITY**

According to the concept of economical study the system should be completed with minimum cost. This is the most important factors for any system. In this study the cost and benefits are considered. We will give you this system in less cost compared to others. Using this system manual work will be reduced. So you will get benefit of giving salary to the employees and also the paper cost will be deducted. By working with the system all the information or records you will keep easily. Whenever you want to find any record you can get the record easily. So the system is Economical for you in various ways.

**MANAGEMENT FEASIBILITY**

It is a determination whether the library management system will be acceptable to the management.

**SOCIAL FEASIBILITY**

Social feasibility is a determination of whether the new system will be acceptable to the people or not.

**TESTING**

Development of a complex client/server three tier application requires that a methodology be developed for more effective application or software testing and quality assurance. Testing is made to find errors in the application. The strategy adopted for the testing in my application is as below.

**UNIT LEVEL TESTING**

The aim of unit testing is to find errors in the functionality of the unit module. At unit level testing each transaction and master screens was checked for is correctness in accepting data, modification and detection of each field then record. Entering a large number of records checked the logic of screens. Check was done that data is saved in the respective tables. The code was checked for all events that no extra code or repeated codes are written. Check was done that proper comments are given wherever required.

**SYSTEM LEVEL TESTING**

After unit level testing the system was checked as a whole for its correctness. Check was done that each transaction interfaced properly with the other. Check was done that the business objects were working properly and the data is posted in the tables properly.

**IMPLEMENTATION DEATAILS**

Implementation of system includes all those activities that take place to convert the old system into a new one from the old system to the new. Mainly the implementation phase of system consists of

* Training of personnel
* Conversion
* Documentation

**TRAINING**

Training may be for systems operators and users. This is done with a view to providing hands-on experience with the new system with interactive systems; users can try out software directly. In fact training should include

* Overview of how the system functions.
* How it will affect their jobs
* How it will relate to current manual procedures.

In any system training should be given to the user for efficient use of the system. There are mainly four steps which should specifically take care of: User involvement in the equipment use, instruction to individuals in troubleshooting the systems and coming out unscathed from troubles, data maintenance. Each and every system is implemented after passing many stages like system analysis, design, testing, documentation etc, successfully. All the instructions which are prepared for the system implementation are arranged in specific order in systems documentation and then according to it and requirement the main stage of system, implementation is taking placed by the system designer.

**CONVERSION**

It is the process of changing from the old system to the new one. There are four methods of handling a system conversion, which are the parallel-systems method, the dual system method or phase- method, the direct cutover method and last is the pilot approach. Here, for this we have used the direct cutover method, because in this method the conversion takes place abruptly. The drawback of this method is that it requires careful planning and training sessions must be scheduled and maintained.

**DOCUMENTATION**

Anything that is written about how a system is designed or functions is documentation. In documentation there are many types such as system documentation, programming documentation, operations documentation, training documentation, implementation documentation and appendix. From those types of documentation system documentation describes the overall system design and includes flowchart, I/O formats, file descriptions, control requirements and report specification. Programming documentation includes programming specifications, descriptions of program logic including graphic aids such as flowcharts. Operating documentation is deal with operating schemes and problems created in the system. Training documentation includes the user training manuals and materials to be used in the conversion and installation of new system. Implementation plans and the results of implementation must be documented. Appendix contains all other documentation. I.e. feasibility study report, problem definition report, study plan, list of controls etc.

**Diagram Of the Tables**

**Login Table:**

LOGIN

**Product Table:**

Product

**Stock Table:**

Stock

**Supplier Table:**

Supplier

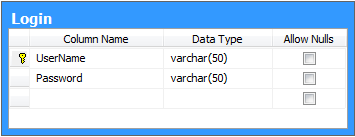
**Warehouse Table:**

Warehouse

**DATA TABLES**

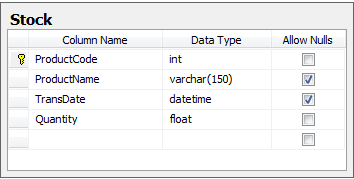
**Login Table:**

Primary key: UserName

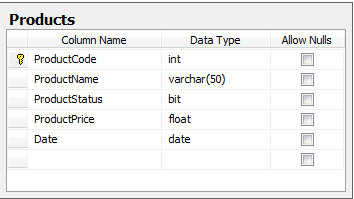
****

**Stock Table:**

Primary key: ProductCode

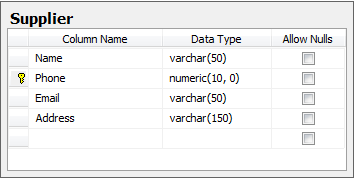
****

**Product Table:**

Primary key: ProductCode

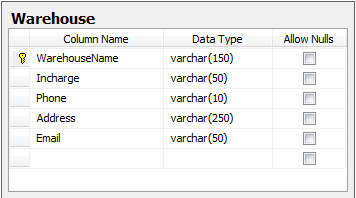
**Supplier Table:**

Primary key: Phone



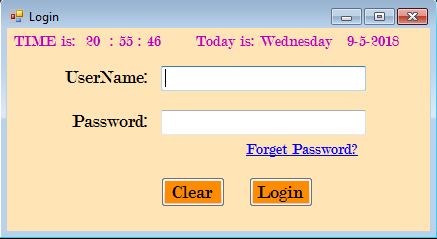
**Warehouse Table:**

Primary key: WarehouseName

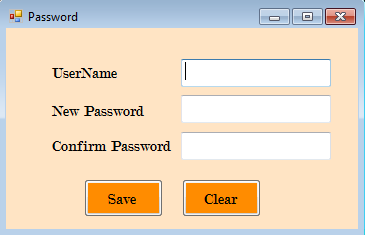


**SCREEN SHOTS OF FORMS**

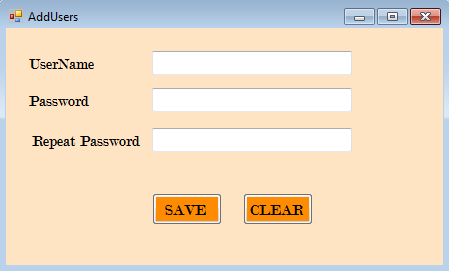
**LOGIN PAGE:**



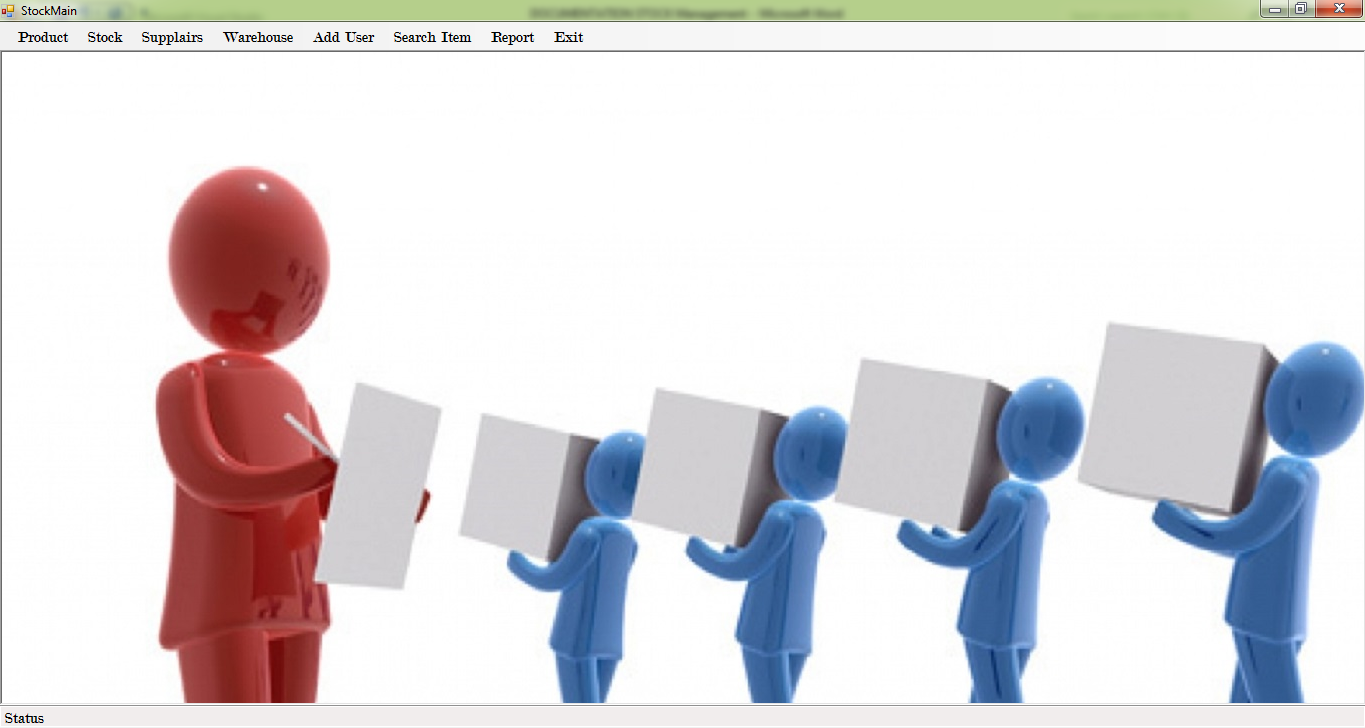
**RESET PASSWORD:**



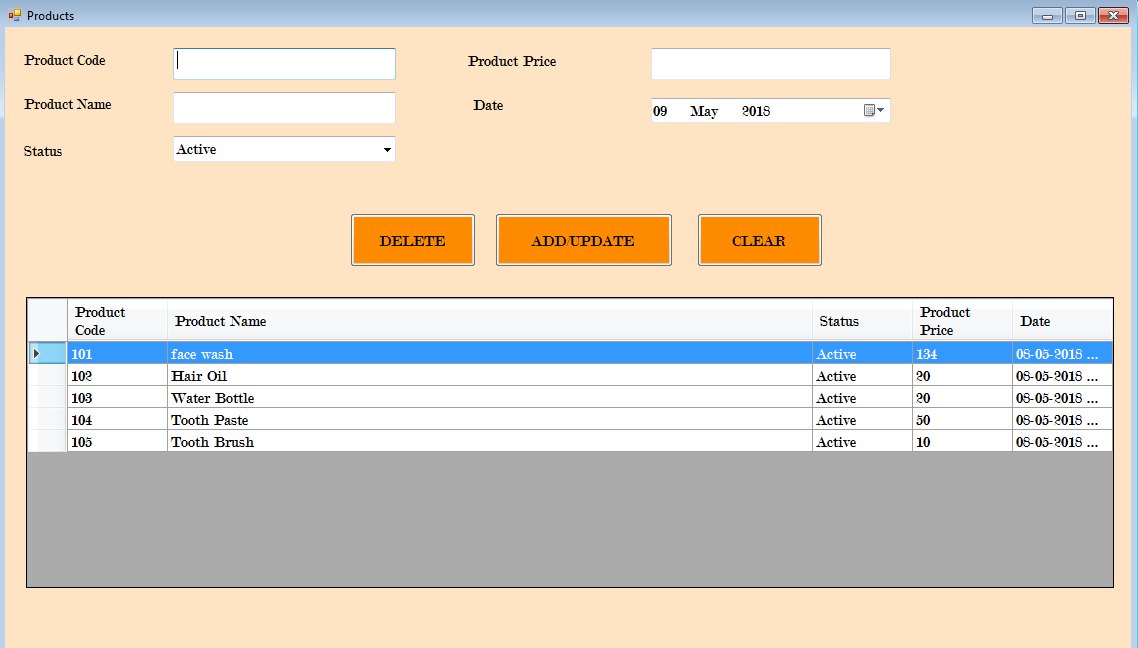
**ADD USER:**

****

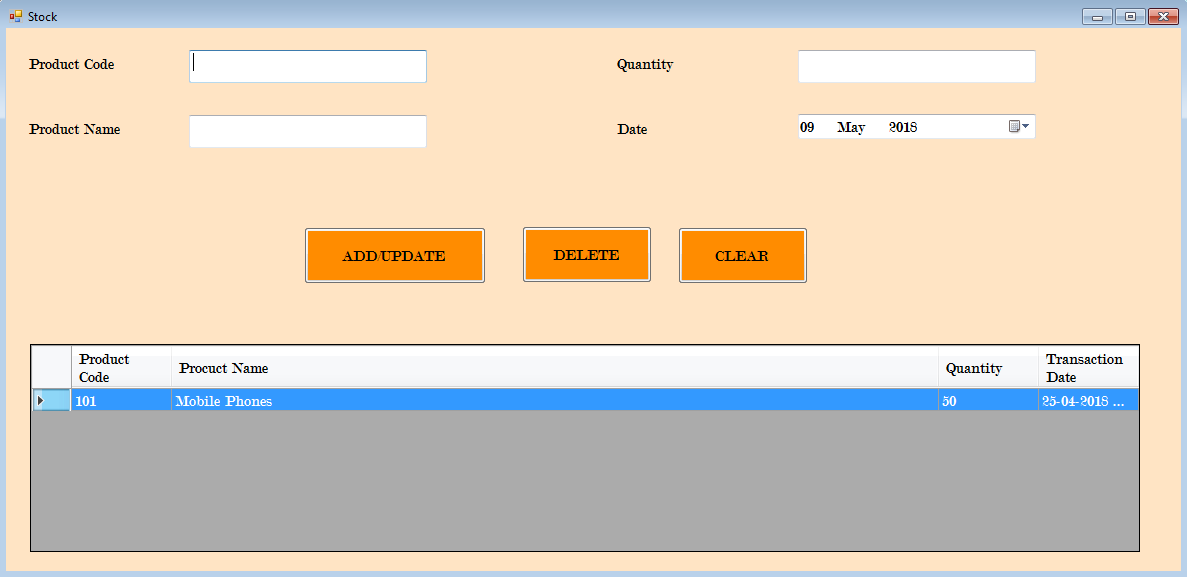
**STOCK MAIN:**

****

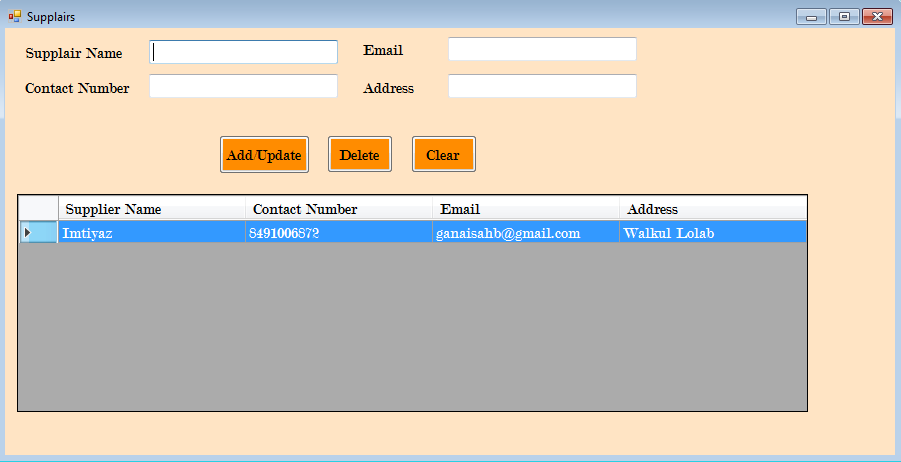
**PRODUCT ADD/UPDATE/DELETE:**

****

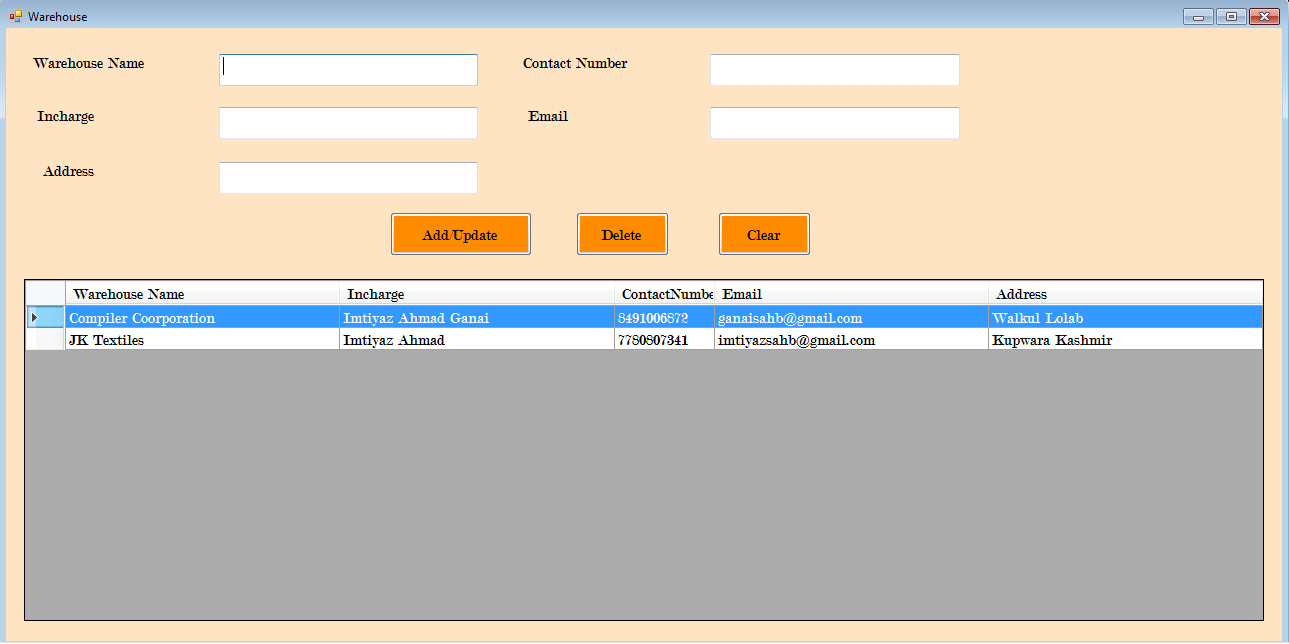
**STOCK ADD/UPDATE/DELETE:**

****

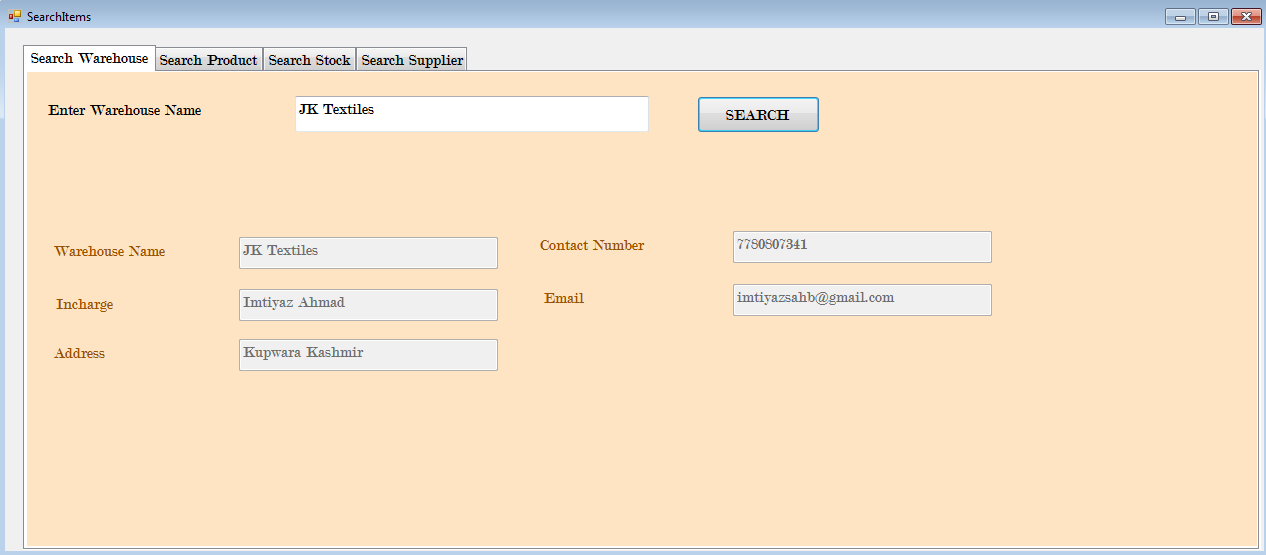
**SUPPLIER ADD/UPDATE/DELETE:**

****

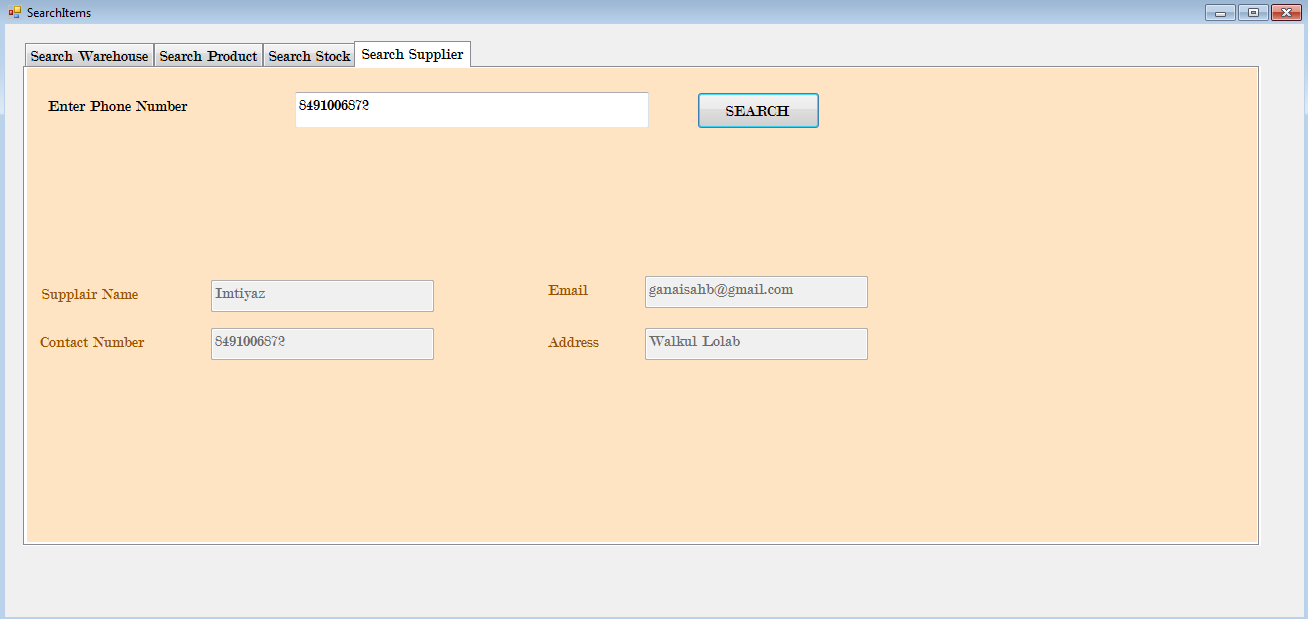
**WAREHOUSE ADD/UPDATE/DELETE:**

****

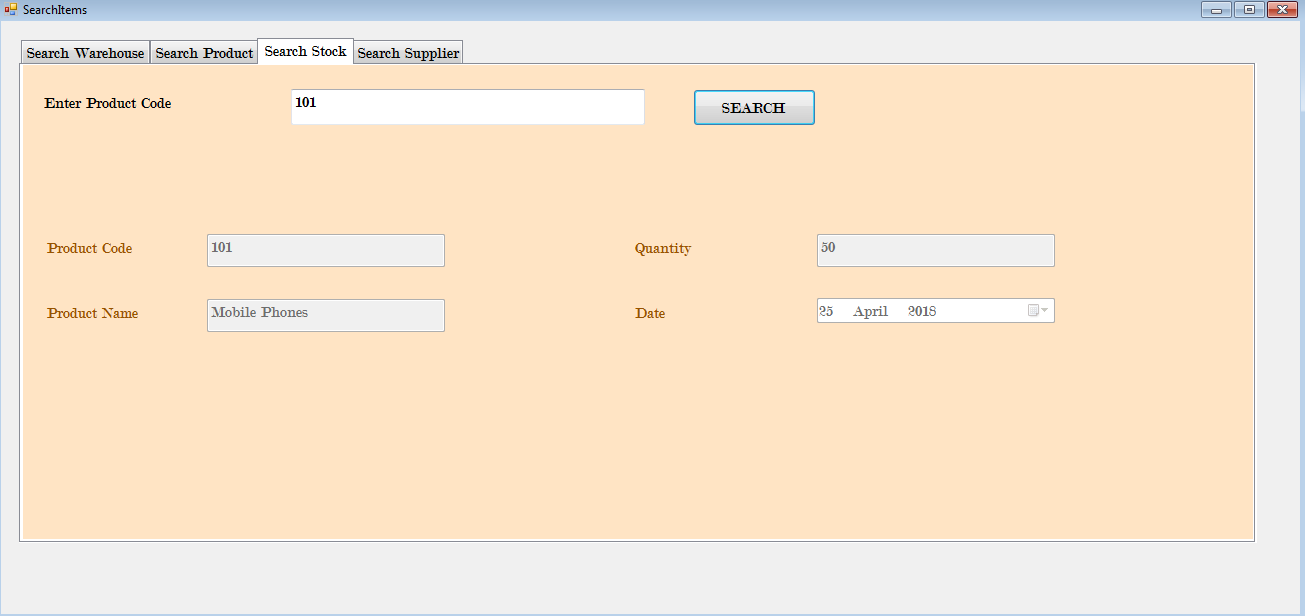
**SEARCH WAREHOUSE:**

****

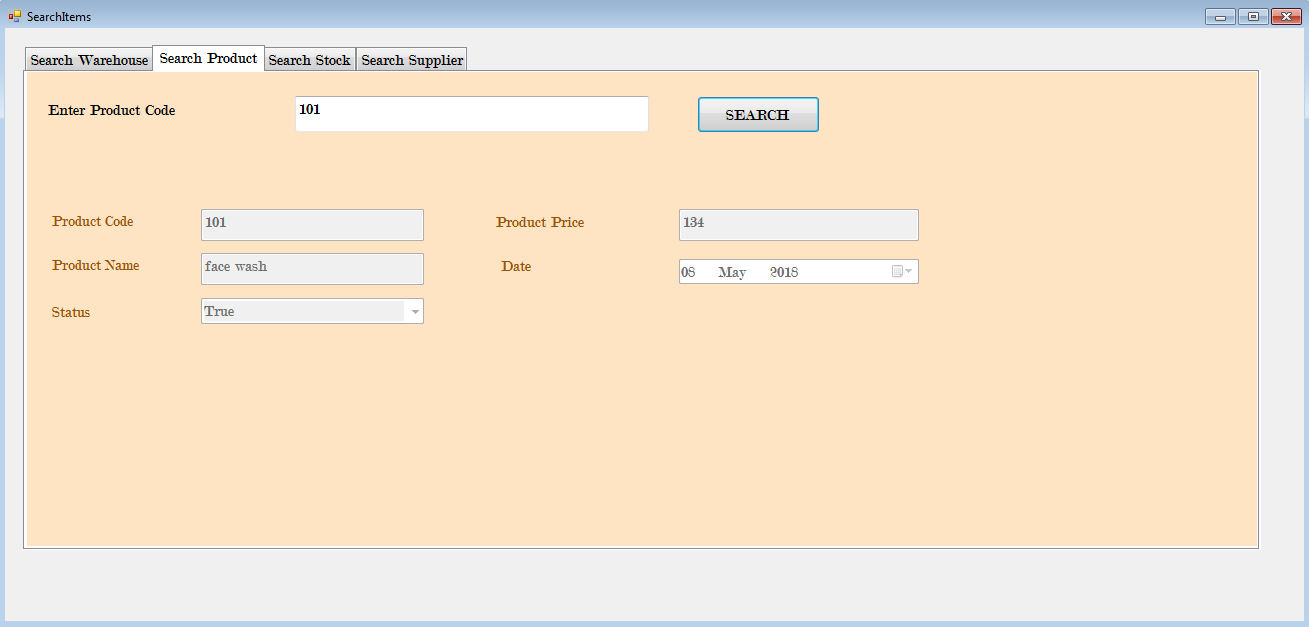
**SEARCH SUPPLIER:**

****

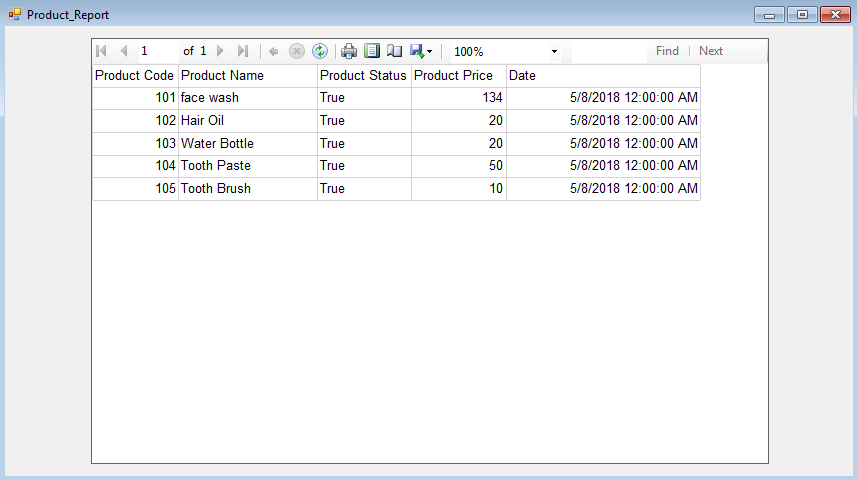
**SEARCH STOCK:**

****

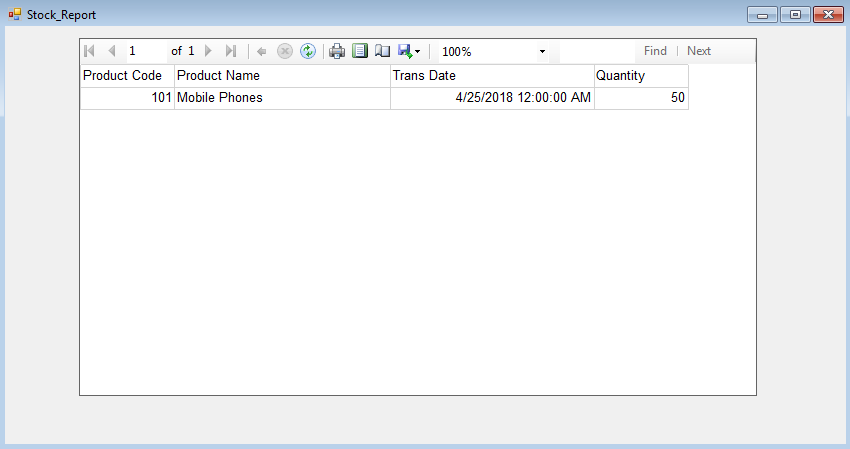
**SEARCH PRODUCT:**

****

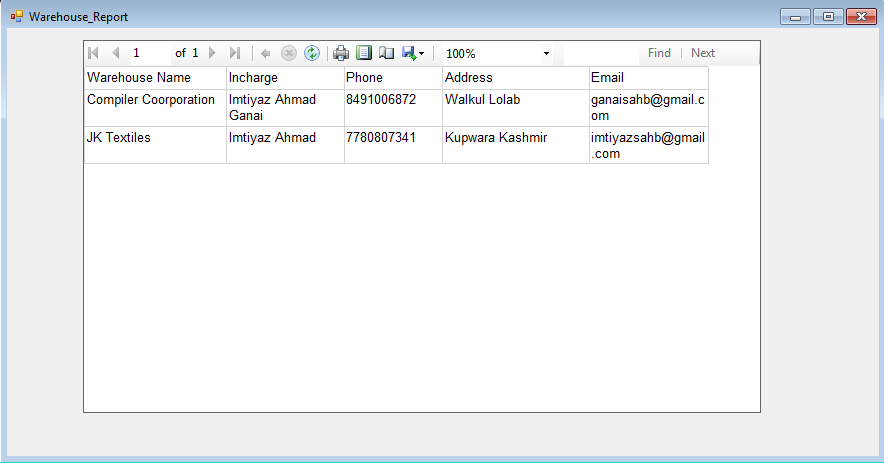
**PRODUCT REPORT:**

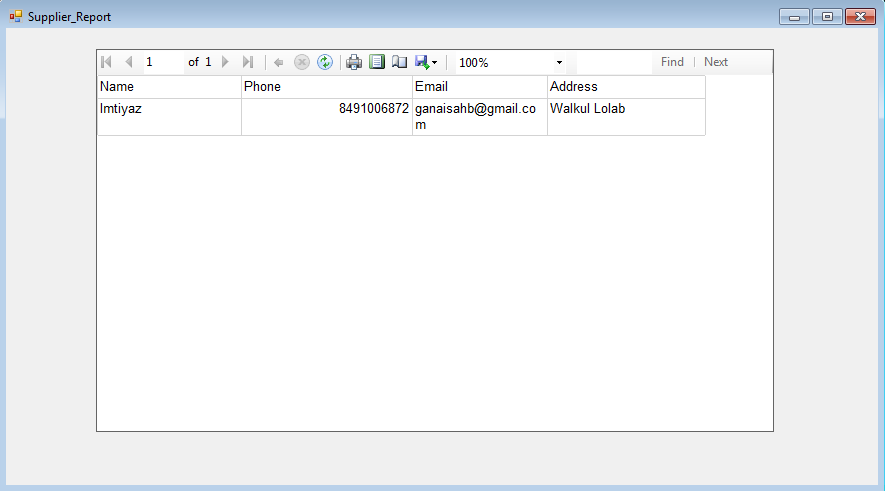
****

**STOCK REPORT:**

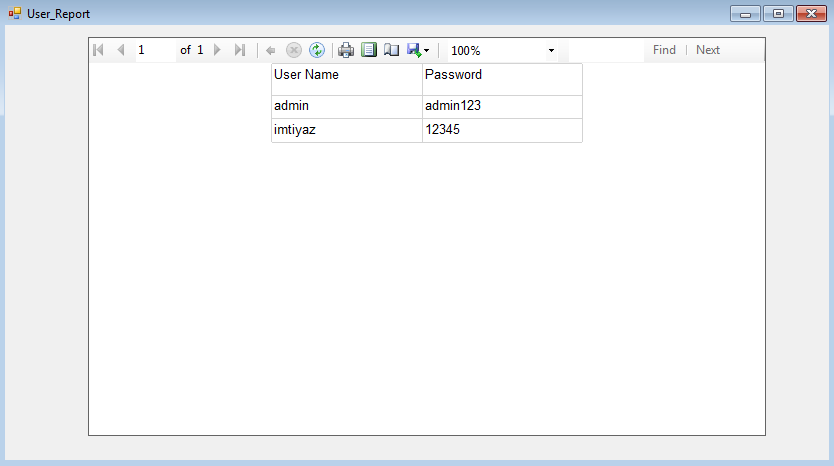
****

**WAREHOUSE REPORT:**

**  
SUPPLIER REPORT:**

****

**USER REPORT:**

****

**SQL SERVER 2008:**

**Create Table:**

USE [Stock]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Products] Script Date: 05/11/2018 15:55:37 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[Products](

[ProductCode] [int] NOT NULL,

[ProductName] [varchar](50) NOT NULL,

[ProductStatus] [bit] NOT NULL,

[ProductPrice] [float] NOT NULL,

[Date] [date] NOT NULL,

CONSTRAINT [PK\_Product] PRIMARY KEY CLUSTERED

(

[ProductCode] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

**INSERT INTO Table:**

INSERT INTO [Stock].[dbo].[Products]

([ProductCode]

,[ProductName]

,[ProductStatus]

,[ProductPrice]

,[Date])

VALUES

(<ProductCode, int,>

,<ProductName, varchar(50),>

,<ProductStatus, bit,>

,<ProductPrice, float,>

,<Date, date,>)

GO

**UPDATE TABLE:**

UPDATE [Stock].[dbo].[Products]

SET [ProductCode] = <ProductCode, int,>

,[ProductName] = <ProductName, varchar(50),>

,[ProductStatus] = <ProductStatus, bit,>

,[ProductPrice] = <ProductPrice, float,>

,[Date] = <Date, date,>

WHERE <Search Conditions,,>

GO

**DELETE ROW FROM TABLE:**

DELETE FROM [Stock].[dbo].[Products]

WHERE <Search Conditions,,>

GO

**DROP AND CREATE TABLE:**

USE [Stock]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Products] Script Date: 05/11/2018 15:58:09 \*\*\*\*\*\*/

IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[Products]') AND type in (N'U'))

DROP TABLE [dbo].[Products]

GO

USE [Stock]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Products] Script Date: 05/11/2018 15:58:11 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[Products](

[ProductCode] [int] NOT NULL,

[ProductName] [varchar](50) NOT NULL,

[ProductStatus] [bit] NOT NULL,

[ProductPrice] [float] NOT NULL,

[Date] [date] NOT NULL,

CONSTRAINT [PK\_Product] PRIMARY KEY CLUSTERED

(

[ProductCode] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

**DROP TABLE:**

USE [Stock]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Products] Script Date: 05/11/2018 15:57:58 \*\*\*\*\*\*/

IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[Products]') AND type in (N'U'))

DROP TABLE [dbo].[Products]

GO

**SELECT DATA FROM TABLE:**

SELECT [ProductCode]

,[ProductName]

,[ProductStatus]

,[ProductPrice]

,[Date]

FROM [Stock].[dbo].[Products]

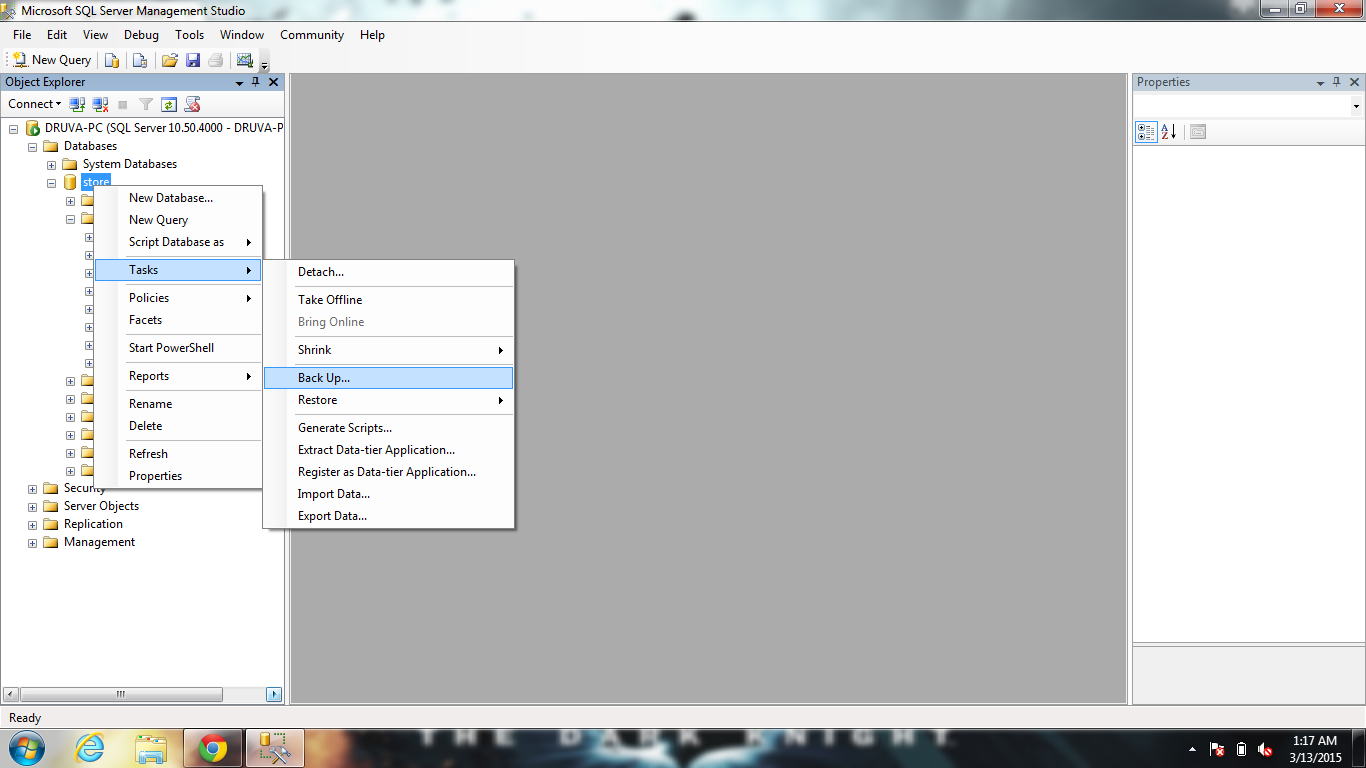
GO

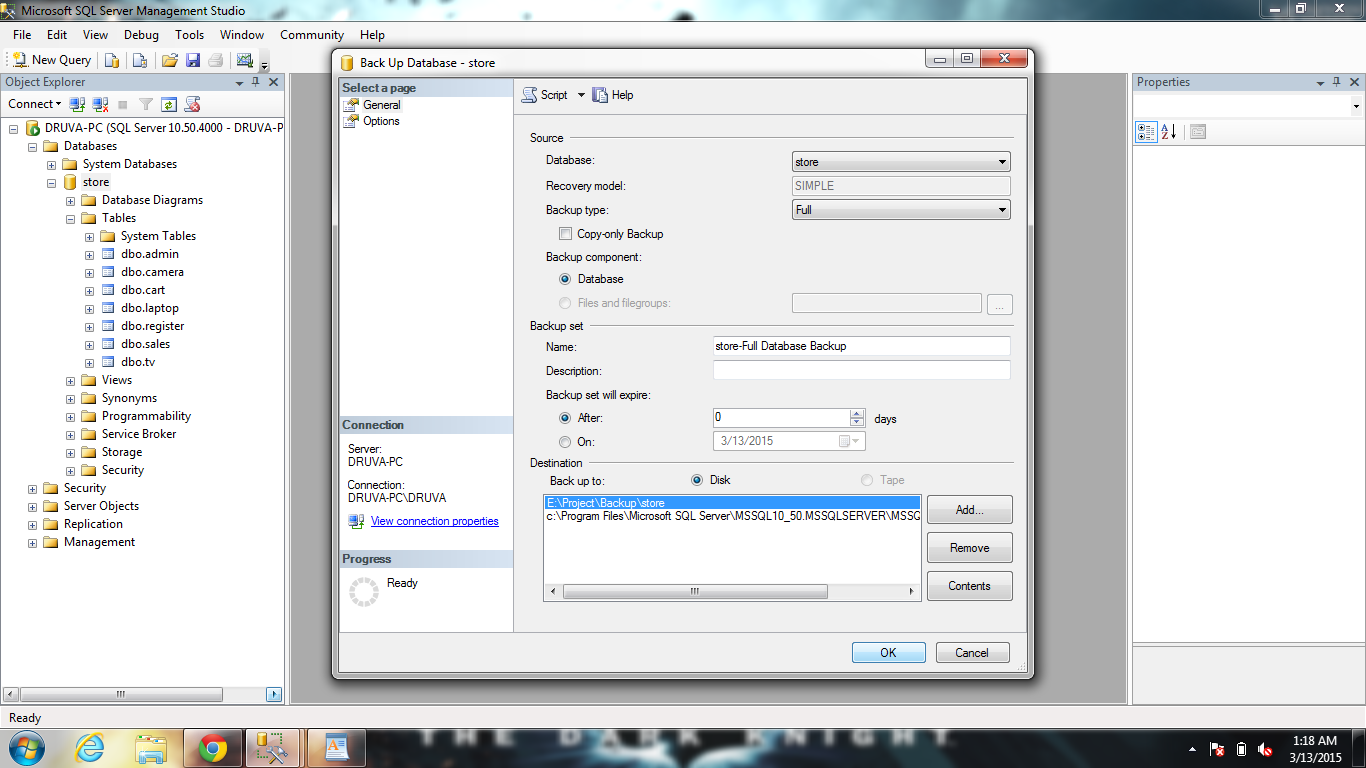
**RECOVERY**

**RECOVERY**

**BACKUP AND RESTORE DATA**

1. **BACKUP**





**ABOUT THE TOOLS**

**ABOUT VISUAL BASIC.NET**

Visual Basic .NET is one of the languages that are directed towards meeting the objectives of the .NET initiative of creating distributed applications.

Visual Basic .NET is a powerful object-oriented language that provides features such as abstraction, encapsulation, inheritance, and polymorphism.

**Features of Visual Basic .NET**

* Some of the key features of Visual Basic .NET are as follows:
  + Inheritance
  + Constructors and destructors
  + Overloading
  + Overriding
  + Structured exception handling
  + Multithreading

**Visual Studio .NET Integrated Development Environment**

* The Visual Studio .NET Integrated Development Environment (IDE) provides you with a common interface for developing various kinds of projects for the .NET Framework.
* In Visual Studio .NET, an application can be made up of one or more items, such as files and folders. To organize these items efficiently, Visual Studio .NET has provided two types of containers:
  + Projects
  + Solutions
* The various components of the Visual Studio .NET IDE are:
  + The Start Page
  + Windows Forms Designer
  + The Solution Explorer Window
  + The Properties Window
  + Toolbox
  + The Output Window
  + The Task List Window
  + The Server Explorer Window
  + The Dynamic Help Window
* Navigational Features in Visual Studio .NET IDE
  + The Class View Window
  + The Code and Text Editor Window
* The Visual Studio .NET IDE also provides three navigational features:
  + Docking
  + Tabbed navigation
  + Auto hide

**Windows Form**

* Is a representation of any window displayed in an application.
* Is used to accept input from a user and display information.

**Advantages of the .NET Framework**

Some advantages of the .NET Framework are:

* + Consistent programming model
  + Multi-platform applications
  + Multi-language integration
  + Automatic resource management
  + Ease of deployment

**Security**

* Evidence-based security (authentication)
* Based on user identity and code identity
* Configurable policies
* Imperative and declarative interfaces

###### INTRODUCTION OF SQL

###### Structured Query Language is a computer language designed for the retrieval and management of data in relational database management systems, database schema creation and modification, and database object access control management.

SQL has been standardized originally designed as a declarative query and data manipulation language, variations of SQL have been created by SQL database management system (DBMS) vendors that add procedural constructs, control-of-flow statements, user-defined data types, and various other language extensions. With the release of the SQL: 1999 standard, many such extensions were formally adopted as part of the SQL language via the SQL Persistent Stored Modules (SQL/PSM) portion of the standard.

Using SQL one can create and maintain data manipulation objects such as table, views, sequence etc. These data manipulation objects will be created and stored on the server's hard disk drive, in a table space, to which the user has been assigned. Once these data manipulation objects are created, they are used extensively in commercial applications.

**Data Manipulation Language:**

First, there are the standard Data Manipulation Language (DML) elements. DML is the subset of the language used to add, update and delete data:

* INSERT is used to add rows (formally tuples) to an existing table.
* UPDATE is used to modify the values of a set of existing table rows.
* MERGE is used to combine the data of multiple tables.
* DELETE removes zero or more existing rows from a table.

**Data Definition Language:**

The second group of keywords is the Data Definition Language (DDL). DDL allows the user to define new tables and associated elements. Most commercial SQL databases have proprietary extensions in their DDL, which allow control over nonstandard features of the database system. The most basic items of DDL are the CREATE, ALTER, RENAME, TRUNCATE and DROP statements:

* CREATE causes an object (a table, for example) to be created within the database.
* DROP causes an existing object within the database to be deleted, usually irretrievably.
* TRUNCATE deletes all data from a table (non-standard, but common SQL statement).
* ALTER statement permits the user to modify an existing object in various ways -- for example, adding a column to an existing table.

Example:

CREATE TABLE my\_table (My\_field1 INT, My\_field2 VARCHAR (50), My\_field3 DATE NOT NULL, PRIMARY KEY (my\_field1, my\_field2));

In addition to the creation of data manipulation objects, the actual manipulation of data within these objects is done using SQL.

**Data Control Language:**

The third group of SQL keywords is the Data Control Language (DCL). DCL handles the authorization aspects of data and permits the user to control who has access to see or manipulate data within the database. Its two main keywords are:

* GRANT authorizes one or more users to perform an operation or a set of operations on an object.
* REVOKE removes or restricts the capability of a user to perform an operation or a set of operations.

The SQL sentences that are used to create these objects are called DDL's or Data Definition Language. The SQL sentences used to manipulate data within these objects are called DML's or Data Manipulation Language. The SQL sentences, which are used to control the behavior of these objects, are called DCL's or Data Control Language.

**DATA FLOW DIAGRAMS**

CONTEXT DIAGRAM

Request to Purchase

PUR DEPT

STORE DEPT

VENDER

Q

U

A

T

Request to send items

Purchase Order

Goods Issue

PURCHASE ACTIVITIES

SEND PURCHASE

ORDER

COST CENTER

VENDOR

SEND RECEIPTS

STORES

RECEIPTS

**INDENT MASTER**

SEND

POSITION FOR PRODUCTS **ITEM**

# RETRIEVE ENQUIRY

PRODUCTS

PRODUCTS

ITEM MAST

QUATATION

STORES

PRODUCTS

STORE ACTIVITIES

VENDOR

COST CENTER

**STORES**

INSPECTION

TOP LEVEL

ITEMS

VENDER

DEPARTMENT

Request for purchase

Indents

Purchase order

Quotation

Goods issue

Goods receipts

E-R DIAGRAM

Send to

Request

Stores dept

Store in

Goods

Purchase Dept

Plays

Request

Sent

Quotation

Vendor

Order

COST CENTER

**CODING OF THE PROJECT**

**AddUser Page Coding:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Stock

{

public partial class AddUsers : Form

{

public AddUsers()

{

InitializeComponent();

}

SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=Stock;Integrated Security=True");

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

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox1.Focus();

}

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

{

conn.Open();

var sqlQuery = "";

if (IfExists(conn, textBox1.Text))

{

MessageBox.Show("User Already exists", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Warning);

button2\_Click(sender, e);

}

else

{

if (textBox2.Text.Trim() == textBox3.Text.Trim())

{

sqlQuery = @"INSERT INTO [Stock].[dbo].[Login]([UserName],[Password]) VALUES ('" + textBox1.Text.Trim() + "','" + textBox2.Text.Trim() + "')";

MessageBox.Show("User Registered Successfully", "Successful Entry", MessageBoxButtons.OK, MessageBoxIcon.Information);

SqlCommand cmd = new SqlCommand(sqlQuery, conn);

cmd.ExecuteNonQuery();

textBox1.Enabled = true;

button2\_Click(sender, e);

}

else

{

MessageBox.Show("Password Did not Match!!!!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

button2\_Click(sender, e);

}

}

conn.Close();

}

private bool IfExists(SqlConnection conn, string user)

{

SqlDataAdapter sda = new SqlDataAdapter(@"select \* from [Login] where [UserName] ='" + user + "'", conn);

DataTable dt = new DataTable();

sda.Fill(dt);

if (dt.Rows.Count > 0)

return true;

else

return false;

}

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

{

}

}

}

**Login Page Coding:**

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 Stock

{

public partial class Login : Form

{

public Login()

{

InitializeComponent();

}

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

{

textBox1.Text = "";

textBox2.Clear();

textBox1.Focus();

}

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

{

SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=Stock;Integrated Security=True");

conn.Open();

SqlDataAdapter sda = new SqlDataAdapter(@"select \* from [Stock].[dbo].[Login] where UserName= '" + textBox1.Text+"' and Password ='"+textBox2.Text+"'",conn);

DataTable dt = new DataTable();

sda.Fill(dt);

if (dt.Rows.Count == 1)

{

this.Hide();

StockMain obj = new StockMain();

obj.Show();

}

else

{

MessageBox.Show("Invalid Username Or Password........", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

button1\_Click(sender, e);

}

conn.Close();

}

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

{

timer1.Enabled = true;

}

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

{

label3.Text = "TIME is: " + " " + DateTime.Now.Hour.ToString() + " : " + DateTime.Now.Minute.ToString() + " : " + DateTime.Now.Second.ToString();

label4.Text = "Today is: " + DateTime.Now.DayOfWeek.ToString() + " " + DateTime.Now.Day.ToString() + "-" + DateTime.Now.Month + "-" + DateTime.Now.Year.ToString();

}

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

{

this.Hide();

Password obj = new Password();

obj.Show();

}

}

}

**Password Page Coding:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Stock

{

public partial class Password : Form

{

public Password()

{

InitializeComponent();

}

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

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox1.Focus();

}

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

{

SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=Stock;Integrated Security=True");

conn.Open();

SqlDataAdapter sda = new SqlDataAdapter(@"select \* from [Stock].[dbo].[Login] where UserName= '" + textBox1.Text + "'", conn);

DataTable dt = new DataTable();

sda.Fill(dt);

var sqlQuery="";

if (dt.Rows.Count == 1)

{

if (textBox2.Text.Trim() == textBox3.Text.Trim())

{

sqlQuery = @"UPDATE [Login]SET[Password] = '" + textBox2.Text.Trim() + "'";

MessageBox.Show("Password Changed successfully", "Password Changed", MessageBoxButtons.OK,MessageBoxIcon.Warning);

SqlCommand cmd = new SqlCommand(sqlQuery, conn);

cmd.ExecuteNonQuery();

this.Close();

Login obj = new Login();

obj.Show();

}

else

{

MessageBox.Show("Password Does not Match", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

button2\_Click(sender,e);

}

}

else

{

MessageBox.Show("Invalid Username Please Provide Correct Details........", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

button2\_Click(sender, e);

}

conn.Close();

}

private void Password\_FormClosed(object sender, FormClosedEventArgs e)

{

Login obj = new Login();

obj.Enabled = true;

obj.Show();

}

}

}

**Product\_Report Page Coding:**

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 Stock

{

public partial class Product\_Report : Form

{

public Product\_Report()

{

InitializeComponent();

}

private void Product\_Report\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'StockDataSet2.Products' table. You can move, or remove it, as needed.

this.ProductsTableAdapter.Fill(this.StockDataSet2.Products);

this.reportViewer1.RefreshReport();

}

}

}

**Product Page Coding:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Stock

{

public partial class Products : Form

{

public Products()

{

InitializeComponent();

}

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

{

comboBox1.SelectedIndex = 0;

LoadData();

}

SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=Stock;Integrated Security=True");

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

{

conn.Open();

bool status = false;

if (comboBox1.SelectedIndex == 0)

{

status = true;

}

else

{

status = false;

}

var sqlQuery = "";

if (IfProductExists(conn,textBox1.Text))

{

sqlQuery = @"UPDATE [Products]SET[ProductName] = '" + textBox2.Text.Trim() + "',[ProductStatus] ='" + status + "',[ProductPrice] ='" + textBox3.Text.Trim() + "',[Date] ='"+ dateTimePicker1.Text + "'WHERE [ProductCode] = " + Convert.ToInt32(textBox1.Text.Trim());

MessageBox.Show("Data Updated Successfully", "Successful Entry", MessageBoxButtons.OK);

textBox1.Enabled = true;

}

else

{

sqlQuery = @"INSERT INTO [Stock].[dbo].[Products]([ProductCode],[ProductName],[ProductStatus],[ProductPrice],[Date]) VALUES

(" +Convert.ToInt32(textBox1.Text.Trim())+",'"+textBox2.Text.Trim()+"','"+status+"','"+textBox3.Text.Trim()+"','"+dateTimePicker1.Text+"')";

MessageBox.Show("Data inserted Successfully", "Successful Entry", MessageBoxButtons.OK);

}

SqlCommand cmd = new SqlCommand(sqlQuery,conn);

cmd.ExecuteNonQuery();

button3\_Click(sender, e);

conn.Close();

LoadData();

}

private bool IfProductExists(SqlConnection conn, string productcode)

{

SqlDataAdapter sda = new SqlDataAdapter(@"select \* from [Products] where [ProductCode] =" +Convert.ToInt32(productcode), conn);

DataTable dt = new DataTable();

sda.Fill(dt);

if (dt.Rows.Count > 0)

return true;

else

return false;

}

public void LoadData()

{

SqlDataAdapter sda = new SqlDataAdapter(@"select \* from [Stock].[dbo].[Products]", conn);

DataTable dt = new DataTable();

sda.Fill(dt);

dataGridView1.Rows.Clear();

foreach (DataRow item in dt.Rows)

{

int n = dataGridView1.Rows.Add();

dataGridView1.Rows[n].Cells[0].Value = item["ProductCode"].ToString();

dataGridView1.Rows[n].Cells[1].Value = item["ProductName"].ToString();

if ((bool)item["ProductStatus"])

{

dataGridView1.Rows[n].Cells[2].Value = "Active";

}

else

{

dataGridView1.Rows[n].Cells[2].Value = "Deactive";

}

dataGridView1.Rows[n].Cells[3].Value = item["ProductPrice"].ToString();

dataGridView1.Rows[n].Cells[4].Value = item["Date"].ToString();

}

}

private void dataGridView1\_MouseDoubleClick(object sender, MouseEventArgs e)

{

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

textBox2.Text = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

if (dataGridView1.SelectedRows[0].Cells[2].Value.ToString() == "Active")

{

comboBox1.SelectedIndex = 0;

}

else

{

comboBox1.SelectedIndex = 1;

}

textBox3.Text = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();

dateTimePicker1.Text = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();

textBox1.Enabled = false;

}

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

{

DialogResult dialog = MessageBox.Show("Do you really want to Delete the Data?", "Exit", MessageBoxButtons.YesNo);

if (dialog == DialogResult.Yes)

{

if (isEmpty())

{

MessageBox.Show("Nothing to detete please check again?????", "ERROR?", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Focus();

}

else

{

SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=Stock;Integrated Security=True");

conn.Open();

var sqlQuery = "";

sqlQuery = @"DELETE FROM [Stock].[dbo].[Products] WHERE [ProductCode] = " + Convert.ToInt32(textBox1.Text);

MessageBox.Show("Data Deleted Successfully", "Successful Entry", MessageBoxButtons.OK);

SqlCommand cmd = new SqlCommand(sqlQuery, conn);

cmd.ExecuteNonQuery();

textBox1.Enabled = true;

button3\_Click(sender, e);

conn.Close();

LoadData();

}

}

}

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

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox1.Enabled = true;

dateTimePicker1.Text = DateTime.Now.ToString();

textBox1.Focus();

}

private bool isEmpty()

{

if (textBox1.Text == "")

{

return true;

}

else

return false;

}

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

{

}

}

}

**Search\_Item Page Coding:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Stock

{

public partial class SearchItems : Form

{

public SearchItems()

{

InitializeComponent();

}

SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=Stock;Integrated Security=True");

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

{

conn.Open();

if (textBox1.Text.Trim() == "")

{

MessageBox.Show("Please enter Details", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Focus();

}

else

{

SqlCommand cmd = new SqlCommand(@"select \* from [Warehouse] where [WarehouseName] = '" + textBox1.Text.Trim() + "'", conn);

SqlDataReader rd = cmd.ExecuteReader();

rd.Read();

if (rd.HasRows)

{

textBox15.Visible = true;

textBox16.Visible = true;

textBox17.Visible = true;

textBox18.Visible = true;

textBox19.Visible = true;

label18.Visible = true;

label19.Visible = true;

label20.Visible = true;

label21.Visible = true;

label22.Visible = true;

textBox15.Text = rd.GetValue(0).ToString();

textBox16.Text = rd.GetValue(1).ToString();

textBox17.Text = rd.GetValue(2).ToString();

textBox18.Text = rd.GetValue(4).ToString();

textBox19.Text = rd.GetValue(3).ToString();

}

else

{

MessageBox.Show("Warehouse Does not exist","Error",MessageBoxButtons.OK,MessageBoxIcon.Warning);

textBox1.Clear();

textBox1.Focus();

}

}

conn.Close();

}

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

{

conn.Open();

if (textBox2.Text.Trim() == "")

{

MessageBox.Show("Please enter Details", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Focus();

}

else

{

SqlCommand cmd = new SqlCommand(@"select \* from [Products] where [ProductCode] = " + Convert.ToInt32(textBox2.Text.Trim()) , conn);

SqlDataReader rd = cmd.ExecuteReader();

rd.Read();

if (rd.HasRows)

{

textBox14.Visible = true;

textBox13.Visible = true;

textBox12.Visible = true;

comboBox1.Visible = true;

dateTimePicker2.Visible = true;

label17.Visible = true;

label16.Visible = true;

label15.Visible = true;

label14.Visible = true;

label13.Visible = true;

textBox14.Text = rd.GetValue(0).ToString();

textBox13.Text = rd.GetValue(1).ToString();

comboBox1.Text = rd.GetValue(2).ToString();

textBox12.Text = rd.GetValue(3).ToString();

dateTimePicker2.Text = rd.GetValue(4).ToString();

}

else

{

MessageBox.Show("Product Is Not Available", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

textBox2.Clear();

textBox2.Focus();

}

}

conn.Close();

}

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

{

conn.Open();

if (textBox3.Text.Trim() == "")

{

MessageBox.Show("Please enter Details", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Focus();

}

else

{

SqlCommand cmd = new SqlCommand(@"select \* from [Stock] where [ProductCode] = " + Convert.ToInt32(textBox3.Text.Trim()) , conn);

SqlDataReader rd = cmd.ExecuteReader();

rd.Read();

if (rd.HasRows)

{

textBox11.Visible = true;

textBox10.Visible = true;

textBox9.Visible = true;

dateTimePicker1.Visible = true;

label12.Visible = true;

label11.Visible = true;

label10.Visible = true;

label9.Visible = true;

textBox11.Text = rd.GetValue(0).ToString();

textBox10.Text = rd.GetValue(1).ToString();

textBox9.Text = rd.GetValue(3).ToString();

dateTimePicker1.Text = rd.GetValue(2).ToString();

}

else

{

MessageBox.Show("Product Code Does not exist", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

textBox3.Clear();

textBox3.Focus();

}

}

conn.Close();

}

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

{

conn.Open();

if (textBox4.Text.Trim() == "")

{

MessageBox.Show("Please enter Details", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Focus();

}

else

{

SqlCommand cmd = new SqlCommand(@"select \* from [Supplier] where [Phone] = " + Convert.ToInt64(textBox4.Text.Trim()), conn);

SqlDataReader rd = cmd.ExecuteReader();

rd.Read();

if (rd.HasRows)

{

textBox5.Visible = true;

textBox6.Visible = true;

textBox7.Visible = true;

textBox8.Visible = true;

label5.Visible = true;

label6.Visible = true;

label7.Visible = true;

label8.Visible = true;

textBox5.Text = rd.GetValue(0).ToString();

textBox6.Text = rd.GetValue(1).ToString();

textBox7.Text = rd.GetValue(2).ToString();

textBox8.Text = rd.GetValue(3).ToString();

}

else

{

MessageBox.Show("Supplier Does not exist", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

textBox11.Clear();

textBox11.Focus();

}

}

conn.Close();

}

}

}

**Stock Page Coding:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Stock

{

public partial class Stock : Form

{

public Stock()

{

InitializeComponent();

}

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

{

LoadData();

}

public void LoadData()

{

SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=Stock;Integrated Security=True");

SqlDataAdapter sda = new SqlDataAdapter(@"select \* from [Stock].[dbo].[Stock]", conn);

DataTable dt = new DataTable();

sda.Fill(dt);

dataGridView1.Rows.Clear();

foreach (DataRow item in dt.Rows)

{

int n = dataGridView1.Rows.Add();

dataGridView1.Rows[n].Cells[0].Value = item["ProductCode"].ToString();

dataGridView1.Rows[n].Cells[1].Value = item["ProductName"].ToString();

dataGridView1.Rows[n].Cells[2].Value = item["Quantity"].ToString();

dataGridView1.Rows[n].Cells[3].Value = item["TransDate"].ToString();

}

}

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

{

SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=Stock;Integrated Security=True");

conn.Open();

var sqlQuery = "";

if (IfProductExists(conn, textBox1.Text))

{

sqlQuery = @"UPDATE [Stock]SET[ProductName] = '" + textBox2.Text.Trim() + "',[Quantity] = '" + Convert.ToDouble(textBox3.Text.Trim()) +"', [TransDate] = "+ dateTimePicker1.Text + "' WHERE [ProductCode] = " + Convert.ToInt32(textBox1.Text.Trim());

MessageBox.Show("Data Updated Successfully", "Successful Entry", MessageBoxButtons.OK);

textBox1.Enabled = true;

}

else

{

sqlQuery = @"INSERT INTO [Stock].[dbo].[Stock]([ProductCode],[ProductName],[Quantity],[TransDate]) VALUES

(" + Convert.ToInt32(textBox1.Text.Trim()) + ",'" + textBox2.Text.Trim() + "','" + Convert.ToDouble(textBox3.Text.Trim()) + "','" + dateTimePicker1.Text + "')";

MessageBox.Show("Data inserted Successfully", "Successful Entry", MessageBoxButtons.OK);

}

SqlCommand cmd = new SqlCommand(sqlQuery, conn);

cmd.ExecuteNonQuery();

button3\_Click(sender, e);

conn.Close();

LoadData();

}

private bool IfProductExists(SqlConnection conn, string productcode)

{

SqlDataAdapter sda = new SqlDataAdapter(@"select \* from [Stock] where [ProductCode] =" + Convert.ToInt32(productcode), conn);

DataTable dt = new DataTable();

sda.Fill(dt);

if (dt.Rows.Count > 0)

return true;

else

return false;

}

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

{

}

private void dataGridView1\_MouseDoubleClick(object sender, MouseEventArgs e)

{

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

textBox2.Text = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

textBox3.Text = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();

dateTimePicker1.Text = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();

textBox1.Enabled = false;

}

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

{

DialogResult dialog = MessageBox.Show("Do you really want to exit the Programe?", "Exit", MessageBoxButtons.YesNo);

if (dialog == DialogResult.Yes)

{

if (isEmpty())

{

MessageBox.Show("Nothing to detete please check again???", "ERROR?", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Focus();

}

else

{

SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=Stock;Integrated Security=True");

conn.Open();

var sqlQuery = "";

sqlQuery = @"DELETE FROM [Stock].[dbo].[Stock] WHERE ProductCode=" + textBox1.Text;

MessageBox.Show("Data Deleted Successfully", "Successful Entry", MessageBoxButtons.OK);

SqlCommand cmd = new SqlCommand(sqlQuery, conn);

cmd.ExecuteNonQuery();

button3\_Click(sender, e);

textBox1.Enabled = true;

conn.Close();

LoadData();

}

}

}

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

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

dateTimePicker1.Text = DateTime.Now.ToString();

textBox1.Focus();

textBox1.Enabled = true;

}

private bool isEmpty()

{

if (textBox1.Text == "")

{

return true;

}

else

return false;

}

}

}

**Stock\_Report Page Coding:**

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 Stock

{

public partial class Stock\_Report : Form

{

public Stock\_Report()

{

InitializeComponent();

}

private void Stock\_Report\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'StockDataSet4.Stock' table. You can move, or remove it, as needed.

this.StockTableAdapter.Fill(this.StockDataSet4.Stock);

this.reportViewer1.RefreshReport();

}

}

}

**Stock\_Main Page Coding:**

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 Stock

{

public partial class StockMain : Form

{

public StockMain()

{

InitializeComponent();

}

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

{

Products pro = new Products();

pro.MdiParent = this;

pro.Show();

}

private void StockMain\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

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

{

DialogResult dialog = MessageBox.Show("Do you really want to exit the Programe?", "Exit", MessageBoxButtons.YesNo);

if (dialog == DialogResult.Yes)

{

Application.Exit();

}

}

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

{

Stock pro = new Stock();

pro.MdiParent = this;

pro.Show();

}

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

{

Supplairs pro = new Supplairs();

pro.MdiParent = this;

pro.Show();

}

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

{

Warehouse pro = new Warehouse();

pro.MdiParent = this;

pro.Show();

}

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

{

}

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

{

AddUsers pro = new AddUsers();

pro.MdiParent = this;

pro.Show();

}

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

{

SearchItems pro = new SearchItems();

pro.MdiParent = this;

pro.Show();

}

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

{

Product\_Report pro = new Product\_Report();

pro.MdiParent = this;

pro.Show();

}

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

{

Stock\_Report pro = new Stock\_Report();

pro.MdiParent = this;

pro.Show();

}

private void supplairsToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Supplier\_Report pro = new Supplier\_Report();

pro.MdiParent = this;

pro.Show();

}

private void warehouseToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Warehouse\_Report pro = new Warehouse\_Report();

pro.MdiParent = this;

pro.Show();

}

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

{

User\_Report pro = new User\_Report();

pro.MdiParent = this;

pro.Show();

}

}

}

**Supplier Page Coding:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Stock

{

public partial class Supplairs : Form

{

public Supplairs()

{

InitializeComponent();

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=Stock;Integrated Security=True");

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

{

conn.Open();

var sqlQuery = "";

if (isEmpty())

{

MessageBox.Show("Please Enter all the fields", "ERROR?", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Focus();

}

else

{

if (IfDataExists(conn, textBox2.Text))

{

sqlQuery = @"UPDATE [Supplier] SET [Name] = '" + textBox1.Text.Trim() + "',[Phone] = " + textBox2.Text.Trim() + ",[Email] ='" + textBox3.Text.Trim() + "',[Address] ='" + textBox4.Text.Trim() + "'WHERE [Phone] = " + textBox2.Text.Trim();

MessageBox.Show("Data Updated Successfully", "Successful Entry", MessageBoxButtons.OK);

textBox1.Enabled = true;

}

else

{

sqlQuery = @"INSERT INTO [Stock].[dbo].[Supplier]([Name],[Phone],[Email],[Address]) VALUES ('" + textBox1.Text.Trim() + "'," + textBox2.Text.Trim() + ",'" + textBox3.Text.Trim() + "','" + textBox4.Text.Trim() + "')";

MessageBox.Show("Data inserted Successfully", "Successful Entry", MessageBoxButtons.OK);

}

SqlCommand cmd = new SqlCommand(sqlQuery, conn);

cmd.ExecuteNonQuery();

button3\_Click(sender, e);

conn.Close();

LoadData();

}

}

private bool IfDataExists(SqlConnection conn, String phoneno)

{

SqlDataAdapter sda = new SqlDataAdapter(@"select \* from [Supplier] where [Phone] = " + phoneno, conn);

DataTable dt = new DataTable();

sda.Fill(dt);

if (dt.Rows.Count > 0)

return true;

else

return false;

}

public void LoadData()

{

SqlDataAdapter sda = new SqlDataAdapter(@"select \* from [Stock].[dbo].[Supplier]", conn);

DataTable dt = new DataTable();

sda.Fill(dt);

dataGridView1.Rows.Clear();

foreach (DataRow item in dt.Rows)

{

int n = dataGridView1.Rows.Add();

dataGridView1.Rows[n].Cells[0].Value = item["Name"].ToString();

dataGridView1.Rows[n].Cells[1].Value = item["Phone"].ToString();

dataGridView1.Rows[n].Cells[2].Value = item["Email"].ToString();

dataGridView1.Rows[n].Cells[3].Value = item["Address"].ToString();

}

}

private bool isEmpty()

{

if (textBox2.Text == "")

{

return true;

}

else

return false;

}

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

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox1.Focus();

}

private void dataGridView1\_MouseDoubleClick(object sender, MouseEventArgs e)

{

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

textBox2.Text = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

textBox3.Text = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();

textBox4.Text = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();

}

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

{

LoadData();

}

}

}

**Supplier\_Report Page Coding:**

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 Stock

{

public partial class Supplier\_Report : Form

{

public Supplier\_Report()

{

InitializeComponent();

}

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

{

// TODO: This line of code loads data into the 'StockDataSet5.Supplier' table. You can move, or remove it, as needed.

this.SupplierTableAdapter.Fill(this.StockDataSet5.Supplier);

this.reportViewer1.RefreshReport();

}

}

}

**User\_Report Page Coding:**

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 Stock

{

public partial class User\_Report : Form

{

public User\_Report()

{

InitializeComponent();

}

private void User\_Report\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'StockDataSet7.Login' table. You can move, or remove it, as needed.

this.LoginTableAdapter.Fill(this.StockDataSet7.Login);

this.reportViewer1.RefreshReport();

}

}

}

**Warehouse Page Coding:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Stock

{

public partial class Warehouse : Form

{

public Warehouse()

{

InitializeComponent();

}

SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=Stock;Integrated Security=True");

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

{

conn.Open();

var sqlQuery = "";

if (isEmpty())

{

MessageBox.Show("Please Enter all the fields", "ERROR?", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Focus();

}

else

{

if (IfDataExists(conn, textBox1.Text))

{

sqlQuery = @"UPDATE [Warehouse] SET [WarehouseName] = '" + textBox1.Text.Trim() + "',[Incharge] ='" + textBox2.Text.Trim() + "',[Phone] = '" + textBox3.Text.Trim() + "',[Email] ='" + textBox4.Text.Trim() + "',[Address] ='" + textBox5.Text.Trim() + "'WHERE [WarehouseName] = '" + textBox1.Text.Trim() + "'";

MessageBox.Show("Data Updated Successfully", "Successful Entry", MessageBoxButtons.OK);

textBox1.Enabled = true;

}

else

{

sqlQuery = @"INSERT INTO [Stock].[dbo].[Warehouse]([WarehouseName],[Incharge],[Phone],[Address],[Email]) VALUES ('" + textBox1.Text.Trim() + "','" + textBox2.Text.Trim() + "','" + textBox3.Text.Trim() + "','" + textBox5.Text.Trim() + "','" + textBox4.Text.Trim() + "')";

MessageBox.Show("Data inserted Successfully", "Successful Entry", MessageBoxButtons.OK);

}

SqlCommand cmd = new SqlCommand(sqlQuery, conn);

cmd.ExecuteNonQuery();

button3\_Click(sender, e);

conn.Close();

LoadData();

}

}

private bool IfDataExists(SqlConnection conn, String warehousename)

{

SqlDataAdapter sda = new SqlDataAdapter(@"select \* from [Warehouse] where [WarehouseName] = '" + warehousename +"'", conn);

DataTable dt = new DataTable();

sda.Fill(dt);

if (dt.Rows.Count > 0)

return true;

else

return false;

}

public void LoadData()

{

SqlDataAdapter sda = new SqlDataAdapter(@"select \* from [Stock].[dbo].[Warehouse]", conn);

DataTable dt = new DataTable();

sda.Fill(dt);

dataGridView1.Rows.Clear();

foreach (DataRow item in dt.Rows)

{

int n = dataGridView1.Rows.Add();

dataGridView1.Rows[n].Cells[0].Value = item["WarehouseName"].ToString();

dataGridView1.Rows[n].Cells[1].Value = item["Incharge"].ToString();

dataGridView1.Rows[n].Cells[2].Value = item["Phone"].ToString();

dataGridView1.Rows[n].Cells[3].Value = item["Email"].ToString();

dataGridView1.Rows[n].Cells[4].Value = item["Address"].ToString();

}

}

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

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

textBox1.Enabled = true;

textBox1.Focus();

}

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

{

DialogResult dialog = MessageBox.Show("Do you really want to Delete the Data?", "Exit", MessageBoxButtons.YesNo);

if (dialog == DialogResult.Yes)

{

if (isEmpty())

{

MessageBox.Show("Nothing to detete please check again?????", "ERROR?", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Focus();

}

else

{

conn.Open();

var sqlQuery = "";

sqlQuery = @"DELETE FROM [Stock].[dbo].[Warehouse] WHERE [WarehouseName] = '" + textBox1.Text.Trim() + "'";

MessageBox.Show("Data Deleted Successfully", "Successful Entry", MessageBoxButtons.OK);

SqlCommand cmd = new SqlCommand(sqlQuery, conn);

cmd.ExecuteNonQuery();

textBox1.Enabled = true;

button3\_Click(sender, e);

conn.Close();

LoadData();

}

}

}

private bool isEmpty()

{

if (textBox1.Text == "")

{

return true;

}

else

return false;

}

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

{

LoadData();

}

private void dataGridView1\_CellMouseDoubleClick(object sender, DataGridViewCellMouseEventArgs e)

{

}

private void dataGridView1\_MouseDoubleClick\_1(object sender, MouseEventArgs e)

{

}

private void dataGridView1\_MouseDoubleClick(object sender, MouseEventArgs e)

{

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

textBox2.Text = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

textBox3.Text = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();

textBox4.Text = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();

textBox5.Text = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();

textBox1.Enabled = false;

}

}

}

**Warehouse\_Report Page Coding:**

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 Stock

{

public partial class Warehouse\_Report : Form

{

public Warehouse\_Report()

{

InitializeComponent();

}

private void Warehouse\_Report\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'StockDataSet6.Warehouse' table. You can move, or remove it, as needed.

this.WarehouseTableAdapter.Fill(this.StockDataSet6.Warehouse);

this.reportViewer1.RefreshReport();

}

}

}

**FUTURE SCOPE OF APPLICATION:**

This application can be easily implemented under various situations. We can add new features as and when we require. Reusability is possible as and when require in this application. There is flexibility in all the modules.

**SOFTWARE SCOPE:**

• **Extensibility**: This software is extendable in ways that its original developers may not expect. The following principles enhance extensibility like hide data structure, avoid traversing multiple links or methods, avoid case statements on object type and distinguish public and private operations.

• **Reusability**: Reusability is possible as and when require in this application. We can update it next version. Reusable software reduces design, coding and testing cost by amortizing effort over several designs. Reducing the amount of code also simplifies understanding, which increases the likelihood that the code is correct. We follow up both types of reusability.

• **Understandability:** A method is understandable if someone other than the creator of the method can understand the code (as well as the creator after a time lapse). We use the method, which small and coherent helps to accomplish this.

• **Cost-effectiveness:** Its cost is under the budget and make within given time period. It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy the entire requirement. Scope of this document is to put down the requirements, clearly identifying the information needed by the user, the source of the information and outputs expected from the system.

**Conclusion**

While developing the system a conscious effort has been made to create and develop a software package, making use of available tools, techniques and resources – that would generate a proper System While making the system, an eye has been kept on making it as user-friendly, as cost-effective and as flexible as possible. As such one may hope that the system will be acceptable to any user and will adequately meet his/her needs. As in case of any system development processes where there are a number of shortcomings, there have been some shortcomings in the development of this system also. The project is still under modification.

BIBLIOGRAPHY

Books referenced:-

* Visual Basics .NET Programming
* Visual Basics 2010 Programming
* Introduction To Programming with Visual Basic .C By Gary J. Bronson

Weblinks:-

* [http://www.dreamincode.net](http://www.dreamincode.net/)
* [http://www.a1vbcode.com](http://www.a1vbcode.com/)