## SYNOPSIS

## NASC STOCK MAINTENANCE SYSTEM

NASC-Stock Maintenance System. In this VB.net article we will discuss about SMS. This software which used in maintaining the Stock details. Our main focus in this stock software to keep records for office and department stock in our college full details.

In this software Admin and User can operate the system. Admin has to first add the user. user can insert the Stock all the details .

We provide the search stock by stockid wise, date wise, Department wise, staffid wise our project. This software help to us find the already register stock about full records.

**Project Type:** Desktop / windows

**Front end:** Visual basic.net

**Back end:** Microsoft Access

**Project title:** NASC-Stock Maintenance System

# INTRODUCTION

## 1.1.OBJECTIVES

The Stock Maintenance System was developed as an attempt to take a record of register our college Stock Maintenance by restricting such a large database to that of particular Stock and Stock type Date of product. The system provides the facility of viewing both the stock type and date of stock maintenance. It can also search for Stock type and which kind of stocks done. And the particular stocks will done on the day. Also Insertion and Deletion of records by the administrator. The GUI used for the project provide instruction and the various actions performed administrator on button clicks or on selection of items, guide the user through a series of pages as in the project.

This System can be used as an application for the college stock organizer of the college to manage the Stock with regard to register details about product. Staffs logging should be able to upload their information for Stoks. The key feature of this project that, It is a one time registration. It provides the facility of maintaining the details of stock. It also provides a requested list of who is booking the Stock based on given query. This project will private colleges to practice full IT deployment.

This will also help in fast access procedures in stock reports related activities. The users can access easily to this and the data can be retrieved easily in no time. A registered staff can directly login using user ID and password, submit Stock details. In the staff registration form, we can give stock id, stock name, stock type, register date and some other stock details will uploaded.

The stock records of the requesting and register stock dates will be provided by the administrator. The administrator plays an important role in our project. They provide approval of stock registration and updating.

## 1.2.SYSTEM SPECIFICATION

### 1.2.1 HARDWARE CONFIGURATION

| | |
|---|---|
| SYSTEM | : DELL |
| HARD DISK | : 1 TB |
| MONITOR | : 15 VGA colour |
| MOUSE | : Logitech. |
| RAM | : 4 GB |
| KEYBOARD | : 110 keys enhanced. |

### 1.2.2.SOFTWARE CONFIGURATION

| | |
|---|---|
| Operating system | : Windows 7 Professional |
| Front End | : Microsoft Visual Studio .Net 2010 |
| Coding Language | : Vb.Net |
| Database | : MS Access |

## 1.3.SELECTION OF SOFTWARE

### .NET OVERVIEW

The .NET framework is an integral windows component that supports building and running the next generation of applications and XML web services. The key component of the .NET frame work are the common language run time and the .NET frame work class library, which includes ADO.NET, ASP.NET and windows forms. The .NET framework provides a managed execution environment simplified development and deployment and integration with a wide variety of programming languages.

### .NET OVERVIEW

.NET is the Microsoft's development model in which software becomes platform and device independent and data becomes available over the Internet. The .Net Framework is the infrastructure of .NET.

.NET is a platform that can be used for building and running the next generation of Microsoft Windows and Web applications. The goal of the Microsoft .NET platform is to simplify web development.

The .Net Framework is language neutral. Currently it supports C++, C#, Visual Basic, and JS script (The Microsoft version of JavaScript).Microsoft's Visual Studio.NET is a common development environment for the new .NET Framework.

### FEATURES OF VB .NET:

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There's no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

".NET" is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

**THE .NET FRAMEWORK**

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).
2. A hierarchical set of class libraries.

The CLR is described as the "execution engine" of .NET. It provides the environment within which programs run. The most important features are

- Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
- Memory management, notably including garbage collection.
- Checking and enforcing security restrictions on the running code.
- Loading and executing programs, with version control and other such features.
- The following features of the .NET framework are also worth description:

**MANAGED CODE**

The code that target's .NET, and which contains certain extra Information - "metadata" - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

**MANAGED DATA**

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you're using, impose certain constraints on the features

available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn't get garbage collected but instead is looked after by unmanaged code.

**COMMON TYPE SYSTEM**

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn't attempt to access memory that hasn't been allocated to it.

**COMMON LANGUAGE SPECIFICATION**

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

**THE CLASS LIBRARY**

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary.

The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity.

The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

**LANGUAGES SUPPORTED BY .NET**

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications and XML Web services. The .NET framework supports new versions of Microsoft's old favorites Visual Basic and C++ (as VB.NET and Managed C++), but there are also a number of new additions to the family.

Visual Basic .NET has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others. Visual Basic also now supports structured exception handling, custom attributes and also supports multi-threading.

Visual Basic .NET is also CLS compliant, which means that any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET.

Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new .NET Framework.

Active State has created Visual Perl and Visual Python, which enable .NET-aware applications to be built in either Perl or Python. Both products can be integrated into the Visual Studio .NET environment. Visual Perl includes support for Active State's Perl Dev Kit.

Other languages for which .NET compilers are available include

- FORTRAN

<table>
<tr><td>ASP.NET<br>XML WEB SERVICES</td><td>Windows Forms</td></tr>
<tr><td colspan="2">Base Class Libraries</td></tr>
<tr><td colspan="2">Common Language Runtime</td></tr>
<tr><td colspan="2">Operating System</td></tr>
</table>

- COBOL
- Eiffel

C#.NET is also compliant with CLS (Common Language Specification) and supports structured exception handling. CLS is set of rules and constructs that are supported by the CLR (Common Language Runtime). CLR is the runtime environment provided by the .NET Framework; it manages the execution of the code and also makes the development process easier by providing services.

C#.NET is a CLS-compliant language. Any objects, classes, or components that created in C#.NET can be used in any other CLS-compliant language. In addition, we can use objects, classes, and components created in other CLS-compliant languages in C#.NET .

**CONSTRUCTORS AND DESTRUCTORS:**

Constructors are used to initialize objects, whereas destructors are used to destroy them. In other words, destructors are used to release the resources allocated to the object. In C#.NET the sub finalize procedure is available. The sub finalize procedure is used to complete the tasks that must be performed when an object is destroyed. The sub finalize procedure is called automatically when an object is destroyed. In addition, the sub finalize procedure can be called only from the class it belongs to or from derived classes.

**GARBAGE COLLECTION**

Garbage Collection is another new feature in C#.NET. The .NET Framework monitors allocated resources, such as objects and variables. In addition, the .NET Framework automatically releases memory for reuse by destroying objects that are no longer in use.

In C#.NET, the garbage collector checks for the objects that are not currently in use by applications. When the garbage collector comes across an object that is marked for garbage collection, it releases the memory occupied by the object.

**OVERLOADING**

Overloading is another feature in C#. Overloading enables us to define multiple procedures with the same name, where each procedure has a different set of arguments. Besides using overloading for procedures, we can use it for constructors and properties in a class.

**MULTITHREADING:**

C#.NET also supports multithreading. An application that supports multithreading can handle multiple tasks simultaneously, we can use multithreading to decrease the time taken by an application to respond to user interaction.

**STRUCTURED EXCEPTION HANDLING**

C#.NET supports structured handling, which enables us to detect and remove errors at runtime. In C#.NET, we need to use Try…Catch…Finally statements to create exception handlers. Using Try…Catch…Finally statements, we can create robust and effective exception handlers to improve the performance of our application.

**MS ACCESS**

Microsoft Access is used to create simple database solutions. Access tables support a variety of standard field types, indices, and referential integrity. Access also includes a query interface, forms to display and enter data, and reports for printing. The underlying Jet database, which contains these objects, is multi-user-aware and handles record-locking and referential integrity including cascading updates and deletes.

Simple tasks can be automated through macros with point-and-click options. Microsoft Access is very popular among non-programmers who can create visually pleasing and relatively advanced solutions on their own. It is also easy to place a database on a network and have multiple users share and update data without overwriting each other's work. Data is locked at the record level which is significantly different from Excel which locks the entire spreadsheet.

Access also supports the creation of Pass-Through queries. These are queries that can be linked to external data sources through the use of ODBC connections on the local machine. This enables users to interact with data stored outside the Access programme without using linked Tables. The Pass-Through queries are written using the SQL syntax supported by the external data source.

When developing Reports that are linked to Queries placing or moving items in the design view of the Report, Access runs the linked query in the background on any placement or movement of an item in that Report. If the Report is linked to a Query that takes a long time to

return records this forces you to wait until the query has run before you can add/edit or move the next item in the Report (this feature cannot be turned off).

# SYSTEM STUDY

## 2.1 EXISTING SYSTEM

To get the information from an existing system is called data analysis approach. It simply asks the user what information is required. It relies heavily on user to articulate information needs

The system, which is followed at present, is a manual system. The system consists of Book of Accounts that has to be maintained in all aspects. Printing work are difficult. In the existing system each and every time a reference should be made manually. There are high possibilities to commit errors and mistakes, which leads to produce the wrong statements to the management. Report generation is also not an easy task. Another important drawback of existing system is time factor. It will not help the management to solve the problem in time. The management could not act on any issue purchase of production unless getting the reports in time controls cannot be used.

Currently, The purpose of the stock manager registering the stock for our kinds of products.

### 2.1.1 DRAWBACKS

- The existing system provides the manual training with many troubles that has been interrupting for both staff and the stock manager.
- The burden on the Stock Manager is more as the he or she has to maintain bunches of the copies of Stock records.

## 2.2 PROPOSED SYSTEM

After studying the present manual system a planned manual system, a planned computer based system is proposed. The proposed system will take care of all the necessary and sufficient information required for the effective maintenance of accounts. The scope of the project involves development of the processes related to the accounts for each transaction. The application can be used for every trade firm that wants to be computerized.

The project will be very useful to handle all the situations very efficiently. The form is used to get information from the user and until he will not fill the all required information in correct format he would not be able to do transaction. Also the calculations are very easy to perform, within only a few seconds a person can calculate that if handled manually will take a few days. Also list of account can be seen as per the requirements. Cleanliness also can be maintained in the record keeping whenever some updates are required, the old information is deleted and instead of that old information the new is written. Computerized firms will also not cause frustration to the customer, as they will get the quick responses.

### 2.1.1 FEATURES OF PROPOSED SYSTEM

This not only gives quick and readymade access to the information stored. The computerized handling of all banking procedure adds a new advantage to it and that is of its capacity to store volume of data in a small region. One can avoid away the large amount of labor and also saves a lot of time of the customers and employees of the firm.

Some of the important advantages of the project are as follows:

- It saved a lot of time.
- It saved manpower.
- It is easy to understand.

## SYSTEM DESIGN AND DEVELOPMENT

## (CONTENT FOR SYSTEM DESIGN)

### 3.1 INPUT DESIGN

Input Design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system. A large number of problems with a system can usually be tracked backs to fault input design and method. Needless to say, therefore, that the input data is the life blood of a system and have to be analysed and designed with utmost case and consideration. The decisions made during the input design are

- To provide cost effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is understand by the user.

System analysis decide the following input design details like, what data to input, what medium to use, hoe the data should be arranged or coded, data items and transactions needing validations to detect errors and at last the dialogue to guide user in providing input.

Input data of a system may not be necessarily be raw data captured in the system from scratch. These can also be the output of another system or sub system. The design of input covers all phases of input from the creation of initial data to actual entering the data to the system for processing. The design of inputs involves identifying the data needed, specifying the characteristics of each data item, capturing and preparing data for computer processing and ensuring correctness of data.

### 3.2.OUTPUT DESIGN

Output Design generally refers to the results and information's that are generated by the system for many end-users, output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

The objective of a system finds its shape in terms of the output. The analysis of the objective of a system leads to determination of outputs. Outputs of a system can face various forms. The most common are reports, screen displays, printed forms, graphical drawings etc., The output also vary in terms of their contents frequency, timing and format. The user of the output from a system are the justification for its existence. If the output are inadequate in any way, the system are itself is adequate. The basic requirements of output are that it should be accurate, timely and appropriate, in terms of content, medium and layout for its intended purpose.

When designing output, system analysis most accomplish things like, to determine what information to be present, to decide whether to display or print the information and select the output medium and to decide how to distribute the output to intended recipients.

External outputs are those whose destination will be outside the organization and which require special attention as they project the image of the organization. Internal outputs are those whose destination is within the organization. it is to be carefully designed as they are the users main interface with the system.

### 3.3.DATABASE DESIGN

The database design converts the data model developed in logical design to a database definition that is supported by database software. Database design proceeds through a number of steps.

The first step is independent of the kind of DBMS used. This step converts the conceptual entity relationships model to a set of record type is known as the logical record structures. (LRS)

The next database design step converts the LRS to a database definition. These steps use techniques that depend on the DBMS. DBMS dependent techniques are needed here because different DBMS support different kind of links between the records. Such links are used to retrieve records by following the link from one record to another. Database design depends on the structure supported by DBMS and uses techniques appropriate to these structures.

DBMS dependent design proceeds in two stages. The first step is logical design. Logical design defines the DBMS record types and the links between them. The next step is physical design. This step chooses a physical organization that supports the methods uses to accesses the databases.

The first thing in system design the input and output screen design according to the needs of the user the input and output design are related with each other in the sense that the accuracy of output data depends on the accuracy of the input data and processing of the input data and processing of the input data. Thus for this proposed system the input and output design are in the form of the forms. In the forms based interface design, the user can easily input data without any difficulty. The forms are also designed in such a way that the blanks which are to be filled up are made in a sequential pattern. It also helps the user while entering the data.

## 3.4.DESCRIPTION OF MODULES

Modules are:

- Staff Details
- Office stock Details.
- Department stock Details

### STAFF

Staff module which helps us to register the stock and details about stock name, stock type, stock date they need. In the staff module they can upload their requesting to the administrator through their LAN. And also get the information about the details of the event. Though this module they receive the acknowledgment to their registering from the stock manager.

### OFFICE STOCK

Office Stock module which helps us to maintenance office stock will be maintained. and when the product was maintenance. What is the name of the office product. How many products registered is maintain. This kind of information can get then the office stock module

### DEPARTMENT STOCK

Department Stock module which helps us to maintenance department stock will be maintenance. and when the product was maintained. What is the name of the office product. How many products registered is maintain. This kind of information can get then the department stock module.

# TESTING AND IMPLEMENTATION

## 4.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

## 4.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

## 4.3 VALIDATION TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

## 4.4 WHITEBOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

## 4.5 BLACKBOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

## 4.6 SYSTEM IMPLEMENTATION

Implementation includes all those activities that take place to convert from the old system to the new. The old system consists of manual operations, which is operated in a very different manner from the proposed new system. A proper implementation is essential to provide a reliable system to meet the requirements of the organizations. An improper installation may affect the success of the computerized system.

There are several methods for handling the implementation and the consequent conversion from the old to the new computerized system.

The most secure method for conversion from the old system to the new system is to run the old and new system in parallel. In this approach, a person may operate in the manual older processing system as well as start operating the new computerized system. This method offers high security, because even if there is a flaw in the computerized system, we can depend upon the manual system. However, the cost for maintaining two systems in parallel is very high. This outweighs its benefits.

Another commonly method is a direct cut over from the existing manual system to the computerized system. The change may be with in a week or with in a day. There are no parallel activities. However, there is no remedy in case of a problem. This strategy requires careful planning.

A working version of the system can also be implemented in one part of the organization and the personnel will be piloting the system and changes can be made as and when required. But this method is less preferable due to the loss of entirety of the system.

**IMPLEMENTATION PLAN**

The implementation plan includes a description of all the activities that must occur to implement the new system and to put it into operation. It identifies the personnel responsible for the activities and prepares a time chart for implementing the system.

The implementation plan consists of the following steps.

- List all files required for implementation.
- Identify all data required to build new files during the implementation.
- List all new documents and procedures that go into the new system.
- The implementation plan should anticipate possible problems and must be able to deal with them.
- The usual problems may be missing documents; mixed data formats between current and files, errors in data translation, missing data etc.

# CONCLUSION AND FUTURE ENCHANCEMENT

## 5.1 CONCLUSION

"NASC Stock Maintenance System" has been used in the Stock Manager. Although the project is just a small part of the whole system, but an effort has been made to improve upon the existing GUI (Graphical User Interface) and to make it more attractive and user friendly and has been appreciated by the users who tested it. The Project is user friendly in the sense that there are buttons for navigation to each page and it has benefitted the developer in the sense that various concepts of Vb.net could be implemented as well as newer concepts of MS Access could be learnt while working on the project.

## 5.2 FUTURE ENCHANCEMENT

- Further enhancements can be made to the application, so that the web site functions very attractive and useful manner than the present one.
- NASC Stock Maintenance System used in the Stock admin.
- Enhanced system is created in a user-friendly manner with appropriate message guiding the user.
- In future using Biometric system Login the users.
- The site works according to the restrictions provided in their respective browsers.
- Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature.

## BIBLIOGRAPHY

### BOOK REFERENCE

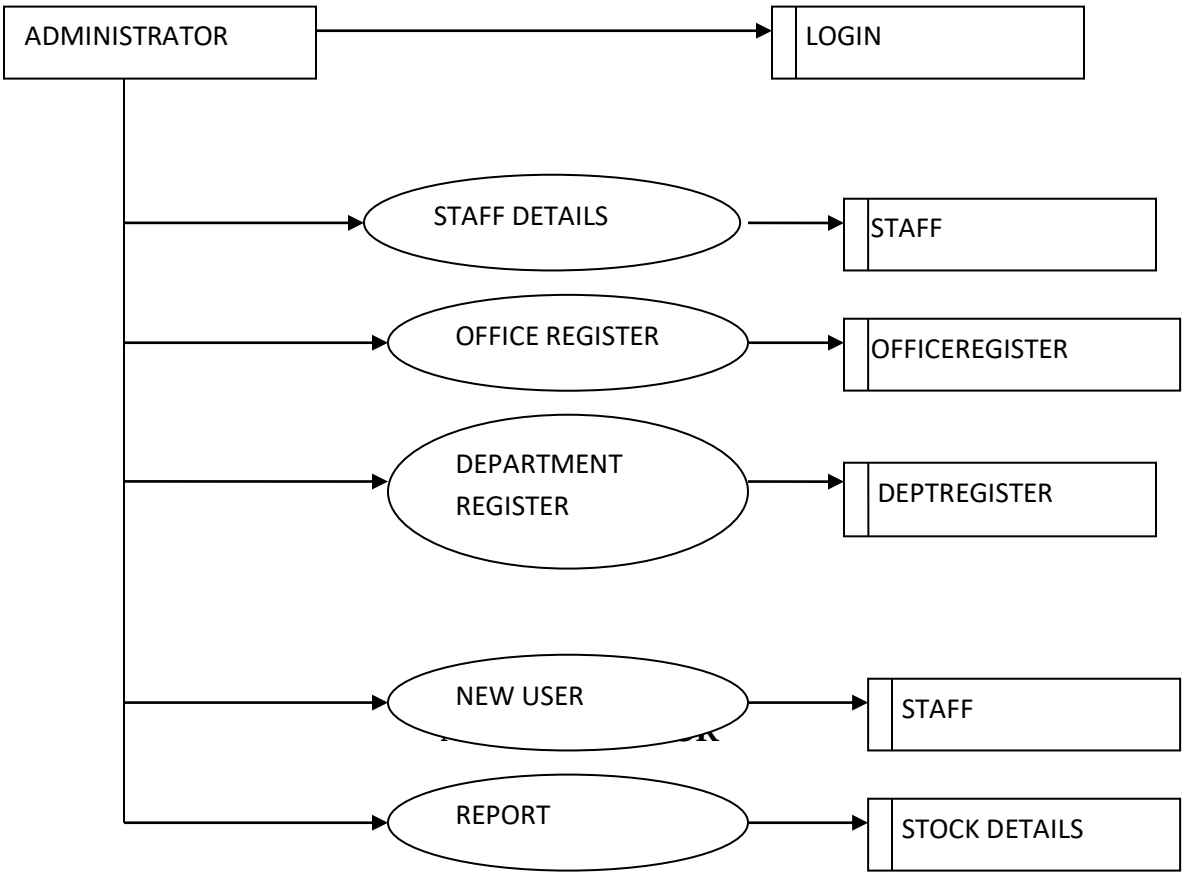

1. Microsoft Visual Basic .NET Programming for the Absolute Beginner, Jonathan S. Harbour
2. Elmasri & Navathe "Fundamentals of DATABASE SYSTEMS", fourth edition (2004) , Pearson Education Asia, Part2: "Relational Model: Concepts, Constraints, Languages, Design, and Programming", Part4: "Data Storage, Indexing, Query Processing, and Physical design"


### WEBSITE


* w3school.com
* code-projects.org

# APPENDICES

## A.DATAFLOW DIAGRAM



**B.TABLE STRUCTURE**

**TABLE 1**

**TABLE NAME : LOGIN**

**PURPOSE : ADMIN/USER LOGIN**

| FIELD | DATA TYPE | SIZE | DESCRIPTION | CONSTRAINTS |
|---|---|---|---|---|
| StaffID | text | 10 | STAFF ID | Primarykey |
| StaffName | text | 20 | STAFF NAME | Not null |
| Password | Text | 8 | PASSWORD | Not null |
| Usertype | Text | 5 | USER TYPE | Not null |

**TABLE 2**

**TABLE NAME : Staff**

**PURPOSE : STAFF DETAILS**

| FIELD | DATA TYPE | SIZE | DESCRIPTION | CONSTRAINTS |
|---|---|---|---|---|
| Department | Text | 25 | DEPARTMENT | Not null |
| Staffid | Text | 10 | STAFF ID | Primarykey |
| Staffname | Text | 25 | STAFF NAME | Not null |
| phonenumber | Text | 10 | Phone Number | Not null |

**TABLE 3**

**TABLE NAME : officeregister**

**PURPOSE : OFFICE STOCK REGISTER**

| FIELD | DATA TYPE | SIZE | DESCRIPTION | CONSTRAINTS |
|---|---|---|---|---|
| DateTime | Text | 25 | DATE AND TIME | Not null |
| Company | Text | 20 | COMPANY | Not null |
| Location | Text | 20 | LOCATION | Not null |
| Staffid | Text | 10 | STAFF ID | Not null |
| StaffName | Text | 20 | STAFF NAME | Not null |
| ProductID | Text | 8 | PRODUCT ID | Primarykey |
| ProductName | Text | 20 | PRODUCT NAME | Not null |
| Producttype | Text | 15 | PRODUCT TYPE | Not null |
| Quantity | Text | 10 | QUANTITY | Not null |
| ProductPrize | Text | 10 | PRODUCT PRIZE | Not null |
| Total | Text | 10 | TOTAL | Not null |
| Amount | Text | 10 | AMOUNT | Not null |

**TABLE 4**

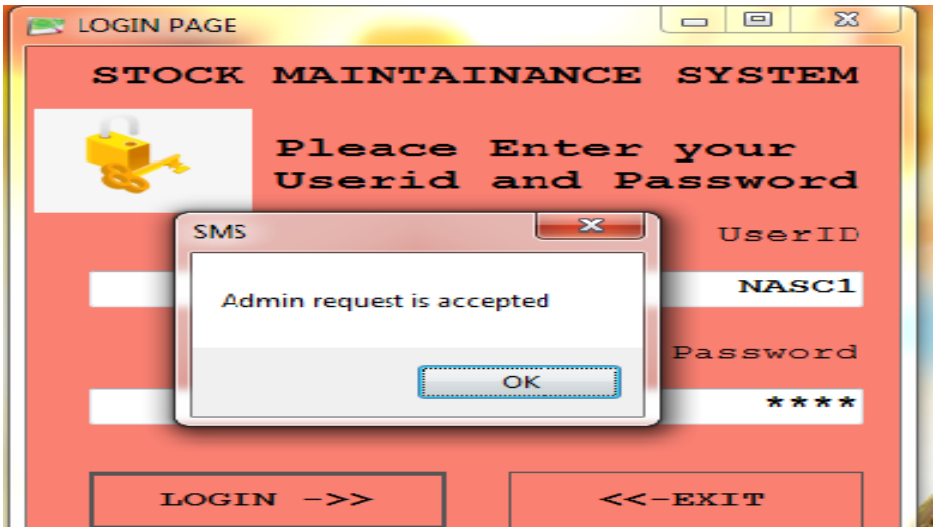
**TABLE NAME : deptregister**

**PURPOSE : DEPARTMENT STOCK REGISTER**

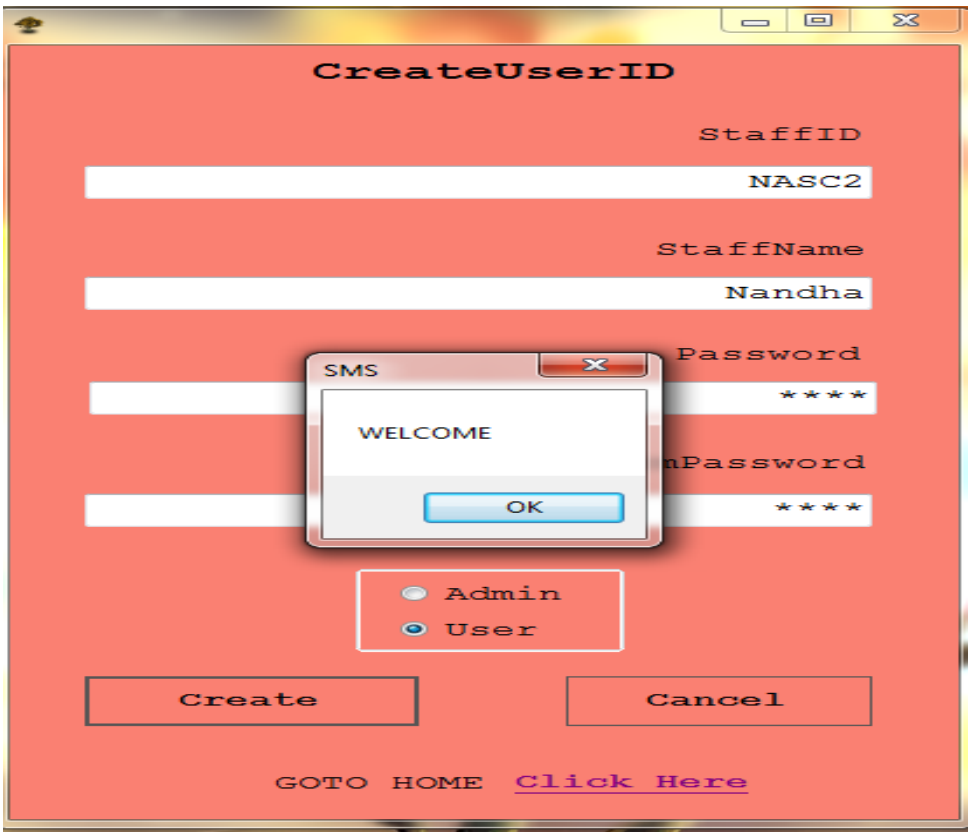
| FIELD | DATA TYPE | SIZE | DESCRIPTION | CONSTRAINTS |
|---|---|---|---|---|
| DateTime | Text | 25 | DATE AND TIME | Not null |
| Company | Text | 20 | COMPANY | Not null |
| Location | Text | 20 | LOCATION | Not null |
| department | Text | 25 | DEPARTMENT | Not null |
| StaffID | Text | 10 | STAFF ID | Not null |
| StaffName | Text | 20 | STAFF NAME | Not null |
| ProductID | Text | 8 | PRODUCT ID | Primarykey |
| ProductName | Text | 20 | PRODUCT NAME | Not null |
| Producttype | Text | 15 | PRODUCT TYPE | Not null |
| Quantity | Text | 10 | QUANTITY | Not null |
| ProductPrize | Text | 10 | PRODUCTPRIZE | Not null |
| Total | Text | 10 | TOTAL | Not null |
| Amount | Text | 10 | AMOUNT | Not null |

## C.SAMPLE INPUT

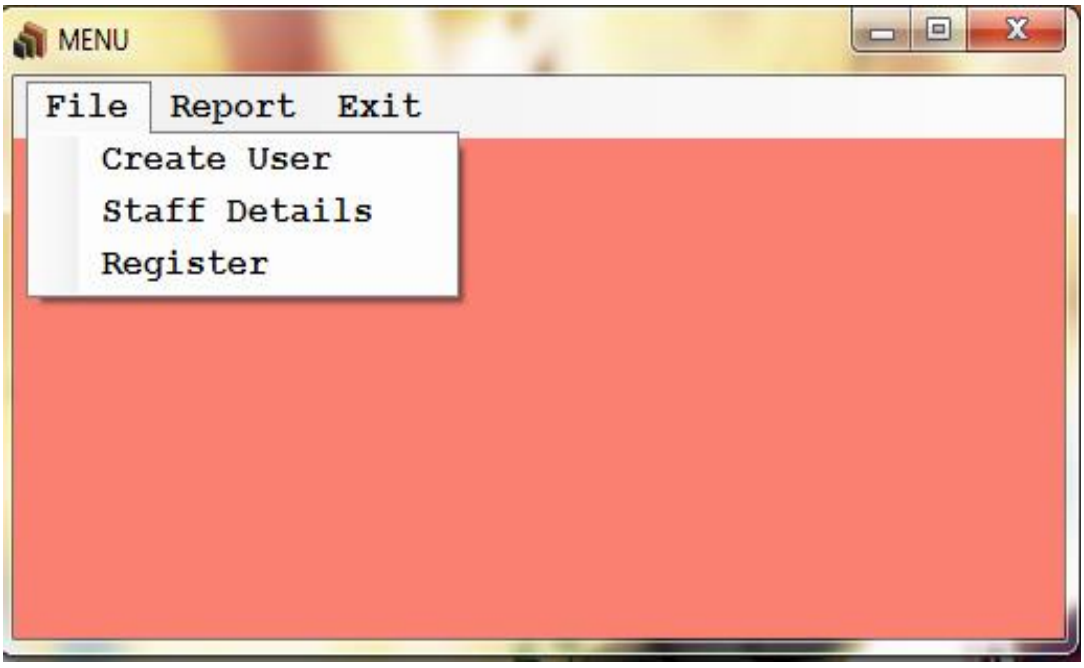
## LOGIN

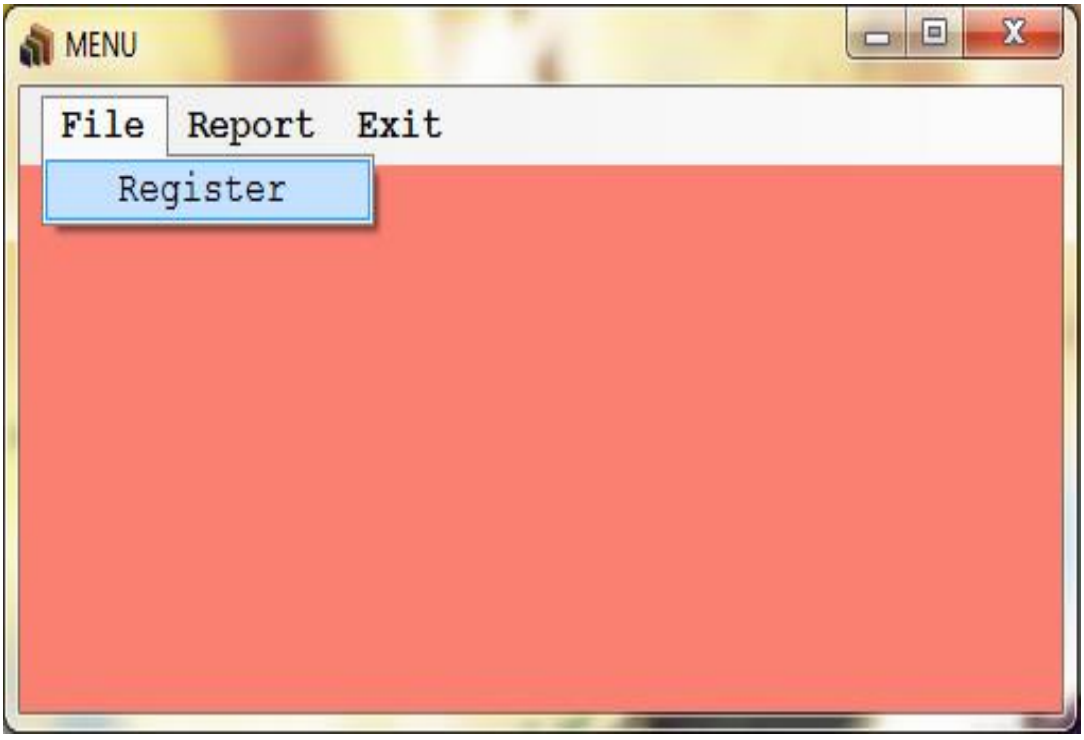



## LOGIN REGISTRATION

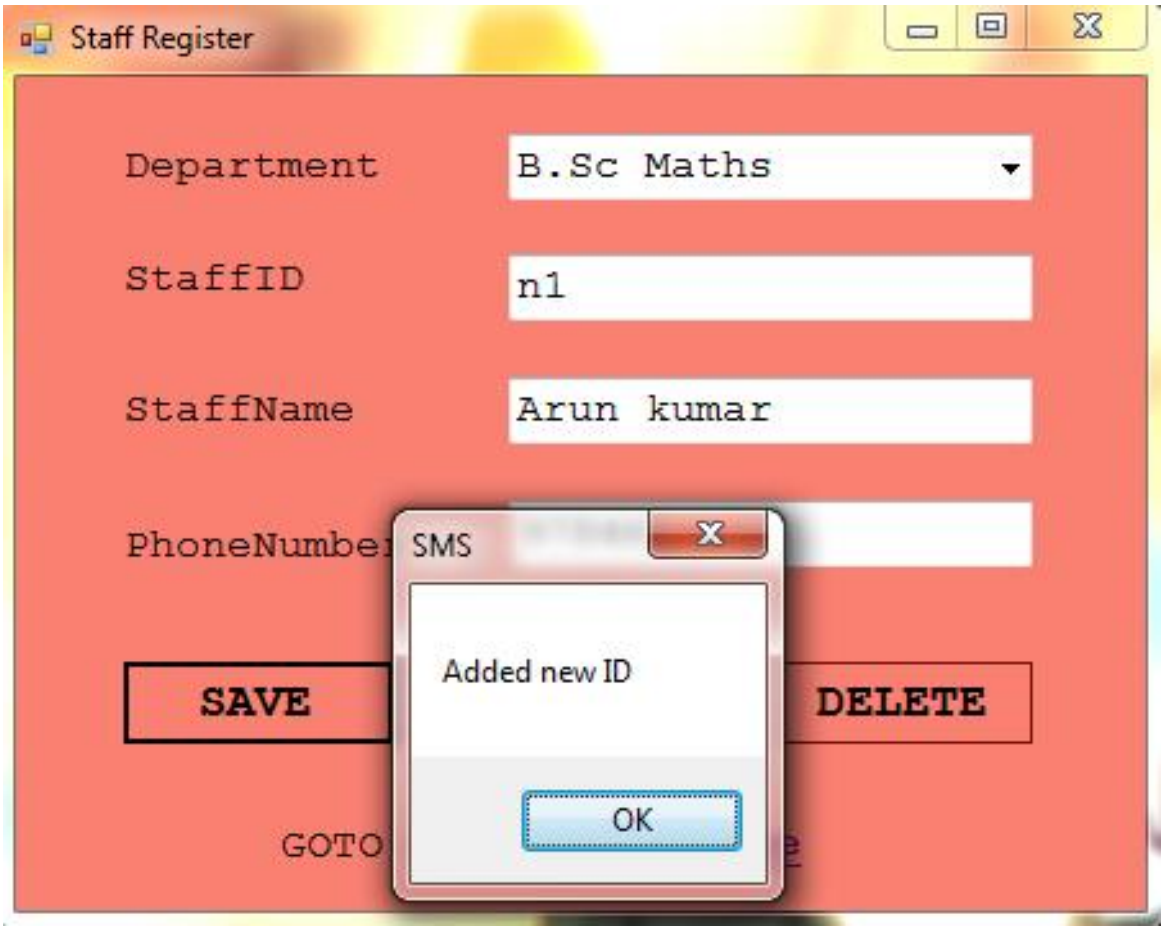


**HOME**

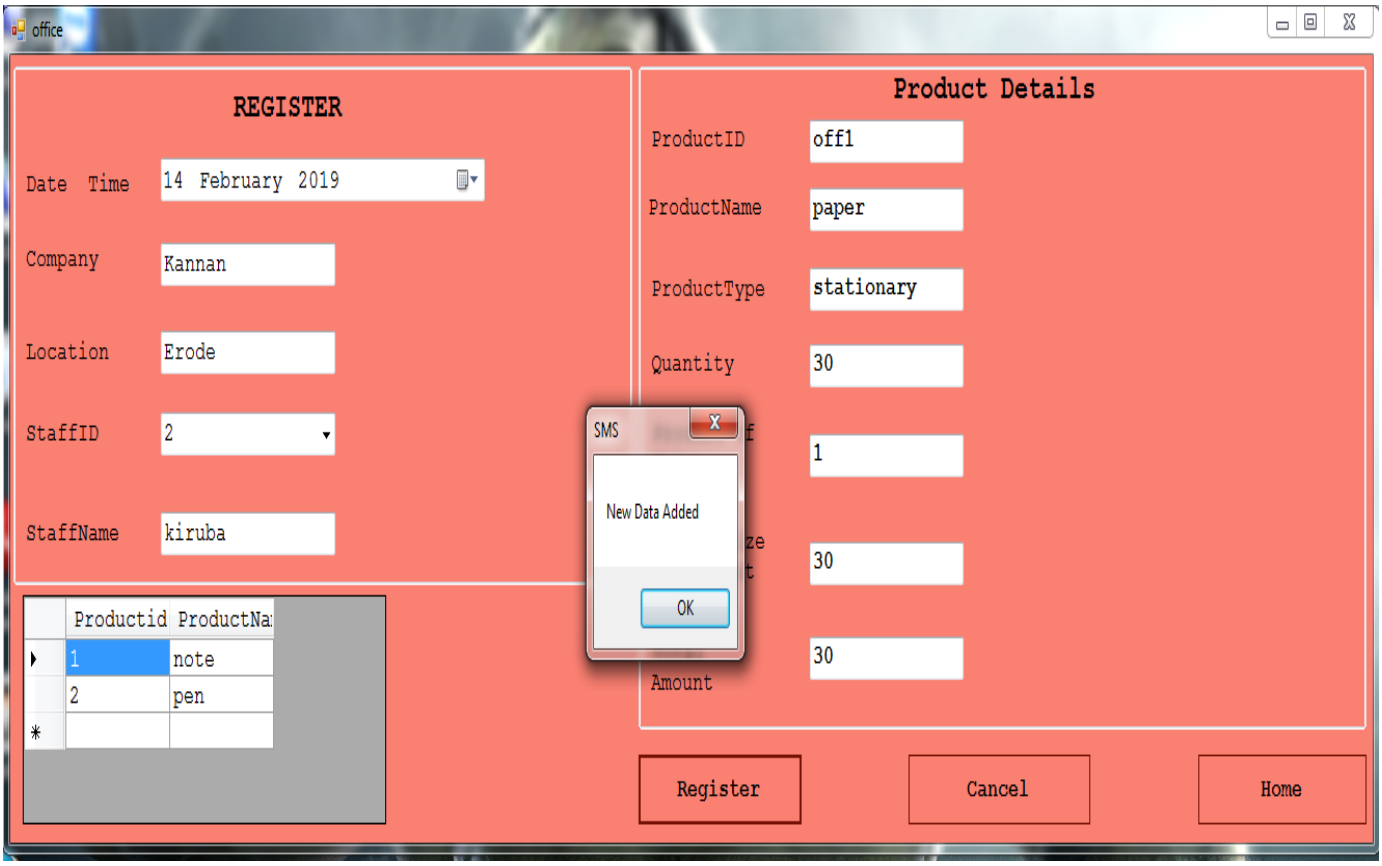

MENU
File
Report
Exit
Create User
Staff Details
Register

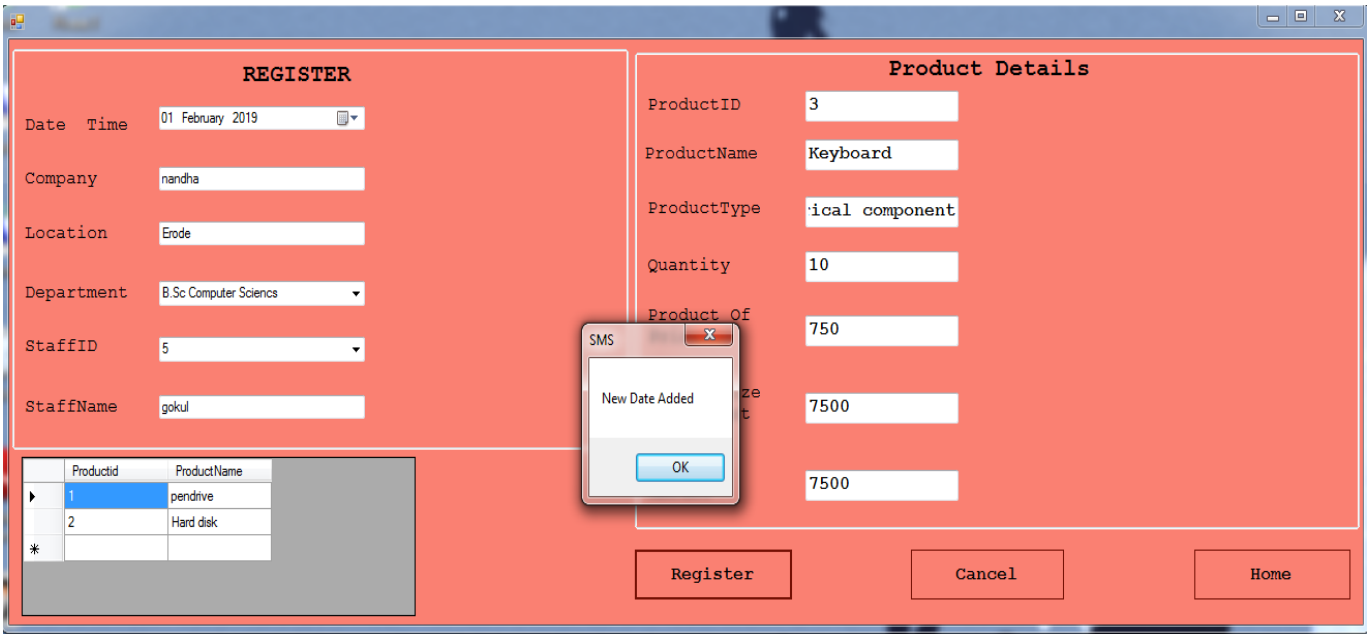

MENU
File
Report
Exit
Register

**Staff Register**

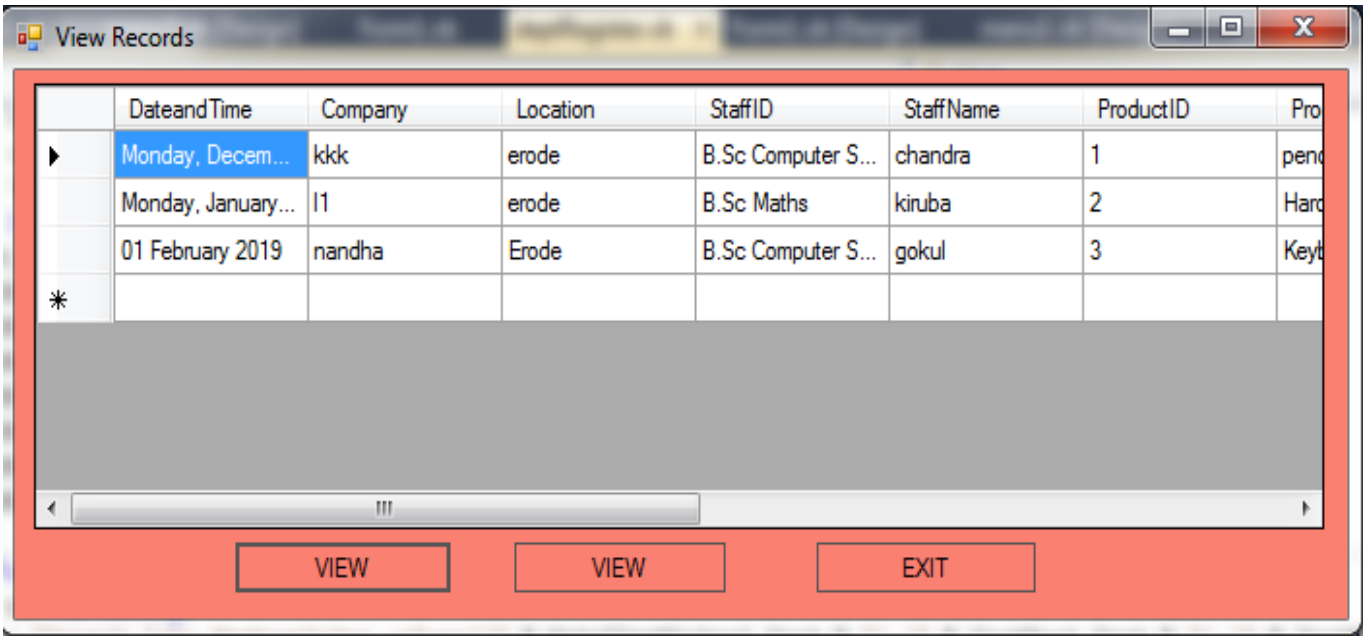


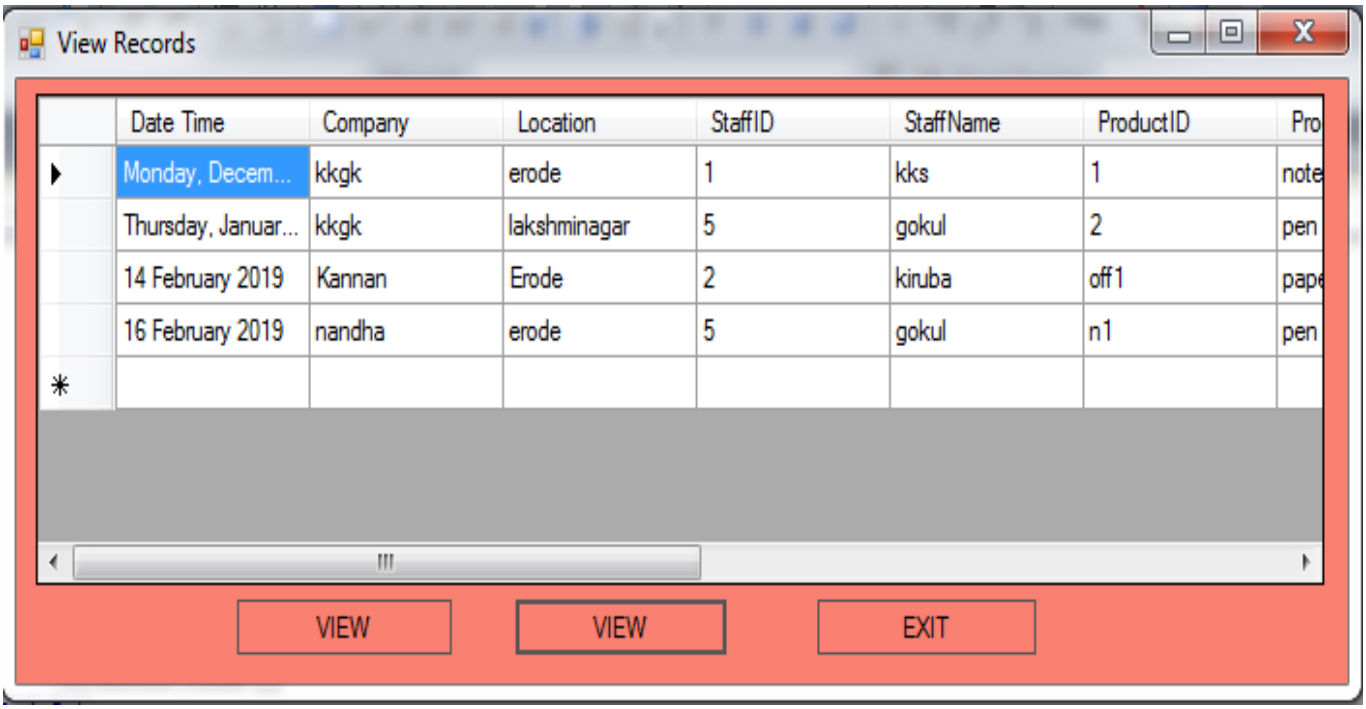
## REGISTRATION FORM



## VIEW REGISTER STOCK DETAILS



**D.SAMPLE CODING**

**LOGIN**

```
Imports System.Data.OleDb
Public Class Login
   Dim con As New OleDbConnection ("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=D:\project\SMS.mdb;Persist Security Info=True")
   Dim cmd As New OleDbCommand
   Dim dr As OleDbDataReader
   Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
      con.Open()
   End Sub

   Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
      End
   End Sub

   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
      Dim userfound As Boolean = False
      Dim str As String
      str = "select StaffID,Password,Usertype from LOGIN where StaffID='" & TextBox1.Text &
"' and Password='" & TextBox2.Text & "'"
      cmd = New OleDbCommand(str, con)
      cmd.ExecuteNonQuery()
      dr = cmd.ExecuteReader()
      While dr.Read()
         userfound = True
         TextBox3.Text = dr("Usertype").ToString()
```

```
        End While
        If userfound = True Then
            If TextBox3.Text <> "Admin" Then
                MsgBox("User request is accepted")
                menu2.Show()
                TextBox1.Clear()
                TextBox2.Clear()
                Me.Hide()
            Else
                MsgBox("Admin request is accepted")
                menu1.Show()
                TextBox1.Clear()
                TextBox2.Clear()
                Me.Hide()
            End If
        Else
            MsgBox("Userid or Password Incorrectly")
            TextBox1.Text = ""
            TextBox2.Text = ""
            TextBox1.Focus()
        End If
    End Sub
```

**REGISTER USER**

```
Imports System.Data.OleDb
Public Class createuser
    Dim con As New OleDbConnection ("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=D:\project\SMS.mdb;Persist Security Info=True")
    Dim cmd As New OleDbCommand
```

```
    Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        con.Open()
        Label10.Text = " "
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        TextBox1.Clear()
        TextBox2.Clear()
        TextBox3.Clear()
        TextBox4.Clear()
        TextBox1.Focus()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim str As String
        str = "insert into LOGIN values('" & TextBox1.Text & "','" & TextBox2.Text & "','" &
TextBox3.Text & "','" & TextBox5.Text & "')"
        cmd = New OleDbCommand(str, con)
        cmd.ExecuteNonQuery()
        MsgBox(" WELCOME ")
        TextBox1.Clear()
        TextBox2.Clear()
        TextBox3.Clear()
        TextBox4.Clear()
        TextBox1.Focus()
    End Sub

    Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked
```

```
        menu1.Show()
        Me.Hide()
    End Sub

    Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RadioButton1.CheckedChanged
        If RadioButton1.Checked = True Then
            TextBox5.Text = RadioButton1.Text
        End If
    End Sub

    Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RadioButton2.CheckedChanged
        If RadioButton2.Checked = True Then
            TextBox5.Text = RadioButton2.Text
        End If
    End Sub

    Private Sub TextBox4_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs)
Handles TextBox4.LostFocus
        If Val(TextBox3.Text) = Val(TextBox4.Text) Then
            Button1.Focus()
            Label10.Text = " "
        Else
            Label10.Text = "Password is Incorrect"
            TextBox3.Clear()
            TextBox4.Clear()
            TextBox3.Focus()
        End If
    End Sub
End Class
```

**STAFF DETAILS REGISTRATION**

```
Imports System.Data.OleDb
Public Class staff
    Dim con As New OleDbConnection ("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=D:\project\SMS.mdb;Persist Security Info=True")
    Dim cmd As New OleDbCommand
    Dim dr As OleDbDataReader
    Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Label1.Click

    End Sub
    Private Sub Label3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Label3.Click

    End Sub

    Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        con.Open()
        ComboBox1.Items.Add("B.Sc Agriculture")
        ComboBox1.Items.Add("B.Sc Physics")
        ComboBox1.Items.Add("B.Sc Chemistry")
        ComboBox1.Items.Add("B.Sc Maths")
        ComboBox1.Items.Add("B.Sc Zoology")
        ComboBox1.Items.Add("B.Sc Statistics")
        ComboBox1.Items.Add("B.Sc Botany")
        ComboBox1.Items.Add("B.Sc Computer Sciencs")
        ComboBox1.Items.Add("B.Sc Fahion technology")
        ComboBox1.Items.Add("B.Com Cosmputer Science")
        ComboBox1.Items.Add("B.Sc Computer Application")
```

```
        ComboBox1.Items.Add("B.Com Computer Science")
        ComboBox1.Items.Add("B.Com Computer Application")
    End Sub

    Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked
        menu1.Show()
        Me.Hide()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        ComboBox1.Text = " "
        TextBox2.Clear()
        TextBox3.Clear()
        TextBox4.Clear()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim str As String
        str = "insert into staff values('" & ComboBox1.SelectedItem & "','" & TextBox2.Text & "','"
& TextBox3.Text & "','" & TextBox4.Text & "')"
        cmd = New OleDbCommand(str, con)
        cmd.ExecuteNonQuery()
        MsgBox("Added new ID")
        ComboBox1.Text = " "
        TextBox2.Clear()
        TextBox3.Clear()
        TextBox4.Clear()
    End Sub
```

```
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        Dim str As String
        str = "delete from staff where staffid='" & TextBox2.Text & "'"
        cmd = New OleDbCommand(str, con)
        cmd.ExecuteReader()
        ComboBox1.Text = ""
        TextBox2.Clear()
        TextBox3.Clear()
        TextBox4.Clear()
    End Sub
End Class
```

**STOCK REGISTER**

**1.Office Register**

```
Imports System.Data.OleDb
Public Class officeRegister
    Dim con As New OleDbConnection ("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=D:\project\SMS.mdb;Persist Security Info=True;")
    Dim cmd As New OleDbCommand
    Dim dr As OleDbDataReader
    Dim adp As OleDbDataAdapter
    Dim ds As New DataSet
    Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs)

    End Sub
```

```
    Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    End Sub
    Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    End Sub
    Private Sub TextBox7_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox7.TextChanged

    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        menu1.Show()
        Me.Hide()
    End Sub

    Private Sub Form5_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        con.Open()

        TextBox1.Clear()
        TextBox2.Clear()
        ComboBox1.Text = " "
        TextBox3.Clear()
        TextBox4.Clear()
        TextBox5.Clear()
        TextBox6.Clear()
        TextBox7.Clear()
```

```
        TextBox8.Clear()
        TextBox9.Clear()
        TextBox10.Clear()

        Dim s As String
        s = "select Productid,ProductName from pofficeregister "
        adp = New OleDbDataAdapter(s, con)
        adp.Fill(ds)
        DataGridView1.DataSource = ds.Tables(0)



        Dim str As String
        str = "select staffid from staff"
        cmd = New OleDbCommand(str, con)
        dr = cmd.ExecuteReader
        While dr.Read
            ComboBox1.Items.Add(dr("staffid").ToString())
        End While


    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        TextBox1.Clear()
        TextBox2.Clear()
        ComboBox1.Text = " "
        TextBox3.Clear()
        TextBox4.Clear()
        TextBox5.Clear()
        TextBox6.Clear()
```

```
        TextBox7.Clear()
        TextBox8.Clear()
        TextBox9.Clear()
        TextBox10.Clear()
    End Sub

    Private Sub ToolTip1_Popup(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PopupEventArgs)
    End Sub

    Private Sub DataGridView1_CellContentClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs)

    End Sub

    Private Sub Label18_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

        Dim str As String
        Dim userfound As Boolean = False
        str = "select ProductID,Quantity from pofficeregister where ProductID='" & TextBox4.Text
& "'"
        cmd = New OleDbCommand(str, con)
        dr = cmd.ExecuteReader()
        Dim i As Integer
        While dr.Read
            i = dr(1)
            userfound = True
```

```
        End While
        If userfound = True Then
            Dim k As String
            k = "update pofficeregister set Quantity ='" & i + Val(TextBox7.Text) & "' where
ProductID='" & TextBox4.Text & "' "
            cmd = New OleDbCommand(k, con)
            cmd.ExecuteNonQuery()
            MsgBox("updated")
        Else
            Dim g As String
            g = "insert into pofficeregister values('" & DateTimePicker1.Text & "','" &
TextBox1.Text & "','" & TextBox2.Text & "','" & ComboBox1.Text & "','" & TextBox3.Text &
"','" & TextBox4.Text & "','" & TextBox5.Text & "','" & TextBox6.Text & "','" & TextBox7.Text
& "','" & TextBox8.Text & "','" & TextBox9.Text & "','" & TextBox10.Text & "');"
            cmd = New OleDbCommand(g, con)
            cmd.ExecuteNonQuery()
            MsgBox("New Data Added")
        End If
        sample()
        TextBox4.Clear()
        TextBox5.Clear()
        TextBox6.Clear()
        TextBox7.Clear()
        TextBox8.Clear()
        TextBox9.Clear()
        TextBox10.Clear()
    End Sub

    Private Sub ComboBox1_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs)
Handles ComboBox1.LostFocus
        Dim str As String
```

```
        str = "select staffid,staffname from staff where staffid='" & ComboBox1.SelectedItem & "'"
        cmd = New OleDbCommand(str, con)
        dr = cmd.ExecuteReader
        While dr.Read
            TextBox3.Text = dr("staffname").ToString
        End While
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ComboBox1.SelectedIndexChanged

    End Sub

    Private Sub TextBox3_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs)
Handles TextBox3.LostFocus

    End Sub

    Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox3.TextChanged

    End Sub

    Private Sub TextBox9_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox9.TextChanged
        TextBox9.Text = Val(TextBox6.Text) * Val(TextBox8.Text)
    End Sub

    Private Sub TextBox8_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox8.TextChanged
```

```
    End Sub
    Public Function sample()
        Dim str As String
        str = "insert into officeregister values('" & DateTimePicker1.Text & "','" & TextBox1.Text
& "','" & TextBox2.Text & "','" & ComboBox1.SelectedItem & "','" & TextBox3.Text & "','" &
TextBox4.Text & "','" & TextBox5.Text & "','" & TextBox6.Text & "','" & TextBox7.Text & "','"
& TextBox8.Text & "','" & TextBox9.Text & "','" & TextBox10.Text & "')"
        cmd = New OleDbCommand(str, con)
        cmd.ExecuteNonQuery()
    End Function

    Private Sub GroupBox1_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GroupBox1.Enter

    End Sub

    Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RadioButton3.CheckedChanged
        If RadioButton3.Checked = True Then
            TextBox6.Text = (TextBox6.Text + RadioButton3.Text)
        End If
    End Sub

    Private Sub RadioButton4_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RadioButton4.CheckedChanged
        If RadioButton4.Checked = True Then
            TextBox6.Text = (TextBox6.Text + RadioButton4.Text)
        End If
    End Sub
```

```
    Private Sub RadioButton5_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles RadioButton5.CheckedChanged
        If RadioButton5.Checked = True Then
            TextBox6.Text = (TextBox6.Text + RadioButton5.Text)
        End If
    End Sub
End Class
```

**2.Department Register**

```
Imports System.Data.OleDb
Public Class deptRegister
    Dim con As New OleDbConnection ("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=D:\project\SMS.mdb;Persist Security Info=True")
    Dim cmd As New OleDbCommand
    Dim dr As OleDbDataReader
    Dim adp As OleDbDataAdapter
    Dim ds As New DataSet
    Private Sub TextBox3_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs)

    End Sub
    Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    End Sub
    Private Sub deptpurchase_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    End Sub
    Private Sub deptpurchase_Load_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
```

```
con.Open()

ComboBox1.Items.Add("B.Sc Agriculture")
ComboBox1.Items.Add("B.Sc Physics")
ComboBox1.Items.Add("B.Sc Chemistry")
ComboBox1.Items.Add("B.Sc Maths")
ComboBox1.Items.Add("B.Sc Zoology")
ComboBox1.Items.Add("B.Sc Statistics")
ComboBox1.Items.Add("B.Sc Botany")
ComboBox1.Items.Add("B.Sc Computer Sciencs")
ComboBox1.Items.Add("B.Sc Fahion technology")
ComboBox1.Items.Add("B.Com Cosmputer Science")
ComboBox1.Items.Add("B.Sc Computer Application")
ComboBox1.Items.Add("B.Com Computer Science")
ComboBox1.Items.Add("B.Com Computer Application")


TextBox1.Clear()
TextBox2.Clear()
ComboBox1.Text = " "
TextBox3.Clear()
TextBox4.Clear()
TextBox5.Clear()
TextBox6.Clear()
TextBox7.Clear()
TextBox8.Clear()
TextBox9.Clear()
TextBox11.Clear()

Dim s As String
s = "select Productid,ProductName from pdeptregister "
```

```
        adp = New OleDbDataAdapter(s, con)
        adp.Fill(ds)
        DataGridView1.DataSource = ds.Tables(0)

        Dim str As String
        str = "select staffid from staff"
        cmd = New OleDbCommand(str, con)
        dr = cmd.ExecuteReader
        While dr.Read
            ComboBox2.Items.Add(dr("staffid").ToString())
        End While

    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim userfound As Boolean = False
        Dim str As String
        str = "select ProductID,Quantity from pdeptregister where ProductID='" & TextBox4.Text
& "'"
        cmd = New OleDbCommand(str, con)
        dr = cmd.ExecuteReader
        Dim i As Integer
        While dr.Read
            i = dr(1)
            userfound = True
        End While
        If userfound = True Then
            Dim j As String
            j = "update pdeptregister set Quantity='" & i + Val(TextBox7.Text) & "' where
ProductID='" & TextBox4.Text & "'"
            cmd = New OleDbCommand(j, con)
```

```
            cmd.ExecuteNonQuery()
            MsgBox("updated")
        Else
            Dim s As String
            s = "insert into pdeptregister values('" & DateTimePicker1.Text & "','" & TextBox1.Text & "','" & TextBox2.Text & "','" & ComboBox1.SelectedItem & "','" & TextBox3.Text & "','" & TextBox4.Text & "','" & TextBox5.Text & "','" & TextBox6.Text & "','" & TextBox7.Text & "','" & TextBox8.Text & "','" & TextBox9.Text & "','" & TextBox11.Text & "')"
            cmd = New OleDbCommand(s, con)
            cmd.ExecuteNonQuery()
            MsgBox("New Date Added")
        End If
        dept()
        TextBox4.Clear()
        TextBox5.Clear()
        TextBox6.Clear()
        TextBox7.Clear()
        TextBox8.Clear()
        TextBox9.Clear()
        TextBox11.Clear()
    End Sub
    Public Function dept()
        Dim str As String
        str = "insert into deptregister values('" & DateTimePicker1.Text & "','" & TextBox1.Text & "','" & TextBox2.Text & "','" & ComboBox1.SelectedItem & "','" & TextBox3.Text & "','" & TextBox4.Text & "','" & TextBox5.Text & "','" & TextBox6.Text & "','" & TextBox7.Text & "','" & TextBox8.Text & "','" & TextBox9.Text & "','" & TextBox11.Text & "')"
        cmd = New OleDbCommand(str, con)
        cmd.ExecuteNonQuery()
    End Function
```

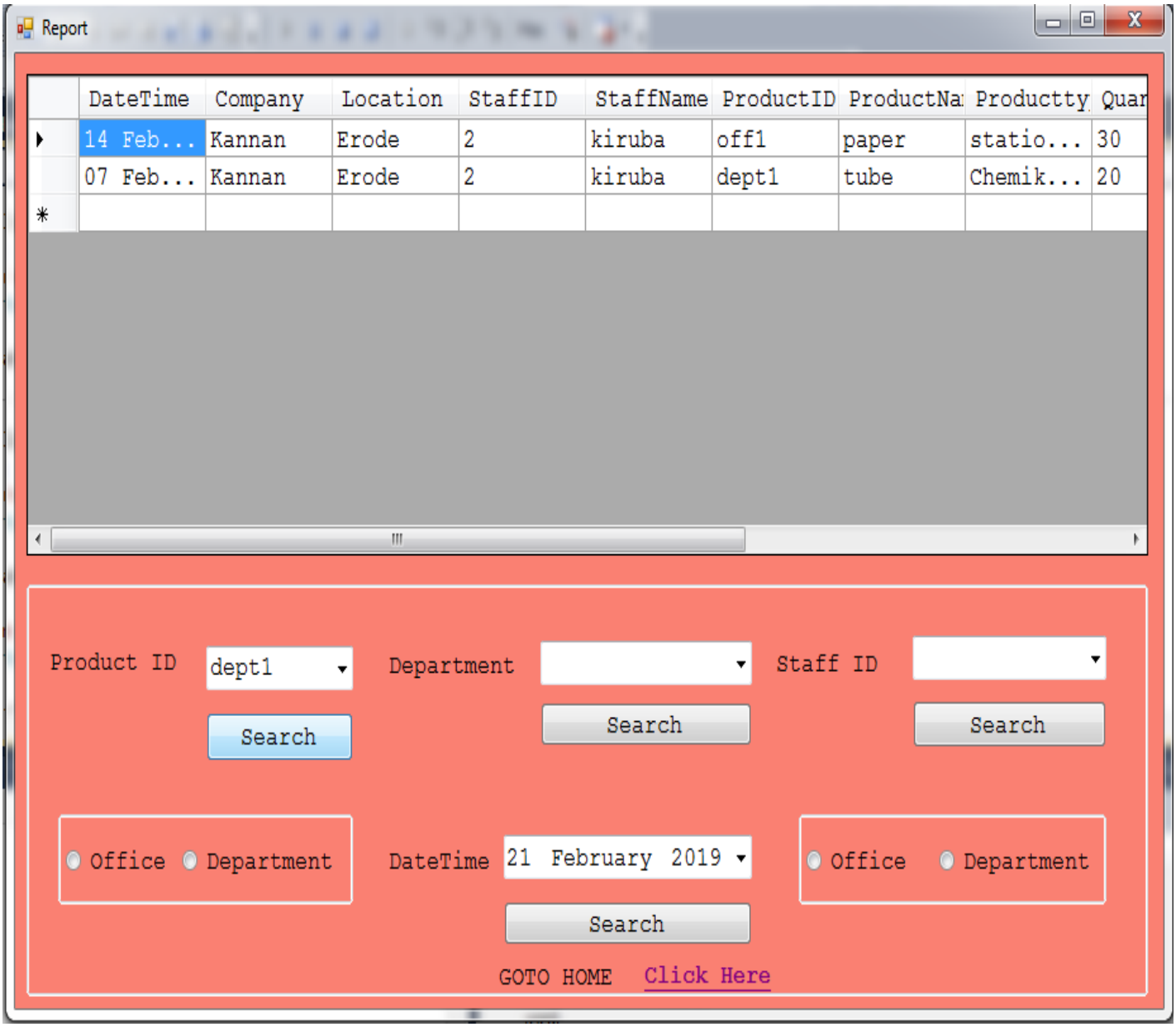
```
    Private Sub TextBox3_LostFocus1(ByVal sender As Object, ByVal e As System.EventArgs)
Handles TextBox3.LostFocus

    End Sub

    Private Sub TextBox3_TextChanged_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox3.TextChanged

    End Sub

    Private Sub TextBox8_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox8.TextChanged
        TextBox9.Text = Val(TextBox6.Text) * Val(TextBox8.Text)
    End Sub

    Private Sub Button3_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        menu2.Show()
        Me.Hide()
    End Sub

    Private Sub ComboBox2_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs)
Handles ComboBox2.LostFocus
        Dim str As String
        str = "select staffid,staffname from staff where staffid='" & ComboBox2.SelectedItem & "'"
        cmd = New OleDbCommand(str, con)
        dr = cmd.ExecuteReader
        While dr.Read
            TextBox3.Text = dr("staffname").ToString
        End While
    End Sub
```

```
    Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ComboBox2.SelectedIndexChanged

    End Sub

    Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RadioButton3.CheckedChanged
        If RadioButton3.Checked = True Then
            TextBox6.Text = (TextBox6.Text + RadioButton3.Text)
        End If
    End Sub

    Private Sub RadioButton4_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles RadioButton4.CheckedChanged
        If RadioButton4.Checked = True Then
            TextBox6.Text = (TextBox6.Text + RadioButton4.Text)
        End If
    End Sub

    Private Sub RadioButton5_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles RadioButton5.CheckedChanged
        If RadioButton5.Checked = True Then
            TextBox6.Text = (TextBox6.Text + RadioButton5.Text)
        End If
    End Sub
```

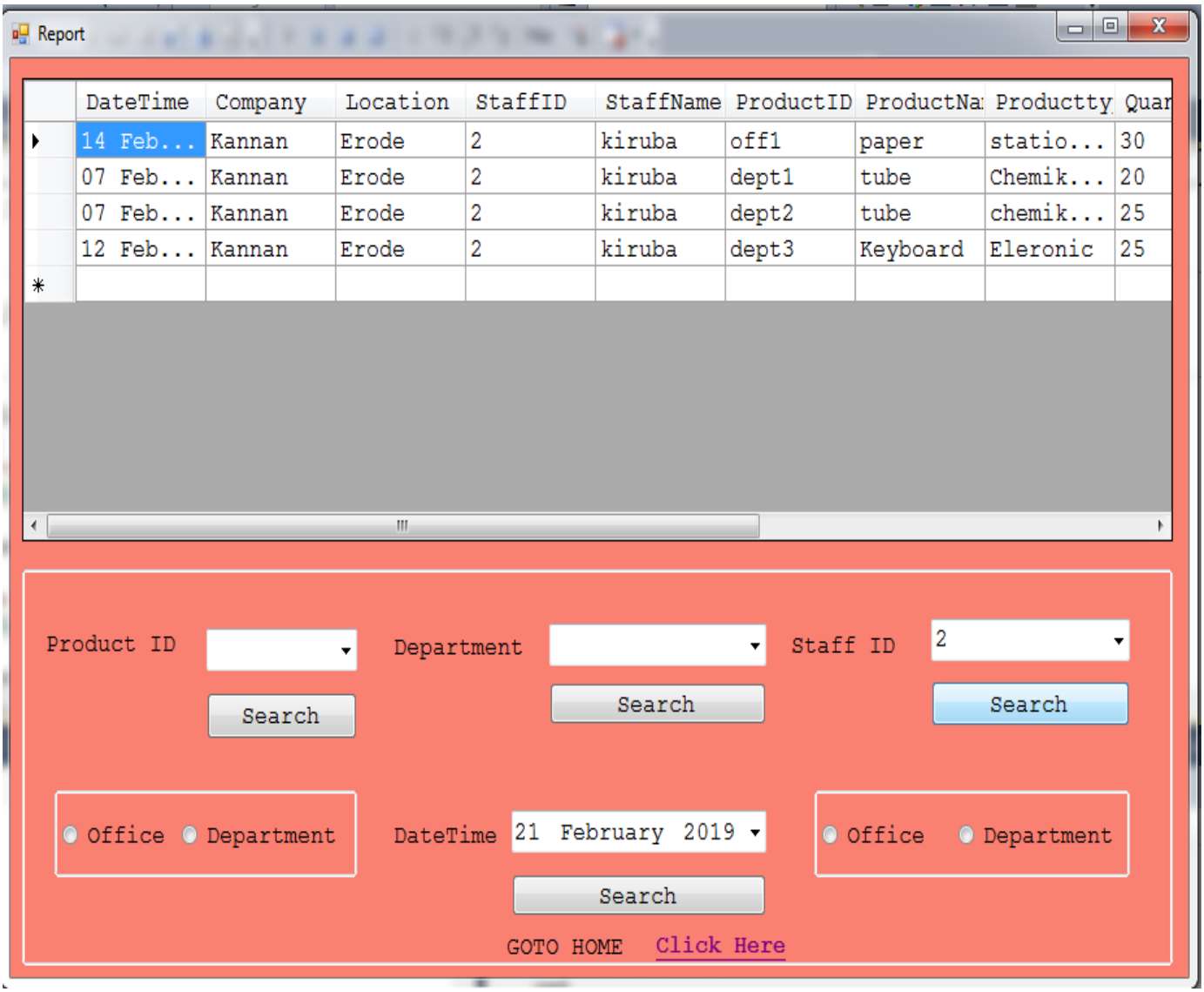
## E.SAMPLE OUTPUT

## SEARCH PRODUCT

### I. Productidwise



## II. Staffidwise



**III. Datewise**