**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 Ms.sheetal Ms.Harshini A S

(professor)

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 fulfillment of the requirements for the award of degree of MCA 2nd sem, from Presidency college Bangloru, 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 Bangloru) 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. There are many departments of administration for the maintenance of college information and student databases in any institution. All these departments provide various records regarding students. Most of these records need to maintain information about the students. This information could be the general details like student name, address, performance, attendance etc or specific information related to departments like collection of data. All the modules in the college administration are inter-dependent. They are maintained manually. So they need to be automated and centralized as, information from one module will be needed by other modules. Library system is a part of the complete college management system and is also available individually .It offers a simple yet powerful solution to automate college libraries completely. The software helps us to manage the print materials like books, journals and magazines along with the non printed materials. College transport management module will make the process of assigning and maintaining the buses and other vehicles easier this module will keep the records of students using conveyance, routes etc. Hostel management module will offer a full features system to efficiently manage the entire residential facility in the institution. Reducing the staff and paper work, this hostel module will keep the update records of students, their meal, lodging, transfer room and other facilities.

INTRODUCTION

The project entitled product stock Management is developed as part of the MCA 2nd sem Project for the fulfillment 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.

**DATA TABLES**

Table Name: VENDOR

Primary Key: VNO

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.NO | FIELD NAME | DATA TYPE | WIDTH | CONSTRAINTS | DESCRIPTION |
| 1.  2.  3. | VNO  VNAME  VADDR | NUMBER  VARCHAR2  VARCHAR2 | 8  10  20 | NOT NULL | VENDOR NO  VENDOR NAME  VENDOR ADDRESS |

Table Name: ITEM

Primary Key: ITEM CODE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.NO | FIELD NAME | DATA TYPE | WIDTH | CONSTRAINTS | DESCRIPTION |
| 1.  2.  3.  4.  5. | ITEM CODE  DESCRIPTION  QUANTITY  VALUE  RATE | VARCHAR2  VARCHAR2  NUMBER  NUMBER  NUMBER | 10  15  6,3  6,2  6,2 | NOT NULL | ITEM CODE  DESCRIPTION  QUANTITY  VALUE  RATE |

Table Name: IND\_HDR

Primary Key: INDENTNO

Foreign Key: ITEMCODE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.NO | FIELD NAME | DATA TYPE | WIDTH | CONSTRAINTS | DESCRIPTION |
| 1.  2.  3.  4.  5.  6.  7.  8.  9. | INDENTOR  INDATE  INDANTNO  TOT\_EST\_VAL  DEPT  ITEM CODE  QTY  RATE  DT\_REQ | VARCHAR2  DATE  VARCHAR2  NUMBER  VARCHAR2  VARCHAR2  NUMBER  NUMBER  DATE | 6  8  10  8,2  10  10  6,3  6,2  8 | NOT NULL | INDENTOR  INDENT DATE  INDENT NUMBER  TOTAL ESTVALUE  DEPARTMENT  ITEM CODE  QUANTITY  RATE  DATE OF REQUEST |

Table Name: ENQ\_HDR

Primary Key: ENQNO

Foreign Key: ITEM CODE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.NO | FIELD NAME | DATA TYPE | WIDTH | CONSTRAINTS | DESCRIPTION |
| 1.  2.  3.  4.  5.  6.  7. | ENQNO  INDENTNO  ENQDATE  VNO  ITEMCODE  QTY  DT\_REQ | VARCHAR2  VARCHAR2  DATE  NUMBER  VARCHAR2  NUMBER  DATE | 10  10  8  8  10  6,3  8 | NOT NULL | ENQUIRY NUM  INDENT NUMBER  ENQUIRY DATE  VENDOR NUMBER  ITEM CODE  QUANTITY  DATE OF REQUEST |

Table Name: QUTN\_HDR

Primary Key: QUTNO

Foreign Key: ENQNO

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.NO | FIELD NAME | DATA TYPE | WIDTH | CONSTRAINTS | DESCRIPTION |
| 1.  2.  3.  4.  5.  6.  7.  8. | QUTNO  ENQNO  ENQDATE  VNO  ITEMCODE  QTY  RATE  DT\_REQ | VARCHAR2  VARCHAR2  DATE  NUMBER  VARCHAR2  NUMBER  NUMBER  DATE | 10  10  8  8  10  6,3  6,2  8 | NOT NULL | QUATATION NO  ENQUIRY NO  ENQUIRY DATE  VENDOR NO  ITEM CODE  QUANTITY  RATE  DATE OF REQUEST |

Table Name: PO\_HDR

Primary Key: PONO

Foreign Key: QUTNO

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.NO | FIELD NAME | DATA TYPE | WIDTH | CONSTRAINTS | DESCRIPTION |
| 1.  2.  3.  4.  5.  6.  7.  8.  9. | PONO  QUTNO  PODATE  VNO  ITEMCODE  QTY  RATE  DT\_REQ  POVAL | VARCHAR2  VARCHAR2  DATE  NUMBER  VARCHAR2  NUMBER  NUMBER  DATE  NUMBER | 6  6  8  8  6  6,3  6,2  8  8,2 | NOT NULL | PURCHASE NO  QUATATION NO  PURCHASEDATE  VENDOR NO  ITEM CODE  QUANTITY  RATE  DATEOFREQUES  PURCHASE VAL |

Table Name: RECPT\_HDR

Primary Key: RP\_NO

Foreign Key: PONO

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.NO | FIELD NAME | DATA TYPE | WIDTH | CONSTRAINTS | DESCRIPTION |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10. | RP\_NO  PONO  RDATE  SUP\_BY  RECD\_BY  ITEMCODE  QTY  VAL  CHK\_BY  REMKS | VARCHAR2  VARCHAR2  DATE  VARCHAR2  VARCHAR2  VARCHAR2  NUMBER  NUMBER  VARCHAR2  VARCHAR2 | 10  10  8  10  10  10  6,3  6,2  20  20 | NOT NULL | RECEIPT NO  PURCHASE NO  RECEIPT DATE  SUPPLIED BY  RECEIVED BY  ITEM CODE  QUANTITY  VALUE  CHECKED BY  REMARKS |

Table Name: ISUE\_HDR

Primary Key: ISUNO

Foreign Key: ITEMCODE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.NO | FIELD NAME | DATA TYPE | WIDTH | CONSTRAINTS | DESCRIPTION |
| 1.  2.  3.  4.  5.  6.  7.  8.  9. | ISUNO  ISDAT  DEPT  ISU\_BY  ITEMCODE  QTY  VAL  HANDOVR\_TO  REMAK | VARCHAR2  DATE  VARCHAR2  VARCHAR2  VARCHAR2  NUMBER  NUMBER  VARCHAR2  VARCHAR2 | 10  8  10  10  8  6,2  6,2  10  20 | NOT NULL | ISSUE NO  ISSUE DATE  DEPARTMENT  ISSUED BY  ITEM CODE  QUANTITY  VALUE  HANDOVER TO  REMARKS |

Table Name: RETUN\_HDR

Primary Key: RETNO

Foreign Key: ISUNO

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.NO | FIELD NAME | DATA TYPE | WIDTH | CONSTRAINTS | DESCRIPTION |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10. | RETNO  RTDAT  ISUNO  DEPT  RETN\_TO  RETN\_BY  ITEMCODE  QTY  VAL  REMKS | VARCHAR2  DATE  VARCHAR2  VARCHAR2  VARCHAR2  VARCHAR2  VARCHAR2  NUMBER  NUMBER  VARCHAR2 | 5  8  10  10  10  10  10  6,3  6,2  10 | NOT NULL | RETURN NO  RETURN DATE  ISSUE NO  DEPARTMENT  RETURN TO  RETURN BY  ITEM CODE  QUANTITY  VALUE  REMARKS |

**SCREEN SHOTS OF FORMS**

**Enquiry**

**Indent**

**Issue**

**Item Code**

**Login Page**

**Purchase Order**



**Quotation**

**Receipts**

**Return**

**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

COST CENTER

ORDER

**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**

**Login page Coding**

Private Sub Command1\_Click ()

If Trim (UCase (txtID)) = "TANVEER" And Trim (UCase (txtPWD)) = "PASSWORD" Then

Unload Me

MDIForm1.Show

MDIForm1.WindowState = 2

Else

MsgBox "Your Username or Password is wrong"

End If

End Sub

Private Sub Command2\_Click ()

Unload Me

End

End Sub

Private Sub Form\_Load ()

Form14.Caption = "STOCK MANAGEMENT" & " " & Date & " " & Time

End Sub

**Home Page Coding**

Private Sub eenqurydatewise\_Click ()

ItemEnquiryReportByDateWise.Show

End Sub

Private Sub eenqurynumberwise\_Click ()

ItemEnquiryReportByNumberWise.Show

End Sub

Private Sub eexit\_Click ()

Call closing

End

End Sub

Private Sub iindentdatewise\_Click ()

ItemIndentReportbydatewise.Show

End Sub

Private Sub iindentnumberwise\_Click ()

ItemIndenReport.Show

End Sub

Private Sub iissuedetails\_Click ()

ItemIssueReport.Show

End Sub

Private Sub iitem\_Click ()

Form8.Show

End Sub

Private Sub iitemissues\_Click ()

Form11.Show

End Sub

Private Sub iitemreceipt\_Click ()

form10.Show

End Sub

Private Sub iitemreturns\_Click ()

Form12.Show

End Sub

Private Sub MDIForm\_Load ()

MDIForm1.Caption = "SEC INDUSTRIES" & " " & Date & " " & Time

End Sub

Private Sub newquatation\_Click ()

Form5.Show

End Sub

Private Sub nnewenquery\_Click ()

Form3.Show

End Sub

Private Sub nnewindent\_Click ()

Form1.Show

End Sub

Private Sub ppurchasedetails\_Click ()

PurchaseDetailsReport.Show

End Sub

Private Sub ppurchaseorder\_Click ()

form15.Show

End Sub

Private Sub qquatationdatewise\_Click ()

End Sub

Private Sub qquatationnumberwise\_Click ()

QuatationReportByNoWise.Show

End Sub

Private Sub rreceiptdetails\_Click ()

ItemReceiptReport.Show

End Sub

Private Sub rreturndetails\_Click ()

ItemReturnReport.Show

End Sub

Private Sub sstockstatusdetails\_Click ()

stockstatusreport.Show

End Sub

Private Sub ttransactiondetails\_Click ()

itemtransactionreport.Show

End Sub

Private Sub vven\_Click ()

Form7.Show

End Sub

**Vendor Master Page Coding**

Option Explicit

Dim ed1%

Dim bool As Boolean

Dim clmod As New inventclass

Private Sub cmddel\_Click()

Dim d%, n%

Call cmdfind\_Click

If bool = False Then Exit Sub

n = clmod.nul(Me)

If n <> 1 Then

MsgBox "Fields are null so cannot be delete"

Exit Sub

End If

d = MsgBox("Dou You want to Delete then Current Record", vbYesNo)

If d = vbYes Then

venderrs.Delete

venderrs.Requery

Call clmod.clall(Me)

End If

End Sub

Private Sub cmdexit\_Click()

Unload Me

End Sub

Private Sub cmdfind\_Click()

Dim f$

bool = True

f = InputBox("Enter Vendor No that you want Find", vbYesNo)

If Not IsNumeric(f) And f = "" Then Exit Sub

venderrs.MoveFirst

venderrs.Find "vno=" & f

If Not venderrs.EOF Then

txtvno = venderrs("vno")

txtvname = venderrs("vname")

txtvadd = venderrs("vaddr")

Else

MsgBox "Record Does Not Exists"

bool = False

Exit Sub

End If

End Sub

Private Sub cmdmodify\_Click()

cmdfind\_Click

Call clmod.enbt(Me)

txtvno.Enabled = False

ed1 = 0

End Sub

Private Sub cmdnew\_Click()

Call clmod.clall(Me)

Dim a%

If venderrs.RecordCount = 0 Then

a = 1

Else

venderrs.MoveLast

a = venderrs("vno") + 1

End If

txtvno = a

Call clmod.enbt(Me)

txtvno.Enabled = False

txtvname.SetFocus

ed1 = 1

End Sub

Private Sub cmdSAVE\_Click()

Dim x%

x = clmod.nul(Me)

If x <> 1 Then Exit Sub

If ed1 = 1 Then

venderrs.AddNew

venrec

ElseIf ed1 = 0 Then

venrec

End If

End Sub

Private Sub Form\_Load()

'Form7.Caption = "SEC INDUSTRIES" & " " & Date & " " & Time

ed1 = 0

End Sub

Private Sub txtvadd\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txtvname\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Sub venrec()

venderrs("vno") = Trim(txtvno)

venderrs("vname") = Trim(txtvname)

venderrs("vaddr") = Trim(txtvadd)

venderrs.Update

MsgBox "updated"

Call clmod.enbf(Me)

End Sub

**Item Code Page Coding**

Dim ed1%

Dim bool As Boolean

Dim clmod As New inventclass

Private Sub cmddel\_Click()

Dim d%, n%

cmdfind\_Click

If bool = False Then Exit Sub

n = clmod.nul(Me)

If n <> 1 Then

MsgBox "Fields are null so cannot be delete"

Exit Sub

End If

d = MsgBox("Dou You want to Delete the Current Record", vbYesNo)

If d = vbYes Then

itemrs.Delete

itemrs.Requery

Call clmod.clall(Me)

End If

End Sub

Private Sub cmdexit\_Click()

Unload Me

End Sub

Private Sub cmdfind\_Click()

bool = True

Dim f$

f = InputBox("Enter Item Code that you want Find", vbYesNo)

If Not IsNumeric(i) And i = "" Then Exit Sub

itemrs.MoveFirst

itemrs.Find "itemcode=" & f

If Not itemrs.EOF Then

txtCODE = itemrs("itemcode")

txtIDES = itemrs("description")

txtRATE = itemrs("rate")

txtQTY = itemrs("quantity")

txtVAL = itemrs("value")

Else

MsgBox "Record Does Not Exists"

bool = False

Exit Sub

End If

End Sub

Private Sub cmdmodify\_Click()

cmdfind\_Click

Call clmod.enbt(Me)

txtCODE.Enabled = False

ed1 = 0

End Sub

Private Sub cmdnew\_Click()

Call clmod.clall(Me)

Dim a%

If itemrs.RecordCount = 0 Then

a = 1001

Else

itemrs.MoveLast

a = itemrs("itemcode") + 1

End If

txtCODE = a

Call clmod.enbt(Me)

txtCODE.Enabled = False

txtIDES.SetFocus

ed1 = 1

End Sub

Private Sub cmdSAVE\_Click()

Dim x%

x = clmod.nul(Me)

If x <> 1 Then Exit Sub

If ed1 = 1 Then

itemrs.AddNew

itemrec

ElseIf ed1 = 0 Then

itemrec

End If

End Sub

Private Sub Form\_Load()

'Form8.Caption = "SEC INDUSTRIES" & " " & Date & " " & Time

ed1 = 0

End Sub

Private Sub txtIDES\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txtrate\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Sub itemrec()

itemrs("itemcode") = Trim(txtCODE)

itemrs("description") = Trim(txtIDES)

itemrs("rate") = Trim(txtRATE)

itemrs("quantity") = Trim(txtQTY)

itemrs.Update

MsgBox "updated"

Call clmod.enbf(Me)

End Sub

**Quotation Page Coding**

Dim ed1%

Dim clmod As New inventclass

Dim bool As Boolean

Private Sub cmddelete\_Click()

Dim d%

cmdfind\_Click

If bool = False Then Exit Sub

d = MsgBox("Dou You want to Delete then Current Record", vbYesNo)

If d = vbYes Then

qutnrs.Delete

qutnrs.Requery

Call clmod.clall(Me)

End If

End Sub

Private Sub cmdexit\_Click()

Unload Me

End Sub

Private Sub cmdfind\_Click()

Dim f$

bool = True

f = InputBox("Enter Quation No that you want Find", vbYesNo)

If Not IsNumeric(i) And i = "" Then Exit Sub

qutnrs.MoveFirst

qutnrs.Find "qutno=" & f

If Not qutnrs.EOF Then

txtqno = qutnrs("qutno")

DataCombo1.Text = qutnrs("enqno")

txtdate = qutnrs("enqdate")

txtvno = qutnrs("vno")

txtcode = qutnrs("itemcode")

txtqty = qutnrs("qty")

txtrate = qutnrs("rate")

txtdreq = qutnrs("dt\_req")

Else

MsgBox "Record Does Not Exists"

bool = False

Exit Sub

End If

End Sub

Private Sub cmdmodify\_Click()

cmdfind\_Click

Call clmod.enbt(Me)

txtqno.Enabled = False

ed1 = 0

End Sub

Private Sub cmdnew\_Click()

Call clmod.clall(Me)

Dim a%

If qutnrs.RecordCount = 0 Then

a = 101

Else

qutnrs.MoveLast

a = qutnrs("qutno") + 1

End If

txtqno = a

Call clmod.enbt(Me)

txtqno.Enabled = False

DataCombo1.SetFocus

ed1 = 1

End Sub

Private Sub cmdSAVE\_Click()

Dim x%

x = clmod.nul(Me)

If x <> 1 Then Exit Sub

If ed1 = 1 Then

qutnrs.AddNew

qutrec

ElseIf ed1 = 0 Then

qutrec

End If

End Sub

Private Sub DataCombo1\_Click(Area As Integer)

Dim ino#

If DataCombo1.MatchedWithList = True Then

enqrs.MoveFirst

enqrs.Find "enqno=" & DataCombo1

If Not enqrs.EOF Then

ino = enqrs("indentno")

txtdate = enqrs("enqdate")

txtvno = enqrs("vno")

txtcode = enqrs("itemcode")

txtqty = enqrs("qty")

txtdreq = enqrs("dt\_req")

indrs.MoveFirst

indrs.Find "indantno=" & ino

If Not indrs.EOF Then

txtrate = indrs("rate")

End If

End If

End If

Call clmod.enbf(Me)

End Sub

Private Sub Form\_Load()

Form5.Caption = "STOCK MANAGEMENT GOVT. DEGREE COLLEGE KUPWARA" & " " & Date & " " & Time

ed1 = 0

End Sub

Private Sub txtdreq\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txtqno\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtqty\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtrate\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Sub qutrec()

qutnrs("qutno") = txtqno

qutnrs("enqno") = DataCombo1.Text

qutnrs("enqdate") = txtdate

qutnrs("vno") = txtvno

qutnrs("itemcode") = txtcode

qutnrs("qty") = txtqty

qutnrs("rate") = txtrate

qutnrs("dt\_req") = txtdreq

qutnrs.Update

MsgBox "updated"

Call clmod.enbf(Me)

End Sub

**Enquiry Page Coding**

Option Explicit

Dim ed1%

Dim clmod As New inventclass

Dim bool As Boolean

Private Sub cmddelete\_Click()

Dim d%

cmdfind\_Click

If bool = False Then Exit Sub

d = MsgBox("Dou You want to Delete then Current Record", vbYesNo)

If d = vbYes Then

enqrs.Delete

enqrs.Requery

Call clmod.clall(Me)

txtENO.Enabled = False

txtDATE.Enabled = False

End If

End Sub

Private Sub cmdexit\_Click()

Unload Me

End Sub

Private Sub cmdfind\_Click()

Dim f$

f = InputBox("Enter Enquiry No that you want Find", vbYesNo)

If Not IsNumeric(f) And f = "" Then Exit Sub

enqrs.MoveFirst

enqrs.Find "enqno=" & f

If Not enqrs.EOF Then

txtENO = enqrs("enqno")

txtDATE = enqrs("enqdate")

DataCombo1.Text = enqrs("indentno")

DataCombo2.Text = enqrs("vno")

txtqty = enqrs("qty")

txtdreq = enqrs("dt\_req")

txtcode = enqrs("itemcode")

Else

MsgBox "Record Does Not Exists"

bool = False

Exit Sub

End If

End Sub

Private Sub cmdmodify\_Click()

cmdfind\_Click

Call clmod.enbt(Me)

txtENO.Enabled = False

txtDATE.Enabled = False

ed1 = 0

End Sub

Private Sub cmdnew\_Click()

Call clmod.clall(Me)

Dim a%

If enqrs.RecordCount = 0 Then

a = 101

Else

enqrs.MoveLast

a = enqrs("enqno") + 1

End If

txtENO = a

txtDATE.Text = Date

Call clmod.enbt(Me)

txtENO.Enabled = False

txtDATE.Enabled = False

ed1 = 1

End Sub

Private Sub cmdSAVE\_Click()

Call clmod.nul(Me)

If clmod.nul(Me) = 0 Then Exit Sub

If ed1 = 1 Then

enqrs.AddNew

enqrec

ElseIf ed1 = 0 Then

enqrec

End If

End Sub

Private Sub DataCombo1\_Click(Area As Integer)

If DataCombo1.MatchedWithList = True Then

indrs.MoveFirst

indrs.Find "indantno=" & DataCombo1

If Not indrs.EOF Then

txtcode = indrs("itemcode")

txtqty = indrs("qty")

txtdreq = indrs("dt\_req")

End If

End If

txtcode.Enabled = False

txtqty.Enabled = False

txtdreq.Enabled = False

End Sub

Private Sub Form\_Load()

'Form3.Caption = "SEC INDUSTRIES" & " " & Date & " " & Time

ed1 = 0

txtENO.Enabled = False

txtDATE.Enabled = False

End Sub

Private Sub txtCODE\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtDATE\_LostFocus()

If txtDATE <> Date Then

MsgBox "ENTER TODAY'S DATE"

Exit Sub

End If

End Sub

Private Sub txtENO\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtqty\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Sub enqrec()

enqrs("enqno") = txtENO

enqrs("enqdate") = txtDATE

enqrs("indentno") = DataCombo1.Text

enqrs("vno") = DataCombo2.Text

enqrs("qty") = txtqty

enqrs("dt\_req") = txtdreq

enqrs("itemcode") = txtcode

enqrs.Update

MsgBox "updated"

Call clmod.enbf(Me)

End Sub

**Purchase Order Page Coding**

Dim ed1%

Dim clmod As New inventclass

Dim bool As Boolean

Private Sub cmdexit\_Click()

Unload Me

End Sub

Private Sub cmdfind\_Click()

Dim f$

bool = True

f = InputBox("Enter Purchase No To Find?", vbYesNo)

If Not IsNumeric(f) And f = "" Then Exit Sub

pors.MoveFirst

pors.Find "pono=" & f

If Not pors.EOF Then

txtpono = pors("pono")

DataCombo1.Text = pors("qutno")

txtpodate = pors("qutno")

txtvno = pors("vno")

txticode = pors("itemcode")

txtqty = pors("qty")

txtrate = pors("rate")

txtdreq = pors("dt\_req")

txtpoval = pors("poval")

Else

MsgBox "RECORD DOESNOT EXIT"

bool = False

End If

End Sub

Private Sub cmdmodify\_Click()

Call cmdfind\_Click

txtpono.Enabled = False

txtpodate.Enabled = False

ed1 = 0

End Sub

Private Sub cmdnew\_Click()

Dim a%

Call clmod.clall(Me)

If pors.RecordCount = 0 Then

a = 101

Else

pors.MoveLast

a = pors("pono") + 1

End If

txtpono = a

txtpodate.Text = Date

Call clmod.enbt(Me)

txtpono.Enabled = False

txtpodate.Enabled = False

ed1 = 1

End Sub

Private Sub cmdSAVE\_Click()

Dim k%

k = clmod.nul(Me)

If k <> 1 Then Exit Sub

If ed1 = 1 Then

pors.AddNew

porec

ElseIf ed1 = 0 Then

porec

End If

End Sub

Private Sub DataCombo1\_Click(Area As Integer)

If DataCombo1.MatchedWithList = True Then

qutnrs.MoveFirst

qutnrs.Find "qutno=" & DataCombo1

If Not qutnrs.EOF Then

txticode = qutnrs("itemcode")

txtqty = qutnrs("qty")

txtrate = qutnrs("rate")

txtvno = qutnrs("vno")

txtdreq = qutnrs("dt\_req")

txtpoval = Val(txtrate) \* Val(txtqty)

End If

End If

Call clmod.enbf(Me)

End Sub

Private Sub Form\_Load()

'form15.Caption = "SEC INDUSTRIES" & " " & Date & " " & Time

ed1 = 0

txtpono.Enabled = False

txtpodate.Enabled = False

End Sub

Private Sub txticode\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txtpno\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtpodate\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtpoval\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtqty\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtrate\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtvno\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Sub porec()

pors("pono") = txtpono

pors("qutno") = DataCombo1.Text

pors("podate") = txtpodate

pors("vno") = txtvno

pors("itemcode") = txticode

pors("qty") = txtqty

pors("rate") = txtrate

pors("dt\_req") = txtdreq

pors("poval") = txtpoval

pors.Update

MsgBox "updated"

Call clmod.enbf(Me)

End Sub

**Issue Returns Page Coding**

Dim clmod As New inventclass

Dim rate%, ioqty%

Private Sub cmdexit\_Click()

Unload Me

End Sub

Private Sub cmdfind\_Click()

Dim f$

f = InputBox("Enter Return No that you want Find", vbYesNo)

If Not IsNumeric(i) And i = "" Then Exit Sub

retunrs.MoveFirst

retunrs.Find "retno=" & f

If Not retunrs.EOF Then

txtRNO = retunrs("retno")

txtRDATE = retunrs("rtdat")

DataCombo1.Text = retunrs("isuno")

txtRETUNT = retunrs("retn\_to")

txtRETUNB = retunrs("retn\_by")

txticode = retunrs("itemcode")

txtQTY = retunrs("qty")

txtVAL = retunrs("val")

txtREM = retunrs("remks")

txtDEPT = retunrs("dept")

Else

MsgBox "Record Does Not Exists"

Exit Sub

End If

End Sub

Private Sub cmdmodify\_Click()

cmdfind\_Click

Call clmod.enat(Me)

txtRNO.Enabled = False

End Sub

Private Sub cmdnew\_Click()

Call clmod.clr(Me)

Dim a%

If retunrs.RecordCount = 0 Then

a = 101

Else

retunrs.MoveLast

a = retunrs("retno") + 1

End If

txtRNO = a

txtRDATE = Date

txtRNO.Enabled = False

txtRDATE.Enabled = False

End Sub

Private Sub cmdsave1\_Click()

Dim k%

k = clmod.nul(Me)

If k <> 1 Then Exit Sub

stockrs.MoveFirst

stockrs.Find "itemcode=" & txticode

If Not stockrs.EOF Then

oqty = stockrs("qty")

dval = stockrs("sdate")

Else

MsgBox "Corresponding Item Code does not Exists"

Exit Sub

End If

nqty = oqty + Val(txtQTY)

stockrs("itemcode") = txticode

stockrs("des") = itemrs("description")

stockrs("rate") = itemrs("rate")

stockrs("qty") = nqty

stockrs("val") = nqty \* (itemrs("rate"))

stockrs("sdate") = Date

stockrs("transaction") = "Returns"

stockrs.Update

MsgBox "stock updated"

retunrs.AddNew

retunrs("retno") = txtRNO

retunrs("rtdat") = txtRDATE

retunrs("dept") = txtDEPT

retunrs("isuno") = DataCombo1.Text

retunrs("retn\_to") = txtRETUNT

retunrs("retn\_by") = txtRETUNB

retunrs("itemcode") = txticode

retunrs("qty") = txtQTY

retunrs("val") = txtVAL

retunrs("remks") = txtREM

retunrs.Update

MsgBox "RECORD SAVED"

End Sub

Private Sub DataCombo1\_Click(Area As Integer)

If DataCombo1.MatchedWithList = True Then

isuers.MoveFirst

isuers.Find "isuno=" & DataCombo1

If Not isuers.EOF Then

txtDEPT = isuers("dept")

ioqty = isuers("qty")

txticode = isuers("itemcode")

End If

txticode.Enabled = False

txtDEPT.Enabled = False

txtQTY.SetFocus

End If

End Sub

Private Sub Form\_Load()

'Form12.Caption = "SEC INDUSTRIES" & " " & Date & " " & Time

End Sub

Private Sub txtdept\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txticode\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtqty\_Change()

itemrs.MoveFirst

itemrs.Find "itemcode=" & txticode

' If Not itemrs.EOF Then

' rate = itemrs("rate")

' txtVAL = Val(txtQTY) \* rate

' End If

txtVAL.Enabled = False

End Sub

Private Sub txtqty\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtQTY\_LostFocus()

If Trim(txtQTY) = "" And Val(txtQTY) = 0 Then

MsgBox "Enter Quantity"

txtQTY.SetFocus

Exit Sub

End If

If txtQTY > ioqty Then

MsgBox " returns are more then isues"

txtQTY.SetFocus

Exit Sub

End If

End Sub

Private Sub txtREM\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txtRETUNB\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txtRETUNT\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txtRNO\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtVAL\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

**Issue Stock Page Coding**

Dim clmod As New inventclass

Dim bool As Boolean

Dim rate!

Private Sub cmddelete\_Click()

Dim d%

cmdfind\_Click

If bool = False Then Exit Sub

d = MsgBox("Dou You want to Delete then Current Record", vbYesNo)

If d = vbYes Then

isuers.Delete

isuers.Requery

Call clmod.clr(Me)

End If

End Sub

Private Sub cmdexit\_Click()

Unload Me

End Sub

Private Sub cmdfind\_Click()

Dim f$

bool = True

f = InputBox("Enter Issue No that you want Find", vbYesNo)

If Not IsNumeric(f) And f = "" Then Exit Sub

isuers.MoveFirst

isuers.Find "isuno=" & f

If Not isuers.EOF Then

txtINO = isuers("isuno")

txtIDATE = isuers("isdat")

txtDEPT = isuers("dept")

txtisdb = isuers("isu\_by")

txticode = isuers("itemcode")

txtQTY = isuers("qty")

txtVAL = isuers("val")

txtHAND = isuers("handovr\_to")

txtREM = isuers("remak")

Else

MsgBox "Record Does Not Exists"

bool = False

Exit Sub

End If

End Sub

Private Sub cmdmodify\_Click()

cmdfind\_Click

Call clmod.enbt(Me)

txtINO.Enabled = False

txtIDATE.Enabled = False

End Sub

Private Sub cmdnew1\_Click()

Dim a%

Call clmod.clall(Me)

If isuers.RecordCount = 0 Then

a = 101

Else

isuers.MoveLast

a = isuers("isuno") + 1

End If

txtINO = a

txtIDATE.Text = Date

Call clmod.enbt(Me)

txtIDATE.Enabled = False

txtINO.Enabled = False

End Sub

Private Sub cmdsave1\_Click()

Dim k%, oqty%, nqty%

k = clmod.nul(Me)

If k <> 1 Then Exit Sub

stockrs.MoveFirst

stockrs.Find "itemcode=" & txticode

If Not stockrs.EOF Then

oqty = stockrs("qty")

dval = stockrs("sdate")

If txtQTY > oqty Then

MsgBox "STOCK NOT AVAILABLE"

txtQTY = ""

txtQTY.Enabled = True

txtqty\_Change

Exit Sub

End If

Else

MsgBox "Corresponding ItemCode does not Exists"

Exit Sub

End If

isuers.AddNew

isuers("isuno") = txtINO

isuers("isdat") = txtIDATE

isuers("dept") = txtDEPT

isuers("isu\_by") = txtISDBY

isuers("itemcode") = txticode

isuers("qty") = txtQTY

isuers("val") = txtVAL

isuers("handovr\_to") = txtHAND

isuers("remak") = txtREM

isuers.Update

MsgBox "RECORD SAVED"

Call invent(oqty)

Call clmod.enbf(Me)

End Sub

Private Sub DataCombo2\_Click(Area As Integer)

If DataCombo2.MatchedWithList = True Then

indrs.MoveFirst

indrs.Find "indantno=" & DataCombo2

If Not indrs.EOF Then

txtDEPT = indrs("dept")

txticode = indrs("itemcode")

txtQTY = indrs("qty")

End If

End If

itemrs.MoveFirst

itemrs.Find "itemcode=" & txticode

If Not itemrs.EOF Then

rate = itemrs("rate")

txtVAL = Val(txtQTY) \* rate

End If

txtDEPT.Enabled = False

txticode.Enabled = False

txtVAL.Enabled = False

txtQTY.Enabled = False

End Sub

Private Sub Form\_Load()

'Form11.Caption = "SEC INDUSTRIES" & " " & Date & " " & Time

End Sub

Private Sub txtdept\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txtHAND\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txticode\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtINO\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtISDBY\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

If KeyAscii = 13 Then txtQTY.SetFocus

End Sub

Private Sub txtqty\_Change()

itemrs.MoveFirst

' itemrs.Find "itemcode=" & txticode

' If Not itemrs.EOF Then

' rate = itemrs("rate")

' txtVAL = Val(txtQTY) \* rate

' End If

End Sub

Private Sub txtqty\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtQTY\_LostFocus()

If Trim(txtQTY) = "" And Val(txtQTY) = 0 Then

MsgBox "Enter Quantity"

txtQTY.SetFocus

Exit Sub

End If

End Sub

Private Sub txtREM\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txtVAL\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Public Sub invent(ByVal oqty)

nqty = oqty - Val(txtQTY)

stockrs("itemcode") = txticode

stockrs("des") = itemrs("description")

stockrs("rate") = itemrs("rate")

stockrs("qty") = nqty

stockrs("val") = nqty \* (itemrs("rate"))

stockrs("sdate") = Date

stockrs("transaction") = "Issue"

stockrs.Update

MsgBox "stock updated"

End Sub

**Item Receipt Page Coding**

Dim clmod As New inventclass

Dim bool As Boolean

Private Sub cmddel\_Click()

Dim d%

cmdfind\_Click

If bool = False Then Exit Sub

d = MsgBox("Dou You want to Delete then Current Record", vbYesNo)

If d = vbYes Then

recptrs.Delete

recptrs.Requery

Call clmod.clall(Me)

End If

End Sub

Private Sub cmdexit\_Click()

Unload Me

End Sub

Private Sub cmdfind\_Click()

Dim f$

bool = True

f = InputBox("Enter Receipt No that you want Find", vbYesNo)

If Not IsNumeric(f) And f = "" Then Exit Sub

recptrs.MoveFirst

recptrs.Find "rp\_no=" & f

If Not recptrs.EOF Then

txtRNO = recptrs("rp\_no")

DataCombo1.Text = recptrs("pono")

txtDATE = recptrs("rdate")

txtSUPP = recptrs("sup\_by")

txtRECD = recptrs("recd\_by")

txtCODE = recptrs("itemcode")

txtqty = recptrs("qty")

txtval = recptrs("val")

txtchk = recptrs("chk\_by")

txtrem = recptrs("remks")

Else

MsgBox "Record Does Not Exists"

bool = False

Exit Sub

End If

Call clmod.enbf(Me)

End Sub

Private Sub cmdmodify\_Click()

cmdfind\_Click

Call clmod.enbt(Me)

txtRNO.Enabled = False

txtDATE.Enabled = False

End Sub

Private Sub cmdnew\_Click()

Call clmod.clall(Me)

Dim a%

If recptrs.RecordCount = 0 Then

a = 101

Else

recptrs.MoveLast

a = recptrs("rp\_no") + 1

End If

txtRNO = a

txtDATE.Text = Date

Call clmod.enbt(Me)

txtRNO.Enabled = False

txtDATE.Enabled = False

End Sub

Private Sub cmdsave1\_Click()

Dim x%, oqty%

oqty = 0

x = clmod.nul(Me)

If x <> 1 Then Exit Sub

recptrs.AddNew

reciprec

If stockrs.BOF = True Then

stockrs.AddNew

stockrec (oqty)

Else

stockrs.MoveFirst

stockrs.Find "itemcode=" & txtCODE

If Not stockrs.EOF Then

oqty = stockrs("qty")

stockrec (oqty)

Else

stockrs.AddNew

stockrec (oqty)

End If

End If

End Sub

Private Sub DataCombo1\_Click(Area As Integer)

If DataCombo1.MatchedWithList = True Then

pors.MoveFirst

pors.Find "pono=" & DataCombo1

If Not pors.EOF Then

txtCODE = pors("itemcode")

txtqty = pors("qty")

txtval = pors("poval")

End If

End If

txtCODE.Enabled = False

txtqty.Enabled = False

txtval.Enabled = False

End Sub

Private Sub Form\_Load()

'form10.Caption = "SEC INDUSTRIES" & " " & Date & " " & Time

End Sub

Private Sub txtchk\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txtCODE\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtDATE\_LostFocus()

If txtDATE <> Date Then

MsgBox " enter todays date"

txtDATE.SetFocus

Exit Sub

End If

End Sub

Private Sub txtqty\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtRECD\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txtREM\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txtRNO\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Private Sub txtSUPP\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

End Sub

Private Sub txtVAL\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

End Sub

Sub reciprec()

recptrs("rp\_no") = txtRNO

recptrs("rdate") = txtDATE

recptrs("pono") = DataCombo1.Text

recptrs("sup\_by") = txtSUPP

recptrs("recd\_by") = txtRECD

recptrs("itemcode") = txtCODE

recptrs("qty") = txtqty

recptrs("val") = txtval

recptrs("chk\_by") = txtchk

recptrs("remks") = txtrem

recptrs.Update

MsgBox "RECORD SAVED"

Call clmod.enbf(Me)

End Sub

Sub stockrec(ByVal oqty)

Dim nqty%

nqty = (oqty + Val(txtqty))

stockrs("itemcode") = txtCODE

stockrs("des") = itemrs("description")

stockrs("rate") = itemrs("rate")

stockrs("qty") = nqty

stockrs("val") = nqty \* (itemrs("rate"))

stockrs("sdate") = txtDATE

stockrs("transaction") = "Receipt"

stockrs.Update

MsgBox "stock updated"

End Sub

**Item indent Page Coding**

Option Explicit

Dim clmod As New inventclass

Dim bool As Boolean

Dim ed1%

Private Sub cmddelete\_Click()

Dim d%

cmdfind\_Click

If bool = False Then Exit Sub

d = MsgBox("Dou You want to Delete then Current Record", vbYesNo)

If d = vbYes Then

indrs.Delete

indrs.Requery

Call clmod.clr(Me)

txtINO.Enabled = False

txtindate.Enabled = False

txttotval.Enabled = False

End If

End Sub

Private Sub cmdexit\_Click()

Unload Me

End Sub

Private Sub cmdfind\_Click()

bool = True

Dim f$

f = InputBox("Enter Indent No that you want Find", vbYesNo)

If Not IsNumeric(f) And f = "" Then Exit Sub

indrs.MoveFirst

indrs.Find "indantno=" & f

If Not indrs.EOF Then

txtINO = indrs("INDaNTNO")

txtindate = indrs("INDATE")

txtindor = indrs("INDENTOR")

txttotval = indrs("TOT\_EST\_VAL")

txtDEPT = indrs("dept")

DataCombo1.Text = indrs("itemcode")

txtqty = indrs("qty")

txtdreq = indrs("dt\_req")

txtrate = indrs("rate")

Else

MsgBox "Record Does Not Exists"

bool = False

Exit Sub

End If

End Sub

Private Sub cmdmodify\_Click()

cmdfind\_Click

Call clmod.enbt(Me)

txtINO.Enabled = False

txtindate.Enabled = False

txttotval.Enabled = False

ed1 = 0

End Sub

Private Sub cmdnew\_Click()

Call clmod.clall(Me)

Dim a%

If indrs.RecordCount = 0 Then

a = 101

Else

indrs.MoveLast

a = indrs("INDaNTNO") + 1

End If

txtINO = a

txtindate.Text = Date

Call clmod.enbt(Me)

txtINO.Enabled = False

txtindate.Enabled = False

txttotval.Enabled = False

txtDEPT.SetFocus

ed1 = 1

End Sub

Private Sub cmdSAVE\_Click()

Dim x%

x = clmod.nul(Me)

If x <> 1 Then Exit Sub

If ed1 = 1 Then

indrs.AddNew

indentrec

ElseIf ed1 = 0 Then

indentrec

End If

End Sub

Private Sub Form\_Load()

'Form1.Caption = "SEC INDUSTRIES" & " " & Date & " " & Time

ed1 = 0

txtindate = Date

txtINO.Enabled = False

txtindate.Enabled = False

txttotval.Enabled = False

End Sub

Private Sub txtdept\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

If KeyAscii = 13 Then

If txtDEPT = "PRODUCTION" Or txtDEPT = "MAINTENANCE" Or txtDEPT = "ADMISSION" Or txtDEPT = "OTHERS" Then

txtindor.SetFocus

Else

MsgBox "Give the department as 'PRODUCTION' or 'MAINTENANCE' or'ADMISSION' or 'OTHERS'"

End If

End If

End Sub

Private Sub txtdept\_LostFocus()

If txtDEPT = "PRODUCTION" Or txtDEPT = "MAINTENANCE" Or txtDEPT = "ADMISSION" Or txtDEPT = "OTHERS" Then

txtindor.SetFocus

Else

MsgBox "Give the department as 'PRODUCTION' or 'MAINTENANCE' or'ADMISSION' or 'OTHERS'"

End If

End Sub

Private Sub txtdreq\_KeyPress(KeyAscii As Integer)

If KeyAscii = 13 Then txttotval.SetFocus

End Sub

Private Sub txtindate\_KeyPress(KeyAscii As Integer)

If KeyAscii = 13 Then txttotval.SetFocus

End Sub

Private Sub txtindor\_KeyPress(KeyAscii As Integer)

Call clmod.char(KeyAscii)

If KeyAscii >= 97 And KeyAscii <= 122 Then

KeyAscii = KeyAscii - 32

End If

If KeyAscii = 13 Then txtindate.SetFocus

End Sub

Private Sub txtINO\_KeyPress(KeyAscii As Integer)

If KeyAscii = 13 Then txtDEPT.SetFocus

End Sub

Private Sub txtqty\_KeyPress(KeyAscii As Integer)

Call clmod.num(KeyAscii)

If KeyAscii = 13 Then txtrate.SetFocus

End Sub

Private Sub txtrate\_KeyPress (KeyAscii As Integer)

Call clmod.num(KeyAscii)

If KeyAscii = 13 Then

If txtqty = "" Then

MsgBox "the quantity cannot be null"

txtqty.SetFocus

Exit Sub

Else

txttotval = txtqty \* txtrate

txtdreq.SetFocus

End If

End If

End Sub

Private Sub txttotval\_KeyPress (KeyAscii As Integer)

Call clmod.num (KeyAscii)

If KeyAscii = 13 Then cmdsave.SetFocus

End Sub

Private Sub txtrate\_LostFocus ()

If txtqty = "" Then

MsgBox "the quantity cannot be null"

txtqty.SetFocus

Exit Sub

Else

txttotval = txtqty \* txtrate

txtdreq.SetFocus

End If

End Sub

Sub indentrec ()

indrs("INDaNTNO") = txtINO

indrs("INDATE") = txtindate

indrs("INDENTOR") = txtindor

indrs("TOT\_EST\_VAL") = txttotval

indrs("dept") = txtDEPT

indrs ("itemcode") = DataCombo1.Text

Indrs ("qty") = txtqty

indrs ("dt\_req") = txtdreq

indrs ("rate") = txtrate

indrs.Update

MsgBox "RECORD SAVED"

Call clmod.enbf (Me)

End Sub

**Project Module Coding**

Public con As New ADODB.Connection

Public indrs As New ADODB.Recordset

'Public inddtlrs As New ADODB.Recordset

Public itemrs As New ADODB.Recordset

Public venderrs As New ADODB.Recordset

Public enqrs As New ADODB.Recordset

Public qutnrs As New ADODB.Recordset

Public pors As New ADODB.Recordset

Public recptrs As New ADODB.Recordset

Public isuers As New ADODB.Recordset

Public retunrs As New ADODB.Recordset

Public stockrs As New ADODB.Recordset

Public Sub main ()

con.Open "Provider=MSDASQL.1;Password=password;Persist Security Info=True;User ID=tanveer;Extended Properties=DSN=MS Access Database;DBQ=C:\Users\Ganai Imtiyaz Sahb\Desktop\projectstock.mdb;DefaultDir=C:\Users\Ganai Imtiyaz Sahb\Desktop;DriverId=281;FIL=MS Access;MaxBufferSize=2048;PageTimeout=5;PWD=password;UID=admin;"

indrs.Open "select \* from ind\_hdr order by indantno", con, adOpenKeyset, adLockOptimistic

retunrs.Open "select \* from retun\_hdr order by retno", con, adOpenKeyset, adLockOptimistic

itemrs.Open "select \* from item order by itemcode", con, adOpenKeyset, adLockOptimistic

venderrs.Open "select \* from vendor order by vno", con, adOpenKeyset, adLockOptimistic

enqrs.Open "select \* from ENq\_HDR order by enqno", con, adOpenKeyset, adLockOptimistic

qutnrs.Open "select \* from qutn\_hdr order by qutno", con, adOpenKeyset, adLockOptimistic

pors.Open "select \* from po\_hdr order by pono", con, adOpenKeyset, adLockOptimistic

recptrs.Open "select \* from rept\_hdr order by rp\_no", con, adOpenKeyset, adLockOptimistic

isuers.Open "select \* from isue\_hdr", con, adOpenKeyset, adLockOptimistic

'stockrs.Open "select \* from stock", con, adOpenKeyset, adLockOptimistic

Form13.Show

End Sub

Sub closing ()

indrs.Close

retunrs.Close

itemrs.Close

venderrs.Close

enqrs.Close

qutnrs.Close

pors.Close

recptrs.Close

'isuers.Close

'stockrs.Close

End Sub

**Class Module Coding**

Dim a As Object

Dim f as Form

Public Sub clr (ByVal f)

For Each a In f.Controls

If TypeOf a Is TextBox Then

a.Text = ""

End If

Next a

End Sub

Public Function nul (ByVal f)

For Each a In f.Controls

If TypeOf a Is TextBox Then

If a.Text = "" Then

MsgBox "Enter values"

Exit Function

End If

End If

Next a

nul = 1

End Function

Public Sub char (KeyAscii)

If KeyAscii >= 65 And KeyAscii <= 90 Or KeyAscii = 8 Then

ElseIf KeyAscii >= 97 And KeyAscii <= 122 Then

KeyAscii = KeyAscii - 32

Else

KeyAscii = 0

End If

End Sub

Public Sub num (KeyAscii)

If KeyAscii >= 48 And KeyAscii <= 57 Or KeyAscii = 32 Or KeyAscii = 13 Or KeyAscii = 8 Then

Else

KeyAscii = 0

MsgBox "only numbers"

End If

End Sub

Public Sub enbf (ByVal f)

For Each a In f.Controls

If TypeOf a Is TextBox Then

a.Enabled = False

End If

Next a

End Sub

Public Sub enbt (ByVal f)

For Each a In f.Controls

If TypeOf a Is TextBox Then

a.Enabled = True

End If

Next a

End Sub

Public Sub enbcmdf (ByVal f)

For Each a In f.Controls

If TypeOf a Is CommandButton Then

a.Enabled = False

End If

Next a

End Sub

Public Sub enbcmdt (ByVal f)

For Each a In f.Controls

If TypeOf a Is CommandButton Then

a.CommandButton .Enabled = True

End If

Next a

End Sub

Public Sub clall (ByVal f)

For Each a In f.Controls

If TypeOf a Is TextBox Then

a.Text = ""

End If

Next a

End Sub

**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 Programming Using vb 6.0 By
* Visual Basics.C programming By Steven Holzner
* 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/)