

CODING

MODULE:

```
Public c As ADODB.Connection
```

```
Public r As ADODB.Recordset
```

```
Public sql As String
```

```
Public Function abc()
```

```
Set c = New ADODB.Connection
```

```
c.Open "provider = msdaora.1;user id = altered/carbon; persist  
security info = true"
```

```
Set r = New ADODB.Recordset
```

```
End Function
```

```
Public Function tt()
```

```
KeyAscii = Asc(UCCase(Chr(KeyAscii)))
```

```
End Function
```

FRMSPLASH:

```
Private Sub Form_Load()
```

```
Timer1.Enabled = True
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
ProgressBar1.Value = ProgressBar1.Value + 1
```

```
Label1.Caption = "Loading..."
```

```
Label2.Caption = ProgressBar1.Value & "%"
```

```
If (ProgressBar1.Value = ProgressBar1.Max) Then
Timer1.Enabled = False
Unload Me
MDIForm1.Show
End If
End Sub
```

INFOFRM:

```
Dim resetbtn As Boolean
Public Function isValid(box As String)
If (box = "text1") Then
    If (IsNumeric(Text1.Text) = False) Then
        Text1.Text = ""
        Text1.SetFocus
        MsgBox "NUMERIC VALUE ONLY"
    ElseIf Len(Text1.Text) <> 15 Then
        Text1.Text = ""
        Text1.SetFocus
        MsgBox "NUMBER SHOULD BE OF 15 DIGITS! "
    End If
End If

If (box = "text2") Then
    If (IsNumeric(Text2.Text) = False) Then
        Text2.Text = ""
        Text2.SetFocus
        MsgBox "NUMERIC VALUE ONLY"
```

```
ElseIf Len(Text2.Text) <> 5 Then
    Text2.Text = ""
    Text2.SetFocus
    MsgBox "NUMBER SHOULD BE OF 5 DIGITS! "
End If
End If
```

```
If (box = "text3") Then
    If (IsNumeric(Text3.Text) = False) Then
        Text3.Text = ""
        Text3.SetFocus
        MsgBox "NUMERIC VALUE ONLY"
    ElseIf Len(Text3.Text) <> 13 Then
        Text3.Text = ""
        Text3.SetFocus
        MsgBox "NUMBER SHOULD BE OF 13 DIGITS! "
    End If
End If
```

```
If (box = "text9") Then
    If (IsNumeric(Text9.Text) = False) Then
        Text9.Text = ""
        Text9.SetFocus
        MsgBox "NUMERIC VALUE ONLY"
    ElseIf Len(Text9.Text) <> 10 Then
        Text9.Text = ""
        Text9.SetFocus
        MsgBox "INVALID MOBILE NUMBER! "
```

```
End If  
End If  
End Function
```

```
Public Function clean()  
Text1.Text = ""  
Text2.Text = ""  
Text3.Text = ""  
Text4.Text = ""  
Text5.Text = ""  
Text6.Text = ""  
Text7.Text = ""  
Text8.Text = ""  
Text9.Text = ""  
Text10.Text = ""  
Text11.Text = ""  
End Function
```

```
Private Sub add_Click()  
clean  
Text1.Enabled = True  
Text2.Enabled = True  
Text3.Enabled = True  
Text4.Enabled = True  
Text5.Enabled = True  
Text6.Enabled = True  
Text7.Enabled = True  
Text8.Enabled = True
```

```
Text9.Enabled = True
Text10.Enabled = True
Text11.Enabled = True
Combo2.Enabled = True
Combo3.Enabled = True
delete.Enabled = False
update.Enabled = False
showdata.Enabled = False
save.Enabled = True
srch.Enabled = False
reset.Enabled = True
Text1.SetFocus
End Sub
```

```
Private Sub Combo2_Click()
Text7.Text = Combo2.Text
Text8.SetFocus
End Sub
```

```
Private Sub Combo3_Click()
Text8.Text = Combo3.Text
Text9.SetFocus
End Sub
```

```
Private Sub delete_Click()
abc
On Error GoTo ERRHANDLE
    sql = "delete from stud where regno=" + Text1.Text
```

```

Set r = c.Execute(sql)
MsgBox "RECORD DELETED"
clean
Exit Sub
ERRHANDLE:
If (Err.Description <> "") Then
    MsgBox Err.Description
    Exit Sub
End If
End Sub

```

```

Private Sub Form_Load()
abc
MonthView1.Visible = False
infobox.Enabled = False
Combo2.AddItem "MALE"
Combo2.AddItem "FEMALE"
Combo3.AddItem "3-1"
Combo3.AddItem "3-2"
Combo3.AddItem "3-3"
Combo3.AddItem "3-4"
Text4.Locked = True
Text7.Locked = True
Text8.Locked = True
Text10.Locked = True
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False

```

```
Text4.Enabled = False
Text5.Enabled = False
Text6.Enabled = False
Text7.Enabled = False
Text8.Enabled = False
Text9.Enabled = False
Text10.Enabled = False
Text11.Enabled = False
Combo2.Enabled = False
Combo3.Enabled = False
save.Enabled = False
update.Enabled = False
delete.Enabled = False
End Sub
```

```
Private Sub MonthView1_DateClick(ByVal DateClicked As Date)
Text10.Text = MonthView1.Value
T1 = Left(Text11.Text, 4)
T2 = Right(Text10.Text, 4)
If (T1 - T2) < 18 Then
    MsgBox " INVALID DOB OR SESSION "
    MsgBox T1 & "COMB" & T2
    Text10.SetFocus
End If
MonthView1.Visible = False
Text11.SetFocus
End Sub
```

```

Private Sub reset_Click()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
Text9.Text = ""
Text10.Text = ""
Text11.Text = ""
delete.Enabled = False
showdata.Enabled = True
update.Enabled = False
save.Enabled = False
srch.Enabled = True
End Sub

```

```

Private Sub SAVE_Click()
abc
If (Text1.Text = "" Or Text2.Text = "" Or Text4.Text = "" Or Text5.Text =
"" Or Text6.Text = "" Or Text7.Text = "" Or Text8.Text = "" Or Text9.Text
= "" Or Text10.Text = "" Or Text11.Text = "") Then
    MsgBox "ALL FIELDS ARE MANDATORY !"
Else
On Error GoTo jiyen
    sql = "insert into stud values(" + Text1.Text + "," + Text2.Text + "," +
Text3.Text + "," + Text4.Text + "," + Text5.Text + "," + Text6.Text +

```



```
"," + Text7.Text + "," + Text8.Text + "," + Text9.Text + "," +  
Text10.Text + "," + Text11.Text + ")"
```

```
Set r = c.Execute(sql)
```

```
MsgBox "RECORD SAVED"
```

```
jiyan:
```

```
If (Err.Description <> "") Then
```

```
    MsgBox Err.Description
```

```
    Exit Sub
```

```
End If
```

```
End If
```

```
End Sub
```

```
Private Sub showdata_click()
```

```
SEARCHFRM.Show
```

```
End Sub
```

```
Private Sub srch_Click()
```

```
infobox.Enabled = True
```

```
update.Enabled = True
```

```
delete.Enabled = True
```

```
MsgBox "PLEASE ENTER ADDMISSION NO. :)"
```

```
Text12.SetFocus
```

```
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then Text2.SetFocus
```

```
End Sub
```

```
Private Sub Text1_LostFocus()
```

```
IsValid ("text1")
```

```
End Sub
```

```
Private Sub Text10_GotFocus()
```

```
MonthView1.Visible = True
```

```
End Sub
```

```
Private Sub Text10_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then Text11.SetFocus
```

```
End Sub
```

```
Private Sub Text11_Change()
```

```
If (Len(Text11.Text) = 4) Then
```

```
Text11 = Text11.Text & "-" & Text11.Text + 3
```

```
End If
```

```
'Text10.Text = Int(Format(Date, "YYYY")) - 1
```

```
End Sub
```

```
Private Sub Text12_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Text1.Enabled = True
```

```
Text2.Enabled = True
```

```
Text3.Enabled = True
```

```
Text4.Enabled = True
```

```
Text5.Enabled = True
```

```
Text6.Enabled = True
```

```
Text7.Enabled = True
```

```
Text8.Enabled = True
```

```

Text9.Enabled = True
Text10.Enabled = True
Text11.Enabled = True
abc
sql = "select * from stud where admno=" + Text12.Text + ""
Set r = c.Execute(sql)
If Not r.EOF Then
Text1.Text = r.Fields(0)
Text2.Text = r.Fields(1)
If Not IsNull(r.Fields(2)) Then
Text3.Text = r.Fields(2)
End If
Text4.Text = r.Fields(3)
Text5.Text = r.Fields(4)
Text6.Text = r.Fields(5)
Text7.Text = r.Fields(6)
Text8.Text = r.Fields(7)
Text9.Text = r.Fields(8)
Text10.Text = r.Fields(9)
Text11.Text = r.Fields(10)
Else
MsgBox "NO RECORD FOUND ! "
End If
End If
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then Text3.SetFocus

```

End Sub

Private Sub Text2_LostFocus()

IsValid ("text2")

End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)

If KeyAscii = 13 Then Text4.SetFocus

End Sub

Private Sub Text3_LostFocus()

IsValid ("text3")

End Sub

Private Sub Text4_GotFocus()

Text4.Text = "BCA"

End Sub

Private Sub text4_KeyPress(KeyAscii As Integer)

If KeyAscii = 13 Then Text5.SetFocus

End Sub

Private Sub Text5_KeyPress(KeyAscii As Integer)

If (KeyAscii < 97 Or KeyAscii > 122) And (KeyAscii < 65 Or KeyAscii > 90) And KeyAscii <> 8 And KeyAscii <> 32 And KeyAscii <> 13 Then

MsgBox " NUMBER AND SPECIAL CHARACTER ARE NOT ALLOW ! "

KeyAscii = 0

End If

KeyAscii = Asc(UCase(Chr(KeyAscii)))

```
If KeyAscii = 13 Then
'Text5.Text = UCase(Text5.Text)
Text6.SetFocus
End If
End Sub
```

```
Private Sub Text6_KeyPress(KeyAscii As Integer)
If (KeyAscii < 97 Or KeyAscii > 122) And (KeyAscii < 65 Or KeyAscii > 90) And KeyAscii <> 8 And KeyAscii <> 32 And KeyAscii <> 13 Then
MsgBox " NUMBER AND SPECIAL CHARACTER ARE NOT ALLOW !"
KeyAscii = 0
End If
KeyAscii = Asc(UCase(Chr(KeyAscii)))
If KeyAscii = 13 Then
'Text6.Text = UCase(Text6.Text)
Text7.SetFocus
End If
End Sub
```

```
Private Sub Text7_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then Text8.SetFocus
End Sub
```

```
Private Sub Text8_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then Text9.SetFocus
End Sub
```

```
Private Sub Text9_KeyPress(KeyAscii As Integer)
If KeyAscii = 32 Then
```

```

MsgBox "SPACE NOT ALLOWED"
KeyAscii = 0
Else
If (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 8 And KeyAscii <>
13 Then
MsgBox "ONLY DIGIT ALLOWED"
KeyAscii = 0
End If
If (KeyAscii = 13) Then
If (Text9.Text <> "") Then
Text11.SetFocus
Else
KeyAscii = 0
MsgBox "INPUT CONTACT NO. FIRST"
End If
If (Len(Text9.Text) <> 10) Then
MsgBox "Enter 10 digits"
Text9.SetFocus
End If
Text11.SetFocus
End If
End If
End Sub

```

```

Private Sub Text9_LostFocus()
l = 0
l = Left(Text9.Text, 1)
M = (Len(Text9.Text))
If (((l = 6) Or (l = 7) Or (l = 8) Or (l = 9)) And (M = 10)) Then

```

```

Text11.SetFocus
Else
MsgBox "INVALID NUMBER"
Text9.Text = ""
Text9.SetFocus
End If
End Sub

```

```

Private Sub update_Click()
If Len(Text3.Text) <> 13 Then
    MsgBox "UNIVERSITY ROLL NO. SHOULD BE 13 DIGIT"
    Text3.SetFocus
ElseIf Len(Text1.Text) <> 15 Then
    MsgBox " REGISTRATION NO. SHOULD BE 15 DIGIT"
    Text1.SetFocus
ElseIf Len(Text2.Text) <> 5 Then
    MsgBox "ADMMISION NO. SHOULD BE 5DIGIT"
    Text2.SetFocus
ElseIf Len(Text9.Text) <> 10 Then
    MsgBox "MOBILE NO. SHOULD BE 10 DIGIT"
    Text9.SetFocus
Else

```

```

abc

```

```

On Error GoTo ERRHANDLE

```

```

If Text3.Text = "" Then

```

```

    sql = "update stud set REGno=" + Text1.Text + ", course=" +
Text4.Text + ",nm=" + Text5.Text + ",fnm=" + Text6.Text + ",gender="
+ Text7.Text + ",batch = " + Text8.Text + ",phnno = " + Text9.Text +

```

```
",dob = "" + Text10.Text + "", ses="" + Text11.Text + "" Where ADMNO = "
+ Text2.Text + " "
```

```
Else
```

```
    sql = "update stud set REGno=" + Text1.Text + ",univno=" +
Text3.Text + ",course=" + Text4.Text + ",nm=" + Text5.Text + ",fnm="
+ Text6.Text + ",gender=" + Text7.Text + ",batch = " + Text8.Text +
",phnno = " + Text9.Text + ",dob = " + Text10.Text + ", ses=" +
Text11.Text + "" Where ADMNO = " + Text2.Text + " "
```

```
End If
```

```
Set r = c.Execute(sql)
```

```
MsgBox "RECORD UPDATED"
```

```
clean
```

```
Exit Sub
```

```
ERRHANDLE:
```

```
    If (Err.Description <> "") Then
```

```
        MsgBox Err.Description
```

```
        Exit Sub
```

```
    End If
```

```
End If
```

```
End Sub
```

PRJ_DTLFRM:

```
Dim TEXT7F As Boolean
```

```
Dim TEXT6F As Boolean
```

```
Private Sub ADDNEW_Click()
```

```
    Frame1.Enabled = True
```

```
    Text1.Enabled = True
```

```
    Text2.Enabled = True
```



```
Text3.Enabled = True
Text4.Enabled = True
Text5.Enabled = True
Text6.Enabled = True
Text7.Enabled = True
text21.Enabled = True
Text8.Enabled = False
Combo1.Enabled = True
Combo2.Enabled = False
save.Enabled = True
delete.Enabled = False
update.Enabled = False
search.Enabled = False
reset.Enabled = True
clean
MonthView1.Visible = False
End Sub
```

```
Private Sub Combo1_Click()
Text1.Text = Combo1.Text
If Text1.Text = "" Then
Combo1.SetFocus
Else
Combo2.Enabled = True
Combo2.SetFocus
End If
End Sub
```

```
Private Sub Combo1_LostFocus()
```

```
Combo2.Enabled = True
```

```
End Sub
```

```
Private Sub Combo2_Click()
```

```
text21.Text = Combo2.Text
```

```
Text2.SetFocus
```

```
End Sub
```

```
Private Sub Combo2_LostFocus()
```

```
abc
```

```
'sql = "select max(to_number(substr(prjid,8,1))) from prjdtl"
```

```
sql = "select max(substr(prjid,8,1)) from prjdtl where batch='" +  
Text1.Text + "'and year='" + text21.Text + """
```

```
Set r = c.Execute(sql)
```

```
'MsgBox r.Fields(0)
```

```
If IsNull((r.Fields(0))) Then
```

```
Text8.Text = "PRJ" & Right(text21.Text, 2) & Left(Text1.Text, 1) &  
Right(Text1.Text, 1) & "A"
```

```
ElseIf (Asc(r.Fields(0)) < 92) Then
```

```
Text8.Text = "PRJ" & Right(text21.Text, 2) & Left(Text1.Text, 1) &  
Right(Text1.Text, 1) & Chr(Asc(r.Fields(0)) + 1)
```

```
Else
```

```
MsgBox "Can not add more project"
```

```
End If
```

```
'Label1.Caption = "PRJ" & Right(Text21.Text, 2) & Left(Text1.Text, 1) &  
Right(Text1.Text, 1)
```

```
End Sub
```

```

Private Sub delete_Click()
If (Text1.Text = "" Or Text2.Text = "" Or Text3.Text = "" Or Text4.Text =
"" Or Text5.Text = "" Or Text6.Text = "" Or Text7.Text = "") Then
MsgBox "PLEASE ENTER THE REGISTRATION NUMBER !"
Text8.SetFocus
Else
On Error GoTo ERRHADLE
sql = "delete from memdtl where prjid=" + Text8.Text + ""
Set r = c.Execute(sql)
sql = "delete from prjdtl where prjid = " + Text8.Text + ""
Set r = c.Execute(sql)
MsgBox "RECORD DELETED"
Exit Sub
ERRHANDLE:
If (Err.Description <> "") Then
    MsgBox Err.Description
    Exit Sub
End If
End If
End Sub

Private Sub rn_KeyPress(Index As Integer, KeyAscii As Integer)
If (KeyAscii = 13) Then
    getbyroll Index, rn(Index).Text
    If Index < ind Then
        End If
End If
End Sub

```

```
Private Sub SAVE_Click()
```

```
abc
```

```
If (Text8.Text = "" Or Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Or Text5.Text = "" Or Text6.Text = "" Or Text7.Text = "") Then
```

```
    MsgBox "ALL FIELDS ARE MANDATORY !"
```

```
Else
```

```
    If (T1(0).Text = "" Or r1(0).Text = "" Or m1(0).Text = "") Then
```

```
        MsgBox "enter at least two member"
```

```
    Else
```

```
        Text8.Text = Trim(Text8.Text)
```

```
        On Error GoTo ERRHANDLE
```

```
        sql = "insert into prjdtl values('" + Text1.Text + "','" + text21.Text + "','" + Text8.Text + "','" + Text2.Text + "','" + Text3.Text + "','" + Text4.Text + "','" + Text5.Text + "','" + Text6.Text + "','" + Text7.Text + "')"

```

```
        'MsgBox sql
```

```
        'MsgBox "RECORD SAVED"
```

```
        Set r = c.Execute(sql)
```

```
        saveToMember
```

```
        MsgBox "RECORD SAVED"
```

```
    End If
```

```
Exit Sub
```

```
ERRHANDLE:
```

```
    If (Err.Description <> "") Then
```

```
        MsgBox Err.Description
```

```
    Exit Sub
```

```
    End If
```

```
End If
```

End Sub

Private Sub Form_Load()

abc

Dim SES As Integer

Combo1.AddItem "3-1"

Combo1.AddItem "3-2"

Combo1.AddItem "3-3"

Combo1.AddItem "3-4"

On Error GoTo ERRHANDLE

sql = "SELECT DISTINCT SES FROM STUD ORDER BY SES "

Set r = c.Execute(sql)

While r.EOF = False

SES = 0

SES = Int(Right(r.Fields(0), 6)) - 1

Combo2.AddItem SES

r.MoveNext

Wend

ERRHANDLE:

If (Err.Description <> "") Then

MsgBox Err.Description

Exit Sub

End If

Text1.Locked = True

text21.Locked = True

Text6.Locked = True

Text7.Locked = True

```
Text8.Enabled = False
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text5.Enabled = False
Text6.Enabled = False
Text7.Enabled = False
text21.Enabled = False
Frame1.Enabled = False
Combo1.Enabled = False
Combo2.Enabled = False
save.Enabled = False
update.Enabled = False
delete.Enabled = False
reset.Enabled = False
MonthView1.Visible = False
End Sub
```

```
Private Sub Text10_KeyPress(KeyAscii As Integer)
If (KeyAscii = 13) Then
getbyroll 1, Text10.Text
Text11.SetFocus
End If
End Sub
```

```
Private Sub Text11_Change()
If (KeyAscii = 13) Then
```

```
getbyroll 2, Text11.Text  
Text12.SetFocus  
End Sub
```

```
Private Sub Text12_Change()  
If (KeyAscii = 13) Then  
getbyroll 3, Text12.Text  
Text4.SetFocus  
End Sub
```

```
Private Sub MonthView1_DateClick(ByVal DateClicked As Date)  
If TEXT7F Then  
Text7.Text = Format(MonthView1.Value, "dd mmm yyyy")  
ElseIf TEXT6F Then  
Text6.Text = Format(MonthView1.Value, "dd mmm yyyy")  
End If  
MonthView1.Visible = False  
  
End Sub
```

```
Private Sub reset_Click()  
clean  
search.Enabled = True  
update.Enabled = False  
delete.Enabled = False  
save.Enabled = False  
End Sub
```

```

Private Sub search_Click()
Text8.Enabled = True
MsgBox " ENTER PRJID !"
Text8.SetFocus
update.Enabled = True
delete.Enabled = True
Text1.Enabled = True
Text2.Enabled = True
Text3.Enabled = True
Text4.Enabled = True
Text5.Enabled = True
Text6.Enabled = True
Text7.Enabled = True
text21.Enabled = True
reset.Enabled = True
Combo1.Enabled = False
Combo2.Enabled = False
End Sub

```

```

Private Sub Text2_KeyPress(KeyAscii As Integer)
If (KeyAscii < 97 Or KeyAscii > 122) And (KeyAscii < 65 Or KeyAscii > 90) And KeyAscii <> 8 And KeyAscii <> 32 And KeyAscii <> 13 Then
MsgBox " NUMBER AND SPECIAL CHARACTER ARE NOT ALLOW !"
KeyAscii = 0
End If
KeyAscii = Asc(UCase(Chr(KeyAscii)))
If KeyAscii = 13 Then
Text3.SetFocus
End If

```


End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)

If (KeyAscii < 97 Or KeyAscii > 122) And (KeyAscii < 65 Or KeyAscii > 90) And KeyAscii <> 8 And KeyAscii <> 32 And KeyAscii <> 13 Then

MsgBox " NUMBER AND SPECIAL CHARACTER ARE NOT ALLOW ! "

KeyAscii = 0

End If

KeyAscii = Asc(UCase(Chr(KeyAscii)))

If KeyAscii = 13 Then

rn(0).SetFocus

End If

End Sub

Private Sub text4_KeyPress(KeyAscii As Integer)

KeyAscii = Asc(UCase(Chr(KeyAscii)))

If KeyAscii = 13 Then Text5.SetFocus

End Sub

Private Sub Text5_KeyPress(KeyAscii As Integer)

KeyAscii = Asc(UCase(Chr(KeyAscii)))

If KeyAscii = 13 Then Text6.SetFocus

End Sub

Private Sub Text6_GotFocus()

MonthView1.Visible = True

TEXT6F = True

TEXT7F = False

End Sub

```
Private Sub Text6_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then Text7.SetFocus
End Sub
```

```
Private Sub Text7_GotFocus()
MonthView1.Visible = True
TEXT6F = False
TEXT7F = True
End Sub
```

```
Private Sub Text8_KeyPress(KeyAscii As Integer)
On Error GoTo pqr
KeyAscii = Asc(UCase(Chr(KeyAscii)))
If KeyAscii = 13 Then
abc
sql = "select * from prjdtl where prjid='" + Text8.Text + "'"
Set r = c.Execute(sql)
If r.EOF = False Then
Text1.Text = r.Fields(0)
text21.Text = r.Fields(1)
Text2.Text = r.Fields(3)
Text3.Text = r.Fields(4)
Text4.Text = r.Fields(5)
Text5.Text = r.Fields(6)
Text6.Text = Format(r.Fields(7), "dd mmm yyyy")
Text7.Text = Format(r.Fields(8), "dd mmm yyyy")
Text8.Text = r.Fields(2)
```

```

'-----

sql = "select * from memdt1 where prjid='" + Text8.Text + "'"
Set r = c.Execute(sql)
Dim I As Integer
I = 0
While r.EOF = False
    rn(I) = r.Fields(0)
    T1(I) = r.Fields(1)
    r1(I) = r.Fields(2)
    m1(I) = r.Fields(3)
    r.MoveNext
    I = I + 1
Wend
Else
    MsgBox " NO RECORD FOUND ! "
End If
Frame1.Enabled = True
Exit Sub

pqr:
MsgBox " NO RECORD FOUND ! "
End If
End Sub

Private Sub update_Click()
    abc
    If (Text1.Text = "" Or Text2.Text = "" Or Text3.Text = "" Or Text4.Text =
    "" Or Text5.Text = "" Or Text6.Text = "" Or Text7.Text = "" Or Text8.Text
    = "" Or text21.Text = "") Then
        MsgBox "PLEASE ENTER THE PROJECT ID !"
    
```

Else

If (T1(0).Text = "" Or r1(0).Text = "" Or m1(0).Text = "") Then

MsgBox "enter at least two member"

Else

Text8.Text = Trim(Text8.Text)

On Error GoTo ERRHADLE

sql = "update prjdtl set prjtitle='" + Text2.Text + "',compnm='" + Text3.Text + "',fend='" + Text4.Text + "',bend='" + Text5.Text + "', stdt='" + Text6.Text + "',sbdtd='" + Text7.Text + "where prjid='" + Text8.Text + ""

Set r = c.Execute(sql)

sql = "delete from memdtdl where prjid='" + Text8.Text + ""

Set r = c.Execute(sql)

saveToMember

MsgBox "RECORD UPDATED"

clean

End If

Exit Sub

ERRHANDLE:

If (Err.Description <> "") Then

MsgBox Err.Description

Exit Sub

End If

End If

End Sub

Function saveToMember()

Dim ind As Integer

ind = 0

```

If (T1(0).Text <> "" And r1(0).Text <> "" And m1(0) <> "") Then
ind = ind + 1
End If
If (T1(1).Text <> "" And r1(1).Text <> "" And m1(1) <> "") Then
ind = ind + 1
End If
If (T1(2).Text <> "" And r1(2).Text <> "" And m1(2) <> "") Then
ind = ind + 1
End If
If (T1(3).Text <> "" And r1(3).Text <> "" And m1(3) <> "") Then
ind = ind + 1
End If
MsgBox "Total Member " & ind
For I = 0 To ind - 1 Step 1
sql = "insert into memdtl values(" + rn(I).Text + "," & T1(I).Text & "," &
r1(I).Text & "," & m1(I).Text & "," & Text8.Text & ")"
Set r = c.Execute(sql)
Next
End Function

```

```

Function getbyroll(I As Integer, txt As String)
If (txt <> "" And Text1.Text <> "") Then
    sql = "select nm,regno,phnno from stud where admno=" & txt & "
and batch=" & Text1.Text & ""
    Set r = c.Execute(sql)
    If r.EOF = False Then
        T1(I).Text = r.Fields("nm")
        r1(I).Text = r.Fields(1)
        m1(I).Text = r.Fields(2)
    End If
End Function

```

```

    If I < 3 Then rn(I + 1).SetFocus
    Else
    MsgBox "record not found"
    rn(I).SetFocus
    T1(I).Text = ""
    r1(I).Text = ""
    m1(I).Text = ""
    End If
End If
End Function

```

```

Function clean()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
text21.Text = ""
Dim I As Integer
For I = 0 To 3
rn(I).Text = ""
T1(I).Text = ""
r1(I).Text = ""
m1(I).Text = ""
Next

```

End Function

GUIDE_DATAFRM:

Private Sub Combo1_Click()

sql = "SELECT * FROM GUIDE WHERE code=" + Combo1.Text + ""

Set r = c.Execute(sql)

Text1.Text = r.Fields(0)

Text2.Text = r.Fields(1)

Text3.Text = r.Fields(2)

Text4.Text = r.Fields(3)

Text5.Text = r.Fields(4)

Text6.Text = r.Fields(5)

Text7.Text = r.Fields(6)

End Sub

Private Sub delete_Click()

abc

If (Text1.Text = "" Or Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Or Text5.Text = "" Or Text6.Text = "" Or Text7.Text = "") Then

MsgBox "PLEASE SELECT THE CODE !"

Combo1.SetFocus

Else

On Error GoTo ERRHANDLE

sql = "delete from guide where code=" + Combo1.Text + ""

Set r = c.Execute(sql)

MsgBox "RECORD DELETED"

Exit Sub

ERRHANDLE:

```
If (Err.Description <> "") Then
    MsgBox Err.Description
    Exit Sub
End If
End If
clean
End Sub
```

```
Private Sub Form_Load()
abc
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text5.Enabled = False
Text6.Enabled = False
Text7.Enabled = False
save.Enabled = False
update.Enabled = False
delete.Enabled = False
reset.Enabled = True
Frame1.Enabled = False
End Sub
```

```
Private Sub new_Click()
clean
Text1.Enabled = True
Text2.Enabled = True
```



```

Text3.Enabled = True
Text4.Enabled = True
Text5.Enabled = True
Text6.Enabled = True
Text7.Enabled = True
save.Enabled = True
reset.Enabled = True
update.Enabled = False
delete.Enabled = False
abc
sql = "select max(substr(code,2,length(code))) from guide "
Set r = c.Execute(sql)
If IsNull(r.Fields(0)) Then
    Text1.Text = "G001"
    ElseIf ((r.Fields(0)) < 9) Then
        Text1.Text = "G" & "00" & r.Fields(0) + 1
        ElseIf ((r.Fields(0)) < 99) Then
            Text1.Text = "G" & "0" & r.Fields(0) + 1

Else

    MsgBox "Can not add more project"
End If
Text1.Text = Trim(Text1.Text)
End Sub

Private Sub reset_Click()
clean

```

End Sub

Private Sub SAVE_Click()

abc

If (Text1.Text = "" Or Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Or Text5.Text = "" Or Text6.Text = "" Or Text7.Text = "") Then

MsgBox "ALL FIELDS ARE MANDATORY !"

Else

Text1.Text = Trim(Text1.Text)

On Error GoTo ERRHANDLE

sql = "insert into guide values(" + Text1.Text + "," + Text2.Text + "," + Text3.Text + "," + Text4.Text + "," + Text5.Text + "," + Text6.Text + "," + Text7.Text + ")"

Set r = c.Execute(sql)

MsgBox "RECORD SAVED"

Exit Sub

ERRHANDLE:

If (Err.Description <> "") Then

MsgBox Err.Description

Exit Sub

End If

End If

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Text5.Text = ""

Text6.Text = ""

Text7.Text = ""

```
search.Enabled = True
```

```
reset.Enabled = True
```

```
End Sub
```

```
Private Sub search_Click()
```

```
abc
```

```
clean
```

```
update.Enabled = True
```

```
delete.Enabled = True
```

```
Frame1.Enabled = True
```

```
Combo1.SetFocus
```

```
save.Enabled = False
```

```
Combo1.Clear
```

```
sql = "SELECT code FROM GUIDE "
```

```
Set r = c.Execute(sql)
```

```
Do While Not r.EOF
```

```
Combo1.AddItem r.Fields(0)
```

```
r.MoveNext
```

```
Loop
```

```
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
If (KeyAscii = 13) Then Text2.SetFocus
```

```
End Sub
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)
```

```
If (KeyAscii < 97 Or KeyAscii > 122) And (KeyAscii < 65 Or KeyAscii >  
90) And KeyAscii <> 8 And KeyAscii <> 32 And KeyAscii <> 13 Then
```

```
MsgBox " NUMBER AND SPECIAL CHARACTER ARE NOT ALLOW ! "  
KeyAscii = 0  
End If  
If KeyAscii = 13 Then  
Text2.Text = UCase(Text2.Text)  
Text3.SetFocus  
End If  
End Sub
```

```
Private Sub Text3_KeyPress(KeyAscii As Integer)  
If (KeyAscii < 97 Or KeyAscii > 122) And (KeyAscii < 65 Or KeyAscii >  
90) And KeyAscii <> 8 And KeyAscii <> 32 And KeyAscii <> 13 Then  
MsgBox " NUMBER AND SPECIAL CHARACTER ARE NOT ALLOW ! "  
KeyAscii = 0  
End If  
If KeyAscii = 13 Then  
Text3.Text = UCase(Text3.Text)  
Text4.SetFocus  
End If  
End Sub
```

```
Private Sub text4_KeyPress(KeyAscii As Integer)  
If KeyAscii = 32 Then  
MsgBox "SPACE NOT ALLOWED"  
KeyAscii = 0  
Else  
If (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 8 And KeyAscii <>  
13 Then  
MsgBox "ONLY DIGIT ALLOWED"
```

```

KeyAscii = 0
End If
If (KeyAscii = 13) Then
If (Text4.Text <> "") Then
Else
KeyAscii = 0
MsgBox "INPUT CONTACT NO. FIRST"
End If
If (Len(Text4.Text) <> 10) Then
MsgBox "Enter 10 digits"
End If
Text5.SetFocus
End If
End If
End Sub

```

```

Private Sub Text4_LostFocus()
l = 0
l = Left(Text4.Text, 1)
M = (Len(Text4.Text))
If (((l = 6) Or (l = 7) Or (l = 8) Or (l = 9)) And (M = 10)) Then
Text5.SetFocus
Else
MsgBox "INVALID NUMBER"
Text4.Text = ""
Text4.SetFocus
End If
End Sub

```

```
Private Sub Text5_KeyPress(KeyAscii As Integer)
If (KeyAscii = 13) Then Text6.SetFocus
End Sub
```

```
Private Sub Text6_KeyPress(KeyAscii As Integer)
If (KeyAscii = 13) Then Text7.SetFocus
End Sub
```

```
Private Sub Text7_KeyPress(KeyAscii As Integer)
KeyAscii = Asc(UCase(Chr(KeyAscii)))
End Sub
```

```
Function clean()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Combo1.Text = ""
```

```
End Function
```

```
Private Sub update_Click()
abc
If (Text1.Text = "" Or Text2.Text = "" Or Text3.Text = "" Or Text4.Text =
"" Or Text5.Text = "" Or Text6.Text = "" Or Text7.Text = "") Then
```

```

MsgBox "PLEASE SELECT THE CODE !"
Combo1.SetFocus
Else
Text1.Text = Trim(Text1.Text)
On Error GoTo ERRHANDLE
sql = "update guide set nm=" + Text2.Text + ",fnm=" + Text3.Text +
",phnno=" + Text4.Text + ",email=" + Text5.Text + ",exp=" +
Text6.Text + ",addr=" + Text7.Text + " Where code =" + Combo1.Text
+ " "
Set r = c.Execute(sql)
MsgBox "RECORD UPDATED"
Exit Sub
ERRHANDLE:
If (Err.Description <> "") Then
    MsgBox Err.Description
    Exit Sub
End If
clean
End If
End Sub

GUIDE_DTLFRM:

Private Sub Combo1_Click()
abc
sql = "SELECT NM,PHNnO FROM GUIDe WHERE code=" +
Combo1.Text + ""
Set r = c.Execute(sql)
Text4.Text = r.Fields(0)
Text5.Text = r.Fields(1)

```

End Sub

Private Sub Combo2_Click()

sql = "select distinct year from prjdtl where batch='" + Combo2.Text +
" order by year"

Set r = c.Execute(sql)

Combo3.Clear

While r.EOF = False

Combo3.AddItem r.Fields(0)

r.MoveNext

Wend

End Sub

Private Sub Combo3_Click()

sql = "select prjid from prjdtl where batch='" & Combo2.Text & "' and
year='" & Combo3.Text & "'"

Set r = c.Execute(sql)

Combo4.Clear

While r.EOF = False

Combo4.AddItem r.Fields(0)

r.MoveNext

Wend

End Sub

Private Sub Combo4_Click()

sql = "select compnm,prjtitle from prjdtl where prjid='" + Combo4.Text
+ "'"


```
Set r = c.Execute(sql)
If r.EOF = False Then
Text1.Text = r.Fields(1)
Text2.Text = r.Fields(0)
End If
```

```
End Sub
```

```
Private Sub delete_Click()
abc
On Error GoTo ERRHANDLE
sql = "delete from prjguide where prjid='" + Text6.Text + "'"
Set r = c.Execute(sql)
MsgBox "RECORD DELETED"
Exit Sub
ERRHANDLE:
If (Err.Description <> "") Then
    MsgBox Err.Description
    Exit Sub
End If
cln
End Sub
```

```
Private Sub newadd_Click()
save(1).Enabled = True
update.Enabled = False
delete.Enabled = False
reset.Enabled = True
```

End Sub

```
Private Sub reset_Click()  
newadd.Enabled = True  
update.Enabled = False  
delete.Enabled = False  
search.Enabled = True  
save(1).Enabled = False  
cln  
End Sub
```

```
Private Sub search_Click()  
abc  
update.Enabled = True  
delete.Enabled = True  
save(1).Enabled = False  
On Error GoTo ERRORHANDLE  
sql = "select * from prjguide where prjid='" + Text6.Text + "'"<br>  
Set r = c.Execute(sql)  
Text1.Text = r.Fields(2)  
Text2.Text = r.Fields(1)  
Combo1.Text = r.Fields(3)  
Text4.Text = r.Fields(4)  
Text5.Text = r.Fields(5)  
Exit Sub  
ERRORHANDLE:  
If (Err.Description <> "") Then  
    MsgBox Err.Description
```

```

Exit Sub
End If
End If
Exit Sub

End Sub

Private Sub Form_Load()
abc
sql = "SELECT code FROM GUIDe "
Set r = c.Execute(sql)
Do While Not r.EOF
Combo1.AddItem r.Fields(0)
r.MoveNext
Loop
save(1).Enabled = False
delete.Enabled = False
update.Enabled = False
reset.Enabled = True
search.Enabled = True
Combo2.AddItem "3-1"
Combo2.AddItem "3-2"
Combo2.AddItem "3-3"
Combo2.AddItem "3-4"

End Sub

Function cln()
Text1.Text = ""

```

```

Text2.Text = ""
Text4.Text = ""
Text5.Text = ""
Combo1.Text = ""
Combo2.Text = ""
Combo3.Text = ""
Combo4.Text = ""
End Function

Private Sub SAVE_Click(Index As Integer)
    abc

    On Error GoTo ERRHANDLE

    sql = "insert into prjguide values('" + Text6.Text + "','" + Text2.Text +
    "','" + Text1.Text + "','" + Combo1.Text + "','" + Text4.Text + "','" +
    Text5.Text + "')"

    Set r = c.Execute(sql)

    MsgBox "RECORD SAVED"

    Exit Sub

ERRHANDLE:

    If (Err.Description <> "") Then
        MsgBox Err.Description
        Exit Sub
    End If
End If

Exit Sub

cln

End Sub

Private Sub Text6_KeyPress(KeyAscii As Integer)

```

```

On Error GoTo lmn
KeyAscii = Asc(UCase(Chr(KeyAscii)))
If KeyAscii = 13 Then
abc
sql = "select prjtitle,compnm from prjdtl where prjid='" + Text6.Text +
""
Set r = c.Execute(sql)
Text1.Text = r.Fields(1)
Text2.Text = r.Fields(0)
End If
Exit Sub
lmn:
MsgBox "RECORD NOT FOUND"
Text6.Text = ""
Text6.SetFocus
End Sub

Private Sub update_Click()
abc
On Error GoTo ERRHADLE
sql = "update prjguide set gid = '" + Combo1.Text + "',gnm='" +
Text4.Text + "',gphno='" + Text5.Text + "' where prjid='" + Text6.Text +
""
Set r = c.Execute(sql)
MsgBox "RECORD UPDATED"
Exit Sub
ERRHANDLE:
If (Err.Description <> "") Then
MsgBox Err.Description

```

```
Exit Sub
End If
cln
End Sub
```

DOCUMNT_MARKING:

```
Dim UPDATE1 As Boolean
```

```
Private Sub Combo1_Click()
sql = "select distinct year from prjdtl where batch='" + Combo1.Text +
"" order by year"
Set r = c.Execute(sql)
Combo2.Clear
While r.EOF = False
    Combo2.AddItem r.Fields(0)
    r.MoveNext
Wend
End Sub
```

```
Private Sub Combo2_Click()
sql = "select prjid from prjdtl where batch='" & Combo1.Text & "' and
year='" & Combo2.Text & "'"
Set r = c.Execute(sql)
Combo3.Clear
While r.EOF = False
    Combo3.AddItem r.Fields(0)
    r.MoveNext
Wend
```

End Sub

Private Sub Combo3_Click()

RESET.Enabled = True

Frame1.Enabled = True

Frame2.Enabled = True

Dim I As Integer

Dim DocExist, MemExist As Boolean

UPDATE.Enabled = False

delete.Enabled = False

SAVE.Enabled = False

MemExist = False

UPDATE1 = False

sql = "SELECT DISTINCT * FROM DOCMARK WHERE PRJID=" +
Combo3.Text + " ORDER BY SBMDT"

Set r = c.Execute(sql)

I = 0

If (r.EOF = False) Then

UPDATE1 = True

UPDATE.Enabled = True

SAVE.Enabled = False

Else

UPDATE1 = False

UPDATE.Enabled = False

SAVE.Enabled = True

End If

While r.EOF = False

LVL(I).Text = r.Fields(1)

```

    STCOMB(I) = r.Fields(2)
    DTPicker1(I).Value = r.Fields(3)
    GRD(I) = r.Fields(4)
    r.MoveNext
    I = I + 1
Wend
End Sub

```

```

Private Sub delete_Click()
MARKING_REPORTFRM.Show
End Sub

```

```

Private Sub Form_Load()
abc
SAVE.Enabled = False
UPDATE.Enabled = False
RESET.Enabled = False
Frame1.Enabled = False
Frame2.Enabled = False
For I = 0 To 3
    GRD(I).Enabled = False
    STCOMB(I).Enabled = False
    DTPicker1(I).Enabled = False
    rn(I).Locked = True
    T1(I).Locked = True
    r1(I).Locked = True
Next
Frame2.Enabled = False

```



```

Combo1.AddItem "3-1"
Combo1.AddItem "3-2"
Combo1.AddItem "3-3"
Combo1.AddItem "3-4"
For I = 0 To 3 Step 1
    STCOMB(I).AddItem "SUBMITTED"
    STCOMB(I).AddItem "NOT SUBMITTED"
Next
End Sub

```

```

Function saveToDOCUMENT()
Dim ind As Integer
ind = 0
If (LVL(0).Text <> "" And STCOMB(0).Text <> "" And GRD(0) <> "") Then
ind = ind + 1
End If
If (LVL(1).Text <> "" And STCOMB(1).Text <> "" And GRD(1) <> "") Then
ind = ind + 1
End If
If (LVL(2).Text <> "" And STCOMB(2).Text <> "" And GRD(2) <> "") Then
ind = ind + 1
End If
If (LVL(3).Text <> "" And STCOMB(3).Text <> "" And GRD(3) <> "") Then
ind = ind + 1
End If
For I = 0 To ind - 1 Step 1
    sql = "insert into docmark values('" + Combo3.Text + "', '" +
LVL(I).Text + "', '" & STCOMB(I).Text & "', '" &
Format(DTPicker1(I).Value, "dd-MMM-yyyy") & "', '" & GRD(I).Text & "')"

```

```

    MsgBox sql
    Set r = c.Execute(sql)
Next
End Function

Function saveToMember()
If ((Option1.Value = False) And (Option2.Value = True Or
Option3.Value = True Or Option4.Value = True)) Then
    Dim ind As Integer
    Dim lvltext As String
    lvltext = ""
    If (Option2.Value = True) Then
        lvltext = LVL(1).Text
    ElseIf (Option3.Value = True) Then
        lvltext = LVL(2).Text
    ElseIf (Option4.Value = True) Then
        lvltext = LVL(3).Text
    End If
    ind = 0
    If (rn(0).Text <> "" And T1(0).Text <> "" And r1(0).Text <> "" And
m1(0) <> "") Then
        ind = ind + 1
    End If
    If (T1(1).Text <> "" And r1(1).Text <> "" And m1(1) <> "") Then
        ind = ind + 1
    End If
    If (T1(2).Text <> "" And r1(2).Text <> "" And m1(2) <> "") Then
        ind = ind + 1
    End If

```

```

If (T1(3).Text <> "" And r1(3).Text <> "" And m1(3) <> "") Then
    ind = ind + 1
End If

For I = 0 To ind - 1 Step 1
    sql = "insert into MEMMARK values('" & Combo3.Text & "','" &
    lvltext & "','" & rn(I).Text & "','" & T1(I).Text & "','" & r1(I).Text & "','" &
    m1(I).Text & "')"
    Set r = c.Execute(sql)
Next
MsgBox "RECORD SAVED ;) "
End If
End Function

```

```

Public Function clean()
    Combo1.Text = ""
    Combo2.Text = ""
    Combo3.Text = ""
    For I = 0 To 3
        STCOMB(I).Text = ""
        LVL(I).Text = ""
        GRD(I).Text = ""
        rn(I).Text = ""
        T1(I).Text = ""
        r1(I).Text = ""
        m1(I).Text = ""
    Next
End Function

```

```

Private Sub GRD_KeyPress(Index As Integer, KeyAscii As Integer)

```

```
KeyAscii = Asc(UCase(Chr(KeyAscii)))  
End Sub
```

```
Private Sub LVL_KeyPress(Index As Integer, KeyAscii As Integer)  
KeyAscii = Asc(UCase(Chr(KeyAscii)))  
End Sub
```

```
Private Sub Option1_Click()  
CLNMARK  
If (UPDATE1) Then  
UPDATE.Enabled = True  
SAVE.Enabled = False  
Else  
UPDATE.Enabled = False  
SAVE.Enabled = True  
End If  
GRD(0).Enabled = True  
STCOMB(0).Enabled = True  
DTPicker1(0).Enabled = True  
Frame2.Enabled = False  
End Sub
```

```
Private Sub Option2_Click()  
GRD(0).Enabled = False  
GRD(1).Enabled = True  
STCOMB(0).Enabled = False  
STCOMB(1).Enabled = True  
DTPicker1(0).Enabled = False
```

```

DTPicker1(1).Enabled = True
Frame2.Enabled = True
GETMEMMARK (2)
End Sub

Private Sub Option3_Click()
GRD(0).Enabled = False
GRD(1).Enabled = False
GRD(2).Enabled = True
STCOMB(0).Enabled = False
STCOMB(1).Enabled = False
STCOMB(2).Enabled = True
DTPicker1(0).Enabled = False
DTPicker1(1).Enabled = False
DTPicker1(2).Enabled = True
Frame2.Enabled = True
GETMEMMARK (2)
End Sub

Function CLNMARK()
For I = 0 To 3
rn(I).Text = ""
T1(I).Text = ""
r1(I).Text = ""
m1(I).Text = ""
Next
End Function

Private Sub Option4_Click()
GRD(0).Enabled = False
GRD(1).Enabled = False

```

```
GRD(2).Enabled = False
GRD(3).Enabled = True
STCOMB(0).Enabled = False
STCOMB(1).Enabled = False
STCOMB(2).Enabled = False
STCOMB(3).Enabled = True
DTPicker1(0).Enabled = False
DTPicker1(1).Enabled = False
DTPicker1(2).Enabled = False
DTPicker1(3).Enabled = True
Frame2.Enabled = True
GETMEMMARK (2)
End Sub
```

```
Private Sub reset_Click()
clean
End Sub
```

```
Private Sub SAVE_Click()
saveToDOCUMENT
saveToMember
clean
End Sub
```

```
Private Sub update_Click()
Dim lvltext As String
On Error GoTo ERRHANDLE
sql = "DELETE FROM DOCMARK WHERE PRJID = '" + Combo3.Text +
"""
```

Set r = c.Execute(sql)

saveToDOCUMENT

Dim ind As Integer

ind = 0

If (rn(0).Text <> "" And T1(0).Text <> "" And r1(0).Text <> "" And m1(0) <> "") Then

ind = ind + 1

End If

If (T1(1).Text <> "" And r1(1).Text <> "" And m1(1) <> "") Then

ind = ind + 1

End If

If (T1(2).Text <> "" And r1(2).Text <> "" And m1(2) <> "") Then

ind = ind + 1

End If

If (T1(3).Text <> "" And r1(3).Text <> "" And m1(3) <> "") Then

ind = ind + 1

End If

lvltext = ""

If (Option2.Value = True) Then

lvltext = LVL(1).Text

ElseIf (Option3.Value = True) Then

lvltext = LVL(2).Text

ElseIf (Option4.Value = True) Then

lvltext = LVL(3).Text

End If

If ((Option1.Value = False) And (Option2.Value = True Or Option3.Value = True Or Option4.Value = True)) Then

MsgBox "Total Member " & ind

```

    For I = 0 To ind - 1 Step 1
        sql = "update memmark set GRADE='" + m1(I).Text + "' where
prjid='" + Combo3.Text + "' and clsroll='" + rn(I).Text + "' AND LVL='" &
lvltext & "'"
        Set r = c.Execute(sql)
    Next
End If
MsgBox " RECORD UPDATED :) "
Exit Sub
ERRHANDLE:
If (Err.Description <> "") Then
    MsgBox Err.Description
    Exit Sub
End If
clean
End Sub

```

```

Function GETMEMMARK(opt As Integer)
Dim lvltext As String
Dim MemExist As Boolean
Dim I As Integer
MemExist = False
If (opt = 2 Or opt = 3 Or opt = 4) Then
    If (Option2.Value = True) Then
        lvltext = LVL(1).Text
    ElseIf (Option3.Value = True) Then
        lvltext = LVL(2).Text
    ElseIf (Option4.Value = True) Then
        lvltext = LVL(3).Text
    End If
End If

```


End If

sql = "select DISTINCT * from memmark where prjid = '" +
Combo3.Text + "' and LVL='" & lvltext & "'"

Set r = c.Execute(sql)

I = 0

While r.EOF = False

rn(I) = r.Fields(2)

T1(I) = r.Fields(3)

r1(I) = r.Fields(4)

m1(I) = r.Fields(5)

r.MoveNext

I = I + 1

MemExist = True

UPDATE.Enabled = True

delete.Enabled = True

SAVE.Enabled = False

Wend

If (Not MemExist) Then

sql = "select * from memdtl where prjid='" + Combo3.Text + "'"

Set r = c.Execute(sql)

I = 0

While r.EOF = False

rn(I) = r.Fields(0)

T1(I) = r.Fields(1)

r1(I) = r.Fields(2)

m1(I) = ""

r.MoveNext

I = I + 1

UPDATE.Enabled = False

```
        delete.Enabled = False
        SAVE.Enabled = True
    Wend
End If
End If
End Function
```

CERTIFICATE_ISSUEDFRM:

```
Private Sub Combo1_Click()
    sql = "select distinct year from prjdtl where batch='" + Combo1.Text +
    "' order by year"
    Set r = c.Execute(sql)
    Combo2.Clear
    While r.EOF = False
        Combo2.AddItem r.Fields(0)
        r.MoveNext
    Wend
End Sub
```

```
Private Sub Combo2_Click()
    sql = "select prjid from prjdtl where batch='" & Combo1.Text & "' and
    year='" & Combo2.Text & "'"
    Set r = c.Execute(sql)
    Combo3.Clear
    While r.EOF = False
        Combo3.AddItem r.Fields(0)
        r.MoveNext
    Wend
```

End Sub

Private Sub Combo3_Click()

sql = "select compnm,prjtitle from prjdtl where prjid='" + Combo3.Text + "'"

Set r = c.Execute(sql)

If r.EOF = False Then

Text1.Text = r.Fields(1)

Text2.Text = r.Fields(0)

End If

sql = "select COUNT(REGNO) from MEMDTL where prjid='" + Combo3.Text + "'"

Set r = c.Execute(sql)

If r.EOF = False Then

Text3.Text = r.Fields(0)

End If

End Sub

Private Sub Command1_Click()

CERTIFICATE_ISSUEDFRM.Show

End Sub

Private Sub Form_Load()

abc

Combo1.AddItem "3-1"

Combo1.AddItem "3-2"

Combo1.AddItem "3-3"

Combo1.AddItem "3-4"

End Sub

```

Private Sub PRINT_Click()
If (DataEnvironment1.Connection1.State) Then
DataEnvironment1.Connection1.Close
End If
DataEnvironment1.Connection1.Open
DataEnvironment1.Command9_Grouping Combo3.Text
DataReport11.Show
DataReport12.Show
End Sub

```

```

Private Sub SAVE_Click()
If Option1.Value = True Then
    M = "YES"
Else
    M = "NO"
End If
sql = "insert into CISSUED values('" + Combo3.Text + "','" + Text1.Text
+ "','" + Text2.Text + "','" + Text3.Text + "','" + M + "')"
Set r = c.Execute(sql)
MsgBox " RECORD SAVED ;)"
End Sub

```

SEARCHFRM:

```

Private Sub back_Click()
INFOFRM.Show
End Sub

```

```
Private Sub Combo1_Click()  
Text2.Text = Combo1.Text  
Text1.Text = Combo1.Text  
End Sub
```

```
Private Sub Form_Load()  
abc  
With Combo1  
    .AddItem "3-1"  
    .AddItem "3-2"  
    .AddItem "3-3"  
    .AddItem "3-4"  
End With  
Combo1.Visible = False  
Text2.Visible = False  
Text2.Locked = True  
End Sub
```

```
Private Sub Option1_Click()  
Text1.Text = ""  
Text1.SetFocus  
End Sub
```

```
Private Sub Option2_Click()  
Text1.Text = ""  
Text1.SetFocus  
End Sub
```

```
Private Sub Option3_Click()
```

```
Text1.Text = ""
```

```
Text1.SetFocus
```

```
Combo1.Visible = True
```

```
Text2.Visible = True
```

```
End Sub
```

```
Private Sub Text1_Change()
```

```
If Option1.Value = True Then
```

```
    Adodc1.RecordSource = "select * from stud where ADMNO like" +  
Text1.Text + "%"
```

```
    Adodc1.Refresh
```

```
End If
```

```
If Option2.Value = True Then
```

```
    Adodc1.RecordSource = "select * from stud where NM like" +  
Text1.Text + "%"
```

```
    Adodc1.Refresh
```

```
End If
```

```
If Option3.Value = True Then
```

```
    Adodc1.RecordSource = "select * from stud where batch LIKE" +  
Text1.Text + "%"
```

```
    Adodc1.Refresh
```

```
End If
```

```
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

```
End Sub
```

STUD_INFO_REPORTFRM:

```
Private Sub Combo1_Click()  
Text2.Text = Combo1.Text  
Text1.Text = Combo1.Text  
End Sub
```

```
Private Sub Command1_Click()  
If (Option2.Value = True) Then  
    If (DataEnvironment1.Connection1.State) Then  
        DataEnvironment1.Connection1.Close  
    End If  
    DataEnvironment1.Connection1.Open  
    DataEnvironment1.Command1 Text1.Text  
    DataReport1.Show  
ElseIf (Option1.Value = True) Then  
    If (DataEnvironment1.Connection1.State) Then  
        DataEnvironment1.Connection1.Close  
    End If  
    DataEnvironment1.Connection1.Open  
    DataEnvironment1.Command2 Text1.Text  
    DataReport2.Show  
ElseIf (Option3.Value = True) Then  
    If (DataEnvironment1.Connection1.State) Then  
        DataEnvironment1.Connection1.Close  
    End If  
    DataEnvironment1.Connection1.Open
```

```

        DataEnvironment1.Command3 Text1.Text
        DataReport3.Show
Else
    If (DataEnvironment2.Connection1.State) Then
        DataEnvironment2.Connection1.Close
    End If
    DataEnvironment2.Connection1.Open
    DataEnvironment2.Command1
    DataReport8.Show
End If
End Sub

```

```

Private Sub Form_Load()
abc
With Combo1
    .AddItem "3-1"
    .AddItem "3-2"
    .AddItem "3-3"
    .AddItem "3-4"
End With
Combo1.Visible = False
Text2.Visible = False
Text2.Locked = True
End Sub

```

```

Private Sub Option1_Click()
Text1.Text = ""
Text1.SetFocus

```


End Sub

Private Sub Option2_Click()

Text1.Text = ""

Text1.SetFocus

End Sub

Private Sub Option3_Click()

Text1.Text = ""

Text1.SetFocus

Combo1.Visible = True

Text2.Visible = True

End Sub

Private Sub Text1_Change()

If Option1.Value = True Then

Adodc1.RecordSource = "select * from stud where admno like" +
Text1.Text + "%"

Adodc1.Refresh

End If

If Option2.Value = True Then

Adodc1.RecordSource = "select * from stud where nm like" +
Text1.Text + "%"

Adodc1.Refresh

End If

If Option3.Value = True Then

Adodc1.RecordSource = "select * from stud where batch LIKE" +
Text1.Text + "%"

Adodc1.Refresh

```
End If  
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)  
KeyAscii = Asc(UCase(Chr(KeyAscii)))  
End Sub
```

PROJECT_DTL_REPORTFRM:

```
Private Sub Combo1_Click()  
Text2.Text = Combo1.Text  
Text1.Text = Combo1.Text  
End Sub
```

```
Private Sub Combo2_Click()  
Text3.Text = Combo2.Text  
Text1.Text = Combo2.Text  
End Sub
```

```
Private Sub Command1_Click()  
If (Option1.Value = True) Then  
    If (DataEnvironment1.Connection1.State) Then  
        DataEnvironment1.Connection1.Close  
    End If  
    DataEnvironment1.Connection1.Open  
    DataEnvironment1.Command6_Grouping Text1.Text  
    DataReport6.Show
```

```

ElseIf (Option2.Value = True) Then
    If (DataEnvironment1.Connection1.State) Then
        DataEnvironment1.Connection1.Close
    End If
    DataEnvironment1.Connection1.Open
    DataEnvironment1.Command5 Text1.Text
    DataReport5.Show
ElseIf (Option3.Value = True) Then
    If (DataEnvironment1.Connection1.State) Then
        DataEnvironment1.Connection1.Close
    End If
    DataEnvironment1.Connection1.Open
    DataEnvironment1.Command4 Text1.Text
    DataReport4.Show
ElseIf (Option4.Value = True) Then
    If (DataEnvironment2.Connection1.State) Then
        DataEnvironment2.Connection1.Close
    End If
    DataEnvironment2.Connection1.Open
    DataEnvironment2.Command2_Grouping
    DataReport9.Show
End If
End Sub
Private Sub Form_Load()
    abc
    With Combo2
        .AddItem "3-1"
        .AddItem "3-2"
    End With
End Sub

```

```

        .AddItem "3-3"
        .AddItem "3-4"
End With
sql = "select distinct year from prjdtl order by year"
Set r = c.Execute(sql)
Combo1.Clear
While r.EOF = False
    Combo1.AddItem r.Fields(0)
    r.MoveNext
Wend
Combo1.Visible = False
Combo2.Visible = False
Text2.Visible = False
Text3.Visible = False
Text2.Locked = True
Text3.Locked = True
End Sub

```

```

Private Sub Option1_Click()
Text1.Text = ""
Text1.SetFocus
End Sub

```

```

Private Sub Option2_Click()
Text1.Text = ""
Text1.SetFocus
Combo1.Visible = True
Text2.Visible = True

```

End Sub

Private Sub Option3_Click()

Text1.Text = ""

Text1.SetFocus

Combo2.Visible = True

Text3.Visible = True

End Sub

Private Sub Text1_Change()

If Option1.Value = True Then

Adodc1.RecordSource = "SELECT DISTINCT
P.BATCH,P.YEAR,P.PRJID,P.PRJTITLE,P.COMPNM,M.ROLLNO,M.NM,
M.REGNO,M.PHNNO FROM PRJDTL P, MEMDTL M WHERE P.PRJID
= M.PRJID AND P.PRJID LIKE '" + Text1.Text + "%'"

Adodc1.Refresh

End If

If Option2.Value = True Then

Adodc1.RecordSource = "SELECT DISTINCT
P.BATCH,P.YEAR,P.PRJID,P.PRJTITLE,P.COMPNM,M.ROLLNO,M.NM,
M.REGNO,M.PHNNO FROM PRJDTL P, MEMDTL M WHERE P.PRJID
= M.PRJID AND YEAR LIKE'" + Text1.Text + "%'"

Adodc1.Refresh

End If

If Option3.Value = True Then

Adodc1.RecordSource = "SELECT DISTINCT
P.BATCH,P.YEAR,P.PRJID,P.PRJTITLE,P.COMPNM,M.ROLLNO,M.NM,
M.REGNO,M.PHNNO FROM PRJDTL P, MEMDTL M WHERE P.PRJID
= M.PRJID AND batch LIKE'" + Text1.Text + "%'"

Adodc1.Refresh

End If

End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)

KeyAscii = Asc(UCase(Chr(KeyAscii)))

End Sub

GUIDE_REPORTFRM:

Private Sub Command1_Click()

If (Option1.Value = True) Then

 If (DataEnvironment1.Connection1.State) Then

 DataEnvironment1.Connection1.Close

 End If

 DataEnvironment1.Connection1.Open

 DataEnvironment1.Command8 Text1.Text

 DataReport7.Show

ElseIf (Option2.Value = True) Then

 If (DataEnvironment2.Connection1.State) Then

 DataEnvironment2.Connection1.Close

 End If

 DataEnvironment2.Connection1.Open

 DataEnvironment2.Command3

 DataReport10.Show

End If

End Sub

```
Private Sub Option1_Click()  
Text1.Text = ""  
Text1.SetFocus  
End Sub
```

```
Private Sub Option2_Click()  
Command1.SetFocus  
End Sub
```

```
Private Sub Text1_Change()  
If Option1.Value = True Then  
    Adodc1.RecordSource = "select * from guide where code like" +  
Text1.Text + "%"  
    Adodc1.Refresh  
End If  
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)  
KeyAscii = Asc(UCase(Chr(KeyAscii)))  
End Sub
```

MARKING_REPORTFRM:

```
Private Sub Combo1_Click()  
Text2.Text = Combo1.Text  
Text1.Text = Combo1.Text  
End Sub
```

```
Private Sub Combo2_Click()
```

Text2.Text = Combo1.Text

Text1.Text = Combo2.Text

End Sub

Private Sub Command1_Click()

If (Option1.Value = True) Then

If (DataEnvironment1.Connection1.State) Then

DataEnvironment1.Connection1.Close

End If

DataEnvironment1.Connection1.Open

DataEnvironment1.Command11_Grouping.Text1.Text

sql = "SELECT LBL, GRADE, SBMDT FROM DOCMARK WHERE
PRJID=" & Text1.Text & " ORDER BY SBMDT"

Set r = c.Execute(sql)

MsgBox DataEnvironment1.rsCommand11_Grouping.Fields("PRJID")

DataReport13.Sections("pageHeader").Controls("LBLPRJID").Caption =
DataEnvironment1.rsCommand11_Grouping.Fields("PRJID")

DataReport13.Sections("pageHeader").Controls("LBLPRJTTL").Caption
= DataEnvironment1.rsCommand11_Grouping.Fields("PRJTITLE")

DataReport13.Sections("pageHeader").Controls("LBLPRJCOM").Caption
= DataEnvironment1.rsCommand11_Grouping.Fields("COMPNM")

If (r.EOF = False) Then

DataReport13.Sections("pageHeader").Controls("SYNOPSYS").Caption =
r.Fields(0)

DataReport13.Sections("pageHeader").Controls("LBLSYNOPSYS").Caption
= r.Fields(1)

End If

DataReport13.Show

ElseIf (Option2.Value = True) Then


```
If (DataEnvironment1.Connection1.State) Then
DataEnvironment1.Connection1.Close
End If
DataEnvironment1.Connection1.Open
DataEnvironment1.Command12_Grouping Text1.Text
DataReport14.Show
```

```
ElseIf (Option3.Value = True) Then
If (DataEnvironment1.Connection1.State) Then
DataEnvironment1.Connection1.Close
End If
DataEnvironment1.Connection1.Open
DataEnvironment1.Command13_Grouping Text1.Text
DataReport15.Show
```

```
ElseIf (Option4.Value = True) Then
If (DataEnvironment2.Connection1.State) Then
DataEnvironment2.Connection1.Close
End If
DataEnvironment2.Connection1.Open
DataEnvironment2.Command4_Grouping
DataReport16.Show
End If
End Sub
```

```
Private Sub Form_Load()
abc
With Combo2
```

```

.AddItem "3-1"
.AddItem "3-2"
.AddItem "3-3"
.AddItem "3-4"
End With
sql = "select distinct year from prjdtl order by year"
Set r = c.Execute(sql)
Combo1.Clear
While r.EOF = False
    Combo1.AddItem r.Fields(0)
    r.MoveNext
Wend
Combo1.Visible = False
Combo2.Visible = False
Text2.Visible = False
Text3.Visible = False
Text2.Locked = True
Text3.Locked = True
End Sub

```

```

Private Sub Option1_Click()
Text1.Text = ""
Text1.SetFocus
End Sub

```

```

Private Sub Option2_Click()
Text1.Text = ""
Text1.SetFocus

```

```
Combo1.Visible = True
```

```
Text2.Visible = True
```

```
End Sub
```

```
Private Sub Option3_Click()
```

```
Text1.Text = ""
```

```
Text1.SetFocus
```

```
Combo2.Visible = True
```

```
Text3.Visible = True
```

```
End Sub
```

```
Private Sub Option4_Click()
```

```
Text1.Text = ""
```

```
Text1.SetFocus
```

```
End Sub
```

```
Private Sub Text1_Change()
```

```
If Option1.Value = True Then
```

```
    Adodc1.RecordSource = " SELECT * FROM DOCMARK WHERE  
PRJID LIKE " + Text1.Text + "%"
```

```
    Adodc1.Refresh
```

```
End If
```

```
If Option2.Value = True Then
```

```
    Adodc1.RecordSource = " SELECT DISTINCT  
M.LVL,M.CLSROLL,M.NAME,M.REGNO,M.GRADE,P.PRJID,P.YEAR,P.  
BATCH,P.PRJTITLE,P.COMPNM,D.PRJID FROM DOCMARK D ,  
MEMMARK M , PRJDTL P WHERE D.PRJID = M.PRJID AND D.PRJID  
= P.PRJID AND YEAR LIKE" + Text1.Text + "%"
```

```
    Adodc1.Refresh
```

```
End If
```

If Option3.Value = True Then

```
Adodc1.RecordSource = " SELECT DISTINCT  
M.LVL,M.CLSROLL,M.NAME,M.REGNO,M.GRADE,P.PRJID,P.YEAR,P.  
BATCH,P.PRJTITLE,P.COMPNM,D.PRJID FROM DOCMARK D ,  
MEMMARK M , PRJDTL P WHERE D.PRJID = M.PRJID AND D.PRJID  
= P.PRJID AND BATCH LIKE '" + Text1.Text + "%"
```

```
Adodc1.Refresh
```

End If

End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)

```
KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

End Sub

CERTIFICATE INFOFRM:

Private Sub Form_Load()

```
Adodc1.RecordSource = "select PRJID ,PRJTITLE,COMPNM from  
PRJDTL"
```

```
Adodc1.Refresh
```

```
DataGrid2.Visible = False
```

End Sub

Private Sub Option1_Click()

```
DataGrid2.Visible = False
```

```
DataGrid1.Visible = True
```

If Option1.Value = True Then

```
Adodc1.RecordSource = "select PRJID,PRJTITLE,COMPNM from  
PRJDTL"
```

```
Adodc1.Refresh
```

```
End If
```

```
End Sub
```

```
Private Sub Option2_Click()
```

```
DataGrid1.Visible = False
```

```
DataGrid2.Visible = True
```

```
If Option2.Value = True Then
```

```
Adodc1.RecordSource = "SELECT * FROM CISSUED"
```

```
Adodc1.Refresh
```

```
End If
```

```
End Sub
```

Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specialization, design and code generation. Design and code generation. Once source code has been generated, s\w must be tested to uncover as many errors as possible before delivery.

- **Objectives :-**

1. Testing is a process of executing a program with the intent of finding an error.
2. A good test case is one that has a high probability of finding an uncovered error.
3. A successful test is one that uncovers an yet uncovered error.

- **Principles:-**

- All tests should be traceable to customer requirements.
- Tests should be planned long before testing begins.
- The pareto principles applies to software testing, It states that 80% of all errors uncovered during that is traceable to 20% of all program components.
- Testing should begin “In the small“ and progress toward testing “in the large”.
- Exhausting testing is not possible.
- To be most effective, testing should be conducted by an independent team.
- A good test has a high probability of finding an error.
- A good test is not redundant.
- A good test should be neither too simple nor too complex.

- **Testing Strategy Issues:-**

- specify product requirements in a quantifiable names long before testing commences.
- states testing objectives explicitly.
- Understand the users of the software and develop a profile for each user category.
- Develop a testing plan that emphasizes “rapid cycle testing.”
- Build “robust” software that is designed to test itself.
- Use effective technical reviews as a filter prior to testing.
- Conduct formal technical review to access the test strategy and test cases themselves.
- Develop a continuous improvement approach for the testing process.
- **Black Box Testing :-**

This test is conducted to test software interface i.e. input and output. It is used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced. It is also called behavioral testing which focus on the functional requirements of the software. Some sets of input are derived and exercised will all functional requirement for program. It is used to find errors like incorrect function, interface error, error in data structure, external data error, performance error, initialization and termination error. Black box testing is later stage of testing. It covers following areas :-

- How is functional Validity tested ?
- How is system behavior and performance tested?
- What classes of input will make good test cases?
- It the system particularly sensitive to certain input values?
- How are the boundaries of a data class?
- What data rates and data volume can the system tolerate?

- what effect will specific combinations of data have on system operations ?

- **White Box Testing :-**

It is closed examination of procedure details. It is also called “glass box testing”. It is use to derive test case that uses to control structure of the procedural design. It guarantee that all independent paths within a module have been exercised at least once. It exercises all logical decisions on their true and false sides. It executes all loops at their boundaries and also tests internal data structures to ensure their validity. It assumes that

- logic errors and incorrect assumptions are inversely proportional to the probability that a program path will be executed.
- We often believes that a logical path is not likely to be executed when it may be executed on a regular basis.
- Typographical errors (typing) are random while box testing is also used to test “ Basic path “ of program, which is represented as Flow Graph.

- **Verification and Validation :-**

- Verification refer to the set of activities that ensure that software correctly implements a specific function. It is performed by developes to test “ Are we building the product right” ? During development end.

Verification refers to a different Set activities that ensures that the software that has been built is traceable to customer required. It is done by developer and customer both to test “Are we building the right product ? ”

Both test activity are software quality Assurance (SQL) which includes formal technical review, quality and configuration audits, performance monitoring, simulation, feasibility study, documentation review, database review, algorithm analysis, development testing, qualification testing etc.

- **Unit Testing :-**

Unit testing is White Box testing which focus on verification effort on the smallest unit of software – module or program. The module interface is tested to ensure that information properly flows in and out of the program. The local data structure is examined in an algorithm's execution. Boundary conditions are also tested to ensure that all module operates properly. All independent path is also tested.

common error in computations are :-

- Incorrect arithmetic precedence.
 - Mixed mode operations.
 - Incorrect initialization.
 - precision inaccuracy.
 - Incorrect symbolic representation of an expression,
- **Unit test cases uncovered errors like –**
 - comparison of different data types.
 - Incorrect logical operator or precedence.
 - Expectation of equality.
 - Incorrect comparison of variables.
 - Improper loop termination.
 - Failure to exit.

Unit test is performed during coding step to test each unit individually. Unit testing is simplified when a component with high cohesion is designed. when only one function is performed by one component.

- **Integration Testing :-**

A software is collection of programs or module. Each module is individually tested using unit testing, But an integration or combination it generates interface error, global data structure

problem etc. It is a systematic technique for constructing the program structure while at the same time conducting tests to in cover interface error. It is black box.

Integration testing strategy are :-

- Top down Integration.
- Bottom- up Integration.
- Regression Testing.

• **TOP DOWN INTEGRATION :-**

It is an incremental approach to construct a program structure which starts with highest level of module and procedure to the lowest level of module.

Top down integration is performed in a series of five step:-

- a) The Main control module is used as a test driver.
- b) Depending on the integration approach subordinate models are integrated
- c) Test are conducted as each component is integrated.
- d) On completion of each set of test it is replaced with real module.
- e) Regression testing.

For example integration testing is first performed with m1, m2, m3 and m4, then m3 and m7 and last m4.

• **BOTTOM-UP INTEGRATION :-**

It starts with automatic level and proceed to higher level.

STEPS –

- Low level components are combined into cluster.

- 2) A drivers (a control program for testing) is written to coordinate test is written to co-ordinate test.
- 3) The cluster is tested .
- 4) Higher level is tested

For example m10 + m11 are integrated and tested. After that m5, m6 and m7 are integrated as m2 and tested. m8 and m9 are integrated as m4 and tested finally m2, m3 and m4 are integrated as m1 and tested.

- **REGRESSION TESTING :-**

Regression testing is an important strategy for reducing “side effects”.

Each time a new module is added as part of integration testing, the software changes. New data flow paths are established, new i/o may occurs and new control logic is invoked.

Regression testing is the re-execution of some of test that have been already been conducted to ensure that changes have not propagated side effects. It may be conducted manually.

- **ALPHA AND BETA TESTING :-**

When software is built for specific customer, a series of acceptance tests are conducted to enable the customer to validate all requirements. It is conducted by the end user not by the developers.

Acceptance testing can be conducted over a period of week or months. During this period user collects all uncovered errors which are remover by the developer.

Alpha and Beta testing is acceptance testing. The alpha test is conducted at the developer site by a customer. It is conducted in a controlled environments.

The beta test is conducted at one or more customer sites by the end user of the software. During beta testing developer are not present so it is live application of software. The customer records all problems and reports to developer regularly which is short out.

- **SYSTEM TESTING :-**

Software is a complex system which is integrated of multiple modules. Before delivery of software it must be tested according to s/w requirements specification.

It is actually a series of different tests whose primary purpose is to fully exercise the computer based system. software elements are hardware people and information.

- **LEVEL OF SYSTEM TESTING :-**

1) Recovery testing :- software must be recover from faults and resumes processing within specific time. Recovery testing is a system testing that forces the software to fail and verify the recovery.

2) Security testing :- security testing attempts to verify that protection mechanism built into a system will protect it from unauthorized access.

3) Stress testing :- It execute a system in a manner that demands resources in abnormal quality, frequency or volume. It is also called sensitivity testing.

4) performance testing :- It is designed to test the run time performance of software within the context of an integrated system. Performance tests is applied throughout all steps in the testing specially with stress testing it usually requires both hardware and software.

SYSTEM SECURITY MEASURES

Security as a part of software development process is an ongoing process involving people and practices and insured application confidentiality, integrity and availability secure software is the result of security aware software development process where security is built in thus software is developed with security in mind. Security is most effective if planned and managed throughout every stage of software development life cycle (sdcl) specifically in critical applications are those that process sensitive information effective security measures every business should implement today's business environment has changed a lot we hear about hackers stealing valuable company data all the time and holding it for ransom for pawning it on the dark web to the highest bidder this information might be in the form of messages personal data credit card information and other form of sensitive material from ECommerce site. This is a serious breach of a security and could result in severe losses and loss of confidence from your customers which any business would mean the loss of income.

However there are some precautions of business can take to make themselves safer and less prone to attack.

Taking care of your home and resources :

many people in today's workforce are not aware of the many dangers that lurk on the internet. During office many offers check our mail on company computers computers are linked to the whole network infrastructure of the enterprise as such hackers have become clever and devising new ways to penetrate the system without using too much effort. In this regard you find that malware is packaged in a way that seem less than harmful in the form of pictures documents and so forth embedded within this attachments is malware viruses ransomware among others by a single click employee can leave their whole business infrastructure open to attack to prevent all of this a security firm from outside should be contracted to test the system via bridge exercises and also leverage social engineering tactics to test for weak point in the system this so where the weakness and help in the patching process in employee training.

select for a secure eCommerce Platform:-

since every business needs an e-commerce platform it's important to make that platform as safe as possible. One of the key ways to ensure this happens is by using a platform that uses complex object-oriented programming language. This type of platform is more secure than open-source type due to the complexity of the code it is easier to keep things on a tight leash. The admin panel should also be designed in such a way that it's only available on the internal network and inaccessible from outside. There should be a separation between the internal network and all the public-facing servers.

Offsite Data Storage:-

As your business continues to grow malicious persons are looking to take advantage and steal sensitive information from your system they want to get access to your financial records email customer data or even employee identification data all of these data is toxic in the hands of intruders as they will sell it to your competition or hold it over your heads for payment of sensitive data storage solve this problem for the most sensitive data should be stored outside your business where hackers won't find it apart from making sure that the outside location that the data is sent to is secured a company should also be concerned with how they are sending this data to the off-site facility. When data is being sent over the internet it should be fully-encrypted so that even if it is intercepted in transit it will be useless to the thief. Virtual private networks VPN can also be used when accessing what matters outside the home network.

Use SSL Encryption for your Website:- Getting an SSL certificate for your site goes away in information security for the business especially when it comes to an eCommerce site where people buy goods and share sensitive data such as credit card information as `http` in a must to ensure that your financial transactions are secure one must implement SSL encryption for business and extended validation SSL certificate is appropriate EV SSL make sure that the website does belong to the organisation in question this not only gives your peace of mind that client activities and transactions are secured but it also shows that the business is legitimate and it can be trusted.

Use Strong Password

It is responsibility of the company to keep customer data out of the wrong hands. However, the retailer has a responsibility to the customer to help them made of practices that make it and for an authorised person to access their account and easy way to do this it demands that users make password as secure as possible you can implement this by making sure the password have a minimum number of character and they incorporate numbers and special characters having a longer more complex password will deter criminals from breaching the website via the front end other producers that fall into this category include locking work station when they are not in use encrypting data and storing sensitive data on a secured server.

Don't keep Sensitive

Customer Data Transactions are being processed many stores ask for credit card information for payment purpose. However, this information should not be stored by the business for longer than required later such as credit card numbers that verification value cvv2 and their expiration dates not be kept for longer than necessary according to a standard set forth by the PCI payment card industry security standards council old records about such client data should be removed completely from company database. The only information the company might keep should just be for charge back and refund the danger of keeping such record for out weight the convenience customer enjoy during the checkout process.

Use web application firewalls While SSL encryption secure data transmitted to and from the website the web application firewalls will protect the site from Denial of service attacks brute force attacks cross site scripting among others. This type of protection is inexpensive and does not need a lot of time or expertise to configure.

Printer protection

Printers are some of the most common business device today and people don't usually considered the effect that have a security the problem is that many business are aware of the information that they stored in their limited on board memory. Scanned and printed document may be saved on them which means that pulling

information of a poorly secured printer is relatively easy for a smart cybercriminals to keep your printer secure disable things like printer sharing and property forward connections to your network rather than wireless connectivity you may want to OPD for printers that automatically purge there on board memory to keep your data that much more secure.

Surveillance

knowing who is on and around your premises is key to returning thieves. The best way to conduct this is through the use of surveillance system. Implementing a surveillance system is a smart move because of how cost efficient and compact some system can be imagine that extends the functionality of a log book to cover everyone with a record of their face their clothing and plethora of other qualities that can be used later for identification purpose and treat assessment.

Making your Business Secured

While the outline security measures above will be lost or your business security they are by no means the limit to what you should implement things like better encryption stricter security protocols and Indian physical and digital access to your business as you continue to grow and acquire more customers and employees good security practices are key to your ongoing success.

Future scope

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

SOFTWARE SCOPE

Extensibility: This software is extendable in ways that its original developers may not expect. The following principles enhances 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: Sharing of newly written code within a project and reuse of previously written code on new projects.

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.

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.

BIBLIOGRAPHY

BOOKS REFERRED

BOOKS NAME

- **Mastering Visual Basic 6.0**
- **Guide to Visual Basic 6.0**
- **Oracle 8i Ivan**
- **System Analysis & Design**
- **software engineering**

AUTHOR NAME

Evengauls ptrolus
Peter Nortan & Michael
boyross
Eliaas M.Award
pressman

Conclusion

While developing the system a conscious effort has been made to create and develop a software package, making use of available tools, technique resources-that would generate a proper system.

While making the system, an eye has been kept on making it is 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 processor where there are number of shortcomings, there have been some shortcomings in the development of this system also. The project is still under modification.

