QRCODE & BARCODE

in Visual Basic 2010 with SSMS

CREATE YOUR DATABASE

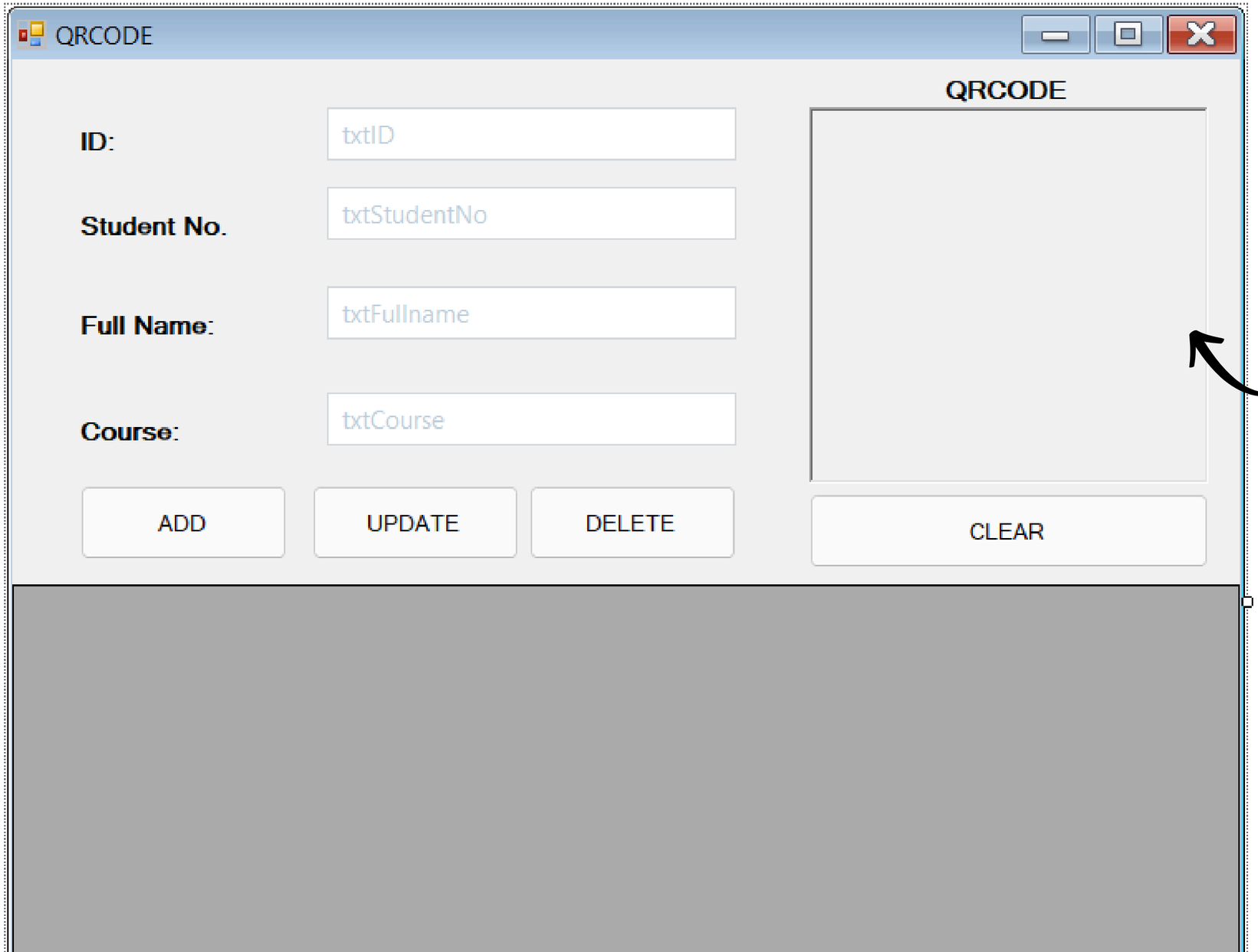
- Go ahead and create your database
- Now create your table and with columns:

```
CREATE TABLE studentinfo(
    id int IDENTITY(1,1) PRIMARY KEY,
    studentno varchar(MAX),
    fullname varchar(MAX),
    course varchar(MAX),
    qr image
);
```

USER INTERFACE

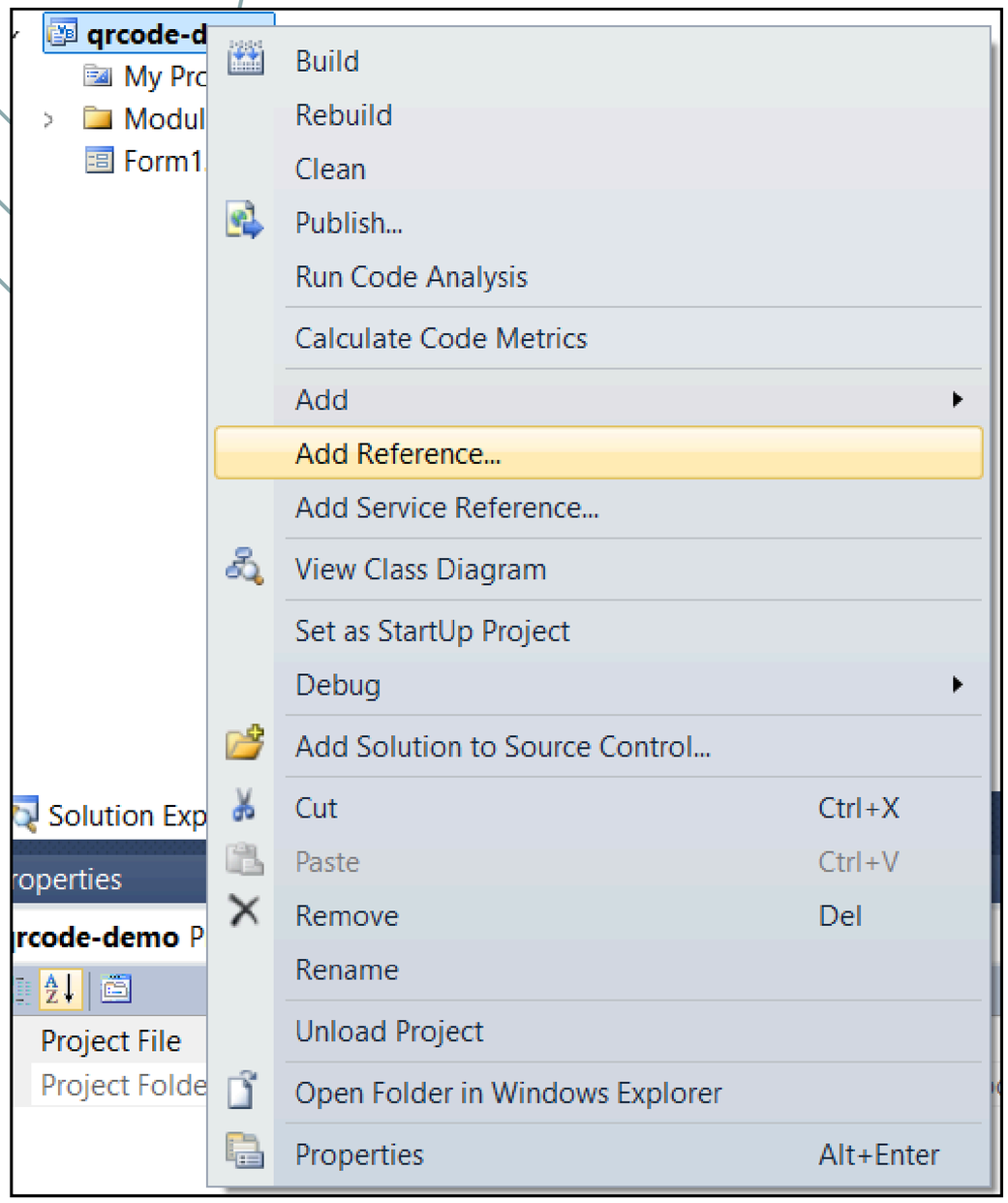
QRCODE

USER INTERFACE

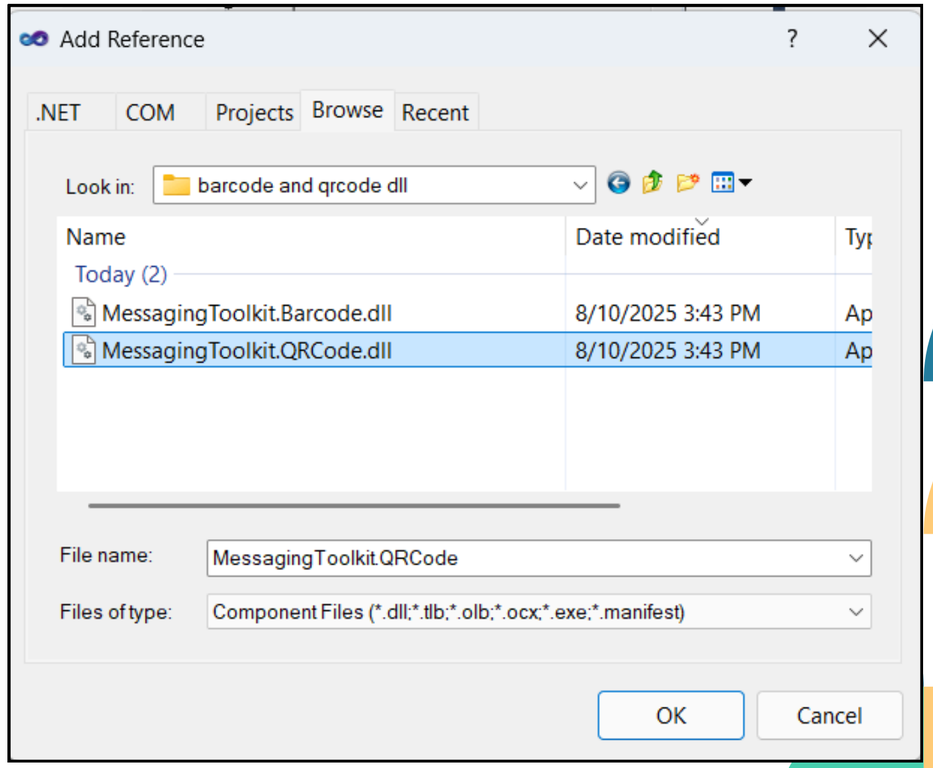


IMPORTING MESSAGING TOOLKIT

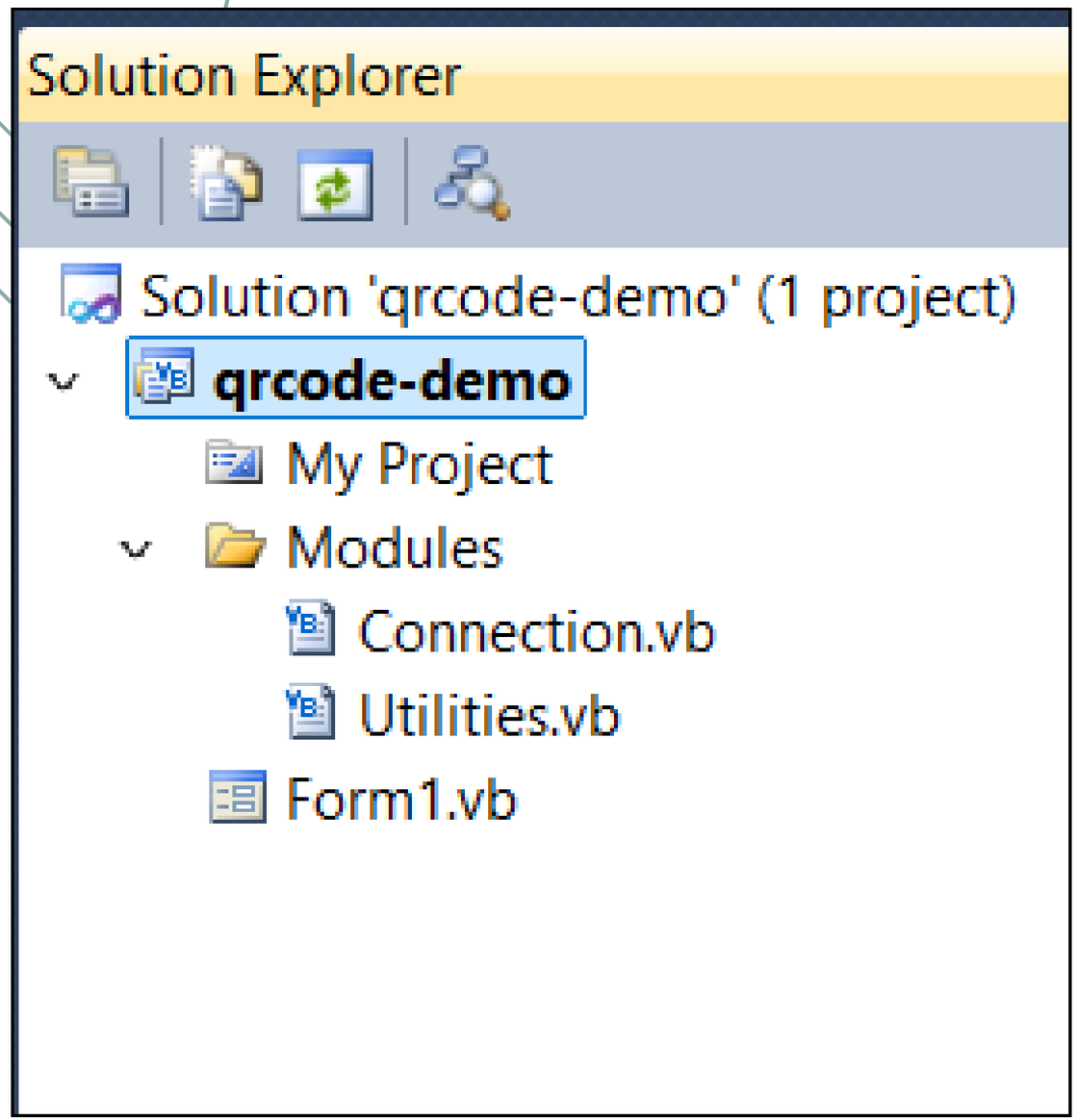
QRCODE



- After creating your Database, Go ahead and create new project
- **Add Reference > Browse** and then locate your **'.dll'** file



QRCODE



- After importing Toolkit, Add Modules on your project:
  - Connection.vb
  - Utilities.vb
- Restructure your project files by creating folders for better readabilty (Optional)

CODING

QRCODE

MODULE: CONNECTION

```
Imports System.Data.SqlClient

Module Connection

    Public connString = "Server=LAPTOP-1CLIFQED\SQLEXPRESS; Database=DBQR; Trusted_Connection=True;"
    Public Connect As New SqlConnection(connString)
    Public Parameters As New List(Of SqlParameter)
    Public Data As DataSet
    Public Datacount As Integer

    Public Sub Open()
        If Connect.State = ConnectionState.Closed Then
            Connect.Open()
        End If
    End Sub

    Public Sub Close()
        If Connect.State = ConnectionState.Open Then
            Connect.Close()
        End If
    End Sub

    Public Sub AddParam(ByVal Key As String, ByVal value As String)
        Parameters.Add(New SqlParameter(Key, value))
    End Sub
```

MODULE: CONNECTION

```
Public Function Insert(ByVal insertQuery As String) As Boolean
    Try
        Open()
        Dim command As New SqlCommand(insertQuery, Connect)

        If Parameters.Count > 0 Then
            For Each param As SqlParameter In Parameters
                command.Parameters.Add(param)
            Next
            Parameters.Clear()
        End If

        Datacount = command.ExecuteNonQuery()
        Return Datacount > 0

    Catch ex As Exception
        MsgBox(ex.Message)
        Return False
    Finally
        Close()
    End Try
End Function
```

INSERT function

```
Public Function Update(ByVal updateQuery As String) As Boolean
    Try
        Open()
        Dim command As New SqlCommand(updateQuery, Connect)
        If Parameters.Count > 0 Then
            For Each param As SqlParameter In Parameters
                command.Parameters.Add(param)
            Next
            Parameters.Clear()
        End If
        Datacount = command.ExecuteNonQuery()
        Return Datacount > 0
    Catch ex As Exception
        MsgBox(ex.Message)
        Return False
    Finally
        Close()
    End Try
End Function
```

UPDATE function

```
Public Function Delete(ByVal deleteQuery As String) As Boolean
    Try
        Open()
        Dim command As New SqlCommand(deleteQuery, Connect)
        If Parameters.Count > 0 Then
            For Each param As SqlParameter In Parameters
                command.Parameters.Add(param)
            Next
            Parameters.Clear()
        End If
        Datacount = command.ExecuteNonQuery()
        Return Datacount > 0
    Catch ex As Exception
        MsgBox(ex.Message)
        Return False
    Finally
        Close()
    End Try
End Function
```

DELETE function

MODULE: CONNECTION

MODULE: UTILITIES

- We can now start coding on Module **Utilities**
- This is where we can store our Queries

```
Imports System.Data.SqlClient
Imports System.Drawing
Imports System.IO


Module Utilities
```

MODULE: UTILITIES

```vb
Public Function InsertStudent(ByVal studentno As String, ByVal fullname As String, ByVal course As String, ByVal qrImage As Byte()) As Boolean
    Try
        Dim query = "INSERT INTO studentinfo ([studentno],[fullname],[course],[qr]) VALUES(@studentno,@fullname,@course,@qr)"
        connection.AddParam("@studentno", studentno)
        connection.AddParam("@fullname", fullname)
        connection.AddParam("@course", course)
        connection.Parameters.Add(New SqlParameter("@qr", qrImage))

        If connection.Insert(query) Then
            MsgBox("Added Successfully", MsgBoxStyle.Information, "Success")
            Return True
        End If
    Catch ex As Exception
        MsgBox("Error adding student: " & ex.Message)
    End Try
    Return False
End Function
```

INSERT Query

```vb
Public Function UpdateStudent(ByVal id As String, ByVal studentno As String, ByVal fullname As String, ByVal course As String, ByVal qrImage As Byte()) As Boolean
    Try
        Dim query = "UPDATE studentinfo SET studentno=@studentno, fullname=@fullname, course=@course, qr=@qr WHERE id=@id"
        connection.AddParam("@id", id)
        connection.AddParam("@studentno", studentno)
        connection.AddParam("@fullname", fullname)
        connection.AddParam("@course", course)
        connection.Parameters.Add(New SqlParameter("@qr", qrImage))

        If connection.Update(query) Then
            MsgBox("Updated Successfully", MsgBoxStyle.Information, "Success")
            Return True
        End If
    Catch ex As Exception
        MsgBox("Error updating student: " & ex.Message)
    End Try
    Return False
End Function
```

UPDATE Query

MODULE: UTILITIES

```
Public Function DeleteStudent(ByVal id As String) As Boolean
    Try
        Dim query = "DELETE FROM studentinfo WHERE id=@id"
        connection.AddParam("@id", id)

        If connection.Delete(query) Then
            MsgBox("Deleted Successfully", MsgBoxStyle.Information, "Success")
            Return True
        End If
    Catch ex As Exception
        MsgBox("Error deleting student: " & ex.Message)
    End Try
    Return False
End Function
```

DELETE Query

```
Public Sub SaveQR(ByVal qrImage As Image)
    If qrImage Is Nothing Then Return

    Dim saveDialog As New SaveFileDialog()
    saveDialog.Filter = "bmp (*.bmp)|*.bmp|jpeg (*.jpeg)|*.jpeg|png (*.png)|*.png|tiff (*.tiff)|*.tiff"
    If saveDialog.ShowDialog() = DialogResult.OK Then
        qrImage.Save(saveDialog.FileName)
    End If
End Sub
```

FORM 1

Now that we are done coding on both modules, we can now start coding on **Form1**

```
Imports System.Data.SqlClient

Public Class Form1

    Dim QR_Generator As New MessagingToolkit.QRCode.Codec.QRCodeEncoder
    Dim arrImage() As Byte

    Sub viewdata()
        Dim con1 As New SqlConnection("Server=LAPTOP-1CLIFQED\SQLEXPRESS; Database=DBQR; Trusted_Connection=True;")
        Dim sql As String = "select * from studentinfo"
        Dim Adapter As New SqlDataAdapter(sql, con1)
        Dim data As New DataTable("studentinfo")
        Adapter.Fill(data)
        dgvInfo.DataSource = data
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        viewdata()
    End Sub
```

```
Private Sub txtStudentNo_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles t
    If txtStudentNo.Text = "" Or txtFullname.Text = "" Or txtCourse.Text = "" Then
        picQR.BackgroundImage = Nothing
    Else
        Dim qrContent As String = txtStudentNo.Text & vbCrLf & txtFullname.Text & vbCrLf & txtCourse.Text
        picQR.BackgroundImage = QR_Generator.Encode(qrContent)
    End If
End Sub
```

*Note: Get the **TextChanged** event of textbox for **StudentNo, Fullname & Course** before proceeding in this code.*

```
Sub clickondgv()
    Dim i As Integer
    i = dgvInfo.CurrentRow.Index

    txtID.Text = Convert.ToString(dgvInfo.Item(0, i).Value)
    txtProdID.Text = Convert.ToString(dgvInfo.Item(1, i).Value)
    txtName.Text = Convert.ToString(dgvInfo.Item(2, i).Value)
    txtQuantity.Text = Convert.ToString(dgvInfo.Item(3, i).Value)
    arrImage = TryCast(dgvInfo.Item(4, i).Value, Byte())
    If arrImage IsNot Nothing Then
        Dim mstream As New System.IO.MemoryStream(arrImage)
        picBarcode.BackgroundImage = Image.FromStream(mstream)
    Else
        picBarcode.BackgroundImage = Nothing
    End If
End Sub

Private Sub dgvInfo_CellClick(ByVal sender As System.Object, ByVal
    clickondgv()
End Sub

Private Sub Clear()
    txtID.Clear()
    txtProdID.Clear()
    txtName.Clear()
    txtQuantity.Clear()
    picBarcode.Image = Nothing
    picBarcode.BackgroundImage = Nothing
    picBarcode.BackColor = Color.Empty
    picBarcode.Invalidate()
End Sub
```

```
Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAdd.Cli
    Try
        Dim mstream As New System.IO.MemoryStream()
        picQR.BackgroundImage.Save(mstream, System.Drawing.Imaging.ImageFormat.Jpeg)
        arrImage = mstream.GetBuffer()
        Dim FileSize As UInt32
        FileSize = mstream.Length
        mstream.Close()

        If Utilities.InsertStudent(txtStudentNo.Text, txtFullname.Text, txtCourse.Text, arrImage) Then
            Utilities.SaveQR(picQR.BackgroundImage)
            Clear()
            viewdata()
        End If
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
```

FORM 1

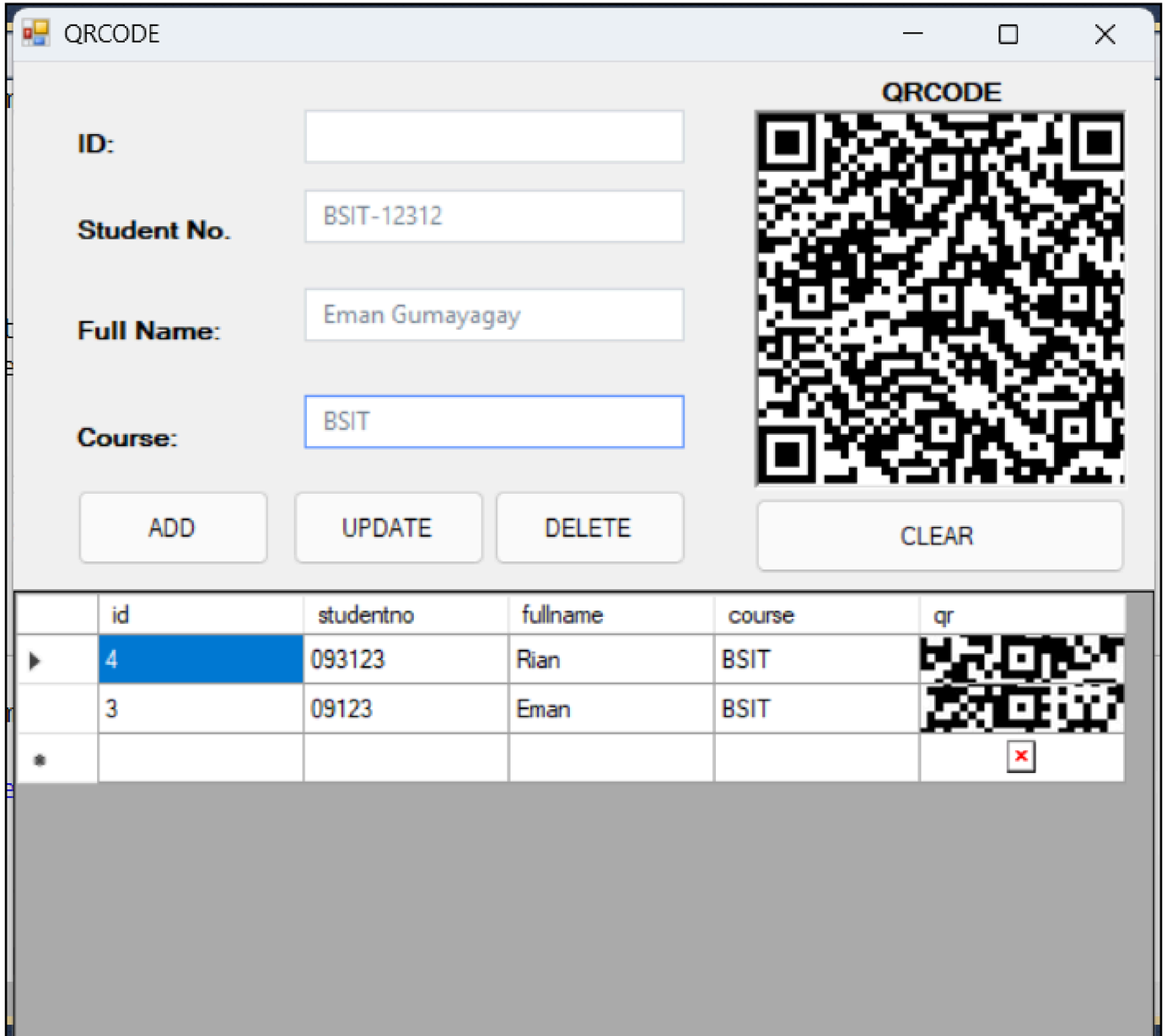
```
Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnUpdate.Click
    Try
        Dim mstream As New System.IO.MemoryStream()
        picQR.BackgroundImage.Save(mstream, System.Drawing.Imaging.ImageFormat.Jpeg)
        arrImage = mstream.GetBuffer()
        Dim FileSize As UInt32
        FileSize = mstream.Length
        mstream.Close()

        If Utilities.UpdateStudent(txtID.Text, txtStudentNo.Text, txtFullname.Text, txtCourse.Text, arrImage) Then
            Utilities.SaveQR(picQR.BackgroundImage)
            Clear()
            viewdata()
        End If
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
```

```
Private Sub btnDelete_Click(ByVal sender As System.Ob
    Try
        If Utilities.DeleteStudent(txtID.Text) Then
            Clear()
            viewdata()
        End If
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
```

```
Private Sub btnClear_Click(ByVal sender As System.Obj
    Clear()
End Sub
```

RESULT


QRCODE
ID:
Student No.
BSIT-12312
Full Name:
Eman Gumayagay
Course:
BSIT
ADD
UPDATE
DELETE
QRCODE
CLEAR
id
studentno
fullname
course
qr
4
093123
Rian
BSIT
3
09123
Eman
BSIT

BARCODE

CREATE YOUR DATABASE

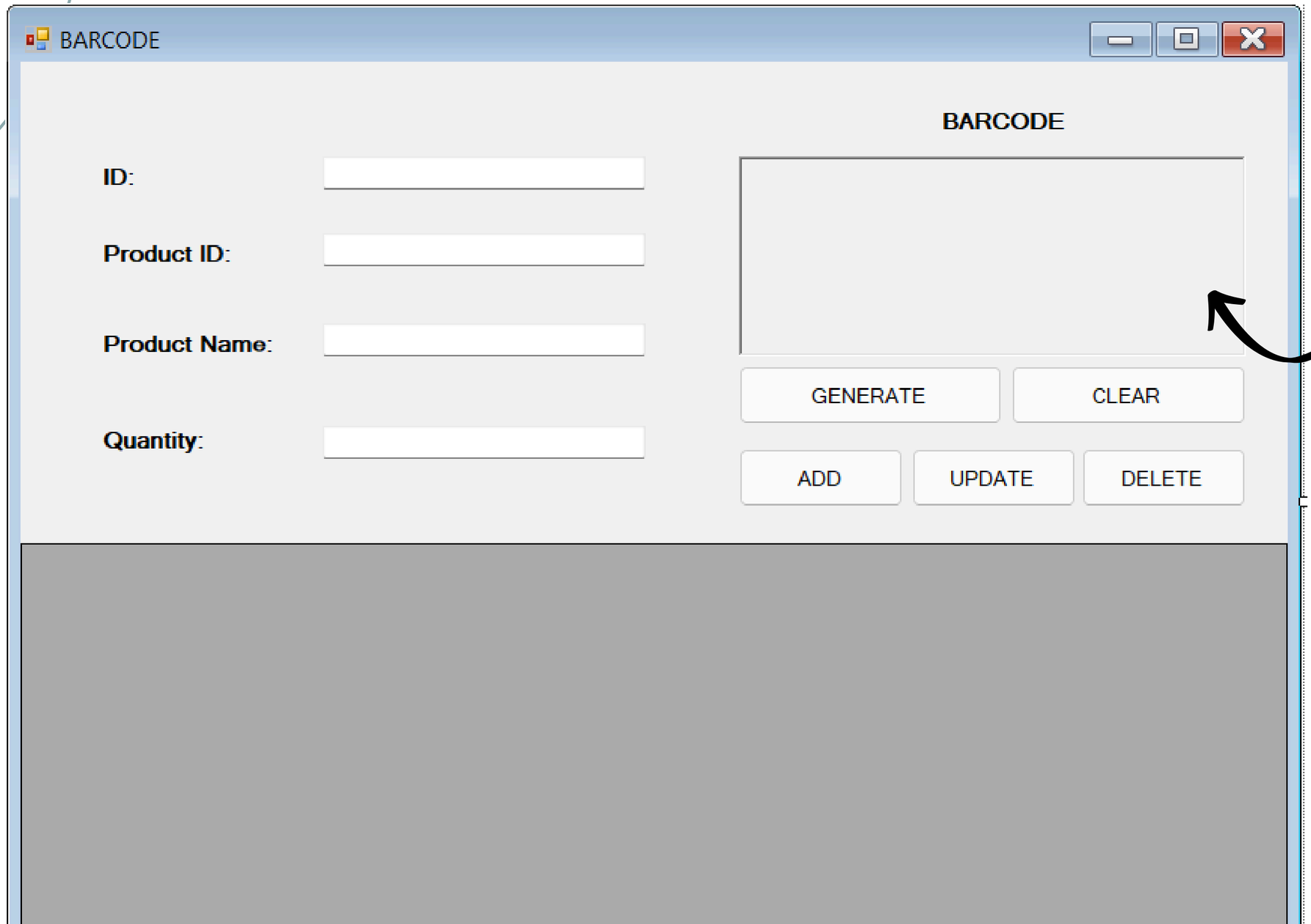
- For Barcode, we will use the same database but different Table and columns.

```
CREATE TABLE productinfo(
    id int IDENTITY(1,1) PRIMARY KEY,
    productid varchar(MAX),
    productname varchar(MAX),
    quantity varchar(MAX),
    barcode image
);
```

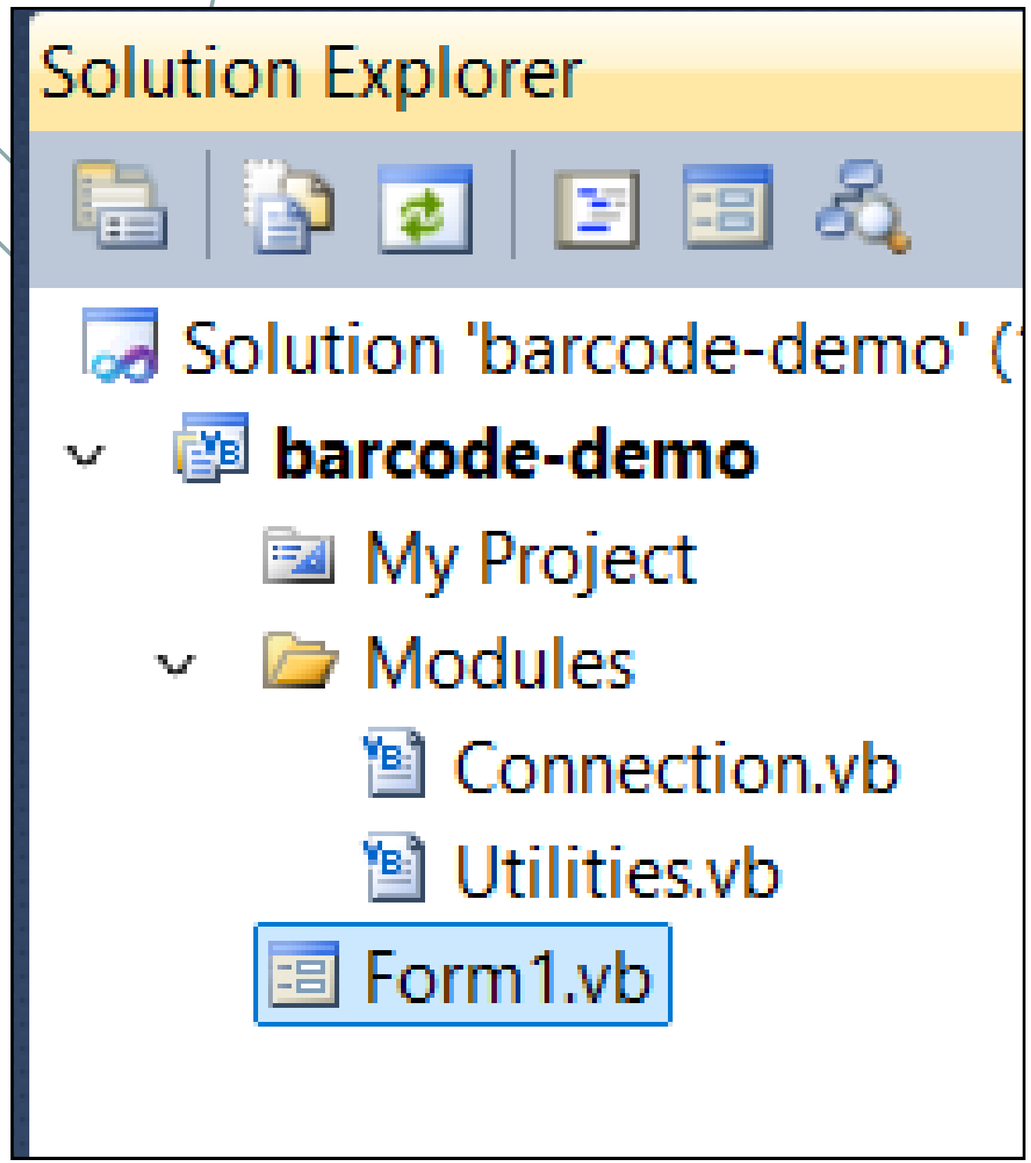
USER INTERFACE

BARCODE

USER INTERFACE



BARCODE



- For Barcode, We will still use the same flow and logic on the Modules.
- As we go, you'll notice that there's a similarity on the two project in terms of coding.

CODING

BARCODE

MODULE: CONNECTION

```
Imports System.Data.SqlClient

Module Connection

    Public connString = "Server=LAPTOP-1CLIFQED\SQLEXPRESS; Database=DBQR; Trusted_Connection=True;"
    Public Connect As New SqlConnection(connString)
    Public Parameters As New List(Of SqlParameter)
    Public Data As DataSet
    Public Datacount As Integer

    Public Sub Open()
        If Connect.State = ConnectionState.Closed Then
            Connect.Open()
        End If
    End Sub

    Public Sub Close()
        If Connect.State = ConnectionState.Open Then
            Connect.Close()
        End If
    End Sub

    Public Sub AddParam(ByVal Key As String, ByVal value As String)
        Parameters.Add(New SqlParameter(Key, value))
    End Sub
```

MODULE: CONNECTION

```
Public Function Insert(ByVal insertQuery As String) As Boolean
    Try
        Open()
        Dim command As New SqlCommand(insertQuery, Connect)

        If Parameters.Count > 0 Then
            For Each param As SqlParameter In Parameters
                command.Parameters.Add(param)
            Next
            Parameters.Clear()
        End If

        Datacount = command.ExecuteNonQuery()
        Return Datacount > 0

    Catch ex As Exception
        MsgBox(ex.Message)
        Return False
    Finally
        Close()
    End Try
End Function
```

INSERT function

```
Public Function Update(ByVal updateQuery As String) As Boolean
    Try
        Open()
        Dim command As New SqlCommand(updateQuery, Connect)
        If Parameters.Count > 0 Then
            For Each param As SqlParameter In Parameters
                command.Parameters.Add(param)
            Next
            Parameters.Clear()
        End If
        Datacount = command.ExecuteNonQuery()
        Return Datacount > 0
    Catch ex As Exception
        MsgBox(ex.Message)
        Return False
    Finally
        Close()
    End Try
End Function
```

UPDATE function

```
Public Function Delete(ByVal deleteQuery As String) As Boolean
    Try
        Open()
        Dim command As New SqlCommand(deleteQuery, Connect)
        If Parameters.Count > 0 Then
            For Each param As SqlParameter In Parameters
                command.Parameters.Add(param)
            Next
            Parameters.Clear()
        End If
        Datacount = command.ExecuteNonQuery()
        Return Datacount > 0
    Catch ex As Exception
        MsgBox(ex.Message)
        Return False
    Finally
        Close()
    End Try
End Function
```

DELETE function

MODULE: CONNECTION

MODULE: UTILITIES

Same as the **'Utilities'** Module for QRCode but since we created a new table, there will be a slight change.

```
Imports System.Data.SqlClient
Imports System.Drawing
Imports System.IO


Module Utilities
```

MODULE: UTILITIES

```
Public Function InsertProduct(ByVal productid As String, ByVal productname As String, ByVal quantity As String, ByVal barcode As Byte()) As Boolean
    Try
        Dim query = "INSERT INTO productinfo ([productid],[productname],[quantity],[barcode]) VALUES(@productid,@productname,@quantity,@barcode)"
        Connection.AddParam("@productid", productid)
        Connection.AddParam("@productname", productname)
        Connection.AddParam("@quantity", quantity)
        Connection.Parameters.Add(New SqlParameter("@barcode", barcode))

        If Connection.Insert(query) Then
            MsgBox("Added Successfully", MsgBoxStyle.Information, "Success")
            Return True
        End If
    Catch ex As Exception
        MsgBox("Error adding student: " & ex.Message)
    End Try
    Return False
End Function
```

INSERT Query

```
Public Function UpdateProduct(ByVal id As String, ByVal productid As String, ByVal productname As String, ByVal quantity As String, ByVal barcode As Byte()) As Boolean
    Try
        Dim query = "UPDATE productinfo SET productid=@productid, productname=@productname, quantity=@quantity, barcode=@barcode WHERE id=@id"
        Connection.AddParam("@id", id)
        Connection.AddParam("@productid", productid)
        Connection.AddParam("@productname", productname)
        Connection.AddParam("@quantity", quantity)
        Connection.Parameters.Add(New SqlParameter("@barcode", barcode))

        If Connection.Update(query) Then
            MsgBox("Updated Successfully", MsgBoxStyle.Information, "Success")
            Return True
        End If
    Catch ex As Exception
        MsgBox("Error updating student: " & ex.Message)
    End Try
    Return False
End Function
```

UPDATE Query

MODULE: UTILITIES

```vb
Public Function DeleteProduct(ByVal id As String) As Boolean
    Try
        Dim query = "DELETE FROM productinfo WHERE id=@id"
        Connection.AddParam("@id", id)

        If Connection.Delete(query) Then
            MsgBox("Deleted Successfully", MsgBoxStyle.Information, "Success")
            Return True
        End If
    Catch ex As Exception
        MsgBox("Error deleting student: " & ex.Message)
    End Try
    Return False
End Function
```

DELETE Query

```vb
Public Sub SaveBarcode(ByVal barcodeImg As Image)
    If barcodeImg Is Nothing Then Return

    Dim saveDialog As New SaveFileDialog()
    saveDialog.Filter = "bmp (*.bmp)|*.bmp|jpeg (*.jpeg)|*.jpeg|png (*.png)|*.png|tiff (*.tiff)|*.tiff"
    If saveDialog.ShowDialog() = DialogResult.OK Then
        barcodeImg.Save(saveDialog.FileName)
    End If
End Sub
```

Now that we are done coding on both modules, we can now start coding on **Form1**

```
Imports System.Data.SqlClient

Public Class Form1

    Dim barcode_generator As New MessagingToolkit.Barcode.BarcodeEncoder
    Dim arrImage() As Byte

    Private Sub btnGenerate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGenerate.Click
        barcode_generator.LabelFont = New Font("Arial", 9, FontStyle.Regular)
        barcode_generator.IncludeLabel = True
        barcode_generator.CustomLabel = txtProdID.Text
        picBarcode.BackgroundImage = New Bitmap(barcode_generator.Encode(MessagingToolkit.Barcode.BarcodeFormat.Code128, txtProdID.Text))
    End Sub
```

FORM 1

```
Sub viewdata()
    Dim con1 As New SqlConnection("Server=LAPTOP-1CLIFQED\SQLEXPRESS; Datab
    Dim sql As String = "select * from productinfo"
    Dim Adapter As New SqlDataAdapter(sql, con1)
    Dim data As New DataTable("productinfo")
    Adapter.Fill(data)
    dgvInfo.DataSource = data
End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.Even
    viewdata()
End Sub

Sub clickondgv()
    Dim i As Integer
    i = dgvInfo.CurrentRow.Index

    txtID.Text = Convert.ToString(dgvInfo.Item(0, i).Value)
    txtProdID.Text = Convert.ToString(dgvInfo.Item(1, i).Value)
    txtName.Text = Convert.ToString(dgvInfo.Item(2, i).Value)
    txtQuantity.Text = Convert.ToString(dgvInfo.Item(3, i).Value)
    arrImage = TryCast(dgvInfo.Item(4, i).Value, Byte())
    If arrImage IsNot Nothing Then
        Dim mstream As New System.IO.MemoryStream(arrImage)
        picBarcode.BackgroundImage = Image.FromStream(mstream)
    Else
        picBarcode.BackgroundImage = Nothing
    End If
End Sub
```

FORM 1

```
Private Sub dgvInfo_CellClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataG
    clickondgv()
End Sub

Private Sub Clear()
    txtID.Clear()
    txtProdID.Clear()
    txtName.Clear()
    txtQuantity.Clear()
    picBarcode.Image = Nothing
    picBarcode.BackgroundImage = Nothing
    picBarcode.BackColor = Color.Empty
    picBarcode.Invalidate()
End Sub

Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAd
    Try
        Dim mstream As New System.IO.MemoryStream()
        picBarcode.BackgroundImage.Save(mstream, System.Drawing.Imaging.ImageFormat.Jpeg)
        arrImage = mstream.GetBuffer()
        Dim FileSize As UInt32
        FileSize = mstream.Length
        mstream.Close()

        If Utilities.InsertProduct(txtProdID.Text, txtName.Text, txtQuantity.Text, arrImage) Then
            Utilities.SaveBarcode(picBarcode.BackgroundImage)
            Clear()
            viewdata()
        End If
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
```

FORM 1

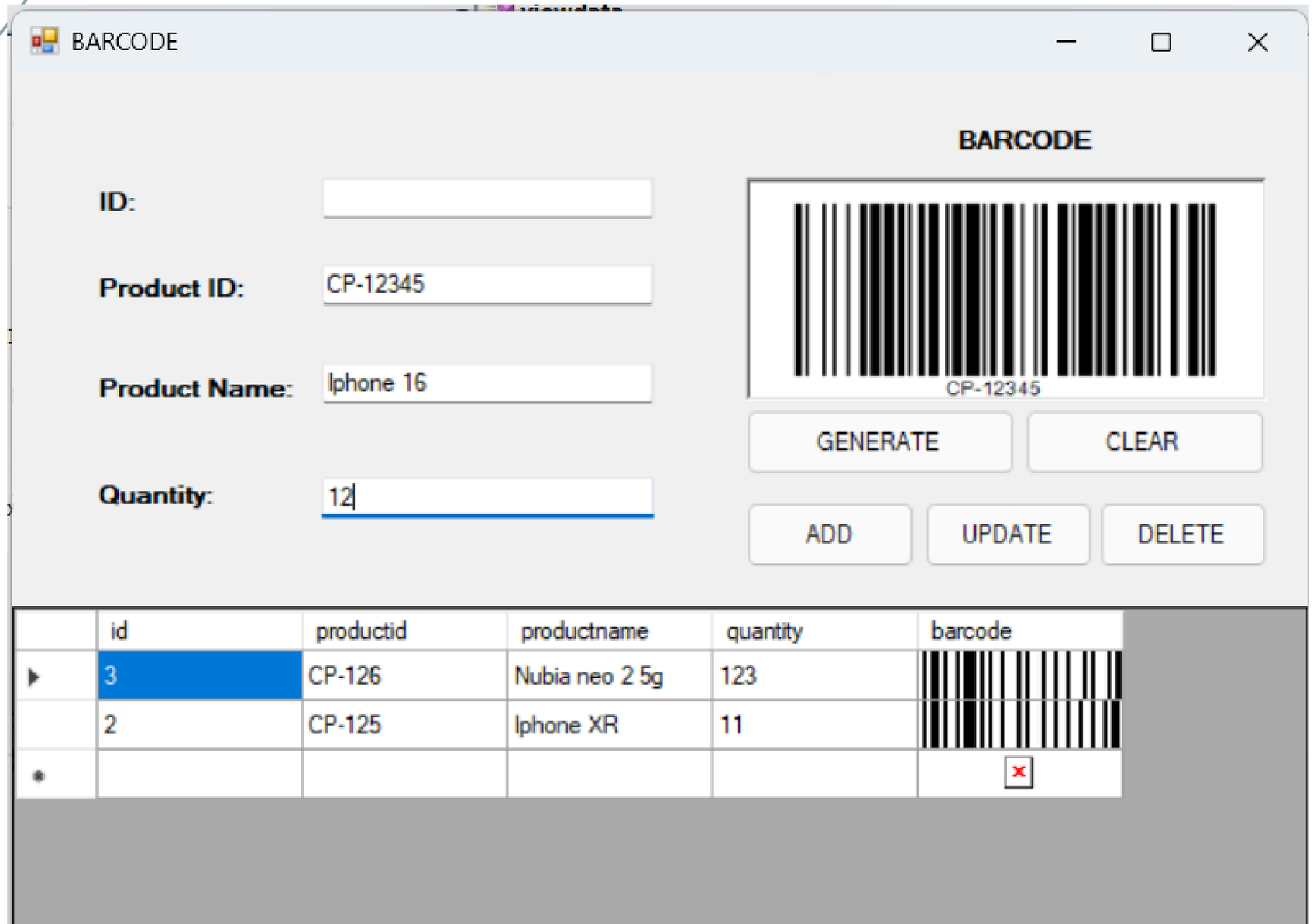
```
Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnUpdate.Clic
    Try
        Dim mstream As New System.IO.MemoryStream()
        picBarcode.BackgroundImage.Save(mstream, System.Drawing.Imaging.ImageFormat.Jpeg)
        arrImage = mstream.GetBuffer()
        Dim FileSize As UInt32
        FileSize = mstream.Length
        mstream.Close()

        If Utilities.UpdateProduct(txtID.Text, txtProdID.Text, txtName.Text, txtQuantity.Text, arrImage) Then
            Utilities.SaveBarcode(picBarcode.BackgroundImage)
            Clear()
            viewdata()
        End If
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDelete.Clic
    Try
        If Utilities.DeleteProduct(txtID.Text) Then
            Clear()
            viewdata()
        End If
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
    Clear()
End Sub
```

RESULT


BARCODE
BARCODE
ID:
Product ID:
CP-12345
Product Name:
Iphone 16
Quantity:
12
CP-12345
GENERATE
CLEAR
ADD
UPDATE
DELETE
id
productid
productname
quantity
barcode
3
CP-126
Nubia neo 2 5g
123
2
CP-125
Iphone XR
11

HAPPY CODING!