
PROJECT REPORT ON Rooster Routine Randomizer Project

Submitted by:

Remlalnghaka

Rollno. 2123BCA014

Ramdinmawia Zadeng

Rollno. 2123BCA013

Rosangpuia Chhakcchuak

Rollno.2123BCA017

**Under the guidance of
Mr.H.Lalruatkima
Assistant Professor
Department of Computer Sciences (HATIM)**



**MIZORAM UNIVERSITY,
TANHRIL: AIZAWL
2023**

HIGHER AND TECHNICAL INSTITUTE, MIZORAM

Kawmzawl, Pukpui, Lunglei – 796701



CERTIFICATE

This is to certify that ***Remlalnghaka , Ramdinmawia Zadeng and Rosangpuia Chhakchhuak*** have fully completed the project entitled “**Rooster Routine Randomizer**” to meet the requirement of the Mizoram University for the V Semester Bachelor Application in the year 2023. It is to certify that all the corrections/suggestions indicated for internal assessment have been incorporated into the project. The project report has been approved as it satisfies the academic requirements in respect of the project. The project report has been approved as it satisfies the academic requirements in respect of the Project work prescribed for the BCA course.

(H.Lalruatkima)
Project Guide
Dept. of Computer Science

(K.Lalmuanpuia)
Head of Department
Dept. of Computer Science

Mr.Vuansanga Vanchhawng
Principal
HATIM

.....
Name of Examiner

.....
Signature with date

ACKNOWLEDGEMENT

First, we would like to give thanks to God for guiding us throughout the process of our project, and giving us good health so that we can get this far. We would also like to give thanks to all our faculty members and our computer science department teacher, our Principal Mr.Vuansanga Vanchhawng, our head of Department K.Lalmuanpuia, and our Hostel Guide Sir H.Lalrinawma for giving us their full co-operation and support. We would also like to give thanks to all the people who helped us and involve in the making of this project, it's mainly because of them that we can complete it in time.

We also would like to give a word of gratitude, especially to our project invigilator, Sir H.Lalruatkima for guiding us from the beginning till the end, providing us with all the necessities required for our project helping us in the completion of our project. Last but not the least; we thank our parents for their support and encouragement.

DECLARATION

We do hereby declare that the project entitled “Rooster Routine Randomizer” is not submitted to any other university or institution for the award of any degree, diploma of fellowship, or published any time before. This project is prepared under the guidance of our project guide Mr. H.Lalruatkima which forms our partial fulfillment of the requirements for the three years, Bachelor’s Degree, in Computer Applications of Mizoram University.

ABSTRACT

The main aim of the project is to make a program that can be used locally in Mizoram where people can easily shuffle a roaster with more simplicity. It will be useful for secretary of any domination from small to large scales. For many secretary, it provides a platform more simple and easier to understand and work with.

To provide a simple and easy to understand, with a graphical user interface friendly where all domination will profit from it. This program will solely be active for Mizoram only. A place where it can provide a sufficient operation with just a click of a button. On this program, people will have to enter their name and password, and from the database they can retrieve the data that they enter.

Secretary of any domination will no longer need spreadsheet or excel to type out the roaster for the next month. For better understanding, it can received data such as the name of the preacher and shuffle them randomly and print it out.

As many secretary are old to this technology and its advancement, they cannot cope with the update of every little thing on Excel/Spreadsheet.

Local Secretary help make the roaster for every month and kind of hectic, to find the order of every preacher, so with the input data of every Sunday morning service, night also included and weekend and special Sunday, it will be more easier to work with and all they have to do is to put the preacher name and shuffle them, print.

It can easily delete a preacher's name or add them into the database, for example the decease person can be one of the preacher

LIST OF FIGURES

1. HOME P	-----AGE 17
2. REGISTRATION	----- 22
3. ACCOUNT	----- 23
4. CART	----- 24

CONTENTS

Acknowledgment.....	2
Declaration.....	3
Abstract.....	4
List of Figures.....	5
1. INTRODUCTION (PROJECT ANALYSIS)	7
2. Objectives of the Project.....	8
3. SYSTEM REQUIREMENTS.....	10
5. Overview of Front-end and back-end.....	10
 	11
7. SYSTEM ANALYSIS.....	13
 	13
 	14
 	15
11. SYSTEM DESIGN AND IMPLEMENTATION.....	17 - 24
12. CODING SYSTEM.....	25- 227
12.1. Home Page	25-26
12.2. Sign up	27-50
12.3. Login	51-96
12.4. Admin	93-125
12.5. About	127-138
12.6. Widgets	139-169
12.7. Payment structure and Security	170-190
12.8. Contact Page	191-209
12.9. Privacy Policy	210-218
13. LIMITATIONS AND DRAWBACKS OF THE PROJECT.....	228
14. FUTURE ENHANCEMENT.....	229
15. CONCLUSION.....	230
16. BIBLIOGRAPHY	231

1. PROJECT ANALYSIS

The project is to make a program that can be used locally in Mizoram where people can enter their preacher's name and shuffle them more easily. It will provide a friendly user interface where everyone can understand. To support the secretary of any domination.

For many newly selected preacher's, they will have a programme to run effortlessly and without any complication of making a preacher's roster.

To have a more simple and user friendly application for which everyone could understand.

The most benefiting from this software application, they can easily compile the preacher's name, and they once they randomize it, it will bring us to the page where it can be print out in a form of PDF, or can be print out as a hardcopy.

Purpose and Objectives of the project

To provide a friendly user experience for the user to minimize the work production of making a roaster. The project is to make a program that can be used locally in Mizoram where people can enter their preacher's name and shuffle them more easily. It will provide a friendly user interface where everyone can understand. To support the secretary of any domination.

System Requirements

Software Requirements

The software used for designing the User Interface (front-end & back-end) is VB.NET and Microsoft SQL Server management studio 19 is used.

Details of Software used

Visual Basic : VB.NET, or Visual Basic .NET, is a programming language developed by Microsoft. It is an evolution of the classic Visual Basic (VB) language and is designed to be more powerful and in line with modern programming practices.

Here are some key details about VB.NET:

1. **Syntax:** VB.NET uses a syntax similar to other languages in the .NET framework, such as C#. It is a case-insensitive language, making it more readable and easier for beginners to learn.
2. **Object-Oriented Programming (OOP):** VB.NET is an object-oriented language, supporting concepts like classes, inheritance, polymorphism, and encapsulation. This allows for the creation of modular and reusable code.
3. **Integrated Development Environment (IDE):** VB.NET is typically developed using Microsoft Visual Studio, a powerful IDE that provides a range of tools for designing, coding, testing, and debugging applications.
4. **Common Language Runtime (CLR):** VB.NET applications run on the .NET Framework's Common Language Runtime, which provides features such as automatic memory management, exception handling, and security.

5. **Database Connectivity:** VB.NET provides robust support for database connectivity through ADO.NET (ActiveX Data Objects). Developers can connect to various databases, execute queries, and manipulate data.

Microsoft SQL Server : Microsoft SQL Server is a relational database management system (RDBMS) developed by Microsoft. It's a comprehensive and feature-rich platform designed for managing and storing data.

Here are some key details:

1. **SQL Server Management Studio (SSMS):** This is the integrated development environment (IDE) for SQL Server. It provides tools for configuring, managing, and administering SQL Server instances.
2. **Integration Services (SSIS):** A platform for building high-performance data integration and workflow solutions. It allows you to solve complex business problems by copying or downloading files, extracting and transforming data from different systems, and loading data into one or multiple destinations.
3. **Analysis Services (SSAS):** Enables data analysis and business intelligence with Online Analytical Processing (OLAP) and data mining capabilities.
4. **Reporting Services (SSRS):** Provides a full range of ready-to-use tools and services to help you create, deploy, and manage reports for your organization.
5. **Calability:** SQL Server supports scalability both vertically (adding resources to a single server) and horizontally (adding more servers to a system).
6. **High Availability:** Features like Always On Availability Groups and Failover Clustering ensure high availability and disaster recovery.

Overview of Front-End and Back-End

VB.NET was not the first program which we intend to design in it, we look on a few different option such as Android Studio but the system requirement does not fit our system so we chose to go with VB.NET as we are more familiar with its function and a great chance to have a insight of what we were learning.

Front End (User Interface):

1. **Windows Forms:** VB.NET is often used for creating Windows desktop applications with a graphical user interface (GUI) using Windows Forms. This involves designing the layout and appearance of forms.
2. **Event-Driven Programming:** The front-end development in VB.NET is event-driven. Actions such as button clicks, mouse movements, or key presses trigger events, and developers write code to respond to these events
3. **Visual Studio IDE:** VB.NET developers typically use Microsoft Visual Studio, an integrated development environment (IDE), for designing and developing the front end. The IDE provides a drag-and-drop interface for designing forms and a code editor

Back End (Logic and Data Processing):

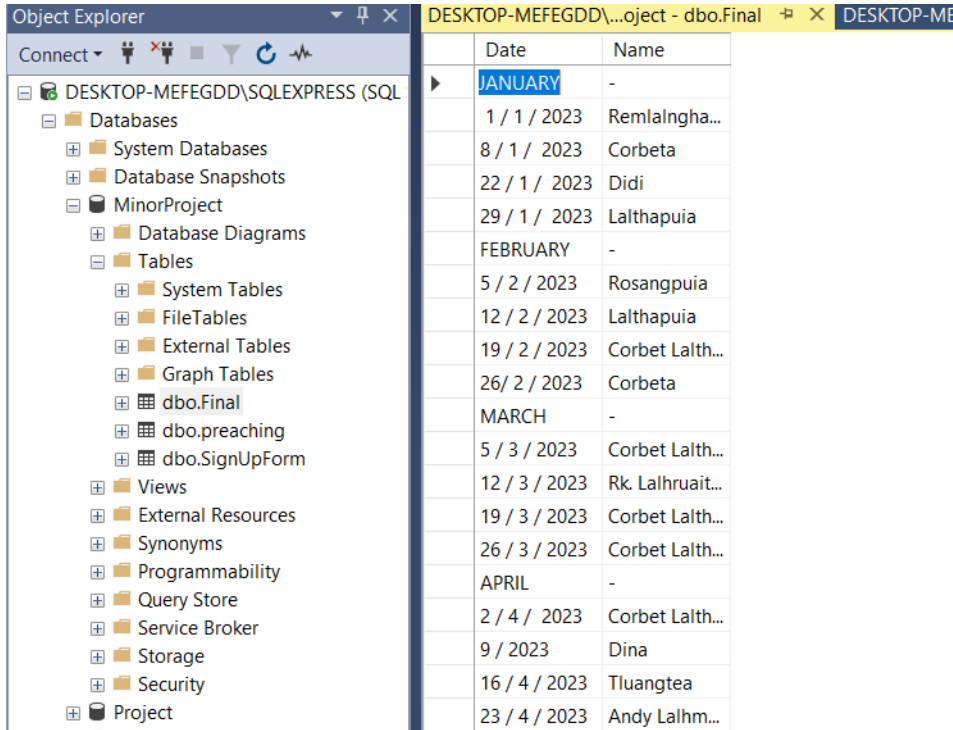
1. **Object-Oriented Programming (OOP):** VB.NET is an object-oriented language, allowing developers to organize code using classes and objects. This is beneficial for creating modular and maintainable code for the back-end logic.
2. **Database Connectivity:** VB.NET applications interact with databases to store and retrieve data. ADO.NET (ActiveX Data Objects) is commonly used for database connectivity in VB.NET, allowing developers to connect to various databases, execute queries, and manage data.
3. **Multithreading:** VB.NET supports multithreading, enabling developers to design applications that can perform multiple tasks concurrently. This is particularly useful for handling background processes without affecting the responsiveness of the user interface.
4. **Web Development (Optional):** While VB.NET is often associated with desktop applications, it can also be used for web development through ASP.NET. In web development, VB.NET is used to create the server-side logic that processes requests, interacts with databases, and generates dynamic web pages.
5. **Error Handling and Security:** The back end of a VB.NET application includes mechanisms for error handling and security. VB.NET applications benefit from the security features provided by the .NET framework, and developers implement error-handling code to ensure robust and secure applications.

Physical and Logical Structure

Database Design

Database and Tables :

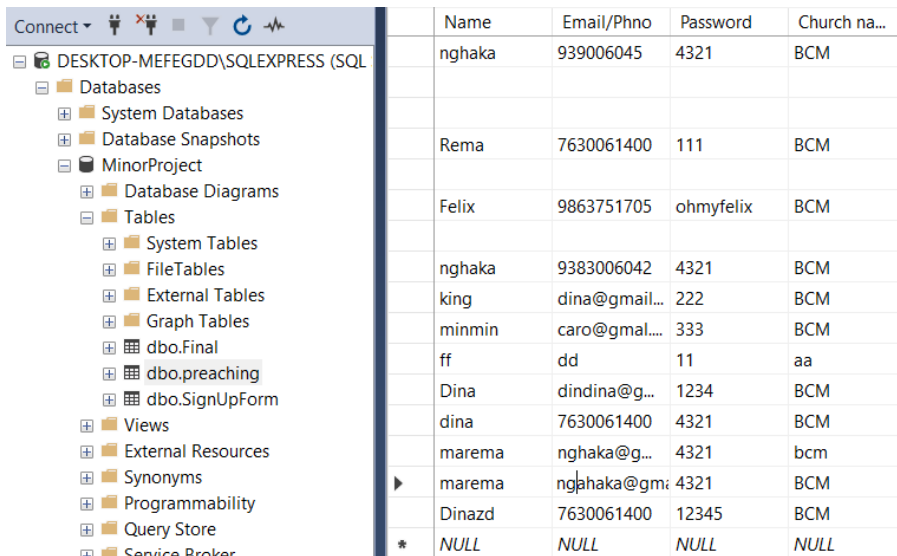
Roster routine-



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the database structure for 'DESKTOP-MELEGDD\SQLEXPRESS (SQL)'. The 'Tables' folder is expanded, showing a list of tables including 'dbo.Final', 'dbo.preaching', and 'dbo.SignUpForm'. The 'dbo.Final' table is selected. On the right, the 'Query Results' pane shows the data for the 'dbo.Final' table. The table has two columns: 'Date' and 'Name'. The data is organized by month, starting with 'JANUARY' and 'FEBRUARY'.

Date	Name
JANUARY	-
1 / 1 / 2023	Remlalngha...
8 / 1 / 2023	Corbeta
22 / 1 / 2023	Didi
29 / 1 / 2023	Lalthapuia
FEBRUARY	-
5 / 2 / 2023	Rosangpuia
12 / 2 / 2023	Lalthapuia
19 / 2 / 2023	Corbet Lalth...
26 / 2 / 2023	Corbeta
MARCH	-
5 / 3 / 2023	Corbet Lalth...
12 / 3 / 2023	Rk. Lalhrui...
19 / 3 / 2023	Corbet Lalth...
26 / 3 / 2023	Corbet Lalth...
APRIL	-
2 / 4 / 2023	Corbet Lalth...
9 / 2023	Dina
16 / 4 / 2023	Tluangtea
23 / 4 / 2023	Andy Lalhm...

SignUpForm table which contain the user's username and password.



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the database structure for 'DESKTOP-MELEGDD\SQLEXPRESS (SQL)'. The 'Tables' folder is expanded, showing a list of tables including 'dbo.Final', 'dbo.preaching', and 'dbo.SignUpForm'. The 'dbo.SignUpForm' table is selected. On the right, the 'Query Results' pane shows the data for the 'dbo.SignUpForm' table. The table has five columns: 'Name', 'Email/Phno', 'Password', and 'Church na...'. The data is organized by user, starting with 'nghaka' and 'Rema'.

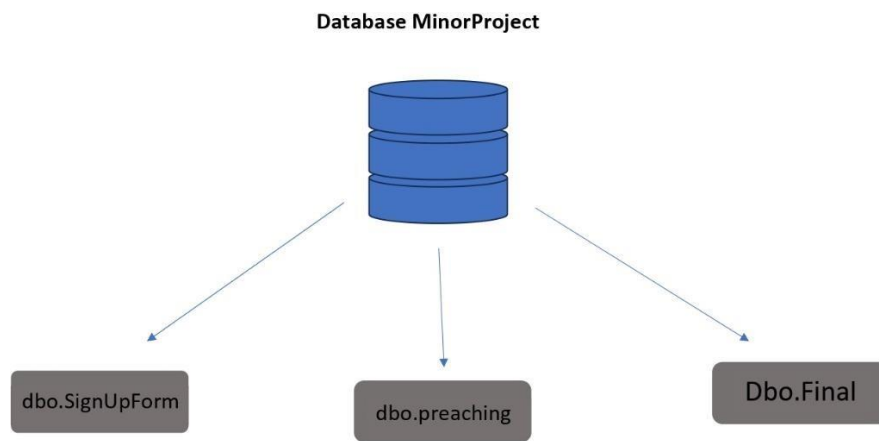
Name	Email/Phno	Password	Church na...
nghaka	939006045	4321	BCM
Rema	7630061400	111	BCM
Felix	9863751705	ohmyfelix	BCM
nghaka	9383006042	4321	BCM
king	dina@gmail...	222	BCM
minmin	caro@gmal...	333	BCM
ff	dd	11	aa
Dina	dindina@g...	1234	BCM
dina	7630061400	4321	BCM
marema	nghaka@g...	4321	bcm
marema	ngahaka@gm...	4321	BCM
Dinazd	7630061400	12345	BCM
*	NULL	NULL	NULL

Preaching table which contain the preacher's name, address, phone

Object Explorer					DESKTOP-MELEGDD\...ject - dbo.Final					DESKTOP-MELEG				
Connect														
DESKTOP-MELEGDD\SQLEXPRESS (SQL)														
Databases														
System Databases														
Database Snapshots														
MinorProject														
Database Diagrams														
Tables														
System Tables														
FileTables														
External Tables														
Graph Tables														
dbo.Final														
dbo.preaching														
dbo.SignUpForm														
Views														
External Resources														

	id	name	address	phone
7	7	RemlalIngha...	s.vanlaiphai	785938
8	8	Ramdinma...	Tlabung	975732
9	9	Rosangpuia	Tlabung	978674
10	10	Andy Lalhm...	Aizawl	87659
11	11	Lalthapuia	Hrangchalk...	887949
12	12	Corbet Lalth...	Lawngtlai	8785743
13	13	Rk. Lalhrui...	Serchhip	7685035
14	14	Corbeta	Lawngtlai	787573
15	15	Tluangtea	Lunglei	988786
16	16	Kimkima	Lawngtlai	567378
17	17	Dina	tlabung	256742
18	18	Didi	Tlabung	6579472
*	NULL	NULL	NULL	NULL

Database Structure :



Tables contained in the database MinorProject.

SYSTEM DESIGN AND IMPLEMENTATION

User Interface and approach

Home Page: This is where you will have to register for new user's or sign in by filling up all the textboxes available in the form.

	Exit	Sign in	Register
Already have account ?			
EMAIL / PHONE NO	<input type="text"/>		
CHURCH'S NAME	<input type="text"/>		
USERNAME	<input type="text"/>		
PASSWORD	<input type="password"/>		

Login Page : For who already register users they can login for the perpose of any kind the application provided by filling up the textboxes display on the form 'Username' & 'Password'.

		Exit	Back	Log in
USERNAME	<input type="text"/>			
PASSWORD	<input type="password"/>			

The home page : This windows form is the center of this software for the work. Provide a key to create new user account(Back to the first form), preaching list(Next form), login another(Login form for other users) account.

Help

Create New Account

Create The Preaching List

Login Another Account

Exit

Preaching list form :This is where we should enter the preacher's name and bio-date which is to fill up. And submit into database after insert all the data in the textboxes. Also available to update, delete and ramdommizing the inserted data for the Preaching Roster.

		Exit	Back	Submit			
<i>Name</i>	<input type="text"/>	<table border="1"><thead><tr><th>name</th><th>address</th><th>phone</th></tr></thead><tbody></tbody></table>			name	address	phone
name	address				phone		
<i>Address</i>	<input type="text"/>						
<i>Phone no</i>	<input type="text"/>						
		Insert	New	Delete	Update	Randomize	

Shuffle Form : This form provide multiple button to shuffle the preacher person to their respective date for preching and able to modify if mistake happened by rewrite the name on the texboxes.

Display button let it to the next form of crystal report to display the Roster routine and Provide a printing methods.

Form5

Back

Clear

Shuffle

Confirm

Submit

Display

Preacher's <div>▼</div> <div>▼</div> <div>▼</div>	January		May		September	
	1 / 1 / 2023		1 / 5 / 2023		1 / 9 / 2023	
	8 / 1 / 2023		8 / 5 / 2023		8 / 9 / 2023	
	22 / 1 / 2023		22 / 5 / 2023		22 / 9 / 2023	
	29 / 1 / 2023		29 / 5 / 2023		29 / 9 / 2023	
	February		June		October	
	5 / 2 / 2023		5 / 6 / 2023		5 / 10 / 2023	
	12 / 2 / 2023		12 / 6 / 2023		12 / 10 / 2023	
	19 / 2 / 2023		19 / 6 / 2023		19 / 10 / 2023	
	26 / 2 / 2023		26 / 6 / 2023		26 / 10 / 2023	
	March		July		November	
	5 / 3 / 2023		5 / 7 / 2023		5 / 11 / 2023	
12 / 3 / 2023		12 / 7 / 2023		12 / 11 / 2023		
19 / 3 / 2023		19 / 7 / 2023		19 / 11 / 2023		
26 / 3 / 2023		26 / 7 / 2023		26 / 11 / 2023		
April		August		December		
2 / 4 / 2023		2 / 8 / 2023		2 / 12 / 2023		
9 / 4 / 2023		9 / 8 / 2023		9 / 12 / 2023		
16 / 4 / 2023		16 / 8 / 2023		16 / 12 / 2023		
23 / 4 / 2023		23 / 8 / 2023		23 / 12 / 2023		

Report form : This form will display the roster which shuffled in the previous windows form .
Let it display in a proper manner for printing as well.

The image shows a screenshot of the SAP Crystal Reports 'Form6' window. The window has a title bar labeled 'Form6'. Below the title bar is a toolbar containing icons for print, refresh, save, and navigation. The main area of the window is split into two panes: a white left pane and a grey right pane. The status bar at the bottom displays 'Current Page No.:', 'Total Page No.:', and 'Zoom Factor: 100%'. The text 'SAP CRYSTAL REPORTS' is visible in the top right corner of the window.

CODING SYSTEM:

As the front-end is created using Wordpress,Html and PHP with CSS, the language used to develop the application is PHP programming. The codes used in the project are given below:

Codes for Form 1:

```
Imports System.Data
```

```
Imports System.Data.SqlClient
```

```
Imports System.Data.OleDb
```

```
Public Class Form1
```

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles btnRegister.Click
```

```
    If txtpassword.Text = "" Then
```

```
        txtpassword.Focus()
```

```
        Label8.Text = "Invalid Password"
```

```
        Label8.ForeColor = Color.Red
```

```
        Exit Sub
```

```
    Else
```

```
        Label8.Text = ""
```

```
    End If
```

```
    If txtusername.Text = "" Then
```

```
        txtusername.Focus()
```

```
        Label7.Text = "Invalid username"
```

```
        Label7.ForeColor = Color.Red
```

```
        Exit Sub
```

```
    Else
```

```
        Label7.Text = ""
```

```
    End If
```

```
    If txtemailphone.Text = "" Then
```

```
        txtemailphone.Focus()
```

```
        Exit Sub
```

```
    End If
```

```
    If txtchurchname.Text = "" Then
```

```
        txtchurchname.Focus()
```

```
        Exit Sub
```

```
    End If
```

```

        Dim con As SqlConnection = New SqlConnection("Data Source=DESKTOP-
MEFEGDD\SQLEXPRESS;Initial Catalog=MinorProject;Integrated Security=True")
        Dim cmd As SqlCommand = New SqlCommand("INSERT INTO [dbo].[SignUpForm]
        ([Name]
        ,[Email/Phno]
        ,[Password]
        ,[Church name])
VALUES
        ('" + txtusername.Text + "','" + txtmailphone.Text + "','" + txtpassword.Text + "','" +
txtchurchname.Text + "')", con)
        Me.Hide()
        Form2.Show()
        con.Open()
        cmd.ExecuteNonQuery()
        MessageBox.Show("You have registered sucessfully ")
    End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles btnSignIn.Click
    Me.Hide()
    Form2.Show()

End Sub

Private Sub Button1_Click_1(sender As Object, e As EventArgs) Handles Button1.Click
    End

End Sub

End Class

```

Codes for form 2:

Imports System.Data

Imports System.Data.SqlClient

Public Class Form2

Dim rdr As SqlDataReader

Dim cmd As SqlCommand

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

Form1.txtchurchname.Clear()

Form1.txtemailphone.Clear()

Form1.txtpassword.Clear()

Form1.txtusername.Clear()

Try

Dim con As SqlConnection = New SqlConnection("Data Source=DESKTOP-MEFEGDD\SQLEXPRESS;Initial Catalog=MinorProject;Integrated Security=True")

cmd = New SqlCommand("select * from SignUpForm where Name ='" & LoginName.Text & "'and Password='" & LoginPassword.Text & "'", con)

Dim sda As SqlDataAdapter = New SqlDataAdapter(cmd)

Dim dt As DataTable = New DataTable()

sda.Fill(dt)

If (dt.Rows.Count > 0) Then

MessageBox.Show("Login Successful", "information", MessageBoxButtons.OK, MessageBoxIcon.Information)

Me.Hide()

Form3.Show()

LoginName.Text = ""

LoginPassword.Text = ""

Else

MessageBox.Show("Login error", "information", MessageBoxButtons.OK, MessageBoxIcon.Information)

End If

' cmd = New SqlCommand("SELECT [Name]

' ,[Email/Phno]

' ,[Password]

' ,[Church name]

' FROM [dbo].[SignUpForm] WHERE Name = '" & LoginName.Text & "', AND Password ='" & LoginPassword.Text & "'")

' cmd.Connection = con

' rdr = cmd.ExecuteReader()

' If (rdr.Read()) Then

' Form3.Show()

' Me.Hide()

```
    ' End If
    'Catch ex As Exception
    'End Try
End Sub
```

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
```

```
    Me.Hide()
    Form1.Show()
    LoginPassword.Text = ""
    LoginName.Text = ""
    Form1.txtchurchname.Clear()
    Form1.txtemailphone.Clear()
    Form1.txtpassword.Clear()
    Form1.txtusername.Clear()
End Sub
```

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
```

```
    End
End Sub
End Class
```

Codes for Form 3:

```
Public Class Form3
```

```
    Private Sub Button1_Click(sender As Object, e As EventArgs)
        End
```

```
End Sub
```

```
Private Sub Button2_Click(sender As Object, e As EventArgs)
```

```
    Me.Hide()
    Form4.Show()
```

```
End Sub
```

```
Private Sub Button5_Click(sender As Object, e As EventArgs)
```

```
    Me.Hide()
    Form1.Show()
```

```
End Sub
```

```
Private Sub Button4_Click(sender As Object, e As EventArgs)
```

```
    Me.Hide()
```

```
    Form2.Show()
```

```
End Sub
```

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
```

```
    Button3.BackColor = Color.Gray
```

```
    MessageBox.Show("CREATE NEW ACCOUNT - if new account require or want new account
```

CREATE PREACHING ROASTER - to create the roaster of the preacher by plotting them into the particular date

LOGIN ANOTHER ACCOUNT - which help to enter another account that we have, along with the data its contain

BACK TO SIGNUP - enable to create or register new account for the application")

```
End Sub
```

```
Private Sub Label4_MouseHover(sender As Object, e As EventArgs) Handles Label4.MouseHover
```

```
    Label4.ForeColor = Color.Blue
```

```
End Sub
```

```
Private Sub Label4_MouseLeave(sender As Object, e As EventArgs) Handles Label4.MouseLeave
```

```
    Label4.ForeColor = Color.Black
```

```
End Sub
```

```
Private Sub Label3_Click(sender As Object, e As EventArgs) Handles lblnewaccount.Click
```

```
    Me.Hide()
```

```
    Form1.Show()
```

```
End Sub
```

```
Private Sub lblnewaccount_MouseHover(sender As Object, e As EventArgs) Handles  
lblnewaccount.MouseHover
```

```
    lblnewaccount.ForeColor = Color.Blue
```

```
End Sub
```

```
Private Sub lblnewaccount_MouseLeave(sender As Object, e As EventArgs) Handles  
lblnewaccount.MouseLeave
```



```
    lblnewaccount.ForeColor = Color.Black  
End Sub
```

```
Private Sub lbllist_MouseHover(sender As Object, e As EventArgs) Handles lbllist.MouseHover  
    lbllist.ForeColor = Color.Blue  
End Sub
```

```
Private Sub lbllist_MouseLeave(sender As Object, e As EventArgs) Handles lbllist.MouseLeave  
    lbllist.ForeColor = Color.Black  
End Sub
```

```
Private Sub lblanotheraccount_MouseHover(sender As Object, e As EventArgs) Handles  
lblanotheraccount.MouseHover  
    lblanotheraccount.ForeColor = Color.Blue  
End Sub
```

```
Private Sub lblanotheraccount_MouseLeave(sender As Object, e As EventArgs) Handles  
lblanotheraccount.MouseLeave  
    lblanotheraccount.ForeColor = Color.Black  
End Sub
```

```
Private Sub lbllist_Click(sender As Object, e As EventArgs) Handles lbllist.Click  
    Me.Hide()  
    Form4.Show()  
End Sub
```

```
Private Sub lblanotheraccount_Click(sender As Object, e As EventArgs) Handles  
lblanotheraccount.Click  
    Me.Hide()  
    Form2.Show()
```

```
End Sub
```

```
End Class
```

CODES FOR Form 4:

Imports System.Data.DataTable

Imports System.IO

Imports System.Data

Imports System.Data.SqlClient

Public Class Form4

Dim table As New DataTable("table")

Dim index As Integer

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

End

End Sub

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

Me.Hide()

Form3.Show()

End Sub

Private Sub Form4_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'TODO: This line of code loads data into the 'MinorProjectDataSet3.preaching' table. You can move, or remove it, as needed.

Me.PreachingTableAdapter2.Fill(Me.MinorProjectDataSet3.preaching)

'TODO: This line of code loads data into the 'MinorProjectDataSet2.preaching' table. You can move, or remove it, as needed.

Me.PreachingTableAdapter1.Fill(Me.MinorProjectDataSet2.preaching)

table.Columns.Add("ID", Type.GetType("System.Int32"))

table.Columns.Add("Name", Type.GetType("System.String"))

table.Columns.Add("Address", Type.GetType("System.String"))

table.Columns.Add("Phone", Type.GetType("System.Int64"))

DataGridView1.DataSource = table

End Sub

Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click

table.Rows.Add(TextBox2.Text, TextBox3.Text, TextBox4.Text)

DataGridView1.DataSource = table

End Sub

Private Sub DataGridView1_CellClick(sender As Object, e As DataGridViewCellEventArgs)
Handles DataGridView1.CellClick

index = e.RowIndex

Dim selectedrow As DataGridViewRow

selectedrow = DataGridView1.Rows(index)

TextBox2.Text = selectedrow.Cells(1).Value.ToString

TextBox3.Text = selectedrow.Cells(2).Value.ToString

TextBox4.Text = selectedrow.Cells(3).Value.ToString

End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click

Dim newdata As DataGridViewRow

newdata = DataGridView1.Rows(index)

newdata.Cells(1).Value = TextBox2.Text

newdata.Cells(2).Value = TextBox3.Text

newdata.Cells(3).Value = TextBox4.Text

End Sub

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click

TextBox2.Text = ""

TextBox3.Text = ""

TextBox4.Text = ""

End Sub

```
Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
    DataGridView1.Rows.RemoveAt(index)
```

```
End Sub
```

```
Private Sub DataGridView1_CellContentClick(sender As Object, e As
DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick
```

```
End Sub
```

```
Dim conn As New SqlConnection("Data Source=DESKTOP-MEFEGDD\SQLEXPRESS;Initial
Catalog=MinorProject;Integrated Security=True")
```

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
```

```
For Each row As DataGridViewRow In DataGridView1.Rows
```

```
    If Not String.IsNullOrEmpty(row.Cells(1).Value.ToString()) Then
```

```
        Dim cmd As New SqlCommand("Insert into preaching(name,address,phone)
values(@name,@address,@phone)", conn)
```

```
        'cmd.Parameters.AddWithValue("id", row.Cells(0).Value)
```

```
        cmd.Parameters.AddWithValue("name", row.Cells(1).Value.ToString())
```

```
        cmd.Parameters.AddWithValue("address", row.Cells(2).Value.ToString())
```

```
        cmd.Parameters.AddWithValue("phone", row.Cells(3).Value)
```

```
        conn.Open()
```

```
        cmd.ExecuteNonQuery()
```

```
        conn.Close()
```

```
    End If
```

```
Next
```

```
MsgBox("Data inserted successfully")
```

```
End Sub
```

```
Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
```

```
    Me.Hide()
```

```
    Form5.Show()
```

```
End Sub
```

```
End Class
```

Code for Form 5:

Imports System.Data.SqlClient

Imports System.Data.OleDb

Imports System.IO

Imports System.Data

Imports System.Random

Public Class Form5

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click

'shuffle the preacher's and place into the textboxes

shuffle()

End Sub

Private Sub shuffle()

Dim random As New Random()

For Each textBox As TextBox In Me.Controls.OfType(Of TextBox)

Dim randomIndex As Integer = random.Next(0, ComboBox1.Items.Count)

textBox.Text = ComboBox1.Items(randomIndex).ToString()

Next

End Sub

Private Sub Form5_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim connection As String = "Data Source=DESKTOP-MEFEGDD\SQLEXPRESS;Initial
Catalog=MinorProject;Integrated Security=True"

' SQL query to retrieve data from a table

Dim query As String = "SELECT name FROM preaching"

' Create a SqlConnection and a SqlCommand

Using connect As New SqlConnection(connection)

Using command As New SqlCommand(query, connect)

connect.Open()

' Create a SqlDataReader to retrieve the data

Dim reader As SqlDataReader = command.ExecuteReader()

' Clear the ComboBox

ComboBox1.Items.Clear()

```

        ' Loop through the data and add it to the ComboBox
        While reader.Read()
            ComboBox1.Items.Add(reader("Name").ToString())
        End While

        ' Close the SqlDataReader and the SqlConnection
        reader.Close()
        connect.Close()
    End Using
End Using
End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
    For Each textBox As TextBox In Me.Controls.OfType(Of TextBox)
        textBox.Clear()

    Next

End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    For Each textBox As TextBox In Me.Controls.OfType(Of TextBox)
        textBox.Clear()
    Next
    Me.Hide()
    Form4.Show()

End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs)
    End
End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    ComboBox2.Items.Add("JANUARY") ' January
    ComboBox2.Items.Add(Label2.Text)
    ComboBox2.Items.Add(Label3.Text)
    ComboBox2.Items.Add(Label4.Text)
    ComboBox2.Items.Add(Label5.Text)
    ComboBox2.Items.Add("FEBRUARY") ' February

```

```
ComboBox2.Items.Add(Label6.Text)
ComboBox2.Items.Add(Label7.Text)
ComboBox2.Items.Add(Label8.Text)
ComboBox2.Items.Add(Label9.Text)
ComboBox2.Items.Add("MARCH") ' March
ComboBox2.Items.Add(Label17.Text)
ComboBox2.Items.Add(Label16.Text)
ComboBox2.Items.Add(Label14.Text)
ComboBox2.Items.Add(Label15.Text)
ComboBox2.Items.Add("APRIL") ' April
ComboBox2.Items.Add(Label10.Text)
ComboBox2.Items.Add(Label11.Text)
ComboBox2.Items.Add(Label12.Text)
ComboBox2.Items.Add(Label13.Text)
ComboBox2.Items.Add("MAY") ' May
ComboBox2.Items.Add(Label42.Text)
ComboBox2.Items.Add(Label41.Text)
ComboBox2.Items.Add(Label40.Text)
ComboBox2.Items.Add(Label33.Text)
ComboBox2.Items.Add("JUNE") ' June
ComboBox2.Items.Add(Label30.Text)
ComboBox2.Items.Add(Label32.Text)
ComboBox2.Items.Add(Label34.Text)
ComboBox2.Items.Add(Label38.Text)
ComboBox2.Items.Add("JULY") ' July
ComboBox2.Items.Add(Label35.Text)
ComboBox2.Items.Add(Label36.Text)
ComboBox2.Items.Add(Label39.Text)
ComboBox2.Items.Add(Label37.Text)
ComboBox2.Items.Add("AUGUST") ' August
ComboBox2.Items.Add(Label27.Text)
ComboBox2.Items.Add(Label28.Text)
ComboBox2.Items.Add(Label29.Text)
ComboBox2.Items.Add(Label31.Text)
ComboBox2.Items.Add("SEPTEMBER") ' September
ComboBox2.Items.Add(Label62.Text)
ComboBox2.Items.Add(Label61.Text)
ComboBox2.Items.Add(Label60.Text)
ComboBox2.Items.Add(Label53.Text)
ComboBox2.Items.Add("OCTOBER") ' October
```

```
ComboBox2.Items.Add(Label50.Text)
ComboBox2.Items.Add(Label52.Text)
ComboBox2.Items.Add(Label54.Text)
ComboBox2.Items.Add(Label58.Text)
ComboBox2.Items.Add("NOVEMBER") ' November
ComboBox2.Items.Add(Label55.Text)
ComboBox2.Items.Add(Label56.Text)
ComboBox2.Items.Add(Label59.Text)
ComboBox2.Items.Add(Label57.Text)
ComboBox2.Items.Add("DECEMBER") ' December
ComboBox2.Items.Add(Label47.Text)
ComboBox2.Items.Add(Label48.Text)
ComboBox2.Items.Add(Label49.Text)
ComboBox2.Items.Add(Label51.Text)
```

'Textboxes

```
ComboBox3.Items.Add("-")
ComboBox3.Items.Add(TextBox1.Text)
ComboBox3.Items.Add(TextBox2.Text)
ComboBox3.Items.Add(TextBox3.Text)
ComboBox3.Items.Add(TextBox4.Text)
ComboBox3.Items.Add("-")
ComboBox3.Items.Add(TextBox5.Text)
ComboBox3.Items.Add(TextBox6.Text)
ComboBox3.Items.Add(TextBox12.Text)
ComboBox3.Items.Add(TextBox11.Text)
ComboBox3.Items.Add("-")
ComboBox3.Items.Add(TextBox10.Text)
ComboBox3.Items.Add(TextBox9.Text)
ComboBox3.Items.Add(TextBox8.Text)
ComboBox3.Items.Add(TextBox7.Text)
ComboBox3.Items.Add("-")
ComboBox3.Items.Add(TextBox16.Text)
ComboBox3.Items.Add(TextBox15.Text)
ComboBox3.Items.Add(TextBox14.Text)
ComboBox3.Items.Add(TextBox13.Text)
ComboBox3.Items.Add("-")
ComboBox3.Items.Add(TextBox32.Text)
ComboBox3.Items.Add(TextBox31.Text)
ComboBox3.Items.Add(TextBox29.Text)
```



```
ComboBox3.Items.Add(TextBox25.Text)
ComboBox3.Items.Add("-")
ComboBox3.Items.Add(TextBox30.Text)
ComboBox3.Items.Add(TextBox28.Text)
ComboBox3.Items.Add(TextBox27.Text)
ComboBox3.Items.Add(TextBox26.Text)
ComboBox3.Items.Add("-")
ComboBox3.Items.Add(TextBox17.Text)
ComboBox3.Items.Add(TextBox18.Text)
ComboBox3.Items.Add(TextBox21.Text)
ComboBox3.Items.Add(TextBox24.Text)
ComboBox3.Items.Add("-")
ComboBox3.Items.Add(TextBox23.Text)
ComboBox3.Items.Add(TextBox22.Text)
ComboBox3.Items.Add(TextBox20.Text)
ComboBox3.Items.Add(TextBox19.Text)
ComboBox3.Items.Add("-")
ComboBox3.Items.Add(TextBox48.Text)
ComboBox3.Items.Add(TextBox47.Text)
ComboBox3.Items.Add(TextBox45.Text)
ComboBox3.Items.Add(TextBox41.Text)
ComboBox3.Items.Add("-")
ComboBox3.Items.Add(TextBox46.Text)
ComboBox3.Items.Add(TextBox44.Text)
ComboBox3.Items.Add(TextBox43.Text)
ComboBox3.Items.Add(TextBox42.Text)
ComboBox3.Items.Add("-")
ComboBox3.Items.Add(TextBox33.Text)
ComboBox3.Items.Add(TextBox34.Text)
ComboBox3.Items.Add(TextBox37.Text)
ComboBox3.Items.Add(TextBox40.Text)
ComboBox3.Items.Add("-")
ComboBox3.Items.Add(TextBox39.Text)
ComboBox3.Items.Add(TextBox38.Text)
ComboBox3.Items.Add(TextBox36.Text)
ComboBox3.Items.Add(TextBox35.Text)
```

End Sub

Private Sub submit()

```
Dim pro As String = "Data Source=DESKTOP-MEFEGDD\SQLEXPRESS;Initial
```

```

Catalog=MinorProject;Integrated Security=True"
    Dim cmd As String = "INSERT INTO Final([Date],[Name]) values (@DATE, @NAME)"
    Dim connect As New SqlConnection(pro)
    connect.Open()
    Try

        For i As Integer = 0 To ComboBox2.Items.Count - 1

            'command = "INSERT INTO Table1 ([DATE],[NAME]) values (@DATE, @NAME)"
            Dim command As New SqlCommand(cmd, connect)
            command.Parameters.AddWithValue("@DATE", ComboBox2.Items(i))
            command.Parameters.AddWithValue("@NAME", ComboBox3.Items(i))
            command.ExecuteNonQuery()
        Next
        connect.Close()
        MsgBox("Data Saved")
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

Private Sub Button2_Click_1(sender As Object, e As EventArgs) Handles Button2.Click
    submit()
End Sub

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click

    Form6.Show()

End Sub
End Class

```

Codes for Form 6:

Public Class Form6

```
Private Sub Form6_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    CrystalReportViewer1.ReportSource = Application.StartupPath + "\CrystalReport1.rpt"
    CrystalReportViewer1.Refresh()
    CrystalReportViewer1.RefreshReport()
```

End Sub

End Class

LIMITATION AND DRAWBACK

Our teamwork was exceptional for this project, we put our utmost effort for this project, applying our knowledge of VB.NET that we learned in fifth semester, its a new experience for us to work on Microsoft SQL Server we learned by ourselves since it is outside our syllabus but as a team we tried to influence our junior that everything is possible when we are dedicated to achieve a goal. When we decided this project we only knew what we wanted to do without the how to do it , but by the help and motivation of our teacher we are able to build this program.

We failed many times building this program as we did research on the go making step by step progress and implementing the knowledge we have gained from our BCA course and self learning on the internet.

We build this program because we felt the need for secretaries who work tirelessly for the church, for the younger secretaries, who don't have any experiences, this could be a huge help as it will be as simple and does not need any complicated formula to have an output.

Because of the Short tenure of a Semester ,we had little Time(only Four Months) for developing this program. The sixth semester is only a period of 5 months including the examination month.

It was taxing work as expected because even though our teachers supported and helped us in whatever the way they can, when. In the end we could only rely on ourselves. We usually consult tutorial videos from Youtube, ChatGPT, Websites that teach about VB.NET and Microsoft SQL Server and how to combine them, to facilitate our project work which gives us and help us gain a lot of knowledge in terms of developing this program.

FUTURE UPGRADES

As of now, we don't have any future upgrades in mind, but we can fix any bug which can occur while using the program.

CONCLUSION

We know that our project “R3” has many drawbacks and shortcomings. But we are proud to present our Work. This project work is the compilations of our ideas, views and thoughts, we have benefited a great deal from our interaction with our teachers, peoples and friends. We extend our sincere thanks to them.

It was developed hoping that people from different parts of Mizoram will be making use of this program as we intended to and improve the productivity of the church services.

It is a great pleasure for us to express our immense regards to our project guide Mr. H. Lalruatkima and Head of Department Mr. K.Lalmuanpuia for their inspirational guidance who helps us in many ways from the beginning till the end.

We would like to place and record sincere thanks and gratitude to our Principal, Mr VuansangaVanchhawng for extending his full support and contentment to this project.

Though this project may not be the best, we hope and pray that the user of this program will find satisfaction in using it. We find happiness and pride to have completed this project. With many efforts this project has come into being even though we are conscious of our limited knowledge and skills. But it would be our request than this project we have undertaken, be dealt with much consideration and acknowledgement.

BIBLIOGRAPHY

www.youtube.com

chat.openai.com