Jake Norris

Professor Featherman

MIS325

12/17/2021
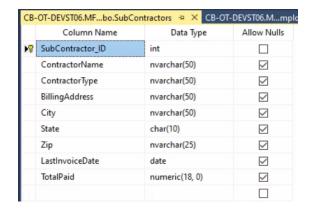
Final Project

Website Link:

http://cb-ot-devst06.ad.wsu.edu/MF13jake.norris/Final%20Project/BigIdeasConstructionAccount

ing.aspx

Screenshots of databases:

CB-OT-DEVST06.MF...bo.SubContractors CB-OT-DEVST06.M...mplo

Column Name	Data Type	Allow Nulls
SubContractor_ID	int	☐
ContractorName	nvarchar(50)	☑
ContractorType	nvarchar(50)	☑
BillingAddress	nvarchar(50)	☑
City	nvarchar(50)	☑
State	char(10)	☑
Zip	nvarchar(25)	☑
LastInvoiceDate	date	☑
TotalPaid	numeric(18, 0)	☑
		☐

Column Name	Data Type	Allow Nulls
Employeelabor_ID	int	☐
EmployeeName	nvarchar(50)	☑
LaborDate	date	☑
Day	nvarchar(50)	☑
Hours	numeric(18, 0)	☑
PayRate	numeric(18, 0)	☑
Earned	numeric(18, 0)	☑
		☐

Column Name	Data Type	Allow Nulls
EmployeeArchive_ID	int	☐
EmployeeName	nvarchar(50)	☑
HRCategory	nvarchar(50)	☑
PayRate	numeric(18, 0)	☑
DaysWorked	numeric(18, 0)	☑
LastDayWorked	date	☑
TotalEarned	numeric(18, 0)	☑
		☐

Column Name	Data Type	Allow Nulls
Customer_ID	int	☐
CustomerName	nvarchar(50)	☑
Address	nvarchar(50)	☑
City	nvarchar(50)	☑
State	char(2)	☑
Zip	nvarchar(20)	☑
LastInvoiceDate	numeric(10, 0)	☑
TotalPaid	numeric(10, 0)	☑

Column Name	Data Type	Allow Nulls
🔑 Records_ID	int	☐
SubContractor_ID	int	☑
ContractorName	nvarchar(50)	☑
ContractorType	nvarchar(50)	☑
Customer_ID	int	☑
CustomerName	nvarchar(50)	☑
StartDate	date	☑
EndDate	date	☑
InvoiceAmount	numeric(18, 0)	☑
		☐

Code:

Imports System.Data

Imports System.Data.SqlClient

Partial Class Final_Project_BigIdeasConstructionAccounting

Inherits System.Web.UI.Page

'con represents the websites connection to me personal database

Private Shared con As New SqlConnection("Data Source = cb-ot-devst06.ad.wsu.edu; initial catalog = MF13jake.norris; Persist Security Info = True; User ID = jake.norris; Password = a1963982")

'Here is the dataAdapters and their commandBuilder to attain information from the databases

Public Shared daSubContractors As New SqlDataAdapter("SELECT * FROM SubContractors", con)

Public Shared cbSubContractors As New SqlCommandBuilder(daSubContractors)

Public Shared daCustomers As New SqlDataAdapter("SELECT * FROM Customers", con)

Public Shared cbCustomers As New SqlCommandBuilder(daCustomers)

Public Shared daEmployees As New SqlDataAdapter("SELECT * FROM EmployeeArchiveMain", con)

Public Shared cbEmployees As New SqlCommandBuilder(daEmployees)

Public Shared daEmployeeLaborMain As New SqlDataAdapter("SELECT * FROM EmployeeLaborMain", con)

Public Shared cbEmployeeLaborMain As New SqlCommandBuilder(daEmployeeLaborMain)

Public Shared daRecords As New SqlDataAdapter("SELECT * FROM BigIdeasRecords", con)

```vb
Public Shared cbRecords As New SqlCommandBuilder(daRecords)

'These are the essential global dataTables
Public Shared gdtEmployee, gdtRecords, gdtSubContractor, gdtCustomer, gdtEmployeeLabor As
New DataTable

Protected Sub page_init(sender As Object, e As EventArgs) Handles Me.Init
'Fill the schema of the dataTables
daRecords.FillSchema(gdtRecords, SchemaType.Mapped)
daEmployeeLaborMain.FillSchema(gdtEmployeeLabor, SchemaType.Mapped)
daEmployees.FillSchema(gdtEmployee, SchemaType.Mapped)
daSubContractors.FillSchema(gdtSubContractor, SchemaType.Mapped)

'Set the image on the page to be the companies logo
Image1.Visible = True
Image1.ImageUrl = "IMG_1666.jpg"

Call updateDDL()
Call fillEmployeeChart()
Call fillSubContractorChart()

End Sub

'Update and initiate the DDL for website
Protected Sub updateDDL()
'Keep all accessing and manipulating of SQL servers inside Try/Catch block to prevent shutdown
Try
'Clear the rows of the dataTables to prevent overlapping information
gdtCustomer.Rows.Clear()
gdtEmployee.Rows.Clear()
gdtSubContractor.Rows.Clear()
'Fill tables with required information
daSubContractors.Fill(gdtSubContractor)
daCustomers.Fill(gdtCustomer)
```

```vbnet
daEmployees.Fill(gdtEmployee)
'Fill employee DDL with employee names and their respective primary key numbers
With DropDownList1
.DataSource = gdtEmployee
.DataTextField = "EmployeeName"
.DataValueField = "EmployeeArchive_ID"
.DataBind()
.Items.Insert(0, "Select Employee")
End With
'Fill subContractor DDL with contractor names and their respective primary key numbers
With DropDownList2
.DataSource = gdtSubContractor
.DataTextField = "ContractorName"
.DataValueField = "SubContractor_ID"
.DataBind()
.Items.Insert(0, "Select SubContractor")
End With
'Fill customer DDL with customer names and their respective primary key numbers
With DropDownList3
.DataSource = gdtCustomer
.DataTextField = "CustomerName"
.DataValueField = "Customer_ID"
.DataBind()
.Items.Insert(0, "Select Customer")
End With

Catch ex As Exception
Response.Write(ex.Message)
End Try


End Sub


'Button to enter employee payroll'
Protected Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
```

```vb
'Initialize the two varibale and timeSpan function needed to preform later tasks
Dim payRate, overtimeRate As New Integer
Dim ts As TimeSpan

gdtEmployeeLabor.Rows.Clear()
Dim dr As DataRow = gdtEmployeeLabor.NewRow
'SQL command to attain specific employee payRate
Dim cmdGrabPayRate As New SqlCommand("SELECT [PayRate] From EmployeeArchiveMain
WHERE [EmployeeArchive_ID] = @p1", con)
'Set command parameters to prevent SQL injection attack
With cmdGrabPayRate.Parameters
.Clear()
.AddWithValue("@p1", DropDownList1.SelectedItem.Value)
End With
'Keep all accessing and manipulating of SQL servers inside Try/Catch block to prevent shutdown
Try
'Since we are using a SQL command we have to open and close the connection
con.Open()
payRate = cmdGrabPayRate.ExecuteScalar()
Catch ex As Exception
Response.Write(ex.Message)
Finally
con.Close()
End Try

ts = DateTime.Parse(TextBox2.Text) - DateTime.Parse(TextBox1.Text)

Select Case Weekday(DateTime.Parse(TextBox3.Text))
Case 1 'Sunday
payRate *= 2
dr("Day") = "Sunday"
Case 2 To 6 'Weekday
dr("Day") = "Weekday"
Case 7 'Saturday
```

```
            payRate *= 1.5
            dr("Day") = "Saturday"
        End Select


            dr.Item("EmployeeName") = DropDownList1.SelectedItem.Text
            dr.Item("LaborDate") = DateTime.Parse(TextBox3.Text)
            dr.Item("Hours") = ts.TotalHours
            dr.Item("PayRate") = payRate
            dr.Item("Earned") = payRate * ts.TotalHours


            Dim cmdEmployeeArchiveUpdate As New SqlCommand("UPDATE [EmployeeArchiveMain]
    SET DaysWorked += 1, LastDayWorked = @p1, TotalEarned += @p2", con)


            With cmdEmployeeArchiveUpdate.Parameters
            .Clear()
            .AddWithValue("@p1", DateTime.Parse(TextBox3.Text))
            .AddWithValue("@p2", payRate * ts.TotalHours)
            End With


            gdtEmployeeLabor.Rows.Add(dr)


            Try
            con.Open()
            daEmployeeLaborMain.Update(gdtEmployeeLabor)
            cmdEmployeeArchiveUpdate.ExecuteNonQuery()
            Catch ex As Exception
            Response.Write(ex.Message)
            Finally
            con.Close()
            End Try
            Call EmployeeLaborGridViewUpdate()
        End Sub


        'Report work done by a Sub-contractor for a customer
```

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

        gdtRecords.Rows.Clear()
        Dim dr As DataRow = gdtRecords.NewRow()

        Dim cmdUpdateCustomers As New SqlCommand("UPDATE Customers SET LastInvoiceDate =
@p1, TotalPaid += @p2", con)
        Dim cmdUpdateSubContractors As New SqlCommand("UPDATE SubContractors SET
LastIncoiceDate = @p1, TotalPaid += @p2", con)
        Dim cmdGetContractorType As New SqlCommand("SELECT [ContractorType] FROM
SubContractors WHERE SubContractor_ID = @p1", con)
        With cmdUpdateCustomers.Parameters
        .Clear()
        .AddWithValue("@p1", Date.Parse(TextBox6.Text))
        .AddWithValue("@p2", Convert.ToDecimal(TextBox4.Text))
        End With
        With cmdUpdateSubContractors.Parameters
        .Clear()
        .AddWithValue("@p1", DateTime.Parse(TextBox6.Text))
        .AddWithValue("@p2", Convert.ToDecimal(TextBox4.Text))
        End With
        With cmdGetContractorType.Parameters
        .Clear()
        .AddWithValue("@p1", DropDownList2.SelectedItem.Value)
        End With

        dr.Item("SubContractor_ID") = DropDownList2.SelectedItem.Value
        dr.Item("ContractorName") = DropDownList2.SelectedItem.Text
        dr.Item("Customer_ID") = DropDownList3.SelectedItem.Value
        dr.Item("CustomerName") = DropDownList3.SelectedItem.Text
        dr.Item("StartDate") = DateTime.Parse(TextBox5.Text)
        dr.Item("EndDate") = DateTime.Parse(TextBox6.Text)
        dr.Item("InvoiceAmount") = Convert.ToDecimal(TextBox4.Text)
```

```vbnet
Try
con.Open()
dr("ContractorType") = cmdGetContractorType.ExecuteScalar()
gdtRecords.Rows.Add(dr)
daRecords.Update(gdtRecords)
cmdUpdateCustomers.ExecuteNonQuery()
cmdUpdateSubContractors.ExecuteNonQuery()
Catch ex As Exception
Response.Write(ex.Message)
Finally
con.Close()
End Try
Call RecordsGridViewUpdate()
Call SubContractorGridViewUpdate()
Call CustomerGridViewUpdate()
Call fillSubContractorChart()
End Sub

'Setup a new Employee
Protected Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click

gdtEmployee.Rows.Clear()
Dim dr As DataRow = gdtEmployee.NewRow()

dr.Item("EmployeeName") = TextBox7.Text
dr.Item("PayRate") = TextBox8.Text
dr.Item("HRCategory") = RadioButtonList1.SelectedItem.Text
dr.Item("DateEmployed") = DateTime.Parse(TextBox10.Text)

gdtEmployee.Rows.Add(dr)

Try
daEmployees.Update(gdtEmployee)
Catch ex As Exception
```

```vbnet
Response.Write(ex.Message)
End Try

Call EmployeeGridViewUpdate()
Call updateDDL()
End Sub

'Setup a new SubContractor
Protected Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click

gdtSubContractor.Rows.Clear()
Dim dr As DataRow = gdtSubContractor.NewRow()

dr.Item("ContractorName") = TextBox11.Text
dr.Item("ContractorType") = TextBox12.Text
dr.Item("BillingAddress") = TextBox13.Text
dr.Item("City") = TextBox14.Text
dr.Item("State") = TextBox15.Text
dr.Item("Zip") = TextBox16.Text

gdtSubContractor.Rows.Add(dr)
Try
daSubContractors.Update(gdtSubContractor)
Catch ex As Exception
Response.Write(ex.Message)
End Try
Call updateDDL()
Call SubContractorGridViewUpdate()
End Sub

'Setup a new Customer
Protected Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click

gdtCustomer.Rows.Clear()
```

```vbnet
        Dim dr As DataRow = gdtCustomer.NewRow()

        dr.Item("CustomerName") = TextBox17.Text
        dr.Item("Address") = TextBox18.Text
        dr.Item("City") = TextBox19.Text
        dr.Item("Zip") = TextBox20.Text

        gdtCustomer.Rows.Add(dr)

        Try
            daCustomers.Update(gdtCustomer)
        Catch ex As Exception
            Response.Write(ex.Message)
        End Try
        Call updateDDL()
        Call CustomerGridViewUpdate()
    End Sub

    Protected Sub EmployeeLaborGridViewUpdate()
        Try
            gdtEmployeeLabor.Rows.Clear()
            daEmployeeLaborMain.Fill(gdtEmployeeLabor)
            GridView7.DataSource = gdtEmployeeLabor
            GridView7.DataBind()
        Catch ex As Exception
            Response.Write(ex.Message)
        End Try
    End Sub

    'Fill the employeeSetup page gridView with all Employees
    Protected Sub EmployeeGridViewUpdate()

        Try
            gdtEmployee.Rows.Clear()
```

```vbnet
daEmployees.Fill(gdtEmployee)

GridView4.DataSource = gdtEmployee

GridView4.DataBind()

Catch ex As Exception

Response.Write(ex.Message)

End Try

End Sub


'Fill the subContractorSetup page gridView with all SubContractors

Protected Sub SubContractorGridViewUpdate()

Try

gdtSubContractor.Rows.Clear()

daSubContractors.Fill(gdtSubContractor)

GridView5.DataSource = gdtSubContractor

GridView5.DataBind()

Catch ex As Exception

Response.Write(ex.Message)

End Try

End Sub


'Fill the customerSetup page gridView with all Customers

Protected Sub CustomerGridViewUpdate()

Try

gdtCustomer.Rows.Clear()

daCustomers.Fill(gdtCustomer)

GridView6.DataSource = gdtCustomer

GridView6.DataBind()

Catch ex As Exception

Response.Write(ex.Message)

End Try

End Sub


Protected Sub RecordsGridViewUpdate()

Try
```

```
gdtRecords.Rows.Clear()

daRecords.Fill(gdtRecords)

GridView3.DataSource = gdtRecords

GridView3.DataBind()

Catch ex As Exception

Response.Write(ex.Message)

End Try

End Sub


'Fill the first bar chart on companyBreakdownCharts page

Protected Sub fillSubContractorChart()


Dim daFetchSubContractors As New SqlDataAdapter("SELECT * FROM SubContractors", con)

Dim dtSubContractors As New DataTable


Try

daFetchSubContractors.Fill(dtSubContractors)

Chart1.DataSource = dtSubContractors

Chart1.DataBind()

Chart1.Titles.Add("Total Cost By Contractor")


With Chart1.ChartAreas(0)

.AxisX.Title = "Contractor Name"

.AxisY.Title = "Total Cost"

End With


With Chart1.Series("Series1")

.ChartType = DataVisualization.Charting.SeriesChartType.Column

.XValueMember = "ContractorName"

.YValueMembers = "TotalPaid"

.IsValueShownAsLabel = True

End With

Catch ex As Exception

Response.Write(ex.Message)
```

```vb
        End Try
    End Sub


    'Fill the second char on companyBreakdownCharts page
    Protected Sub fillEmployeeChart()
        Dim daFetchEmployees As New SqlDataAdapter("SELECT * FROM EmployeeArchiveMain",
con)
        Dim dtEmployees As New DataTable

        Try
            daFetchEmployees.Fill(dtEmployees)
            Chart2.DataSource = dtEmployees
            Chart2.DataBind()
            Chart2.Titles.Add("Total Days Worked By Employee")

            With Chart2.ChartAreas(0)
                .AxisX.Title = "Employee Name"
                .AxisY.Title = "Total Days"
            End With

            With Chart2.Series("Series1")
                .ChartType = DataVisualization.Charting.SeriesChartType.Column
                .XValueMember = "EmployeeName"
                .YValueMembers = "DaysWorked"
                .IsValueShownAsLabel = True
            End With
        Catch ex As Exception
            Response.Write(ex.Message)
        End Try

    End Sub

    'Clear the EmployeePayroll form
    Protected Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
```

```vb
DropDownList1.SelectedIndex = -1

TextBox1.Text = Nothing

TextBox2.Text = Nothing

TextBox3.Text = Nothing

End Sub

'Clear the SubContractor Reporting form

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

DropDownList2.SelectedIndex = -1

DropDownList3.SelectedIndex = -1

TextBox4.Text = Nothing

TextBox5.Text = Nothing

TextBox6.Text = Nothing

End Sub

'Clear the Employee form

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

TextBox7.Text = Nothing

TextBox8.Text = Nothing

RadioButtonList1.SelectedIndex = -1

TextBox10.Text = Nothing

End Sub

'Clear the SubContractor form

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

TextBox11.Text = Nothing

TextBox12.Text = Nothing

TextBox13.Text = Nothing

TextBox14.Text = Nothing

TextBox15.Text = Nothing

TextBox16.Text = Nothing

End Sub

'Clear the CustomerSetup form

Protected Sub Button10_Click(sender As Object, e As EventArgs) Handles Button10.Click

TextBox17.Text = Nothing

TextBox18.Text = Nothing

TextBox19.Text = Nothing
```

```vbnet
        TextBox20.Text = Nothing
        TextBox21.Text = Nothing
    End Sub


    Protected Sub LinkButton1_Click(sender As Object, e As EventArgs) Handles LinkButton1.Click
        MultiView1.ActiveViewIndex = 0
        Call EmployeeLaborGridViewUpdate()
    End Sub


    Protected Sub LinkButton2_Click(sender As Object, e As EventArgs) Handles LinkButton2.Click
        MultiView1.ActiveViewIndex = 1
    End Sub


    Protected Sub LinkButton3_Click(sender As Object, e As EventArgs) Handles LinkButton3.Click
        MultiView1.ActiveViewIndex = 2
        Call RecordsGridViewUpdate()
    End Sub


    Protected Sub LinkButton4_Click(sender As Object, e As EventArgs) Handles LinkButton4.Click
        MultiView1.ActiveViewIndex = 3
        Call EmployeeGridViewUpdate()
    End Sub


    Protected Sub LinkButton5_Click(sender As Object, e As EventArgs) Handles LinkButton5.Click
        MultiView1.ActiveViewIndex = 4
        Call SubContractorGridViewUpdate()
    End Sub


    Protected Sub LinkButton6_Click(sender As Object, e As EventArgs) Handles LinkButton6.Click
        MultiView1.ActiveViewIndex = 5
        Call CustomerGridViewUpdate()
    End Sub
End Class
```