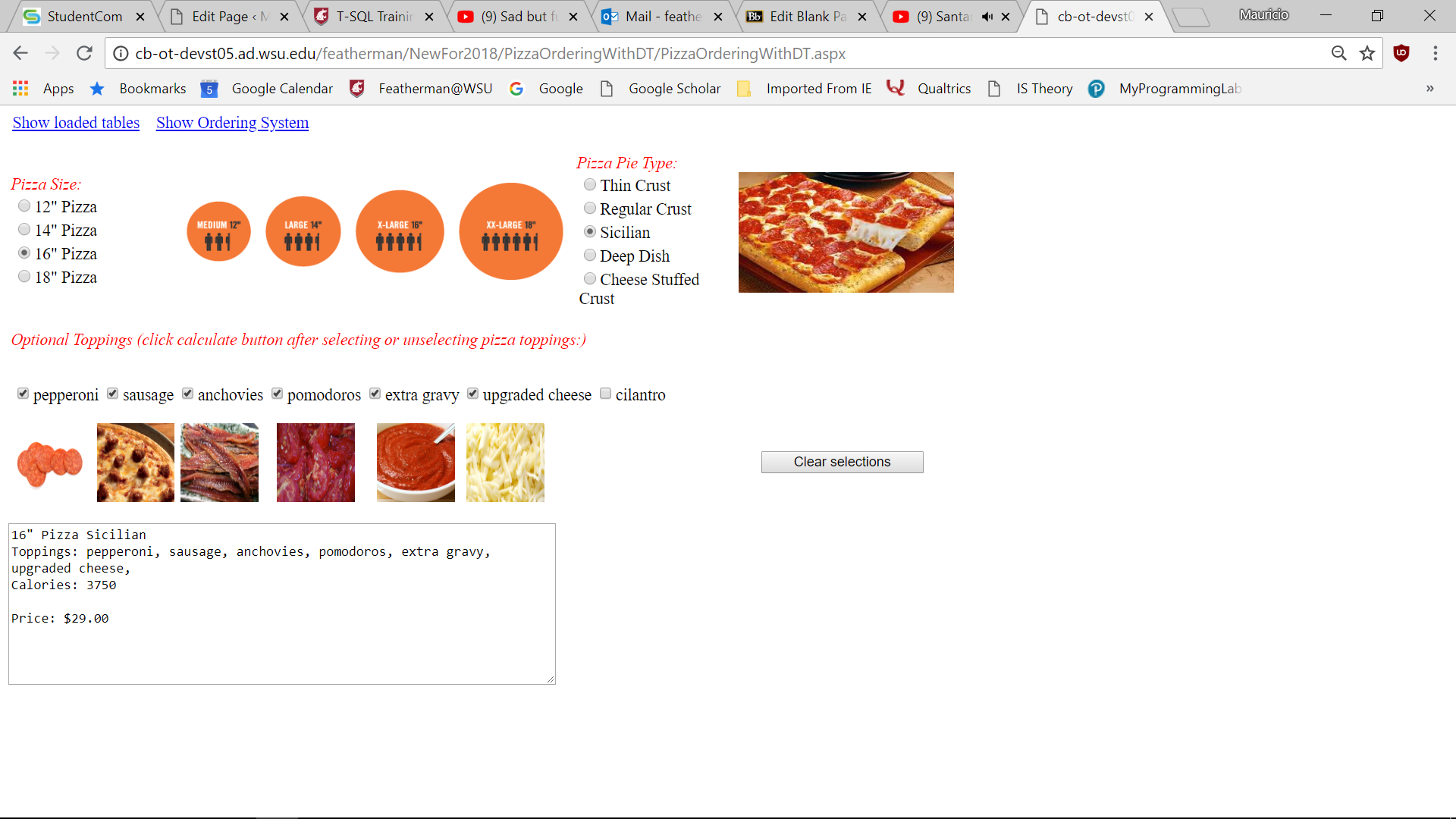
Loading data from a dataBase table into your project

<http://cb-ot-devst05.ad.wsu.edu/featherman/NewFor2018/PizzaOrderingWithDT/PizzaOrderingWithDT.aspx>



This program demonstrates that code for a webpage can be greatly simplified if the webpage has more data loaded into its RAM.   
  
Here the program loads three database tables into the memory of the webpage. The transaction code is simplified because the web page has lots of data loaded into it. The program can access the cost for different items and the calories for different items.

The program then makes the argument that the web developer needs to be competent with designing databases and accessing them for many different reasons and scenarios. Here the demonstration is that pulling more data into memory of the webpage GREATLY reduces the number of lines of code needed.

There are many other productivity enhancements that will be demonstrated in later programs.

Code:

Imports System.Data

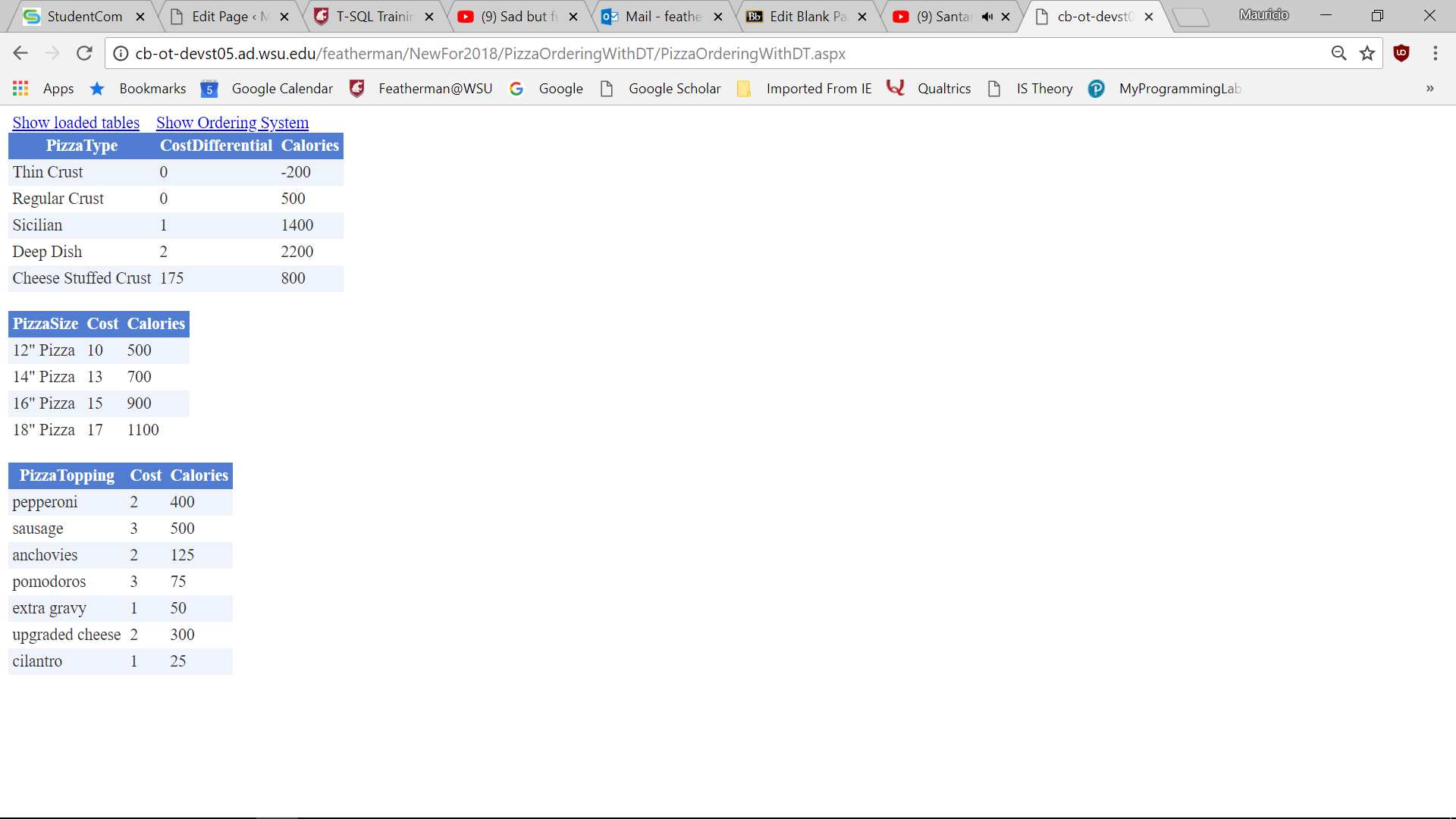
Imports System.Data.SqlClient

'this program calculates the total calories and the price for a shopkeeper to charge for a pizza pie. The program adds more dynamic interaction as the code runs when selections are made in the lists. A prior version of the program had a lot of hard coded prices for toppings and different pizza types and a lot of difficulty since three values needed to be input to the design

' a) textual size of pizza b) textual type of pizza c) textual name of pizza topping,

' b) number of calories for the pizza

'c) cost of different types of pizza



'the list controls can only 2 pieces of information (a .text and .value) not 3, so a lot of code was needed to complete the functionality. This program loads the three pieces of information needed from a database into the web page GREATLY simplifying the code needed.

Partial Class PizzaOrderingWithDT

Inherits System.Web.UI.Page

'this is a connection to a public SQL Server database that Featherman uses for teaching

Public Shared Con As New SqlConnection("Data Source=cb-ot-devst03.ad.wsu.edu;Initial Catalog=featherman\_analytics;Persist Security Info=True;User ID=mfstudent;Password=BIanalyst")

'Here are the 3 datatables that will have data loaded into them. There are 5 different pizz types each with different calories and price. There are 4 different pizza sizes each with different costs and calories. There are 7 different pizza toppings wach with different costs and calories.

Public Shared dtPizzaTypes As New DataTable

Public Shared dtPizzaSizes As New DataTable

Public Shared dtPizzaToppings As New DataTable

#Region "Load Datatables"

Private Sub PizzaOrderingWithDT\_Init(sender As Object, e As EventArgs) Handles Me.Init

'here we load the three tables of data into in-memory data tables. Basically SELECT all the columns and rows from each SQL Server table.

Dim daLoadPizzaTypes As New SqlDataAdapter("SELECT \* FROM featherman.PizzaTypes", Con)

Dim daLoadPizzaSizes As New SqlDataAdapter("SELECT \* FROM featherman.PizzaSizes", Con)

Dim daLoadPizzaToppings As New SqlDataAdapter("SELECT \* FROM featherman.PizzaToppings", Con)

'just in case this procedure had been run previously do not run it again! No need to double load the information

If dtPizzaTypes.Rows.Count > 0 Then Exit Sub

'whenever you make a database call, then put that call into a try catch bolock, to try the connection and interraction with the database. If there are any errors then gracefully catch the error and stop the program without crashing. Load the data from 3 database tables into 3 in RAM data structures

Try

daLoadPizzaTypes.Fill(dtPizzaTypes)

daLoadPizzaSizes.Fill(dtPizzaSizes)

daLoadPizzaToppings.Fill(dtPizzaToppings)

'load the retrieved data into gridview controls that are shown on one of the webpage tabs (views)

GridView1.DataSource = dtPizzaTypes

GridView2.DataSource = dtPizzaSizes

GridView3.DataSource = dtPizzaToppings

GridView1.DataBind()

GridView2.DataBind()

GridView3.DataBind()

'Now we tell the radiobutton lists to display the pizza type and sized from respective columns from the loaded in-memory datatables

With rblPizzaSizes

.DataSource = dtPizzaSizes

.DataTextField = "PizzaSize"

.DataBind()

End With

With rblPizzaTypes

.DataSource = dtPizzaTypes

.DataTextField = "PizzaType"

.DataBind()

End With

With chkToppings

.DataSource = dtPizzaToppings

.DataTextField = "PizzaTopping"

.DataBind()

End With

'catch any errors and show them at the top of the screen

Catch ex As Exception

Response.Write(ex.Message)

Exit Sub

End Try

End Sub

#End Region

Private Sub CalculateCalories()

'here we do most of the work. Notice this code is not attached to a button click event. This means the code is modularized and can be called or invoked from different places in your program. The different list controls call this code when the program user makes a selection in them (for example the radiobutton lists or the checkbox list will call and run this code below). This is called a general procedure.

'This code adds up the calories and prices for the kind of pizza, the size and the # of toppings. We build a description of the pizza and list of toppings for display and for calculations. A difficulty of this program is that we need to compile the sales price of the pizza and we need to compile the number calories.

'we will build strings then concatenate the strings to put into the textbox output.

Dim strToppings As String

Dim StrOutput As String

Dim decTotal As Decimal = 0

Dim decTotalCalories As Decimal = 0

'first some error checking to make sure a pizza size and pizza type are selected.

If rblPizzaSizes.SelectedIndex = -1 Then

txtOutput.Text = "Choose a pizza size"

Exit Sub

End If

If rblPizzaTypes.SelectedIndex = -1 Then

txtOutput.Text = "Choose a pizza size"

Exit Sub

End If

'count up the calories for the pizza size and pizza type. Take a look at the PizzaSizes table to understand the magic here. The item selected in the pizza size radiobutton list is indexed 0,1,2,3 etc. The code here says go to the matching row in the pizza sizes datatable that is selected in the radio button list and go over to the calories column and put that value into the decimal variable (same with the pizza types).

'read this dtPizzaSizes.Rows(rblPizzaSizes.SelectedIndex).Item("Calories") as

' datatable.rows(row #).item("column name)

decTotalCalories = dtPizzaSizes.Rows(rblPizzaSizes.SelectedIndex).Item("Calories")

decTotalCalories += dtPizzaTypes.Rows(rblPizzaTypes.SelectedIndex).Item("Calories")

'now just go over to a different column and pull athat cost into a decimal variable

'here calculate the price for the pizza based on pizza size and type

decTotal = dtPizzaSizes.Rows(rblPizzaSizes.SelectedIndex).Item("Cost")

decTotal += dtPizzaTypes.Rows(rblPizzaSizes.SelectedIndex).Item("CostDifferential")

'build the output text

StrOutput = rblPizzaSizes.SelectedItem.Text & rblPizzaTypes.SelectedItem.Text & " "

'Now we need to add calories and charge for any selected toppings, and grab the topping name for output. we need to examine each and every item in the items list of the checkbox. If the item is selected then increment the calories, cost, and the string output. this is some pretty sweet code!

For intx As Integer = 0 To chkToppings.Items.Count - 1

If chkToppings.Items(intx).Selected = True Then

decTotalCalories += dtPizzaToppings.Rows(intx).Item("Calories")

decTotal += dtPizzaToppings.Rows(intx).Item("Cost")

strToppings &= dtPizzaToppings.Rows(intx).Item("PizzaTopping") & ", "

End If

Next

txtOutput.Text = StrOutput & vbNewLine & "Toppings: " & strToppings & vbNewLine & "Calories: " & decTotalCalories & vbNewLine & vbNewLine & "Price: " & decTotal.ToString("C2")

End Sub

Protected Sub chkToppings\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles chkToppings.SelectedIndexChanged

'make sure program users select a pizza type and crust type. No use chosing toppings before the pizza type and size. We need the foundation before we take the toppings.

If rblPizzaSizes.SelectedIndex = -1 OrElse rblPizzaTypes.SelectedIndex = -1 Then

txtOutput.Text = "Select pizza size and type"

Exit Sub

End If

'show the pictures for the toppings if the item is selected

Image0.Visible = chkToppings.Items(0).Selected 'pepperoni

Image1.Visible = chkToppings.Items(1).Selected 'sausage

Image2.Visible = chkToppings.Items(2).Selected 'anchovies

Image3.Visible = chkToppings.Items(3).Selected 'pomodoros

Image4.Visible = chkToppings.Items(4).Selected 'extra sauce

Image5.Visible = chkToppings.Items(5).Selected 'upgraded cheese

Image6.Visible = chkToppings.Items(6).Selected 'cilantro

'calculate the price of the pizza now that the toppings are selected.

Call CalculateCalories()

End Sub

#Region "Pizza Type pictures"

Protected Sub RadioButtonList2\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles rblPizzaTypes.SelectedIndexChanged

'this program recalculates the calories and charge for the pizza and changes the picture when the progam user changes the style of pizza they want to order. The same image8 control will display a different picture depending on the item selected in the radiobuttonlist.

Select Case rblPizzaTypes.SelectedItem.Text

Case "Thin Crust"

Image8.ImageUrl = "thincrust.jpg"

Case "Regular Crust"

Image8.ImageUrl = "regularcrust.jpg"

Case "Sicilian"

Image8.ImageUrl = "sicilian.jpg"

Case "Deep Dish"

Image8.ImageUrl = "deepdish.jpg"

Case "Cheese Stuffed Crust"

Image8.ImageUrl = "cheese.jpg"

End Select

'different types of pizza have different prices so we call the procedure again.

Call CalculateCalories()

End Sub

#End Region

Protected Sub RadioButtonList1\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles rblPizzaSizes.SelectedIndexChanged

'if the program user changes the size of the size of pizza that they want to order then recalculate

Call CalculateCalories()

End Sub

Protected Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

'clear the radiobutton lists

rblPizzaSizes.SelectedIndex = -1

rblPizzaTypes.SelectedIndex = -1

'clear the toppings list

For Each li As ListItem In chkToppings.Items

li.Selected = False

Next

'clear the toppings pictures

Image0.Visible = False

Image1.Visible = False

Image2.Visible = False

Image3.Visible = False

Image4.Visible = False

Image5.Visible = False

Image6.Visible = False

txtOutput.Text = Nothing

End Sub

'The multiview has two views (tabs) so we need a way to switch between them

Protected Sub LinkButton1\_Click(sender As Object, e As EventArgs) Handles LinkButton1.Click

MultiView1.ActiveViewIndex = 1

End Sub

Protected Sub LinkButton2\_Click(sender As Object, e As EventArgs) Handles LinkButton2.Click

MultiView1.ActiveViewIndex = 0

End Sub

End Class