

Option Strict On

Imports Microsoft.VisualBasic

Imports System.Text

Imports System.Data.SqlClient

Public Class cart

```
'=====
Private ReadOnly Property shoppingCartId() As String
    Get
        Dim context As HttpContext = HttpContext.Current

        ' If the JASON-EXAMPLE_CartID cookie doesn't exist
        ' on client machine we create it with a new GUID
        If context.Request.Cookies("JASON-EXAMPLE_CartID") Is Nothing Then
            ' Generate a new GUID
            Dim cartId As Guid = Guid.NewGuid()

            ' Create the cookie and set its value
            Dim cookie As New HttpCookie("JASON-EXAMPLE_CartID", cartId.ToString)

            ' Current Date
            Dim currentDate As DateTime = DateTime.Now()

            ' Set the time span to 10 days
            Dim ts As New TimeSpan(10, 0, 0, 0)

            ' Expiration Date
            Dim expirationDate As DateTime = currentDate.Add(ts)

            ' Set the Expiration Date to the cookie
            cookie.Expires = expirationDate

            ' Set the cookie on client's browser
            context.Response.Cookies.Add(cookie)
        End If

        Return context.Request.Cookies("JASON-EXAMPLE_CartID").Value
    End If

End Get
End Property

'=====
'SALES TAX SUBJECT TO CHANGE _ BY STATE or BY LAW.
'The actual tax amount could come from a data table or web service
Public Function calculateTax(ByVal price As Decimal) As Decimal
    'Temp Decimal holder
    Dim tax As Decimal

    tax = CType((price) * 0.089, Decimal)

    Return tax 'returns sales tax decimal amount
End Function
'=====
```

```

Public Sub AddProduct(ByVal FkproductId As Integer)
    ' Create the connection object

    Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

    ' Create and initialize the command object
    Dim command As New SqlCommand("jg_AddProductToCart", connection)
    command.CommandType = Data.CommandType.StoredProcedure

    ' Add an input parameter and supply a value for it
    command.Parameters.Add("@PkCartID", Data.SqlDbType.VarChar, 50)
    command.Parameters("@PkCartID").Value = shoppingCartId

    ' Add an input parameter and supply a value for it
    command.Parameters.Add("@FkproductId", Data.SqlDbType.Int, 4)
    command.Parameters("@FkproductId").Value = FkproductId

    ' Open the connection, execute the command and close the connection
    connection.Open()
    command.ExecuteNonQuery()
    connection.Close()
End Sub

```

```

Public Sub UpdateProductQuantity(ByVal FkproductId As String, ByVal quantity As Integer)
    ' Create the connection object
    Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

    ' Create and initialize the command object
    Dim command As New SqlCommand("jg_UpdateCartItem", connection)
    command.CommandType = Data.CommandType.StoredProcedure

    ' Add an input parameter and supply a value for it
    command.Parameters.Add("@PkCartID", Data.SqlDbType.VarChar, 50)
    command.Parameters("@PkCartID").Value = shoppingCartId

    ' Add an input parameter and supply a value for it
    command.Parameters.Add("@FkproductId", Data.SqlDbType.Int, 4)
    command.Parameters("@FkproductId").Value = FkproductId

    ' Add an input parameter and supply a value for it
    command.Parameters.Add("@AmtQuantity", Data.SqlDbType.Int, 4)
    command.Parameters("@AmtQuantity").Value = quantity

    ' Open the connection, execute the command and close the connection
    connection.Open()
    command.ExecuteNonQuery()
    connection.Close()
End Sub

```

```

Public Sub UpdateProductQuantity(ByVal PkUserID As Integer, ByVal PkOrderId As Integer, ByVal
PkPartId As String, ByVal quantity As Integer)
    ' Create the connection object
    Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

    ' Create and initialize the command object
    Dim command As New SqlCommand("jg_UpdateCartLoggedIN", connection)
    command.CommandType = Data.CommandType.StoredProcedure

    ' Add an input parameter and supply a value for it
    command.Parameters.Add("@PkUserID", Data.SqlDbType.Int, 4)
    command.Parameters("@PkUserID").Value = PkUserID

```

```

' Add an input parameter and supply a value for it
command.Parameters.Add("@PkPartId", Data.SqlDbType.Int, 4)
command.Parameters("@PkPartId").Value = PkPartId

' Add an input parameter and supply a value for it
command.Parameters.Add("@PkOrderId", Data.SqlDbType.Int, 4)
command.Parameters("@PkOrderId").Value = PkOrderId

' Add an input parameter and supply a value for it
command.Parameters.Add("@AmtQuantity", Data.SqlDbType.Int, 4)
command.Parameters("@AmtQuantity").Value = quantity

' Open the connection, execute the command and close the connection
connection.Open()
command.ExecuteNonQuery()
connection.Close()
End Sub

Public Sub RemoveProduct(ByVal FkproductId As String)
' Create the connection object
Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

' Create and initialize the command object
Dim command As New SqlCommand("jg_RemoveProductFromCart", connection)
command.CommandType = Data.CommandType.StoredProcedure

' Add an input parameter and supply a value for it
command.Parameters.Add("@PkCartID", Data.SqlDbType.VarChar, 50)
command.Parameters("@PkCartID").Value = shoppingCartId

' Add an input parameter and supply a value for it
command.Parameters.Add("@FkproductId", Data.SqlDbType.Int, 4)
command.Parameters("@FkproductId").Value = FkproductId

' Open the connection, execute the command and close the connection
connection.Open()
command.ExecuteNonQuery()
connection.Close()
End Sub

Public Sub RemoveProduct(ByVal PkOrderID As Integer, ByVal FkPartId As Integer)
' Create the connection object
Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

' Create and initialize the command object
Dim command As New SqlCommand("jg_RemoveProductFromCartLoggedIn", connection)
command.CommandType = Data.CommandType.StoredProcedure

' Add an input parameter and supply a value for it
command.Parameters.Add("@PkOrderID", Data.SqlDbType.Int, 4)
command.Parameters("@PkOrderID").Value = PkOrderID

' Add an input parameter and supply a value for it
command.Parameters.Add("@FkPartId", Data.SqlDbType.Int, 4)
command.Parameters("@FkPartId").Value = FkPartId

' Open the connection, execute the command and close the connection
connection.Open()
command.ExecuteNonQuery()
connection.Close()
End Sub

```

```

Public Function GetProducts() As SqlDataReader
    ' Create the connection object
    Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

    ' Create and initialize the command object
    Dim command As New SqlCommand("jg_GetShoppingCartProducts", connection)
    command.CommandType = Data.CommandType.StoredProcedure

    ' Add an input parameter and supply a value for it
    command.Parameters.Add("@PkCartID", Data.SqlDbType.VarChar, 50)
    command.Parameters("@PkCartID").Value = shoppingCartId

    ' Return the results
    connection.Open()
    Return command.ExecuteReader(Data.CommandBehavior.CloseConnection)
End Function

```

```

Public Function GetProductsLoggedIN(ByVal PkOrderID As Integer) As SqlDataReader
    ' Create the connection object
    Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

    ' Create and initialize the command object
    Dim command As New SqlCommand("jg_GetLoggedInShoppingCartProducts", connection)
    command.CommandType = Data.CommandType.StoredProcedure

    ' Add an input parameter and supply a value for it
    command.Parameters.Add("@PkOrderID", Data.SqlDbType.Int, 4)
    command.Parameters("@PkOrderID").Value = PkOrderID

    ' Return the results
    connection.Open()
    Return command.ExecuteReader(Data.CommandBehavior.CloseConnection)
End Function

```

```

Public Function AdminGetProductsByOrderID(ByVal PkOrderID As Integer) As SqlDataReader
    ' Create the connection object
    Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

    ' Create and initialize the command object
    Dim command As New SqlCommand("jg_AdminShoppingCartProducts", connection)
    command.CommandType = Data.CommandType.StoredProcedure

    ' Add an input parameter and supply a value for it
    command.Parameters.Add("@PkOrderID", Data.SqlDbType.Int, 4)
    command.Parameters("@PkOrderID").Value = PkOrderID

    ' Return the results
    connection.Open()
    Return command.ExecuteReader(Data.CommandBehavior.CloseConnection)
End Function

```

```

Public Function GetProductsEmailer() As String
    ' Create the connection object
    Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

    ' Create and initialize the command object
    Dim command As New SqlCommand("jg_GetShoppingCartProducts", connection)
    command.CommandType = Data.CommandType.StoredProcedure

```

```

' Add an input parameter and supply a value for it
command.Parameters.Add("@PkCartID", Data.SqlDbType.VarChar, 50)
command.Parameters("@PkCartID").Value = ShoppingCartId

' Return the results
connection.Open()
'Create a datareader to read results
Dim dr As SqlDataReader = command.ExecuteReader
'Create a String to build for shopping cart items
Dim builder As StringBuilder = New StringBuilder
'Calling the constructor objects to build String Properties from
'Data reader - Passing in data reader
Dim orderdeatils As New orderdetails(dr)
builder.Append(orderdeatils.ListAsString)

dr.Close()
dr.Dispose()
connection.Close()
connection.Dispose()

Return builder.ToString
End Function

Public Function GetProductsEmailer(ByVal PkOrderID As Integer) As String
' Create the connection object
Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

' Create and initialize the command object
Dim command As New SqlCommand("jg_GetLoggedInShoppingCartProducts", connection)
command.CommandType = Data.CommandType.StoredProcedure

' Add an input parameter and supply a value for it
command.Parameters.Add("@PkOrderID", Data.SqlDbType.Int, 4)
command.Parameters("@PkOrderID").Value = PkOrderID

' Return the results
connection.Open()
'Create a datareader to read results
Dim dr As SqlDataReader = command.ExecuteReader
'Create a String to build for shopping cart items
Dim builder As StringBuilder = New StringBuilder
'Calling the constructor objects to build String Properties from
'Data reader - Passing in data reader
Dim orderdeatils As New orderdetails(dr)
builder.Append(orderdeatils.ListAsString)

dr.Close()
dr.Dispose()
connection.Close()
connection.Dispose()

Return builder.ToString
End Function

Public Function AdminGetCompletedOrderProducts(ByVal PkOrderID As Integer) As String
' Create the connection object
Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

' Create and initialize the command object
Dim command As New SqlCommand("jg_AdminSummaryShoppingCartProducts", connection)
command.CommandType = Data.CommandType.StoredProcedure

```

```

' Add an input parameter and supply a value for it
command.Parameters.Add("@PkOrderID", Data.SqlDbType.Int, 4)
command.Parameters("@PkOrderID").Value = PkOrderID

' Return the results
connection.Open()
'Create a datareader to read results
Dim dr As SqlDataReader = command.ExecuteReader
'Create a String to build for shopping cart items
Dim builder As StringBuilder = New StringBuilder
'Calling the constructor objects to build String Properties from
'Data reader - Passing in data reader
Dim orderdeatils As New orderdetails(dr)
builder.Append(orderdeatils.ListAsString)

dr.Close()
dr.Dispose()
connection.Close()
connection.Dispose()

Return builder.ToString
End Function

Public Function GetTotalAmount() As Decimal
' Create the connection object
Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

' Create and initialize the command object
Dim command As SqlCommand = New SqlCommand("jg_GetTotalAmount", connection)
command.CommandType = Data.CommandType.StoredProcedure

' Add an input parameter and supply a value for it
command.Parameters.Add("@PkCartID", Data.SqlDbType.VarChar, 50)
command.Parameters("@PkCartID").Value = shoppingCartId

' Save the total amount to a variable
Dim amount As Decimal
connection.Open()
amount = CType(command.ExecuteScalar(), Decimal)

' Close the connection
connection.Close()

' Return the amount
Return amount
End Function

Public Function GetTotalAmount(ByVal orderID As Integer) As Decimal
' Create the connection object
Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

' Create and initialize the command object
Dim command As SqlCommand = New SqlCommand("jg_GetTotalAmountloggedIn", connection)
command.CommandType = Data.CommandType.StoredProcedure

' Add an input parameter and supply a value for it
command.Parameters.Add("@PkorderID", Data.SqlDbType.BigInt, 8)
command.Parameters("@PkorderID").Value = orderID

' Save the total amount to a variable
Dim amount As Decimal
connection.Open()
amount = CType(command.ExecuteScalar(), Decimal)

```

```

' Close the connection
connection.Close()

' Return the amount
Return amount
End Function

Sub ClearCart()
'Clear the cart from the DB After a customer orders the products
Dim Conn As New SqlConnection(JASON-EXAMPLEConnStrSQL)
Dim cmd As New SqlCommand
cmd.CommandText = "DELETE FROM tblShoppingCart WHERE PkCartID='" & shoppingCartId & "'"
cmd.Connection = Conn

Conn.Open()
cmd.ExecuteNonQuery()

Conn.Close()
Conn.Dispose()
End Sub

Public Function AddProduct(ByVal PkPartID As Integer, ByVal PkUserID As Integer, ByVal PkOrderID As
Integer) As Integer
' Create the connection object
Dim Conn As New SqlConnection(JASON-EXAMPLEConnStrSQL)

' Create and initialize the command object
Dim command As New SqlCommand("jg_AddProductToCartLoggedIn", Conn)
command.CommandType = Data.CommandType.StoredProcedure

'' Add an input parameter and supply a value for it
'command.Parameters.Add("@PkOrderID", Data.SqlDbType.Int)
'command.Parameters("@PkOrderID").Value = PkOrderID

' Add an input parameter and supply a value for it
command.Parameters.Add("@PkUserID", Data.SqlDbType.Int)
command.Parameters("@PkUserID").Value = PKUserID

' Add an input parameter and supply a value for it
command.Parameters.Add("@PkPartId", Data.SqlDbType.Int, 4)
command.Parameters("@PkPartId").Value = PkPartID

If PkOrderID < 10 Then
    PkOrderID = 0
End If

' Add an input parameter and supply a value for it
command.Parameters.Add("@PkOrderId", Data.SqlDbType.Int, 4)
command.Parameters("@PkOrderId").Value = PkOrderID

Dim OrderID As New SqlParameter("@NewPkOrderID", Data.SqlDbType.Int)
' flag it as an output paramater
OrderID.Direction = Data.ParameterDirection.Output
' add it to the collection, which now
' contains 3 parameters
command.Parameters.Add(OrderID) 'AS ORDER NUMBER

' Open the connection, execute the command and close the connection
Conn.Open()
command.ExecuteNonQuery()
Conn.Close()

Return CType(OrderID.Value, Integer)

```

End Function

```
Public Function CreateOrder(ByVal CustomerID As Integer) As Integer
    Dim orderId As Integer
    ' Create the connection object
    Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)
    'Check out customer open cart status
    If CType(CheckCustomer(CustomerID), Boolean) Then
        'orderId = Mergecart(CustomerID)
        'empty cookie based shopping cart
        Dim cmd As New SqlCommand("jg_EmptyShoppingCart", connection)
        cmd.CommandType = Data.CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        cmd.Parameters.Add("@PkCartID", Data.SqlDbType.VarChar, 50)
        cmd.Parameters("@PkCartID").Value = shoppingCartId

        connection.Open()
        cmd.ExecuteNonQuery()
        connection.Close()
        connection.Dispose()

    Else
        ' Create and initialize the command object
        Dim command As New SqlCommand("jg_CreateOrder", connection)
        command.CommandType = Data.CommandType.StoredProcedure

        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@PkCartID", Data.SqlDbType.VarChar, 50)
        command.Parameters("@PkCartID").Value = shoppingCartId

        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@FkCustID", Data.SqlDbType.BigInt, 8)
        command.Parameters("@FkCustID").Value = CustomerID

        ' Save the value that needs to be returned to a variable

        connection.Open()
        orderId = CType(command.ExecuteScalar(), Integer)

        ' Close the connection
        connection.Close()

        ' Return the saved value

    End If
    Return orderId
End Function

Public Function CheckCustomer(ByVal CustomerID As Integer) As Boolean
    Dim Answer As Boolean
    Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)

    ' Create and initialize the command object
    Dim command As New SqlCommand("jg_OpenCart", connection)
    command.CommandType = Data.CommandType.StoredProcedure

    ' Add an input parameter and supply a value for it
    command.Parameters.Add("@CustID", Data.SqlDbType.BigInt, 8)
    command.Parameters("@CustID").Value = CustomerID

    connection.Open()
    Answer = CType(command.ExecuteScalar(), Boolean)

    ' Close the connection
```

```
connection.Close()
connection.Dispose()
' Return the saved value
```

```
Return Answer
```

```
End Function
```

```
Public Function FillCustInfo(ByVal PkCustID As Integer) As UserCheckOut
```

```
'-----
'get Logged in customer info to Fill the Text Boxes
```

```
'=====
```

```
Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)
```

```
' Create and initialize the command object
```

```
Dim command As New SqlCommand("jg_GetCheckOutInfo", connection)
command.CommandType = Data.CommandType.StoredProcedure
```

```
' Add an input parameter and supply a value for it
```

```
command.Parameters.Add("@PkUserID", Data.SqlDbType.Int, 4)
```

```
command.Parameters("@PkUserID").Value = PkCustID
```

```
connection.Open()
```

```
Dim UCheckOutInfo As New UserCheckOut
```

```
Dim dr As SqlDataReader
```

```
dr = command.ExecuteReader
```

```
If dr.HasRows Then
```

```
dr.Read()
```

```
UCheckOutInfo.PkUserID = PkCustID
```

```
UCheckOutInfo.Company = dr("BusinessName").ToString
```

```
UCheckOutInfo.FName = dr("FName").ToString
```

```
UCheckOutInfo.LName = dr("LName").ToString
```

```
UCheckOutInfo.ShippingAddress = dr("Address").ToString
```

```
UCheckOutInfo.ShippingCity = dr("City").ToString
```

```
UCheckOutInfo.ShippingState = dr("State").ToString
```

```
UCheckOutInfo.ShippingZip = dr("Zipcode").ToString
```

```
UCheckOutInfo.ShippingCountry = dr("Country").ToString
```

```
UCheckOutInfo.Phone = dr("Phone").ToString
```

```
UCheckOutInfo.Extention = dr("Extention").ToString
```

```
UCheckOutInfo.Email = dr("Email").ToString
```

```
UCheckOutInfo.BillingName = dr("BillingName").ToString
```

```
UCheckOutInfo.BillingAddress = dr("BillingAddress").ToString
```

```
UCheckOutInfo.BillingCity = dr("BillingCity").ToString
```

```
UCheckOutInfo.BillingState = dr("BillingState").ToString
```

```
UCheckOutInfo.BillingZip = dr("BillingZipcode").ToString
```

```
UCheckOutInfo.BillingCountry = dr("BillingCountry").ToString
```

```
End If
```

```
'Return the results into the Public Structure
```

```
Return UCheckOutInfo
```

```
End Function
```

```
Public Function ToDB() As ArrayList
```

```
' Create the connection object
```

```
Dim connection As New SqlConnection(JASON-EXAMPLEConnStrSQL)
```

```
' Create and initialize the command object
```

```
Dim command As New SqlCommand("jg_GetShoppingCartProducts", connection)
```

```
command.CommandType = Data.CommandType.StoredProcedure
```

```
Dim SCart As String = shoppingCartId
```

```
' Add an input parameter and supply a value for it
```

```
command.Parameters.Add("@PkCartID", Data.SqlDbType.VarChar, 50)
```

```

command.Parameters("@PkCartID").Value = SCart

' Return the results
connection.Open()

Dim TempCart As New ArrayList

Dim dr As SqlDataReader
dr = command.ExecuteReader
While dr.Read

    Dim cp As New CartProduct
    cp.PkCartID = SCart
    cp.FkPartID = CType(dr("PkPartId"), Integer)
    cp.PartNumber = CType(dr("PartNumber"), String)
    cp.Price = CType(dr("Price"), Decimal)
    cp.AmtQuantity = CType(dr("AmtQuantity"), Integer)
    TempCart.Add(cp)

End While

Return TempCart

End Function
Public Function CustLoginGetOrder(ByVal loggedIn As User) As User

Dim conn As New SqlConnection(JASON-EXAMPLEConnStrSQL)
'Create Command
Dim cmd As New SqlCommand("jg_CustLoginGetOrder", conn)
' Specify Command Type
cmd.CommandType = Data.CommandType.StoredProcedure
' Add an input parameter and supply a value for it
cmd.Parameters.Add("@PkUserID", Data.SqlDbType.Int)
cmd.Parameters("@PkUserID").Value = loggedIn.PkUserID

'Try
conn.Open()
'This auto closes the connection when you close the data reader
'Look for CommandBehavior.SingleRow and .SingleResult
Dim dr As SqlDataReader = cmd.ExecuteReader
'To get an immediate test result to see if information was read after the query
If dr.HasRows = True Then
    'The user must exist
    dr.Read()
    '=====
    loggedIn.Completed = CType(dr("Completed"), Boolean)
    If loggedIn.Completed = False Then
        loggedIn.OrderNumber = CType(dr("PkOrderID"), Integer)
    End If
    '=====
    'RETURN THE STRUCTURE
    'Context.Session("Customer") = loggedIn

End If

Return loggedIn
End Function

Public Function GetIDByEmail(ByVal Email As String) As Integer
'===== QUICK CHECK OUT ID RETURN =====
Dim UserId As Integer

Dim Conn As New SqlConnection(JASON-EXAMPLEConnStrSQL)
'Here we call a SQL SERVER stored procedure, name begins with jg_

```

```

Dim cmd As SqlCommand = New SqlCommand("jg_GetIdByEmail", Conn)
cmd.CommandType = Data.CommandType.StoredProcedure

' Add an input parameter and supply a value for it
cmd.Parameters.Add("@Email", Data.SqlDbType.VarChar)
cmd.Parameters("@Email").Value = Email
'-----
'OUTPUT RETURNING THE DISCOUNT STATUS - true/false
Dim PkUserID As New SqlParameter("@PkUserID", Data.SqlDbType.Int)
cmd.Parameters.Add(PkUserID)
cmd.Parameters("@PkUserID").Direction = Data.ParameterDirection.Output

Conn.Open()
Try
    cmd.ExecuteNonQuery()
    UserId = CType(PkUserID.Value, Integer)
Catch ex As Exception
    UserId = -1
End Try

Conn.Close()
Conn.Dispose()

Return UserId
End Function

'Returns BOOLEAN for part discount eligible (PkPartID) returns True/False
Public Function GetPartsDiscountedTotal(ByVal PkOrderID As Integer) As Decimal

Dim Conn As New SqlConnection(JASON-EXAMPLEConnStrSQL)
'Here we call a SQL SERVER stored procedure, name begins with jg_
Dim cmd As SqlCommand = New SqlCommand("jg_GetPartsDiscountedTotal", Conn)
cmd.CommandType = Data.CommandType.StoredProcedure

' Add an input parameter and supply a value for it
cmd.Parameters.Add("@PkOrderID", Data.SqlDbType.Int, 4)
cmd.Parameters("@PkOrderID").Value = PkOrderID
'-----
'OUTPUT RETURNING THE DISCOUNT STATUS - true/false
Dim DiscountTotal As New SqlParameter("@DiscountTotal", Data.SqlDbType.Decimal)
cmd.Parameters.Add(DiscountTotal)
cmd.Parameters("@DiscountTotal").Direction = Data.ParameterDirection.Output

Conn.Open()

cmd.ExecuteNonQuery()

Dim SubTotal As Decimal = CType(DiscountTotal.Value, Decimal)

Conn.Close()
Conn.Dispose()
'return the value
Return SubTotal

End Function

Public Function completeOrderCheckOut(ByVal cco As completeCheckOut) As String

'<default Result>
Dim result As String = "fail"
' Create the connection object
Dim Conn As New SqlConnection(JASON-EXAMPLEConnStrSQL)

' Create and initialize the command object

```

```

Dim command As New SqlCommand("jg_CustomerOrderCheckOut", Conn)
command.CommandType = Data.CommandType.StoredProcedure

' Add an input parameter and supply a value for it
command.Parameters.Add("@PkOrderID", Data.SqlDbType.Int)
command.Parameters("@PkOrderID").Value = cco.PkOrderID

' Add an input parameter and supply a value for it
command.Parameters.Add("@ShippingCost", Data.SqlDbType.Money)
command.Parameters("@ShippingCost").Value = cco.ShippingCost

' Add an input parameter and supply a value for it
command.Parameters.Add("@ShippingPreference", Data.SqlDbType.VarChar)
command.Parameters("@ShippingPreference").Value = cco.ShippingPreference

' Add an input parameter and supply a value for it
command.Parameters.Add("@SubTotal", Data.SqlDbType.Money)
command.Parameters("@SubTotal").Value = cco.SubTotal

' Add an input parameter and supply a value for it
command.Parameters.Add("@DiscountCode", Data.SqlDbType.VarChar)
command.Parameters("@DiscountCode").Value = cco.DiscountCode

' Add an input parameter and supply a value for it
command.Parameters.Add("@DiscountAmt", Data.SqlDbType.Money)
command.Parameters("@DiscountAmt").Value = cco.DiscountAmt

' Add an input parameter and supply a value for it
command.Parameters.Add("@tax", Data.SqlDbType.Money)
command.Parameters("@tax").Value = cco.tax

' Add an input parameter and supply a value for it
command.Parameters.Add("@TotalPrice", Data.SqlDbType.Money)
command.Parameters("@TotalPrice").Value = cco.TotalPrice

' Add an input parameter and supply a value for it
command.Parameters.Add("@AuthCode", Data.SqlDbType.VarChar)
command.Parameters("@AuthCode").Value = cco.AuthCode

' Add an input parameter and supply a value for it
command.Parameters.Add("@Reference", Data.SqlDbType.VarChar)
command.Parameters("@Reference").Value = cco.Reference

' Add an input parameter and supply a value for it
command.Parameters.Add("@CustomerNotes", Data.SqlDbType.VarChar)
command.Parameters("@CustomerNotes").Value = cco.CustomerNotes

' Add an input parameter and supply a value for it
command.Parameters.Add("@BillingInfo", Data.SqlDbType.VarChar)
command.Parameters("@BillingInfo").Value = cco.BillingInfo

' Add an input parameter and supply a value for it
command.Parameters.Add("@EndingIn", Data.SqlDbType.VarChar)
command.Parameters("@EndingIn").Value = cco.EndingIn

```

Try

```

' Open the connection, execute the command and close the connection
Conn.Open()
command.ExecuteNonQuery()
Conn.Close()
' The command succeeded
result = "success"

```

Catch ex As Exception

```

        result = ex.ToString

    Finally
        Conn.Close()
        Conn.Dispose()

    End Try

    Return result
End Function
End Class

Public Structure OrderItem
    Public ID, QTY As Integer
    Public price As Decimal
    Public ProdName As String
    Public Descript As String
    Public Subtotal As String
End Structure

Public Structure CartProduct
    Dim PkCartID As String
    Dim FkPartID As Integer
    Dim AmtQuantity As Integer
    Dim PartNumber As String
    Dim Price As Decimal
End Structure

Public Structure completeCheckOut
    Dim PkOrderID As Integer
    Dim ShippingCost As Decimal
    Dim ShippingPreference As String
    Dim SubTotal As Decimal
    Dim DiscountCode As String
    Dim DiscountAmt As Decimal
    Dim tax As Decimal
    Dim TotalPrice As Decimal
    Dim AuthCode As String
    Dim Reference As String
    Dim CustomerNotes As String
    Dim BillingInfo As String 'html of checkout info
    Dim EndingIn As String
End Structure

```