Gillian Meshot and Andrew Vernon

**M&V Car Rentals**

<http://cb-ot-devst05.ad.wsu.edu/MF11andrew.vernon/final_proj.aspx>

**Customer:**

The customer is a rental car company, M&V Car Rentals. The company's main service is renting cars to customers. With this service, the company also has other operations that follow being a rental car company. These other operations are the maintenance of the fleet of rental cars, purchasing new cars, and customer relations.

**Problem:**

Car rental companies need a website that allows for easy facilitation of business and operations. The website needs to encompass all aspects of the business in order to perform all functions of the business.

Firstly, the company needs a way to store and retrieve car rental information. With this need, the site will have to retrieve car rental options based on the selections made by the user. Also, past rental information should be able to be retrieved based on a specific customer.

The next need is storing and retrieval of customer information. The company needs a way to store new customer information and retrieve past customer information. Retrieving past customer information should also bring up associated rental information.

Lastly, the company needs a way to facilitate maintenance of their fleet and fleet information. This includes being able to add new car brands, car models and specific cars to their fleet. This includes tracking which cars need what parts.

**Solution:**

Personalized site for a car rental company to store information and retrieve business and operations information. The site includes many aspects that provide different functions for the business.

The first and most important solution is the storing and retrieval of car rental information. The site allows for new car rental information to be inputted and the retrieval of past rental information. Past location information can be retrieved using a CustomerID or CarID.

The next solution is storing and retrieval of customer information. The site allows for new customer information to be inputted. When a customer is added, they are automatically assigned an ID for retrieval of their personal information. This allows for past rental information associated with the customer to be retrieved.

The last solution is storing and retrieval of fleet information. The site allows new cars to be added to the fleet and new car models can be added to the brand. Lastly, the user is able to input the part which allows for maintenance information for cars to be stored.

**Derived benefits:**

This solution provides the company with the added benefit of being able to have a site where all business and operations can be facilitated. With a personalized site that centers around the needs of the company, M&V will be able to increase the efficiency and effectiveness of their business. This is due to the fact that all business operations, like retrieving customer information or rental information, will be in one location, the website.

**Overview of the functionality of each web page and procedure**

**View 1:**

The first view functions to record car rental information. The user will begin by selecting the car brand which filters specific models into the car model dropdownlist. For example, if Subaru is selected, all car models associated with Subaru will be listed. After a car brand and model is selected, the list will filter again to show the cars available for that model. The car models are numbered to track availability and usage information. When a specific car is selected, the starting mileage and color are automatically filled in to correspond to the data in SQL. When selecting Customer Name and the VIP checkbox, the fields are also automatically filled in to correspond to the data in SQL. There are various check-boxes to record if a car needs maintenance.

**View 2:**

The second view functions for ordering more car units. When a new car is purchased, the user can input the car information, like color, brand, and model, to be stored. When the information is inputted the car automatically is assigned an ID. Allows for the retrieval of information based on the unique ID in View 1.

**View 3:**

View 3 functions for adding more car model options. When the company expands their models options for any brand this view allows for their model information to be stored for later retrieval.

**View 4:**

View 4 functions for creating a new customer and reviewing past customer information. Users will be able to enter new customer information to be stored for later retrieval and are automatically assigned a CustomerID. This customer information is used in View 1 when recording car rental information and retrieving rental information. Past customer information and rental history can be viewed after selecting the Customer ID.

**Testing Instructions:**

View 1:

Begin by filling out the fields for a car rental with easy to remember information. Then, below, to view the inputted rental information prior, use the rental number or customer id populated in the dropdown list.

View 2:

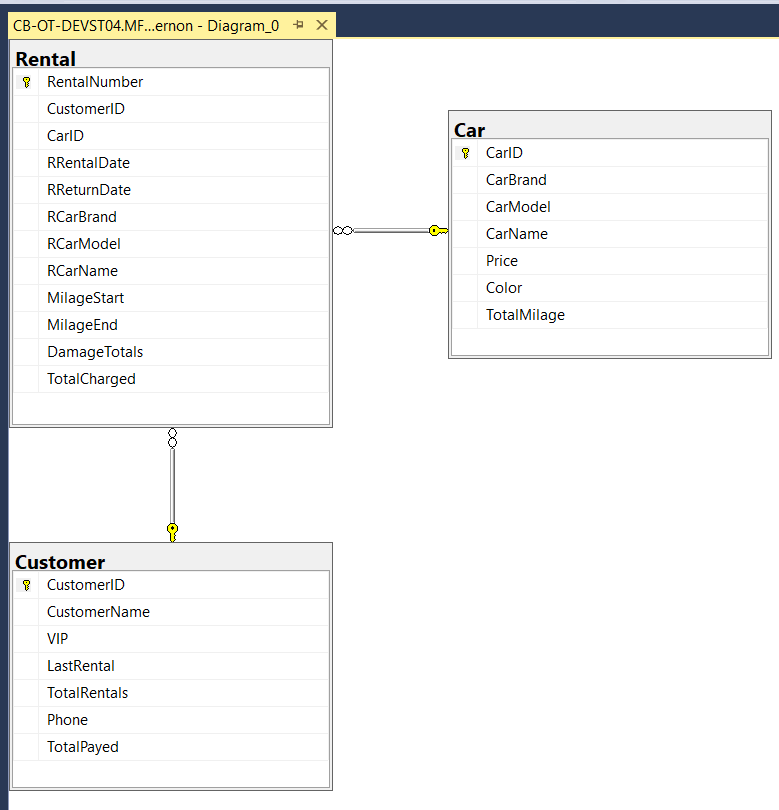
Input your favorite car or your dream rental car. Then to test if the information was stored, go back to view 1. Then review the Car Brand dropdownlist to determine if it is loaded and stored.

View 3:

Similar to view 2, input your favorite car or your dream rental car. Then to test if the information was stored, go back to view 1. Then review the Car Model dropdownlist to determine if it is loaded and stored. Below, enter new information to update the ca

View 4:

Enter customer information. To test if the information was stored, go back to view one to the Customer ID dropdownlist. Below, select the customer you just created and update their information.



**Appendix**

**Imports System.Data**

**Imports System.Data.SqlClient**

**Partial Class final\_proj**

**Inherits System.Web.UI.Page**

**#Region "declare variables"**

**Public Shared Con As New SqlConnection("Data Source=cb-ot-devst04.ad.wsu.edu;Initial Catalog = MF11andrew.vernon; Persist Security Info=True;User ID =andrew.vernon; Password=f6f15df8")**

**Public Shared damageTotals As Decimal**

**Public Shared totalCharged As Decimal**

**Public Shared dtPrice As DataTable**

**Public Shared daNewRental As New SqlDataAdapter("Use [MF11andrew.vernon] Select \* From [Rental]", Con)**

**Public Shared cbNewRental As New SqlCommandBuilder(daNewRental)**

**Public Shared dtRental As New DataTable**

**Public Shared daNewCustomer As New SqlDataAdapter("Use [MF11andrew.vernon] Select \* From [Customer]", Con)**

**Public Shared cbNewCustomer As New SqlCommandBuilder(daNewCustomer)**

**Public Shared dtCustomer As New DataTable**

**Public Shared daNewModel As New SqlDataAdapter("Use [MF11andrew.vernon] Select \* From [Car]", Con)**

**Public Shared cbNewModel As New SqlCommandBuilder(daNewModel)**

**Public Shared dtNewModel As New DataTable**

**Public Shared daNewCar As New SqlDataAdapter("Use [MF11andrew.vernon] Select \* From [Car]", Con)**

**Public Shared cbNewCar As New SqlCommandBuilder(daNewCar)**

**Public Shared dtCarName As New DataTable**

**Public Shared dtCar As New DataTable**

**Public Shared dtCarModel As New DataTable**

**Public Shared gdaGetRentalNum As New SqlDataAdapter("Use [MF11andrew.vernon] SELECT \* FROM [Rental] WHERE [CustomerID] = @p1", Con)**

**Public Shared cmdGetRentalNum As New SqlCommandBuilder(gdaGetRentalNum)**

**Public Shared gdaGetCustomerID As New SqlDataAdapter("Use [MF11andrew.vernon] SELECT \* FROM [Customers] WHERE [CustomerID] = @p1", Con)**

**Public Shared cmdGetCustomerID As New SqlCommandBuilder(gdaGetCustomerID)**

**Public Shared gdaGetCarInfo As New SqlDataAdapter("Use [MF11andrew.vernon] SELECT \* FROM [Car] WHERE [CarID] = @p1", Con)**

**Public Shared cmdGetInfo As New SqlCommandBuilder(gdaGetCarInfo)**

**#End Region**

**#Region "init"**

**Protected Sub final\_proj(sender As Object, e As EventArgs) Handles Me.Init**

**'refreshes upon initialization**

**Call UpdateDDL()**

**End Sub**

**#End Region**

**#Region "load initial ddls"**

**Private Sub UpdateDDL()**

**Dim daCustomerID As New SqlDataAdapter("Select Distinct [CustomerID] From [MF11andrew.vernon].[dbo].[Customer]", Con)**

**Try**

**daCustomerID.Fill(dtCustomer)**

**With ddlCRCustomerID**

**.DataSource = dtCustomer**

**.DataValueField = "CustomerID"**

**.DataTextField = "CustomerID"**

**.DataBind()**

**.Items.Insert(0, "Select a Customer")**

**End With**

**With ddlACSCustomerID**

**.DataSource = dtCustomer**

**.DataValueField = "CustomerID"**

**.DataTextField = "CustomerID"**

**.DataBind()**

**.Items.Insert(0, "Select a Customer")**

**End With**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**daCustomerID.FillSchema(dtCustomer, SchemaType.Mapped)**

**If dtCar.Rows.Count > 0 Then**

**dtCar.Rows.Clear()**

**End If**

**Dim daBrand As New SqlDataAdapter("Select Distinct [CarBrand] From [MF11andrew.vernon].[dbo].[Car]", Con)**

**Try**

**daBrand.Fill(dtCar)**

**With ddlCRBrand**

**.DataSource = dtCar**

**.DataValueField = "CarBrand"**

**.DataTextField = "CarBrand"**

**.DataBind()**

**.Items.Insert(0, "Select a Brand")**

**End With**

**With ddlNCBrand**

**.DataSource = dtCar**

**.DataValueField = "CarBrand"**

**.DataTextField = "CarBrand"**

**.DataBind()**

**.Items.Insert(0, "Select a Brand")**

**End With**

**With ddlNMSBrand**

**.DataSource = dtCar**

**.DataValueField = "CarBrand"**

**.DataTextField = "CarBrand"**

**.DataBind()**

**.Items.Insert(0, "Select a Brand")**

**End With**

**With ddlNMBrand**

**.DataSource = dtCar**

**.DataValueField = "CarBrand"**

**.DataTextField = "CarBrand"**

**.DataBind()**

**.Items.Insert(0, "Select a Brand")**

**End With**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**daBrand.FillSchema(dtCar, SchemaType.Mapped)**

**End Sub**

**#End Region**

**#Region "fill change ddls"**

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

**If dtCarModel.Rows.Count > 0 Then**

**dtCarModel.Rows.Clear()**

**End If**

**Dim daCarModel As New SqlDataAdapter("Select Distinct [CarModel] From [MF11andrew.vernon].[dbo].[Car] WHERE [CarBrand] = @p1", Con)**

**With daCarModel.SelectCommand.Parameters**

**.Clear()**

**.AddWithValue("@p1", ddlCRBrand.SelectedValue)**

**End With**

**Try**

**daCarModel.Fill(dtCarModel)**

**With ddlCRModel**

**.DataSource = dtCarModel**

**.DataValueField = "CarModel"**

**.DataTextField = "CarModel"**

**.DataBind()**

**.Items.Insert(0, "Select a Model")**

**End With**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**daCarModel.FillSchema(dtCarModel, SchemaType.Mapped)**

**End Sub**

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

**If dtCarName.Rows.Count > 0 Then**

**dtCarName.Rows.Clear()**

**End If**

**Dim daCarName As New SqlDataAdapter("Select Distinct [CarName] From [MF11andrew.vernon].[dbo].[Car] WHERE [CarModel] = @p1", Con)**

**With daCarName.SelectCommand.Parameters**

**.Clear()**

**.AddWithValue("@p1", ddlCRModel.SelectedValue)**

**End With**

**Try**

**daCarName.Fill(dtCarName)**

**With ddlCRCarName**

**.DataSource = dtCarName**

**.DataValueField = "CarName"**

**.DataTextField = "CarName"**

**.DataBind()**

**.Items.Insert(0, "Select a Model")**

**End With**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**daCarName.FillSchema(dtCarName, SchemaType.Mapped)**

**End Sub**

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

**If dtCarModel.Rows.Count > 0 Then**

**dtCarModel.Rows.Clear()**

**End If**

**Dim daCarModel As New SqlDataAdapter("Select Distinct [CarModel] From [MF11andrew.vernon].[dbo].[Car] WHERE [CarBrand] = @p1", Con)**

**With daCarModel.SelectCommand.Parameters**

**.Clear()**

**.AddWithValue("@p1", ddlNCBrand.SelectedValue)**

**End With**

**Try**

**daCarModel.Fill(dtCarModel)**

**With ddlNCModel**

**.DataSource = dtCarModel**

**.DataValueField = "CarModel"**

**.DataTextField = "CarModel"**

**.DataBind()**

**.Items.Insert(0, "Select a Model")**

**End With**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**daCarModel.FillSchema(dtCarModel, SchemaType.Mapped)**

**End Sub**

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

**If dtCarModel.Rows.Count > 0 Then**

**dtCarModel.Rows.Clear()**

**End If**

**Dim daCarModel As New SqlDataAdapter("Select Distinct [CarModel] From [MF11andrew.vernon].[dbo].[Car] WHERE [CarBrand] = @p1", Con)**

**With daCarModel.SelectCommand.Parameters**

**.Clear()**

**.AddWithValue("@p1", ddlNMSBrand.SelectedValue)**

**End With**

**Try**

**daCarModel.Fill(dtCarModel)**

**With ddlNMSModel**

**.DataSource = dtCarModel**

**.DataValueField = "CarModel"**

**.DataTextField = "CarModel"**

**.DataBind()**

**.Items.Insert(0, "Select a Model")**

**End With**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**daCarModel.FillSchema(dtCarModel, SchemaType.Mapped)**

**End Sub**

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

**If dtCarName.Rows.Count > 0 Then**

**dtCarName.Rows.Clear()**

**End If**

**Dim daCarName As New SqlDataAdapter("Select Distinct [CarName] From [MF11andrew.vernon].[dbo].[Car] WHERE [CarModel] = @p1", Con)**

**With daCarName.SelectCommand.Parameters**

**.Clear()**

**.AddWithValue("@p1", ddlNMSModel.SelectedValue)**

**End With**

**Try**

**daCarName.Fill(dtCarName)**

**With ddlNMSCarName**

**.DataSource = dtCarName**

**.DataValueField = "CarName"**

**.DataTextField = "CarName"**

**.DataBind()**

**.Items.Insert(0, "Select a Model")**

**End With**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**daCarName.FillSchema(dtCarName, SchemaType.Mapped)**

**End Sub**

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

**Dim daCarPrice As New SqlDataAdapter("Select [Price] From [MF11andrew.vernon].[dbo].[Car] WHERE [CarName] = @p1", Con)**

**With daCarPrice.SelectCommand.Parameters**

**.Clear()**

**.AddWithValue("@p1", ddlCRCarName.SelectedValue)**

**End With**

**Dim ds As New DataSet**

**daCarPrice.Fill(ds)**

**txbCRPrice.Text = Convert.ToString(ds.Tables(0).Rows(0)("Price"))**

**Dim daMilage As New SqlDataAdapter("Select [TotalMilage] From [MF11andrew.vernon].[dbo].[Car] WHERE [CarName] = @p1", Con)**

**With daMilage.SelectCommand.Parameters**

**.Clear()**

**.AddWithValue("@p1", ddlCRCarName.SelectedValue)**

**End With**

**Dim ds1 As New DataSet**

**daMilage.Fill(ds1)**

**txbCRMilageStart.Text = Convert.ToString(ds1.Tables(0).Rows(0)("TotalMilage"))**

**Dim daCarID As New SqlDataAdapter("Select [CarID] From [MF11andrew.vernon].[dbo].[Car] WHERE [CarName] = @p1", Con)**

**With daCarID.SelectCommand.Parameters**

**.Clear()**

**.AddWithValue("@p1", ddlCRCarName.SelectedValue)**

**End With**

**Dim ds2 As New DataSet**

**daCarID.Fill(ds2)**

**txbCRCarID.Text = Convert.ToString(ds2.Tables(0).Rows(0)("CarID"))**

**End Sub**

**#End Region**

**#Region "view links"**

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

**MultiView1.ActiveViewIndex = 0 'first tab (first view)**

**End Sub**

**Protected Sub LinkButton3\_Click(sender As Object, e As EventArgs) Handles LinkButton3.Click**

**MultiView1.ActiveViewIndex = 2 'third tab (third view)**

**End Sub**

**Protected Sub LinkButton4\_Click(sender As Object, e As EventArgs) Handles LinkButton4.Click**

**MultiView1.ActiveViewIndex = 3 'fourth tab (fourth view)**

**End Sub**

**Protected Sub LinkButton5\_Click(sender As Object, e As EventArgs) Handles LinkButton5.Click**

**MultiView1.ActiveViewIndex = 4 'fourth tab (fourth view)**

**End Sub**

**#End Region**

**#Region "insert row - model"**

**Protected Sub btnNMSSave\_Click(sender As Object, e As EventArgs) Handles btnNMSave.Click**

**daNewModel.FillSchema(dtNewModel, SchemaType.Mapped)**

**If dtNewModel.Rows.Count > 0 Then dtCarName.Rows.Clear()**

**gvNewCar.DataSource = Nothing**

**gvNewCar.DataBind()**

**Dim dr As DataRow = dtNewModel.NewRow**

**If txbNMModel.Text = Nothing OrElse txbNMPrice.Text = Nothing Then**

**Response.Write("Enter missing data")**

**Exit Sub**

**End If**

**dr.Item("CarName") = txbNMCarName.Text**

**dr.Item("CarBrand") = ddlNMBrand.SelectedItem**

**dr.Item("CarModel") = txbNMModel.Text**

**dr.Item("Price") = txbNMPrice.Text**

**dr.Item("Color") = txbNMColor.Text**

**dr.Item("TotalMilage") = 0**

**Try**

**dtNewModel.Rows.Add(dr)**

**daNewModel.Fill(dtNewModel)**

**daNewModel.Update(dtNewModel)**

**gvNewModel.DataSource = dtNewModel**

**gvNewModel.DataBind()**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**Call UpdateDDL()**

**End Sub**

**#End Region**

**#Region "insert row - new car"**

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

**daNewCar.FillSchema(dtCarName, SchemaType.Mapped)**

**If dtCarName.Rows.Count > 0 Then dtCarName.Rows.Clear()**

**gvNewCar.DataSource = Nothing**

**gvNewCar.DataBind()**

**Dim dr As DataRow = dtCarName.NewRow**

**If ddlNCBrand.SelectedIndex = 0 OrElse ddlNCModel.SelectedIndex = 0 OrElse txbNCCarName.Text = Nothing OrElse txbNCColor.Text = Nothing Then**

**Response.Write("Enter missing data")**

**Exit Sub**

**End If**

**dr.Item("CarName") = txbNCCarName.Text**

**dr.Item("CarBrand") = ddlNCBrand.SelectedItem**

**dr.Item("CarModel") = ddlNCModel.Text**

**dr.Item("Price") = CDec(txbNCPrice.Text)**

**dr.Item("Color") = txbNCColor.Text**

**dr.Item("TotalMilage") = 0**

**Try**

**dtCarName.Rows.Add(dr)**

**daNewCar.Fill(dtCarName)**

**daNewCar.Update(dtCarName)**

**gvNewCar.DataSource = dtCarName**

**gvNewCar.DataBind()**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**Call UpdateDDL()**

**End Sub**

**#End Region**

**#Region "insert row - customer"**

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

**daNewCustomer.FillSchema(dtCustomer, SchemaType.Mapped)**

**If dtCustomer.Rows.Count > 0 Then dtCustomer.Rows.Clear()**

**gvAddingCustomer1.DataSource = Nothing**

**gvAddingCustomer1.DataBind()**

**Dim dr As DataRow = dtCustomer.NewRow**

**If txbACCustomerName.Text = Nothing OrElse txbACTotalRentals.Text = Nothing OrElse txbACPhone.Text = Nothing Then**

**Response.Write("Enter missing data")**

**Exit Sub**

**End If**

**dr.Item("CustomerName") = txbACCustomerName.Text**

**dr.Item("Phone") = txbACPhone.Text**

**dr.Item("VIP") = chkVIPAC.Checked**

**Try**

**dtCustomer.Rows.Add(dr)**

**daNewCustomer.Fill(dtCustomer)**

**daNewCustomer.Update(dtCustomer)**

**gvAddingCustomer1.DataSource = dtCustomer**

**gvAddingCustomer1.DataBind()**

**Catch ex As Exception**

**End Try**

**Call UpdateDDL()**

**End Sub**

**#End Region**

**#Region "insert row - rental"**

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

**If txbCRRentalDate.Text = Nothing OrElse txbCRReturnDate.Text = Nothing OrElse ddlCRCustomerID.SelectedIndex = 0 OrElse ddlCRBrand.SelectedIndex = 0 OrElse ddlCRModel.SelectedIndex = 0 OrElse ddlCRCarName.SelectedIndex = 0 OrElse txbCRMilageStart.Text = Nothing OrElse txbCRMilageEnd.Text = Nothing Then**

**Response.Write("Please enter sufficient information")**

**End If**

**If dtRental.Rows.Count > 0 Then dtRental.Rows.Clear()**

**daNewRental.FillSchema(dtRental, SchemaType.Mapped)**

**gvRental1.DataSource = Nothing**

**gvRental1.DataBind()**

**Dim dr As DataRow = dtRental.NewRow**

**For Each Part In cblCRParts.Items 'loop to cycle through salad selections**

**If Part.Selected = True Then 'for each item selected...**

**damageTotals += Part.Value '... add selected item(s) value to global varibale ( running calorie count)**

**End If**

**Next**

**Dim dt1 As Date = DateTime.Parse(txbCRRentalDate.Text)**

**Dim dt2 As Date = DateTime.Parse(txbCRReturnDate.Text)**

**Dim ts As TimeSpan = dt2.Subtract(dt1)**

**If ts.TotalDays < 0 OrElse CDec(txbCRMilageStart.Text) > CDec(txbCRMilageEnd.Text) Then**

**Response.Write("Please enter valid info")**

**Exit Sub**

**End If**

**totalCharged = ts.TotalDays \* txbCRPrice.Text + damageTotals**

**If chkVIPCR.Checked Then**

**totalCharged \*= 0.9**

**Exit Sub**

**End If**

**dr.Item("CustomerID") = ddlCRCustomerID.SelectedValue**

**dr.Item("CarID") = txbCRCarID.Text**

**dr.Item("RRentalDate") = txbCRRentalDate.Text**

**dr.Item("RReturnDate") = txbCRReturnDate.Text**

**dr.Item("RCarBrand") = ddlCRBrand.SelectedValue**

**dr.Item("RCarModel") = ddlCRModel.SelectedValue**

**dr.Item("RCarName") = ddlCRCarName.SelectedValue**

**dr.Item("MilageStart") = txbCRMilageStart.Text**

**dr.Item("MilageEnd") = txbCRMilageEnd.Text**

**dr.Item("DamageTotals") = damageTotals**

**dr.Item("TotalCharged") = totalCharged**

**Try**

**dtRental.Rows.Add(dr)**

**daNewRental.Fill(dtRental)**

**daNewRental.Update(dtRental)**

**gvRental1.DataSource = dtRental**

**gvRental1.DataBind()**

**Call UpdateCustomer()**

**Call UpdateCar()**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**Call UpdateDDL()**

**End Sub**

**#End Region**

**#Region "rental clear button"**

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

**txbCRRentalDate.Text = Nothing**

**txbCRReturnDate.Text = Nothing**

**ddlCRCustomerID.SelectedIndex = 0**

**ddlCRBrand.SelectedIndex = 0**

**ddlCRModel.SelectedIndex = 0**

**ddlCRCarName.SelectedIndex = 0**

**txbCRMilageStart.Text = Nothing**

**txbCRMilageEnd.Text = Nothing**

**chkVIPCR.Checked = False**

**For Each Part In cblCRParts.Items 'loop to cycle through salad selections**

**If Part.Selected = True Then 'for each item selected...**

**Part.Selected = False '... add selected item(s) value to global varibale ( running calorie count)**

**End If**

**Next**

**End Sub**

**#End Region**

**#Region "update customers"**

**Protected Sub UpdateCustomer()**

**Dim cmdUpdateRental As New SqlCommand("UPDATE [Customer] set [LastRental] = @p2, [TotalPayed]+=@p3, [TotalRentals] += @p4, [TotalMilage] += @p5 WHERE CustomerID = @p6", Con)**

**With cmdUpdateRental.Parameters**

**.Clear()**

**.AddWithValue("@p2", CDec(txbACLastRental.Text))**

**.AddWithValue("@p3", CDec(txbACTotalPayed.Text))**

**.AddWithValue("@p4", CInt(txbACTotalRentals.Text))**

**.AddWithValue("@p5", CDec(txbACTotalMilage.Text))**

**.AddWithValue("@p6", CDec(ddlACSCustomerID.SelectedValue))**

**End With**

**Try**

**If Con.State = ConnectionState.Closed Then Con.Open()**

**cmdUpdateRental.ExecuteNonQuery()**

**If dtCustomer.Rows.Count > 0 Then dtCustomer.Rows.Clear()**

**gvAddingCustomer1.DataSource = dtCustomer**

**gvAddingCustomer1.DataBind()**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**Finally**

**Con.Close()**

**End Try**

**End Sub**

**#End Region**

**#Region "update car"**

**Protected Sub UpdateCar()**

**Dim cmdUpdateCar As New SqlCommand("UPDATE [Car] set [TotalMilage]+=@p1, WHERE CarName = @p3", Con)**

**With cmdUpdateCar.Parameters**

**.Clear()**

**.AddWithValue("@p1", txbCRMilageEnd.Text)**

**.AddWithValue("@p3", ddlNMSCarName.SelectedItem)**

**End With**

**Try**

**If Con.State = ConnectionState.Closed Then Con.Open()**

**cmdUpdateCar.ExecuteNonQuery()**

**If dtCar.Rows.Count > 0 Then dtCar.Rows.Clear()**

**gvNewModel.DataSource = dtCar**

**gvNewModel.DataBind()**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**Finally**

**Con.Close()**

**End Try**

**End Sub**

**#End Region**

**#Region "change customer"**

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

**If ddlACSCustomerID.SelectedIndex <= 0 Then**

**Response.Write("Please enter all required info")**

**Exit Sub**

**End If**

**Dim daGetCustomerInfo As New SqlDataAdapter("SELECT \* FROM [MF11andrew.vernon].[dbo].[Customer] WHERE [CustomerID] = @p1", Con)**

**Dim dtOneCustomer As New DataTable**

**With daGetCustomerInfo.SelectCommand.Parameters**

**.Clear()**

**.AddWithValue("@p1", ddlACSCustomerID.SelectedItem.Text)**

**End With**

**Try**

**daGetCustomerInfo.Fill(dtOneCustomer)**

**gvAddingCustomer2.DataSource = dtOneCustomer**

**gvAddingCustomer2.DataBind()**

**daGetCustomerInfo.Fill(dtOneCustomer)**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**End Sub**

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

**If ddlACSCustomerID.SelectedIndex = 0 Then**

**Response.Write("Please enter all required info")**

**Exit Sub**

**End If**

**Try**

**Call UpdateCustomerPPT()**

**gdaGetCustomerID.Update(dtCustomer)**

**gvAddingCustomer1.DataSource = dtCustomer**

**gvAddingCustomer1.DataBind()**

**Call UpdateDDL()**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**End Sub**

**Protected Sub UpdateCustomerPPT()**

**Dim cmdUpdatePPT As New SqlCommand("UPDATE [Customer] SET [CustomerName] = @p2, [Phone] = @p3, [LastRental] = @p4, [VIP] = @p5, [TotalPayed] = @p6 WHERE [CustomerID] = @p1", Con)**

**With cmdUpdatePPT.Parameters**

**.Clear()**

**.AddWithValue("@p1", ddlACSCustomerID.SelectedValue)**

**.AddWithValue("@p2", txbACSCustomerName.Text)**

**.AddWithValue("@p3", txbACSPhone.Text)**

**.AddWithValue("@p5", chkVIPACS.Checked)**

**.AddWithValue("@p4", txbACSLastRental.Text)**

**.AddWithValue("@p6", txbACSTotalPayed.Text)**

**End With**

**Try**

**If Con.State = ConnectionState.Closed Then Con.Open()**

**cmdUpdatePPT.ExecuteNonQuery()**

**dtCustomer.Rows.Clear()**

**daNewCustomer.Fill(dtCustomer)**

**gvAddingCustomer1.DataSource = dtCustomer**

**gvAddingCustomer1.DataBind()**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**End Sub**

**#End Region**

**#Region "change car"**

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

**If ddlNMSCarName.SelectedIndex <= 0 Then**

**Response.Write("Please enter all required info")**

**Exit Sub**

**End If**

**Dim daGetCarInfo As New SqlDataAdapter("SELECT \* FROM [MF11andrew.vernon].[dbo].[Car] WHERE [CarName] = @p1", Con)**

**Dim dtOneCar As New DataTable**

**With daGetCarInfo.SelectCommand.Parameters**

**.Clear()**

**.AddWithValue("@p1", ddlNMSCarName.SelectedItem.Text)**

**End With**

**Try**

**daGetCarInfo.Fill(dtOneCar)**

**gvNewModel1.DataSource = dtOneCar**

**gvNewModel1.DataBind()**

**daGetCarInfo.Fill(dtOneCar)**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**End Sub**

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

**If ddlNMSCarName.SelectedIndex = 0 Then**

**Response.Write("Please enter all required info")**

**Exit Sub**

**End If**

**Try**

**Call UpdateCarPPT()**

**gdaGetCarInfo.Update(dtCar)**

**gvNewModel.DataSource = dtCar**

**gvNewModel.DataBind()**

**Call UpdateDDL()**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**End Sub**

**Protected Sub UpdateCarPPT()**

**Dim cmdUpdatePPT As New SqlCommand("UPDATE [Car] SET [TotalMilage] = @p2, [Color] = @p3, [Price] = @p4 WHERE [CarName] = @p1", Con)**

**With cmdUpdatePPT.Parameters**

**.Clear()**

**.AddWithValue("@p1", ddlNMSCarName.Text)**

**.AddWithValue("@p2", txbNMSTotalMilage.Text)**

**.AddWithValue("@p3", txbNMSColor.Text)**

**.AddWithValue("@p4", txbNMSPrice.Text)**

**End With**

**Try**

**If Con.State = ConnectionState.Closed Then Con.Open()**

**cmdUpdatePPT.ExecuteNonQuery()**

**dtCar.Rows.Clear()**

**daNewCar.Fill(dtCar)**

**gvNewModel.DataSource = dtCar**

**gvNewModel.DataBind()**

**Catch ex As Exception**

**Response.Write(ex.Message)**

**End Try**

**End Sub**

**#End Region**

**End Class**