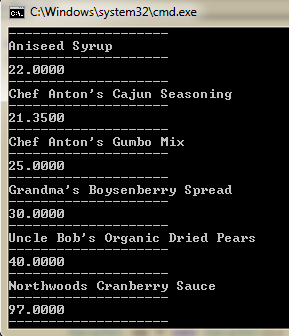
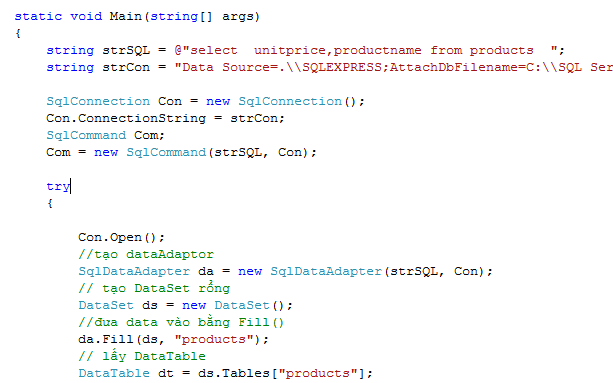
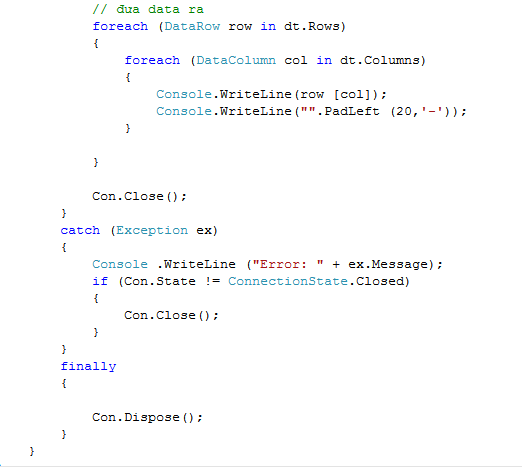
Bài 2 : DataSet và DatView

1. **Viết chương trình lấy dữ liệu từ DataSet :**
2. Output từ DOS :

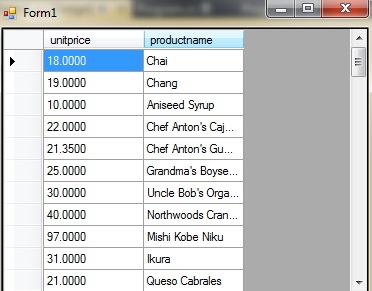


Code :

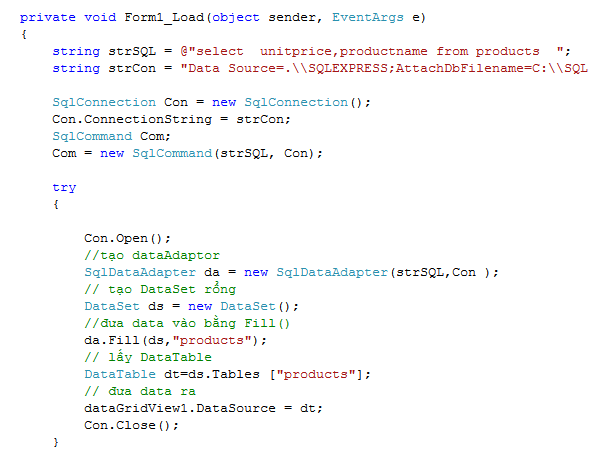


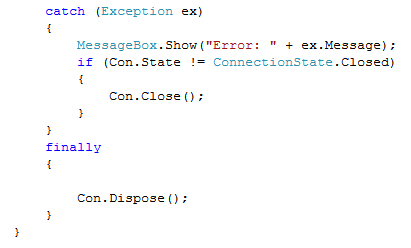


B.Output dùng dataGridvew control:

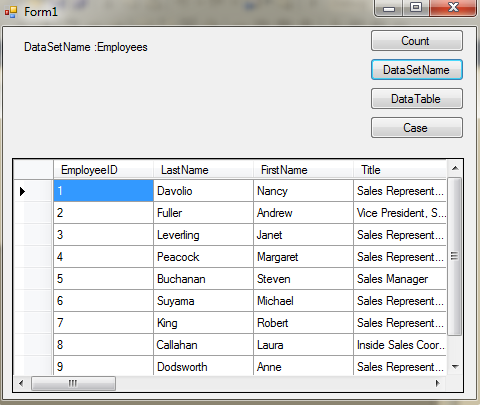


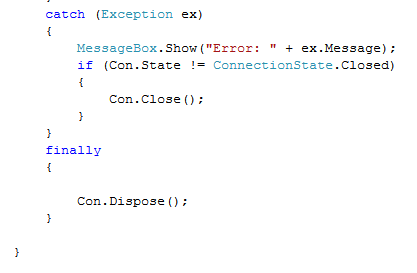
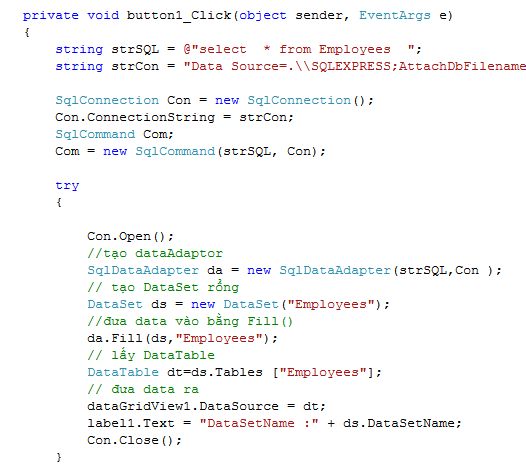
Code :



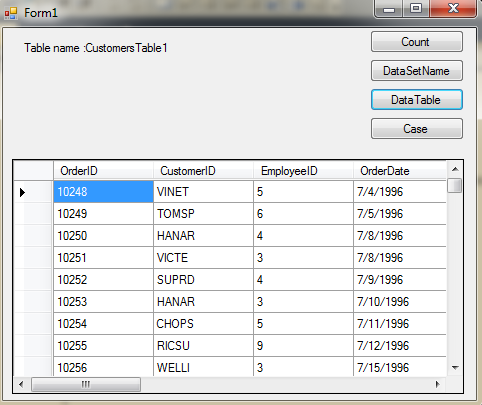


1. Viết chương trình sau:
2. DataSetName:

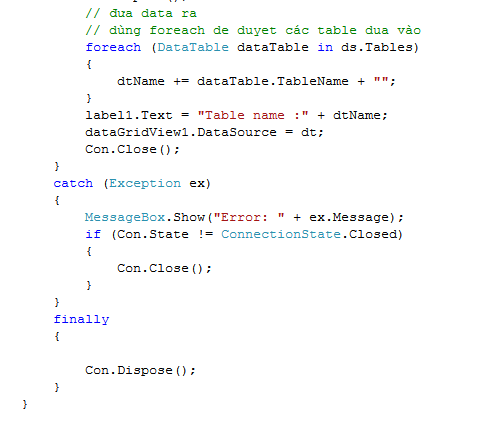
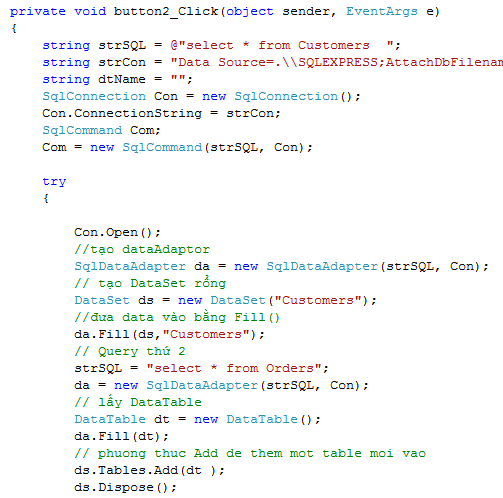




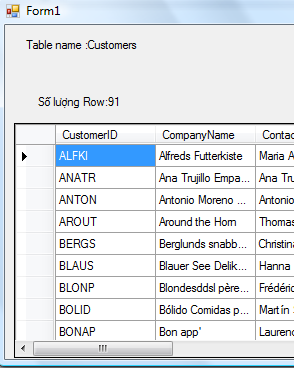
1. DataTable :
   1. Query đến 2 table *Customer và Orders.*
   2. Dùng thuộc tính *Tables để lấy đối tượng DataTable trong DataSet.*



Code:



1. Count:



Code:

string strSQL = @"select \* from Customers ";

string strCon = "Data Source=.\\SQLEXPRESS;AttachDbFilename=C:\\SQL Server 2000 Sample Databases\\NORTHWND.MDF;Integrated Security=True;Connect Timeout=30;User Instance=True";

string dtName = "";

SqlConnection Con = new SqlConnection();

Con.ConnectionString = strCon;

SqlCommand Com;

Com = new SqlCommand(strSQL, Con);

try

{

Con.Open();

//tạo dataAdaptor

SqlDataAdapter da = new SqlDataAdapter(strSQL, Con);

// tạo DataSet rổng

DataSet ds = new DataSet("Customers");

//đưa data vào bằng Fill()

da.Fill(ds, "Customers");

// Query thứ 2

strSQL = "select \* from Orders";

da = new SqlDataAdapter(strSQL, Con);

// lấy DataTable

DataTable dt = new DataTable();

da.Fill(dt);

// phuong thuc Add de them mot table moi vao

ds.Tables.Add(dt);

ds.Dispose();

// đưa data ra

// dùng foreach de duyet các table dua vào

foreach (DataTable dataTable in ds.Tables)

{

dtName += dataTable.TableName + "";

}

label1.Text = "Table name :" + ds.Tables[1].TableName ;

label2.Text = "Số lượng Row:" +ds.Tables [1].Rows .Count .ToString ();

dataGridView1.DataSource = ds.Tables [1];

Con.Close();

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

if (Con.State != ConnectionState.Closed)

{

Con.Close();

}

}

finally

{

Con.Dispose();

}

}

D: CaseSensitive:

string strSQL = @"select \* from Customers ";

string strCon = "Data Source=.\\SQLEXPRESS;AttachDbFilename=C:\\SQL Server 2000 Sample Databases\\NORTHWND.MDF;Integrated Security=True;Connect Timeout=30;User Instance=True";

string dtName = "";

SqlConnection Con = new SqlConnection();

Con.ConnectionString = strCon;

SqlCommand Com;

Com = new SqlCommand(strSQL, Con);

try

{

Con.Open();

//tạo dataAdaptor

SqlDataAdapter da = new SqlDataAdapter(strSQL, Con);

// tạo DataSet rổng

DataSet ds = new DataSet("Customers");

ds.CaseSensitive = true;

//đưa data vào bằng Fill()

da.Fill(ds, "Customers");

// Query thứ 2

strSQL = "select \* from Orders";

da = new SqlDataAdapter(strSQL, Con);

// lấy DataTable

DataTable dt = new DataTable();

da.Fill(dt);

// phuong thuc Add de them mot table moi vao

ds.Tables.Add(dt);

ds.Dispose();

// đưa data ra

// dùng foreach de duyet các table dua vào

foreach (DataTable dataTable in ds.Tables)

{

dtName += dataTable.TableName + "";

}

if (dtName == "Orders")

{

label1.Text = "Case Sensitive";

dataGridView1.DataSource = ds.Tables[0];

}

else

label1.Text = "Case Sentive False";

Con.Close();

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

if (Con.State != ConnectionState.Closed)

{

Con.Close();

}

}

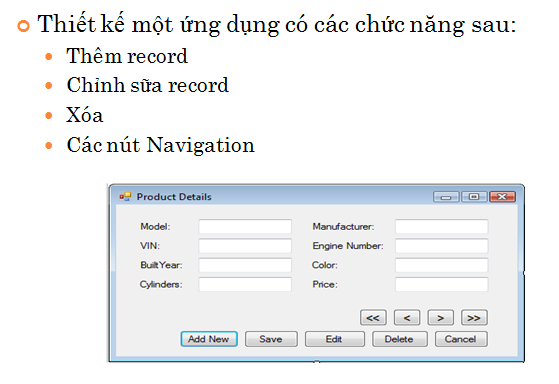
finally

{

Con.Dispose();

}

1. Viết chương trình sau:



Code :

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace AutomobileInventory

{

public partial class frmProductDetails : Form

{

SqlConnection sqlconProductDetails;

SqlCommand sqlcomProductDetails;

SqlDataReader sqldreaderProductDetails;

int rowNumber;

string operation;

string record;

string connectionString;

public frmProductDetails()

{

InitializeComponent();

connectionString = " Data Source=ANHLEVI-PC;Initial Catalog=Motor;Integrated Security=True";

sqlconProductDetails = new SqlConnection(connectionString);

sqlcomProductDetails = new SqlCommand();

}

// Loads data from the database

private void frmProductDetails\_Load(object sender, EventArgs e)

{

try

{

ClearControls();

if (sqlconProductDetails.State == ConnectionState.Open)

sqlconProductDetails.Close();

sqlconProductDetails.Open();

sqlcomProductDetails.CommandText = "SELECT \* FROM ProductDetails";

sqlcomProductDetails.Connection = sqlconProductDetails;

sqldreaderProductDetails = sqlcomProductDetails.ExecuteReader();

if (sqldreaderProductDetails.HasRows)

{

sqldreaderProductDetails.Read();

DisplayResults();

if (sqldreaderProductDetails.Read())

{

btnNext.Enabled = true;

btnLast.Enabled = true;

rowNumber = rowNumber + 1;

btnPrevious\_Click(null, null);

}

else

{

btnNext.Enabled = false;

btnLast.Enabled = false;

btnFirst.Enabled = false;

btnPrevious.Enabled = false;

}

btnEdit.Enabled = true;

btnDelete.Enabled = true;

}

else

DisableButtons();

btnCancel.Enabled = false;

btnSave.Enabled = false;

btnAddNew.Enabled = true;

}

catch (SqlException sqlexcProductDetails)

{

MessageBox.Show("Problem while connecting to database.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

sqlconProductDetails.Close();

}

}

// Displays the record

void DisplayResults()

{

txtModel.Text = sqldreaderProductDetails.GetString(0);

txtManufacturer.Text = sqldreaderProductDetails.GetString(1);

txtBuiltYear.Text = sqldreaderProductDetails.GetInt32(2).ToString();

txtVIN.Text = sqldreaderProductDetails.GetString(3);

txtCylinders.Text = sqldreaderProductDetails.GetInt32(4).ToString();

txtColor.Text = sqldreaderProductDetails.GetString(5);

txtEngineNumber.Text = sqldreaderProductDetails.GetInt32(6).ToString();

txtPrice.Text = sqldreaderProductDetails.GetInt32(7).ToString();

}

// Displays the next record

private void btnNext\_Click(object sender, EventArgs e)

{

if (sqldreaderProductDetails.Read())

{

rowNumber = rowNumber + 1;

DisplayResults();

}

else

{

MessageBox.Show("This is the last record.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

btnNext.Enabled = false;

btnLast.Enabled = false;

}

btnFirst.Enabled = true;

btnPrevious.Enabled = true;

}

// Displays the last record

private void btnLast\_Click(object sender, EventArgs e)

{

while (sqldreaderProductDetails.Read())

{

rowNumber = rowNumber + 1;

DisplayResults();

}

btnNext.Enabled = false;

btnLast.Enabled = false;

btnFirst.Enabled = true;

btnPrevious.Enabled = true;

}

// Displays the first record

private void btnFirst\_Click(object sender, EventArgs e)

{

if (!sqldreaderProductDetails.IsClosed)

sqldreaderProductDetails.Close();

sqldreaderProductDetails = sqlcomProductDetails.ExecuteReader();

if (rowNumber != 0)

{

sqldreaderProductDetails.Read();

DisplayResults();

rowNumber = 0;

}

else

{

MessageBox.Show("This is the first record.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

sqldreaderProductDetails.Read();

}

btnPrevious.Enabled = false;

btnFirst.Enabled = false;

btnNext.Enabled = true;

btnLast.Enabled = true;

}

// Displays the previous record

private void btnPrevious\_Click(object sender, EventArgs e)

{

int currentRowNumber=0;

if (!sqldreaderProductDetails.IsClosed)

sqldreaderProductDetails.Close();

sqldreaderProductDetails = sqlcomProductDetails.ExecuteReader();

if (rowNumber != 0)

{

while (currentRowNumber < rowNumber)

{

currentRowNumber = currentRowNumber + 1;

sqldreaderProductDetails.Read();

}

DisplayResults();

rowNumber = rowNumber - 1;

}

else

{

MessageBox.Show("This is the first record.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

sqldreaderProductDetails.Read();

btnPrevious.Enabled = false;

btnFirst.Enabled = false;

}

btnNext.Enabled = true;

btnLast.Enabled = true;

}

// Adds a new record

private void btnAddNew\_Click(object sender, EventArgs e)

{

ClearControls();

operation = "addnew";

DisableButtons();

btnSave.Enabled = true;

btnCancel.Enabled = true;

txtModel.Focus();

}

// Validates and saves the record in the database

private void btnSave\_Click(object sender, EventArgs e)

{

string addNewString = "";

int index = 0;

if(txtModel.Text == "")

{

MessageBox.Show("Please enter the model.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtModel.Focus();

return;

}

else if(txtManufacturer.Text == "")

{

MessageBox.Show("Please enter the manufacturer.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtManufacturer.Focus();

return;

}

else if (txtBuiltYear.Text == "")

{

MessageBox.Show("Please enter the built year.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtBuiltYear.Focus();

return;

}

else if(txtVIN.Text == "")

{

MessageBox.Show("Please enter the Vehicle Identity Number.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtVIN.Focus();

return;

}

else if(txtCylinders.Text == "")

{

MessageBox.Show("Please enter the number of cylinders.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtCylinders.Focus();

return;

}

else if(txtColor.Text == "")

{

MessageBox.Show("Please enter the color.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtColor.Focus();

return;

}

else if(txtEngineNumber.Text == "")

{

MessageBox.Show("Please enter the engine number.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtEngineNumber.Focus();

return;

}

else if(txtPrice.Text == "")

{

MessageBox.Show("Please enter the price.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtPrice.Focus();

return;

}

if(txtBuiltYear.Text != "")

{

try

{

index = Convert.ToInt32(txtBuiltYear.Text);

if (index < 0)

{

MessageBox.Show("Please enter a positive value for the year.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtBuiltYear.Focus();

return;

}

}

catch(Exception ex)

{

MessageBox.Show("Please enter a numeric value for the year.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtBuiltYear.Focus();

return;

}

}

if (txtCylinders.Text != "")

{

try

{

index = Convert.ToInt32(txtCylinders.Text);

if (index < 0)

{

MessageBox.Show("Please enter a positive value for cylinders.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtCylinders.Focus();

return;

}

}

catch(Exception ex)

{

MessageBox.Show("Please enter a numeric value for cylinders.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtCylinders.Focus();

return;

}

}

if(txtEngineNumber.Text != "")

{

try

{

index = Convert.ToInt32(txtEngineNumber.Text);

if (index < 0)

{

MessageBox.Show("Please enter a positive value for engine number.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtEngineNumber.Focus();

return;

}

}

catch(Exception ex)

{

MessageBox.Show("Please enter a numeric value for engine number.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtEngineNumber.Focus();

return;

}

}

if(txtPrice.Text != "")

{

try

{

index = Convert.ToInt32(txtPrice.Text);

if (index < 0)

{

MessageBox.Show("Please enter a positive value for price.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtPrice.Focus();

return;

}

}

catch (Exception ex)

{

MessageBox.Show("Please enter a numeric value for price.", "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtPrice.Focus();

return;

}

}

if (operation == "addnew")

addNewString = "INSERT INTO ProductDetails VALUES ('" + txtModel.Text + "','" + txtManufacturer.Text + "'," + txtBuiltYear.Text + ",'" + txtVIN.Text + "'," + txtCylinders.Text + ",'" + txtColor.Text + "'," + txtEngineNumber.Text + "," + txtPrice.Text + ")";

else if (operation == "edit")

addNewString = "UPDATE ProductDetails SET Manufacturer='" + txtManufacturer.Text + "',BuiltYear=" + txtBuiltYear.Text + ",VIN='" + txtVIN.Text + "',Cylinders=" + txtCylinders.Text + ",Color='" + txtColor.Text + "',Engine Number=" + txtEngineNumber.Text + ",Price=" + txtPrice.Text + " WHERE Model='" + txtModel.Text+"'";

if (operation != "")

{

if (!sqldreaderProductDetails.IsClosed)

sqldreaderProductDetails.Close();

sqlcomProductDetails.CommandText = addNewString;

sqlcomProductDetails.Connection = sqlconProductDetails;

record = "Model: " + txtModel.Text+"\nManufacturer: " + txtManufacturer.Text + "\nBuiltYear: " + txtBuiltYear.Text + "\nVIN: " + txtVIN.Text + "\nCylinders: " + txtCylinders.Text + "\nColor: " + txtColor.Text + "\nEngine Number: " + txtEngineNumber.Text + "\nPrice: " + txtPrice.Text;

sqlcomProductDetails.ExecuteNonQuery();

MessageBox.Show("The following details have been saved successfully.\n" + record, "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

frmProductDetails\_Load(null, null);

}

}

// Edits the record

private void btnEdit\_Click(object sender, EventArgs e)

{

DisableButtons();

operation = "edit";

btnAddNew.Enabled = false;

btnSave.Enabled = true;

btnCancel.Enabled = true;

}

// Cancels the operation

private void btnCancel\_Click(object sender, EventArgs e)

{

operation = "";

frmProductDetails\_Load(null,null);

}

// Deletes the record

private void btnDelete\_Click(object sender, EventArgs e)

{

record = "Model: " + txtModel.Text + "\nManufacturer: " + txtManufacturer.Text + "\nBuiltYear: " + txtBuiltYear.Text + "\nVIN: " + txtVIN.Text + "\nCylinders: " + txtCylinders.Text + "\nColor: " + txtColor.Text + "\nEngineNumber: " + txtEngineNumber.Text + "\nPrice: " + txtPrice.Text;

if (!sqldreaderProductDetails.IsClosed)

sqldreaderProductDetails.Close();

sqlcomProductDetails.CommandText = "DELETE FROM ProductDetails WHERE Model='" + txtModel.Text + "'";

if (sqlconProductDetails.State == ConnectionState.Open)

sqlconProductDetails.Close();

sqlconProductDetails.Open();

sqlcomProductDetails.ExecuteNonQuery();

sqlconProductDetails.Close();

MessageBox.Show("The record has been deleted successfully.\n" + record, "Product Details", MessageBoxButtons.OK, MessageBoxIcon.Information);

frmProductDetails\_Load(null, null);

}

// Clears the controls

void ClearControls()

{

txtModel.Text = "";

txtManufacturer.Text = "";

txtBuiltYear.Text = "";

txtVIN.Text = "";

txtCylinders.Text = "";

txtColor.Text = "";

txtEngineNumber.Text = "";

txtPrice.Text = "";

}

// Disables the buttons

void DisableButtons()

{

btnEdit.Enabled = false;

btnDelete.Enabled = false;

btnFirst.Enabled = false;

btnPrevious.Enabled = false;

btnNext.Enabled = false;

btnLast.Enabled = false;

btnCancel.Enabled = false;

btnAddNew.Enabled = false;

}

}

}