Model – Products.cs

Purpose – Make a Product object that contains a product ID, product name, unit price, supplier ID, and category ID.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace ProjectProducts.Models

{

public class Products

{

private int productId = -1;

private string name = "n/a";

private string unitPrice = "$9999999.00";

private int supplierId = -1;

private int categoryId = -1;

public int ProductId

{

get

{

return this.productId;

}

set

{

this.productId = value;

}

}

public string Name

{

get

{

return this.name;

}

set

{

this.name = value;

}

}

public string UnitPrice

{

get

{

return this.unitPrice;

}

set

{

this.unitPrice = value;

}

}

public int SupplierId

{

get

{

return this.supplierId;

}

set

{

this.supplierId = value;

}

}

public int CategoryId

{

get

{

return this.categoryId;

}

set

{

this.categoryId = value;

}

}

public Products(int aProductId, string aName, string aUnitPrice, int aSupplierId, int aCategoryId)

{

this.productId = aProductId;

this.name = aName;

this.unitPrice = aUnitPrice;

this.supplierId = aSupplierId;

this.categoryId = aCategoryId;

}

public override string ToString()

{

string aMessage = "";

aMessage += "Product ID: " + this.productId + " ";

aMessage += "Product Name: " + this.name + " ";

aMessage += "Unit Price: " + this.unitPrice + " ";

aMessage += "Supplier ID: " + this.supplierId + " ";

aMessage += "Category ID: " + this.categoryId + " ";

return aMessage;

}

}

}

Controller – HomeController.cs

Purpose – Pull in an Excel file with a list of products, parse the data into separate components, and populate the products in a list.

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using ProjectProducts.Models;

namespace ProjectProducts.Controllers

{

public class HomeController : Controller

{

// GET: Home

public ActionResult Index()

{

List<Products> aListOfProducts = new List<Products>();

string[] items = System.IO.File.ReadAllLines(@"C:\Users\mfras\Documents\CISY 1113\Products.txt");

int[] id = new int[items.Length];

for (int index = 0; index < items.Length; index++)

{

string[] tokens = items[index].Split(',');

int idn = Convert.ToInt32(tokens[0]);

string itemName = tokens[1];

string itemPrice = tokens[2];

int suppId = Convert.ToInt32(tokens[3]);

int catId = Convert.ToInt32(tokens[4]);

Products aProduct = new Products(idn, itemName, itemPrice, suppId, catId);

aListOfProducts.Add(aProduct);

ViewBag.aListOfProducts = aListOfProducts;

}

int counter = 0;

string storeProducts = "";

while (counter < (items.Length - 1))

{

storeProducts += "<pre>" + aListOfProducts[counter] + "</pre>" + "<br />";

counter += 1;

}

ViewBag.StoreProducts = storeProducts;

return View();

}

}

}

View – Index.cshtml

Purpose - Shows user a table filled with the products in the list in a webpage.

@{

ViewBag.Title = "Index";

}

<style>

body{

background-color: aliceblue;

}

table, th, td {

border: 2px double black;

padding: 8px;

}

</style>

<h2>In-Store Products</h2>

<table>

<tr>

<th>Product ID</th>

<th>Product Name</th>

<th>Unit Price</th>

<th>Supplier ID</th>

<th>Category ID</th>

</tr>

@foreach (var products in ViewBag.aListOfProducts)

{

<tr>

<th>@products.ProductId</th>

<th>@products.Name</th>

<th>@products.UnitPrice</th>

<th>@products.SupplierId</th>

<th>@products.CategoryId</th>

</tr>

}

</table>