using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Diagnostics;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using logicProject.Models.DBContext;

using logicProject.Models.EF;

using logicProject.Models.ViewModel;

namespace logicProject.Controllers

{

public class SuppliersController : Controller

{

private LogicEntities db = new LogicEntities();

// GET: Suppliers

public ActionResult SupplierList()

{

return View(db.Supplier.ToList());

}

// GET: Suppliers/Create

public ActionResult CreateSupplier()

{

return View();

}

// POST: Suppliers/Create

// To protect from overposting attacks, please enable the specific properties you want to bind to, for

// more details see https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult CreateSupplier([Bind(Include = "SupplierId,SupplierName,GSTRegNo,ContactName,FaxNo,PhNo,Address")] Supplier supplier)

{

if (ModelState.IsValid)

{

db.Supplier.Add(supplier);

db.SaveChanges();

return RedirectToAction("SupplierList");

}

return View(supplier);

}

// GET: Suppliers/Edit/5

public ActionResult EditSupplier(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Supplier supplier = db.Supplier.Find(id);

if (supplier == null)

{

return HttpNotFound();

}

return View(supplier);

}

// POST: Suppliers/Edit/5

// To protect from overposting attacks, please enable the specific properties you want to bind to, for

// more details see https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult EditSupplier([Bind(Include = "SupplierId,SupplierName,GSTRegNo,ContactName,FaxNo,PhNo,Address")] Supplier supplier)

{

if (ModelState.IsValid)

{

db.Entry(supplier).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("SupplierList");

}

return View(supplier);

}

// GET: Suppliers/Delete/5

public ActionResult DeleteSupplier(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Supplier supplier = db.Supplier.Find(id);

if (supplier == null)

{

return HttpNotFound();

}

return View(supplier);

}

// POST: Suppliers/Delete/5

[HttpPost, ActionName("DeleteSupplier")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmedSupplier(string id)

{

Supplier supplier = db.Supplier.Find(id);

db.Supplier.Remove(supplier);

db.SaveChanges();

return RedirectToAction("SupplierList");

}

public ActionResult ItemSupplied(string supplierid)

{

//SELECT s.ProductId,p.Category,p.[Description],s.Price

//FROM SupplierProducts s, Suppliers sp,Products p

//WHERE s.SupplierId = 'ALPA' AND p.ProductId = s.ProductId AND s.SupplierId = sp.SupplierId;

SupplierProduct suppro = db.SupplierProduct.Where(pro=>pro.SupplierId == supplierid).First();

List<Product> products = db.Product.ToList();

List<Supplier> suppliers = db.Supplier.ToList();

List<SupplierProduct> supplierproducts = db.SupplierProduct.Where(sup => sup.SupplierId == supplierid).ToList();

if (suppro == null)

{

return HttpNotFound();

}

else

{

ViewData["supplierproducts"] = suppro;

}

List<ViewModel> data = (from sp in supplierproducts

join s in suppliers on sp.SupplierId equals s.SupplierId

join p in products on sp.ProductId equals p.ProductId

select new ViewModel { product = p, supplier = s, supplierproduct = sp }).ToList();

return View(data);

}

public ActionResult CreateItemSupplied(string supplierId)

{

Supplier supplier = db.Supplier.Where(x => x.SupplierId == supplierId).SingleOrDefault();

ViewData["supplier"] = supplier;

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult CreateItemSupplied([Bind(Include = "SupplierProductId,ProductId,SupplierId,Price")]SupplierProduct supplierProduct)

{

if (ModelState.IsValid)

{

db.SupplierProduct.Add(supplierProduct);

db.SaveChanges();

return RedirectToAction("ItemSupplied", new { supplierid = supplierProduct.SupplierId });

}

return View(supplierProduct);

}

public ActionResult EditPrice(string supID, string pID)

{

SupplierProduct Product = db.SupplierProduct.Where(x => x.ProductId == pID && x.SupplierId == supID).SingleOrDefault();

ViewData["product"] = Product;

return View(Product);

}

[HttpPost,ActionName("EditPrice")]

[ValidateAntiForgeryToken]

public ActionResult EditPriceConfirmed([Bind(Include = "SupplierProductId,ProductId,SupplierId,Price")] SupplierProduct supplierProduct)

{

if (ModelState.IsValid)

{

db.Entry(supplierProduct).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("ItemSupplied",new {supplierid = supplierProduct.SupplierId });

}

return View(supplierProduct);

}

public ActionResult DeleteItemSupplied(string supId,string pId)

{

SupplierProduct supplierproduct = db.SupplierProduct.Where(x => x.ProductId == pId && x.SupplierId == supId).SingleOrDefault();

return View(supplierproduct);

}

[HttpPost,ActionName("DeleteItemSupplied")]

[ValidateAntiForgeryToken]

public ActionResult DeleteItemSuppliedConfirmed(int supplierproductid)

{

SupplierProduct supplierProduct = db.SupplierProduct.Find(supplierproductid);

db.SupplierProduct.Remove(supplierProduct);

db.SaveChanges();

return RedirectToAction("SupplierList");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}