using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

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 DepartmentsController : Controller

{

private LogicEntities db = new LogicEntities();

//FOR STORE SIDE, STORE CLERK CAN ONLY VIEW THE DEPARTMENT DETAILS AND EDIT SOME DETAILS OF EACH DEPARTMENT

public ActionResult DepartmentDetails()

{

List<ViewModelDepartment> dep = db.Department

.Select(x => new ViewModelDepartment {collectionpoint = x.CollectionPoint, department = x, headname = db.DepartmentStaff.Where(y => y.DeptId == x.DeptId && x.HeadId == y.StaffId).Select(y => y.StaffName).FirstOrDefault(), Repname = db.DepartmentStaff.Where(y => y.DeptId == x.DeptId && x.RepId==y.StaffId).Select(y => y.StaffName).FirstOrDefault() }).ToList();

return View(dep);

}

public ActionResult EditDepartmentDetails(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Department department = db.Department.Find(id);

if (department == null)

{

return HttpNotFound();

}

ViewBag.CollectionPt = new SelectList(db.CollectionPoint, "CollectionPtId", "CollectionPt", department.CollectionPt);

return View(department);

}

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult EditDepartmentDetails([Bind(Include = "DeptId,DeptName,ContactName,PhNo,FaxNo,EmailAddr,HeadId,CollectionPt,RepId")] Department department)

{

if (ModelState.IsValid)

{

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

db.SaveChanges();

return RedirectToAction("DepartmentDetails");

}

ViewBag.CollectionPt = new SelectList(db.CollectionPoint, "CollectionPtId", "CollectionPt", department.CollectionPt);

return View(department);

}

//// GET: Departments

//public ActionResult Index()

//{

// var department = db.Department.Include(d => d.CollectionPoint).Include(d=>d.DepartmentStaffs);

// //var departments=from d in db.Department join ds in db.DepartmentStaff on d.HeadId equals ds.StaffId where ds.StaffType=="head" select new

// return View(department.ToList());

//}

// GET: Departments/Details/5

//public ActionResult Details(string id)

//{

// if (id == null)

// {

// return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

// }

// Department department = db.Department.Find(id);

// if (department == null)

// {

// return HttpNotFound();

// }

// return View(department);

//}

// GET: Departments/Create

//public ActionResult Create()

//{

// ViewBag.CollectionPt = new SelectList(db.CollectionPoint, "CollectionPtId", "CollectionPt");

// return View();

//}

//// POST: Departments/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 Create([Bind(Include = "DeptId,DeptName,ContactName,PhNo,FaxNo,EmailAddr,HeadId,CollectionPt,RepId")] Department department)

//{

// if (ModelState.IsValid)

// {

// db.Department.Add(department);

// db.SaveChanges();

// return RedirectToAction("Index");

// }

// ViewBag.CollectionPt = new SelectList(db.CollectionPoint, "CollectionPtId", "CollectionPt", department.CollectionPt);

// return View(department);

//}

// GET: Departments/Delete/5

//public ActionResult Delete(string id)

//{

// if (id == null)

// {

// return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

// }

// Department department = db.Department.Find(id);

// if (department == null)

// {

// return HttpNotFound();

// }

// return View(department);

//}

// POST: Departments/Delete/5

//[HttpPost, ActionName("Delete")]

//[ValidateAntiForgeryToken]

//public ActionResult DeleteConfirmed(string id)

//{

// Department department = db.Department.Find(id);

// db.Department.Remove(department);

// db.SaveChanges();

// return RedirectToAction("Index");

//}

//protected override void Dispose(bool disposing)

//{

// if (disposing)

// {

// db.Dispose();

// }

// base.Dispose(disposing);

//}

}

}