﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using IP\_Tracking.DAL;

using IP\_Tracking.Models;

namespace IP\_Tracking.Controllers

{

public class AddressController : Controller

{

Address\_DAL \_addressDAL = new Address\_DAL();

// GET: Address

public ActionResult Index(string searchBy, string searchValue)

{

try

{

var addressList = \_addressDAL.GetAllAddress();

if (addressList.Count == 0)

{

TempData["InfoMessage"] = "Currently Data not available in the Database.";

}

else

{

if (string.IsNullOrEmpty(searchValue))

{

TempData["InfoMessage"] = "Please provide the search value.";

return View(addressList);

}

else

{

if(searchBy.ToLower() == "itr\_section")

{

var searchBySection = addressList.Where(p => p.ITR\_Section.ToLower().Contains(searchValue.ToLower()));

return View(searchBySection);

}

else if (searchBy.ToLower() == "empname")

{

var searchByName = addressList.Where(p => p.Employee\_Name.ToLower().Contains(searchValue.ToLower()));

return View(searchByName);

}

else if (searchBy.ToLower() == "ipaddress")

{

var searchByIP = addressList.Where(p => p.IP\_Address.ToLower().Contains(searchValue.ToLower()));

return View(searchByIP);

}

}

}

return View(addressList);

}

catch (Exception ex)

{

TempData["ErrorMessage"] = ex.Message;

return View();

}

}

// GET: Address/Details/5

public ActionResult Details(int id)

{

try

{

var address = \_addressDAL.GetAddressByID(id).FirstOrDefault();

if (address == null)

{

TempData["InfoMessage"] = "Address not available with ID " + id.ToString();

return RedirectToAction("Index");

}

return View(address);

}

catch (Exception ex)

{

TempData["ErrorMessage"] = ex.Message;

return View();

}

}

// GET: Address/Create

private List<string> GetITRSectionSuggestions()

{

List<string> suggestions = new List<string>

{

"Computer Science",

"Information Technology",

"Electronics & Telecommunication",

"Mechanical",

"Civil",

"Production",

"Electrical Engineering",

"Electrical & Electronics"

};

return suggestions;

}

public ActionResult Create()

{

ViewBag.ITRSectionSuggestions = GetITRSectionSuggestions();

return View();

}

// POST: Address/Create

[HttpPost]

public ActionResult Create(Address address)

{

bool IsInserted = false;

try

{

if (ModelState.IsValid)

{

IsInserted = \_addressDAL.InsertData(address);

if (IsInserted)

{

TempData["SuccessMessage"] = "Record Saved Successfully!";

}

else

{

TempData["ErrorMessage"] = "Unable To Save The Record.";

}

}

return RedirectToAction("Index");

}

catch (Exception ex)

{

TempData["ErrorMessage"] = ex.Message;

return View();

}

}

// GET: Address/Edit/5

public ActionResult Edit(int id)

{

var address = \_addressDAL.GetAddressByID(id).FirstOrDefault();

ViewBag.ITRSectionSuggestions = GetITRSectionSuggestions();

if (address == null)

{

TempData["InfoMessage"] = "Address not available with ID " + id.ToString();

return RedirectToAction("Index");

}

return View(address);

}

// POST: Address/Edit/5

[HttpPost,ActionName("Edit")]

public ActionResult UpdateData(Address address)

{

try

{

if (ModelState.IsValid)

{

bool IsUpdated = \_addressDAL.UpdateData(address);

if (IsUpdated)

{

TempData["SuccessMessage"] = "Record Updated Successfully!";

}

else

{

TempData["ErrorMessage"] = "Unable To Update The Record.";

}

}

return RedirectToAction("Index");

}

catch (Exception ex)

{

TempData["ErrorMessage"] = ex.Message;

return View();

}

}

// GET: Address/Delete/5

/\*public ActionResult Delete(int id)

{

try

{

var address = \_addressDAL.GetAddressByID(id).FirstOrDefault();

if (address == null)

{

TempData["InfoMessage"] = "Data not available with ID " + id.ToString();

return RedirectToAction("Index");

}

return View(address);

}

catch (Exception ex)

{

TempData["ErrorMessage"] = ex.Message;

return View();

}

}\*/

// POST: Address/Delete/5

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

public ActionResult Delete(int id)

{

try

{

int r = \_addressDAL.DeleteData(id);

if (r > 0)

{

TempData["SuccessMessage"] = "Record Deleted Successfully!";

}

else

{

TempData["ErrorMessage"] = "Unable To Delete";

}

return RedirectToAction("Index");

}

catch (Exception ex)

{

TempData["ErrorMessage"] = "An error occurred while deleting the record: " + ex.Message;

return View("Index");

}

}

}

}