|  |
| --- |
| using System; |
|  |

|  |
| --- |
| using System.Collections.Generic; |
|  |

|  |
| --- |
| using System.Data; |
|  |

|  |
| --- |
| using System.Data.Entity; |
|  |

|  |
| --- |
| using System.Linq; |
|  |

|  |
| --- |
| using System.Web; |
|  |

|  |
| --- |
| using System.Web.Mvc; |
|  |

|  |
| --- |
| using MvcPaging; |
|  |

|  |
| --- |
| using System.Globalization; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| namespace iloire\_Facturacion.Controllers |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| [Authorize] |
|  |

|  |
| --- |
| public class InvoiceController : Controller |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| private InvoiceDB db = new InvoiceDB(); |
|  |

|  |
| --- |
| private int defaultPageSize = Convert.ToInt32(System.Configuration.ConfigurationManager.AppSettings["DefaultPaginationSize"]); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*CUSTOM\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| private void FillIndexViewBags(IPagedList<Invoice> invoices) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| ViewBag.NetTotal = invoices.Sum(i => i.NetTotal); |
|  |

|  |
| --- |
| ViewBag.TotalWithVAT = invoices.Sum(i => i.TotalWithVAT); |
|  |

|  |
| --- |
| ViewBag.AdvancePaymentTaxAmountTotal = invoices.Sum(i => i.AdvancePaymentTaxAmount); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public ViewResultBase Search(string text, string from, string to, int? page, int? pagesize, bool? proposal = false) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| Session["invoiceText"] = text; |
|  |

|  |
| --- |
| Session["invoiceFrom"] = from; |
|  |

|  |
| --- |
| Session["invoiceTo"] = to; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var invoices = db.Invoices.Include(i => i.InvoiceDetails).Include(i => i.Customer); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (!string.IsNullOrWhiteSpace(from)) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| DateTime fromDate; |
|  |

|  |
| --- |
| if (DateTime.TryParse(from, CultureInfo.CurrentUICulture, DateTimeStyles.AssumeLocal, out fromDate)) |
|  |

|  |
| --- |
| invoices = invoices.Where(t => t.TimeStamp >= fromDate); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| if (!string.IsNullOrWhiteSpace(to)) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| DateTime toDate; |
|  |

|  |
| --- |
| if (DateTime.TryParse(to, CultureInfo.CurrentUICulture, DateTimeStyles.AssumeLocal, out toDate)) |
|  |

|  |
| --- |
| invoices = invoices.Where(t => t.TimeStamp <= toDate); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (!string.IsNullOrWhiteSpace(text)) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| invoices = invoices.Where(t => (t.Notes.ToLower().IndexOf(text.ToLower()) > -1) |
|  |

|  |
| --- |
| || (t.Name.ToLower().IndexOf(text.ToLower()) > -1) |
|  |

|  |
| --- |
| || (t.Customer.Name.ToLower().IndexOf(text.ToLower()) > -1) |
|  |

|  |
| --- |
| ); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ViewBag.IsProposal = proposal; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| int currentPageIndex = page.HasValue ? page.Value - 1 : 0; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| IPagedList<Invoice> invoices\_paged = null; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (proposal == true)//once the data is in memory, i can filter by IsProposal |
|  |

|  |
| --- |
| invoices = invoices.Where(i => i.InvoiceNumber == 0); //we can not use Where(i => i.IsProposal) from within the LINQ db context |
|  |

|  |
| --- |
| else |
|  |

|  |
| --- |
| invoices = invoices.Where(i => i.InvoiceNumber > 0); //we can not use Where(i => i.IsProposal) from within the LINQ db context |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| invoices\_paged = invoices.OrderByDescending(i => i.TimeStamp).ToPagedList(currentPageIndex, (pagesize.HasValue) ? pagesize.Value : defaultPageSize); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| FillIndexViewBags(invoices\_paged); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (Request.IsAjaxRequest()) |
|  |

|  |
| --- |
| return PartialView("Index", invoices\_paged); |
|  |

|  |
| --- |
| else |
|  |

|  |
| --- |
| return View("Index", invoices\_paged); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public PartialViewResult UnPaidInvoices() |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| var invoices = db.Invoices.Include(i => i.Customer).Where(i => i.Paid == false && i.DueDate >= DateTime.Now && i.InvoiceNumber> 0).OrderBy(i => i.DueDate); |
|  |

|  |
| --- |
| return PartialView("InvoicesListPartial", invoices.ToList()); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public PartialViewResult LatestProposals() { |
|  |

|  |
| --- |
| var invoices = db.Invoices.Include(i => i.Customer).Where(i => i.InvoiceNumber == 0).OrderByDescending(i => i.TimeStamp); |
|  |

|  |
| --- |
| return PartialView("InvoicesListPartial", invoices.ToList()); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public PartialViewResult OverDueInvoices() |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| var invoices = db.Invoices.Include(i => i.Customer).Where(i => i.Paid == false && i.DueDate < DateTime.Now && i.InvoiceNumber > 0).OrderBy(i => i.DueDate); |
|  |

|  |
| --- |
| return PartialView("InvoicesListPartial", invoices.ToList()); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public PartialViewResult LastInvoicesByCustomer(int id) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| var invoices = db.Invoices.Include(i => i.Customer).Where(i => i.CustomerID == id && i.InvoiceNumber > 0).OrderByDescending(i => i.TimeStamp); |
|  |

|  |
| --- |
| return PartialView("InvoicesListPartial", invoices.ToList()); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public PartialViewResult LastProposalsByCustomer(int id) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| var invoices = db.Invoices.Include(i => i.Customer).Where(i => i.CustomerID == id && i.InvoiceNumber == 0).OrderByDescending(i=>i.TimeStamp); |
|  |

|  |
| --- |
| return PartialView("InvoicesListPartial", invoices.ToList()); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*END CUSTOM\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| // |
|  |

|  |
| --- |
| // GET: /Invoice/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public ActionResult Index(string filter, int? page, int? pagesize, bool? proposal = false) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| #region remember filter stuff |
|  |

|  |
| --- |
| if (filter == "clear") |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| Session["invoiceText"] = null; |
|  |

|  |
| --- |
| Session["invoiceFrom"] = null; |
|  |

|  |
| --- |
| Session["invoiceTo"] = null; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| if ((Session["invoiceText"] != null) || (Session["invoiceFrom"] != null) || (Session["invoiceTo"] != null)) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| return RedirectToAction("Search", new { text = Session["invoiceText"], from = Session["invoiceFrom"], to = Session["invoiceTo"], proposal=proposal }); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| #endregion |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| int currentPageIndex = page.HasValue ? page.Value - 1 : 0; |
|  |

|  |
| --- |
| var invoices = db.Invoices.Include(i => i.InvoiceDetails).Include(i => i.Customer); |
|  |

|  |
| --- |
| ViewBag.IsProposal = proposal; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| IPagedList<Invoice> invoices\_paged = null; |
|  |

|  |
| --- |
| if (proposal == true) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| invoices = invoices.Where(i => i.InvoiceNumber == 0); //we can not use Where(i => i.IsProposal) from within the LINQ db context |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| invoices = invoices.Where(i => i.InvoiceNumber > 0); //we can not use Where(i => i.IsProposal) from within the LINQ db context |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| invoices\_paged = invoices.OrderByDescending(i => i.TimeStamp).ToPagedList(currentPageIndex, (pagesize.HasValue) ? pagesize.Value : defaultPageSize); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| FillIndexViewBags(invoices\_paged); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return View(invoices\_paged); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| // |
|  |

|  |
| --- |
| // GET: /Invoice/Details/5 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public ViewResult Print(int id, bool? proposal = false) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| if (Request["lan"] != null) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| //valid culture name? |
|  |

|  |
| --- |
| CultureInfo[] cultures = System.Globalization.CultureInfo.GetCultures |
|  |

|  |
| --- |
| (CultureTypes.SpecificCultures); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var selectCulture = from p in cultures |
|  |

|  |
| --- |
| where p.Name == Request["lan"] |
|  |

|  |
| --- |
| select p; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (selectCulture.Count() == 1) |
|  |

|  |
| --- |
| System.Threading.Thread.CurrentThread.CurrentUICulture = new CultureInfo(Request["lan"]); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ViewBag.Print = true; |
|  |

|  |
| --- |
| ViewBag.MyCompany = System.Configuration.ConfigurationManager.AppSettings["MyCompanyName"]; |
|  |

|  |
| --- |
| ViewBag.MyCompanyID = System.Configuration.ConfigurationManager.AppSettings["MyCompanyID"]; |
|  |

|  |
| --- |
| ViewBag.MyCompanyAddress = System.Configuration.ConfigurationManager.AppSettings["MyCompanyAddress"]; |
|  |

|  |
| --- |
| ViewBag.MyCompanyPhone = System.Configuration.ConfigurationManager.AppSettings["MyCompanyPhone"]; |
|  |

|  |
| --- |
| ViewBag.MyEmail = System.Configuration.ConfigurationManager.AppSettings["MyEmail"]; |
|  |

|  |
| --- |
| ViewBag.MyBankAccount = System.Configuration.ConfigurationManager.AppSettings["MyBankAccount"]; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Invoice invoice = db.Invoices.Find(id); |
|  |

|  |
| --- |
| if (proposal == true) |
|  |

|  |
| --- |
| return View("PrintProposal", invoice); |
|  |

|  |
| --- |
| else |
|  |

|  |
| --- |
| return View(invoice); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| // |
|  |

|  |
| --- |
| // GET: /Invoice/Create |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public ActionResult Create(bool? proposal=false) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| Invoice i = new Invoice(); |
|  |

|  |
| --- |
| i.TimeStamp = DateTime.Now; |
|  |

|  |
| --- |
| i.DueDate = DateTime.Now.AddDays(30); //30 days after today |
|  |

|  |
| --- |
| i.AdvancePaymentTax = Convert.ToDecimal(System.Configuration.ConfigurationManager.AppSettings["DefaultAdvancePaymentTax"]); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (!proposal == true) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| //generate next invoice number |
|  |

|  |
| --- |
| var next\_invoice = (from inv in db.Invoices |
|  |

|  |
| --- |
| orderby inv.InvoiceNumber descending |
|  |

|  |
| --- |
| select inv).FirstOrDefault(); |
|  |

|  |
| --- |
| if (next\_invoice != null) |
|  |

|  |
| --- |
| i.InvoiceNumber = next\_invoice.InvoiceNumber + 1; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| ViewBag.IsProposal = proposal; |
|  |

|  |
| --- |
| ViewBag.CustomerID = new SelectList(db.Customers.OrderBy(c=>c.Name), "CustomerID", "Name"); |
|  |

|  |
| --- |
| return View(i); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| // |
|  |

|  |
| --- |
| // POST: /Invoice/Create |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| [HttpPost] |
|  |

|  |
| --- |
| [ValidateInput(false)] |
|  |

|  |
| --- |
| public ActionResult Create(Invoice invoice, bool? proposal = false) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| ViewBag.CustomerID = new SelectList(db.Customers.OrderBy(c => c.Name), "CustomerID", "Name", invoice.CustomerID); |
|  |

|  |
| --- |
| ViewBag.IsProposal = proposal; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (ModelState.IsValid) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| //make sure invoice number doesn't exist |
|  |

|  |
| --- |
| if (proposal == false) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| var invoice\_exists = (from inv in db.Invoices where inv.InvoiceNumber == invoice.InvoiceNumber select inv).FirstOrDefault(); |
|  |

|  |
| --- |
| if (invoice\_exists != null) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| ModelState.AddModelError("InvoiceNumber", "Invoice with that number already exists"); |
|  |

|  |
| --- |
| return View(invoice); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| db.Invoices.Add(invoice); |
|  |

|  |
| --- |
| db.SaveChanges(); |
|  |

|  |
| --- |
| return RedirectToAction("Edit", "Invoice", new { id = invoice.InvoiceID, proposal = proposal }); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| return View(invoice); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| // |
|  |

|  |
| --- |
| // GET: /Invoice/Edit/5 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public ActionResult Edit(int id, bool? proposal = false, bool? makeinvoice = false, bool? makeproposal = false) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| Invoice invoice = db.Invoices.Find(id); |
|  |

|  |
| --- |
| ViewBag.CustomerID = new SelectList(db.Customers.OrderBy(c => c.Name), "CustomerID", "Name", invoice.CustomerID); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (makeinvoice == true) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| //we want to make invoice from this proposal |
|  |

|  |
| --- |
| //generate next invoice number |
|  |

|  |
| --- |
| var next\_invoice = (from inv in db.Invoices |
|  |

|  |
| --- |
| orderby inv.InvoiceNumber descending |
|  |

|  |
| --- |
| select inv).FirstOrDefault(); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (next\_invoice != null) |
|  |

|  |
| --- |
| invoice.InvoiceNumber = next\_invoice.InvoiceNumber + 1; //assign next available invoice number |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| invoice.TimeStamp = DateTime.Now; |
|  |

|  |
| --- |
| invoice.DueDate = DateTime.Now.AddDays(30); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ViewBag.Warning = "The current item is going to be converted on Invoice. A new InvoiceNumber has been pre-assigned. The dates will be modified accordingly. Click on 'Save' to continue."; |
|  |

|  |
| --- |
| ViewBag.ShowMakeInvoice = ViewBag.ShowMakeProposal = false; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else if (makeproposal == true) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| invoice.InvoiceNumber = 0; //reset invoice number |
|  |

|  |
| --- |
| ViewBag.Warning = "The current item is going to be converted on Proposal. You will lose InvoiceNumber. If that's what you want click on 'Save'"; |
|  |

|  |
| --- |
| ViewBag.ShowMakeInvoice = ViewBag.ShowMakeProposal = false; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| if (!invoice.IsProposal && proposal == true) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| //item is invoice, redirect to proper route (hack) |
|  |

|  |
| --- |
| return RedirectToAction("Edit", new { id = id, proposal = false, makeinvoice = false }); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ViewBag.ShowMakeInvoice = invoice.IsProposal; |
|  |

|  |
| --- |
| ViewBag.ShowMakeProposal = !invoice.IsProposal; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ViewBag.IsProposal = invoice.IsProposal; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return View(invoice); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| // |
|  |

|  |
| --- |
| // POST: /Invoice/Edit/5 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| [HttpPost] |
|  |

|  |
| --- |
| [ValidateInput(false)] |
|  |

|  |
| --- |
| public ActionResult Edit(Invoice invoice, bool? proposal = false) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| ViewBag.CustomerID = new SelectList(db.Customers.OrderBy(c => c.Name), "CustomerID", "Name", invoice.CustomerID); |
|  |

|  |
| --- |
| ViewBag.IsProposal = proposal; |
|  |

|  |
| --- |
| if (ModelState.IsValid) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| if (proposal == false) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| //make sure invoice number doesn't exist |
|  |

|  |
| --- |
| var invoice\_exists = (from inv in db.Invoices |
|  |

|  |
| --- |
| where inv.InvoiceNumber == invoice.InvoiceNumber |
|  |

|  |
| --- |
| && inv.InvoiceID != invoice.InvoiceID |
|  |

|  |
| --- |
| select inv).Count(); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (invoice\_exists > 0) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| ModelState.AddModelError("InvoiceNumber", "Invoice with that number already exists"); |
|  |

|  |
| --- |
| return View(invoice); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| db.Entry(invoice).State = EntityState.Modified; |
|  |

|  |
| --- |
| db.SaveChanges(); |
|  |

|  |
| --- |
| return RedirectToAction("Index", new { proposal = proposal }); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| return View(invoice); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| // |
|  |

|  |
| --- |
| // GET: /Invoice/Delete/5 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public ActionResult Delete(int id, bool? proposal = false) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| ViewBag.IsProposal = proposal; |
|  |

|  |
| --- |
| Invoice invoice = db.Invoices.Find(id); |
|  |

|  |
| --- |
| return View(invoice); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| // |
|  |

|  |
| --- |
| // POST: /Invoice/Delete/5 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| [HttpPost, ActionName("Delete")] |
|  |

|  |
| --- |
| public ActionResult DeleteConfirmed(int id, bool? proposal = false) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| ViewBag.IsProposal = proposal; |
|  |

|  |
| --- |
| Invoice invoice = db.Invoices.Find(id); |
|  |

|  |
| --- |
| db.Invoices.Remove(invoice); |
|  |

|  |
| --- |
| db.SaveChanges(); |
|  |

|  |
| --- |
| return RedirectToAction("Index", new { proposal = proposal }); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| protected override void Dispose(bool disposing) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| db.Dispose(); |
|  |

|  |
| --- |
| base.Dispose(disposing); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

}