**BÁO CÁO**

MÔN CÔNG NGHỆ PHẦN MỀM

Nhóm 2

**Tên dự án:** Quản lý chương trình đào tạo

**Thành viên:**

- Tạ Đức Việt

- Trịnh Văn Thử

- Thái Mạnh Tuấn

- Phạm Tiến Thành

- Đào Văn Tiến

**Ngôn ngữ chương trình:** ASP.NET MVC4và C#

**Mục tiêu:**

Quản lý chương trình đào tạo là một công tác quan trọng và phức tạp mà bất kỳ đơn vị đào tạo nào cũng phải thực hiện. Việc quản lý các thông tin về môn học, giảng viên là rất phức tạp và cần được cập nhật liên tục qua từng khóa học. Bởi vậy, nhóm 2 thực hiện dự án **Quản lý chương trình đào tạo** nhằm mục đích giúp việc quản lý thông tin đào tạo trở nên dễ dàng và thuận tiện nhất.

**Mô tả dự án:**

Phần mềm quản lý chương trình đào tạo, giúp các trường đại học xây dựng chương trình đào tạo theo từng ngành học và năm học.

Phần mềm được viết trên nền web, giúp việc quản lý trở nên dễ dàng và tiện lợi hơn.

**Nội dung và các chức năng của chương trình:**

1. *Chức năng thêm thông tin về đội ngũ giáo viên*

**User story:** Thêm đội ngũ cán bộ giảng dạy

**Tác giả:** Tạ Đức Việt

**Mô tả:**

Là User tôi muốn thêm thông tin đội ngũ giảng dạy

**Task:**

- Viết chức năng thêm thông tin giảng viên

- Tạo giao diện chức năng thêm thông tin giảng viên

**Code:**

**GiangVienControler.cs**

// GET: /GiangVien/Create

public ActionResult Create()

{

return View();

}

//

// POST: /GiangVien/Create

[HttpPost]

public ActionResult Create(GiangVien giangvien)

{

if (ModelState.IsValid)

{

db.giangViens.Add(giangvien);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(giangvien);

}

**Create.cshtml** ( Giao diện chức năng thêm mới thông tin giảng viên )

@model SEProject.Models.GiangVien

@{

ViewBag.Title = "Tạo mới";

}

<h2>Tạo mới</h2>

@using (Html.BeginForm()) {

@Html.ValidationSummary(true)

<fieldset>

<legend>Giảng Viên</legend>

<div class="editor-label">

<h4>Họ tên</h4>

</div>

<div class="editor-field">

@Html.EditorFor(model => model.hoTen)

@Html.ValidationMessageFor(model => model.hoTen)

</div>

<div class="editor-label">

<h4>Chức danh</h4>

</div>

<div class="editor-field">

@Html.EditorFor(model => model.chucDanh)

@Html.ValidationMessageFor(model => model.chucDanh)

</div>

<div class="editor-label">

<h4>Chuyên ngành</h4>

</div>

<div class="editor-field">

@Html.EditorFor(model => model.chuyenNganh)

@Html.ValidationMessageFor(model => model.chuyenNganh)

</div>

<div class="editor-label">

<h4>Đơn vị</h4>

</div>

<div class="editor-field">

@Html.EditorFor(model => model.donVi)

@Html.ValidationMessageFor(model => model.donVi)

</div>

<p>

<input type="submit" value="Tạo" />

</p>

</fieldset>

}

<div>

@Html.ActionLink("Quay lại danh sách", "Index")

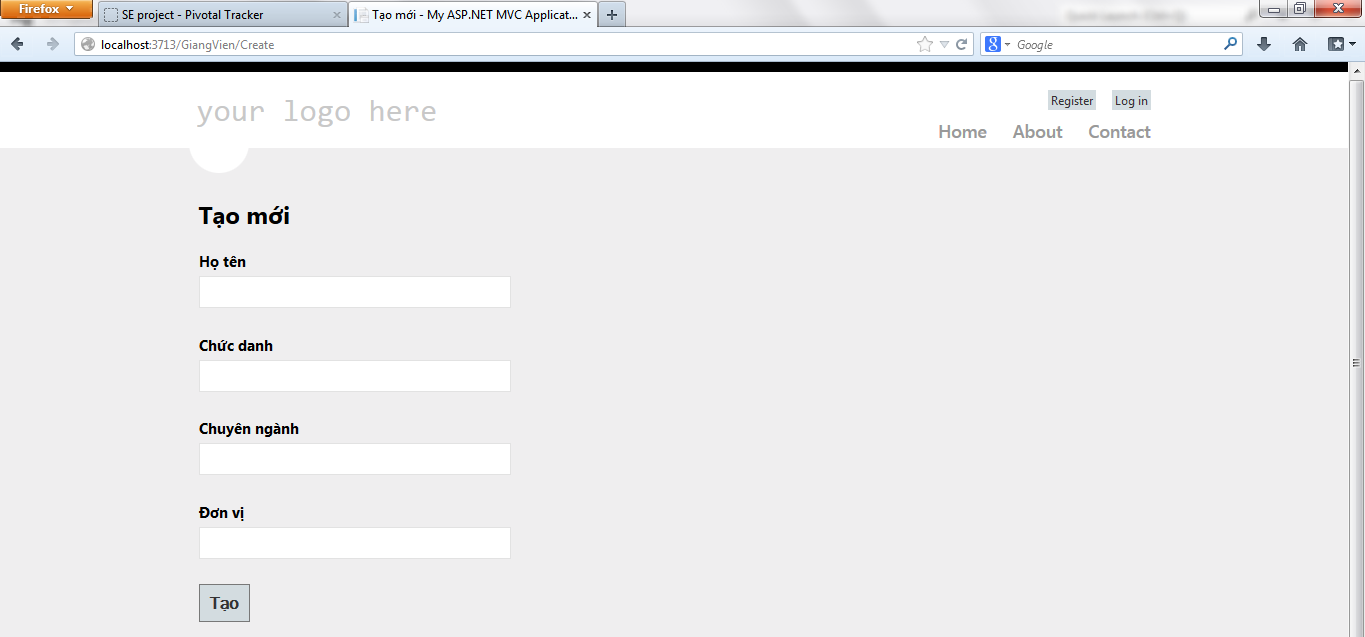
</div>

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

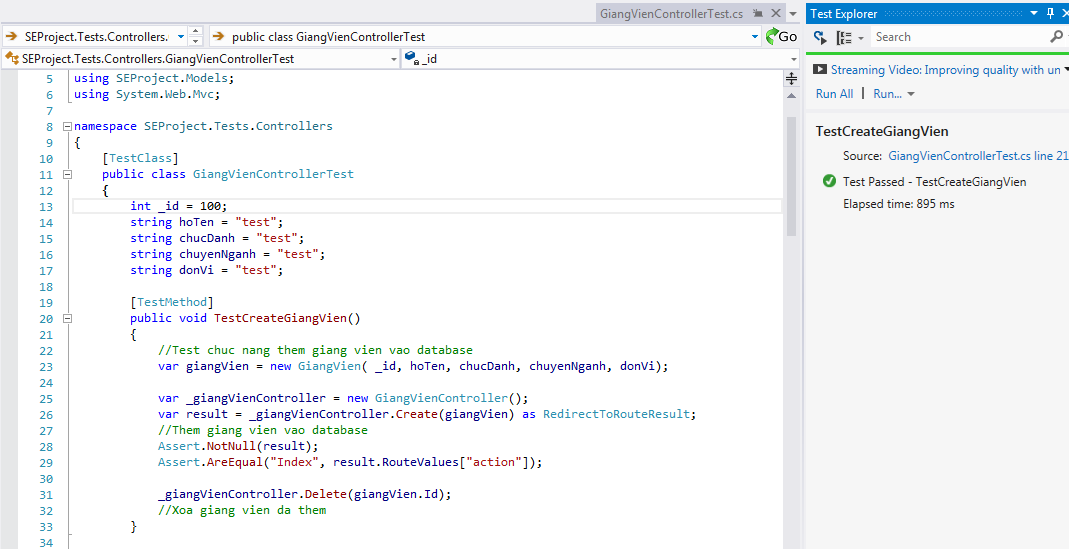
}

**Screenshots:**



Hình 1 – Giao diện chức năng tạo mới thông tin giảng viên

Test chức năng tạo mới thông tin giảng viên:



Hình 2 – Test chức năng tạo mới thông tin giảng viên

1. *Chức năng xem thông tin giảng viên*

**User story:** Xem đội ngũ cán bộ giảng dạy

**Tác giả:** Phạm Tiến Thành

**Mô tả:**

Là User tôi muốn xem thông tin đội ngũ giảng dạy

**Task:**

- In bảng liệt kê danh sách giảng viên

**Code:**

**GiangVienControler.cs:**

//

// GET: /GiangVien/

public ActionResult Index()

{

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

}

**Index.cshtml**( giao diện hiển thị bảng danh sách giảng viên) :

@model IEnumerable<SEProject.Models.GiangVien>

@{

ViewBag.Title = "Giảng viên";

}

<h2>Index</h2>

<p>

@Html.ActionLink("Tạo mới", "Create") |

@Html.ActionLink("Tìm kiếm","SearchGiangVien")

</p>

<table align="center" cellspacing="10">

<tr align="center">

<td>

<h4>Họ tên</h4>

</td>

<td>

<h4>Chức danh</h4>

</td>

<td>

<h4>Chuyên ngành</h4>

</td>

<td>

<h4>Đơn vị</h4>

</td>

<th></th>

</tr>

@foreach (var item in Model) {

<tr align="center">

<td>

@Html.DisplayFor(modelItem => item.hoTen)

</td>

<td>

@Html.DisplayFor(modelItem => item.chucDanh)

</td>

<td>

@Html.DisplayFor(modelItem => item.chuyenNganh)

</td>

<td>

@Html.DisplayFor(modelItem => item.donVi)

</td>

<td>

@Html.ActionLink("Chỉnh sửa", "Edit", new { id=item.Id }) |

@Html.ActionLink("Chi tiết", "Details", new { id=item.Id }) |

@Html.ActionLink("Xóa", "Delete", new { id=item.Id })

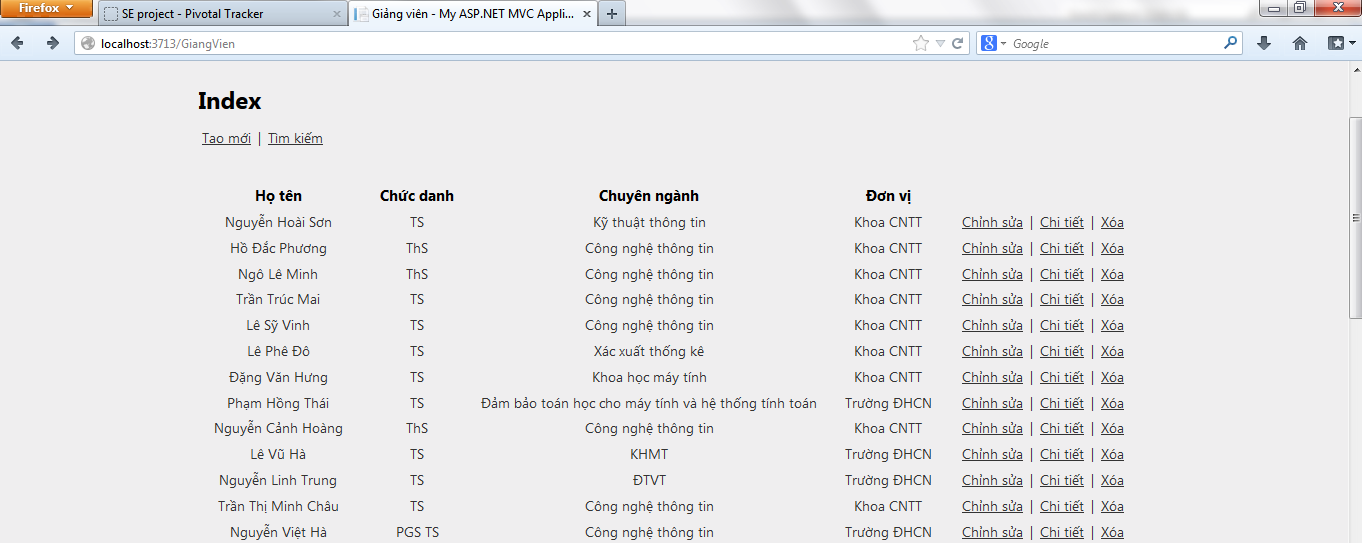
</td>

</tr>

}

</table>

**Screenshots:**



Hình 3 – Giao diện hiển thị bảng danh sách giảng viên

1. *Chức năng tìm kiếm thông tin giảng viên*

**User story:** Tìm kiếm giảng viên

**Tác giả:** Trịnh Văn Thử

**Mô tả:**

Là User tôi muốn tìm kiếm giảng viên theo tên và ngành

**Task:**

- Viết chức năng tìm kiếm giảng viên theo tên và ngành

- In bảng danh sách tìm được

**Code:**

**GiangVienControler.cs**

//Tim kiem giang vien theo ten giang vien va nganh

public ActionResult SearchGiangVien(string searchgv, string nganh)

{

var dsnganh = new List<string>();

var nganhQry = from d in db.giangViens

orderby d.chuyenNganh

select d.chuyenNganh;

dsnganh.AddRange(nganhQry.Distinct());

ViewBag.chuyenNganh = new SelectList(dsnganh);

var giangviens = from m in db.giangViens

select m;

if (!String.IsNullOrEmpty(searchgv))

{

giangviens = giangviens.Where(s => s.hoTen.Contains(searchgv));

}

if (string.IsNullOrEmpty(nganh))

return View(giangviens);

else

{

return View(giangviens.Where(x => x.chuyenNganh == nganh));

}

**SearchGiangVien.cshtml:**

@model IEnumerable<SEProject.Models.GiangVien>

@{

ViewBag.Title = "Tìm kiếm";

}

<h2>Tìm kiếm giảng viên</h2>

<p>

@Html.ActionLink("Tạo mới", "Create")

@using (Html.BeginForm()){

<p> Ngành: @Html.DropDownList("chuyenNganh", "All") Nhập tên giảng viên: @Html.TextBox("searchgv")

<input type="submit" value="Tìm kiếm" /></p>

}

</p>

<table>

<tr align ="center">

<td>

<h4>Họ tên</h4>

</td>

<td>

<h4>Chức danh</h4>

</td>

<td>

<h4>Chuyên ngành</h4>

</td>

<td>

<h4>Đơn vị</h4>

</td>

<th></th>

</tr>

@foreach (var item in Model) {

<tr align ="center">

<td>

@Html.DisplayFor(modelItem => item.hoTen)

</td>

<td>

@Html.DisplayFor(modelItem => item.chucDanh)

</td>

<td>

@Html.DisplayFor(modelItem => item.chuyenNganh)

</td>

<td>

@Html.DisplayFor(modelItem => item.donVi)

</td>

<td>

@Html.ActionLink("Chi tiết", "Details", new { id=item.Id })

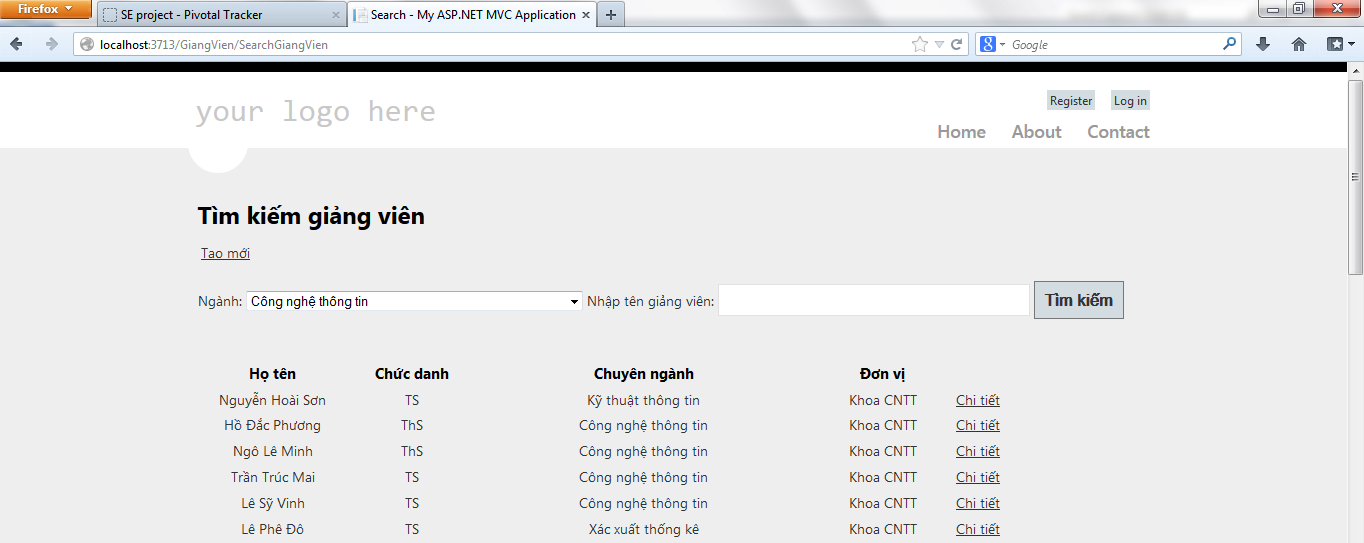
</td>

</tr>

}

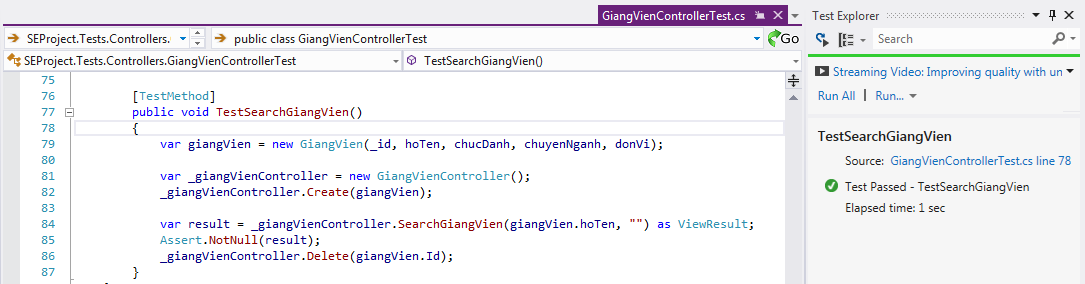
</table>

**Screenshots:**



Hình 4 – Giao diện chức năng tìm kiếm giảng viên

**Test:**



Hình 5 – Test chức năng tìm kiếm giảng viên

1. *Chức năng thêm thông tin môn học*

**User story:** Thêm môn học

**Tác giả:** Trịnh Văn Thử

**Mô tả:**

Là User tôi muốn thêm môn học

**Task:**

- Viết chức năng tạo môn học mới

- Tạo giao diện chức năng tạo môn học mới

**Code:**

**MonHocControler.cs:**

//

// GET: /MonHoc/Create

public ActionResult Create()

{

return View();

}

//

// POST: /MonHoc/Create

[HttpPost]

public ActionResult Create(MonHoc monhoc)

{

if (ModelState.IsValid)

{

db.monHocs.Add(monhoc);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(monhoc);

}

**Create.cshtml**( Giao diện chức năng tạo mới môn học)

@model SEProject.Models.MonHoc

@{

ViewBag.Title = "Tạo mới";

}

<h2>Tạo mới</h2>

@using (Html.BeginForm()) {

@Html.ValidationSummary(true)

<fieldset>

<legend>Môn học</legend>

<div class="editor-label">

Mã môn học

</div>

<div class="editor-field">

@Html.EditorFor(model => model.maMonHoc)

@Html.ValidationMessageFor(model => model.maMonHoc)

</div>

<div class="editor-label">

Tên môn học

</div>

<div class="editor-field">

@Html.EditorFor(model => model.tenMonHoc)

@Html.ValidationMessageFor(model => model.tenMonHoc)

</div>

<div class="editor-label">

Số tín chỉ

</div>

<div class="editor-field">

@Html.EditorFor(model => model.soTinChi)

@Html.ValidationMessageFor(model => model.soTinChi)

</div>

<div class="editor-label">

Số giờ lý thuyết

</div>

<div class="editor-field">

@Html.EditorFor(model => model.gioLyThuyet)

@Html.ValidationMessageFor(model => model.gioLyThuyet)

</div>

<div class="editor-label">

Số giờ thực hành

</div>

<div class="editor-field">

@Html.EditorFor(model => model.gioThucHanh)

@Html.ValidationMessageFor(model => model.gioThucHanh)

</div>

<div class="editor-label">

Số giờ tự học

</div>

<div class="editor-field">

@Html.EditorFor(model => model.gioTuHoc)

@Html.ValidationMessageFor(model => model.gioTuHoc)

</div>

<div class="editor-label">

Mã môn học tiên quyết

</div>

<div class="editor-field">

@Html.EditorFor(model => model.maMonKienQuyet)

@Html.ValidationMessageFor(model => model.maMonKienQuyet)

</div>

<p>

<input type="submit" value="Tạo" />

</p>

</fieldset>

}

<div>

@Html.ActionLink("Back to List", "Index")

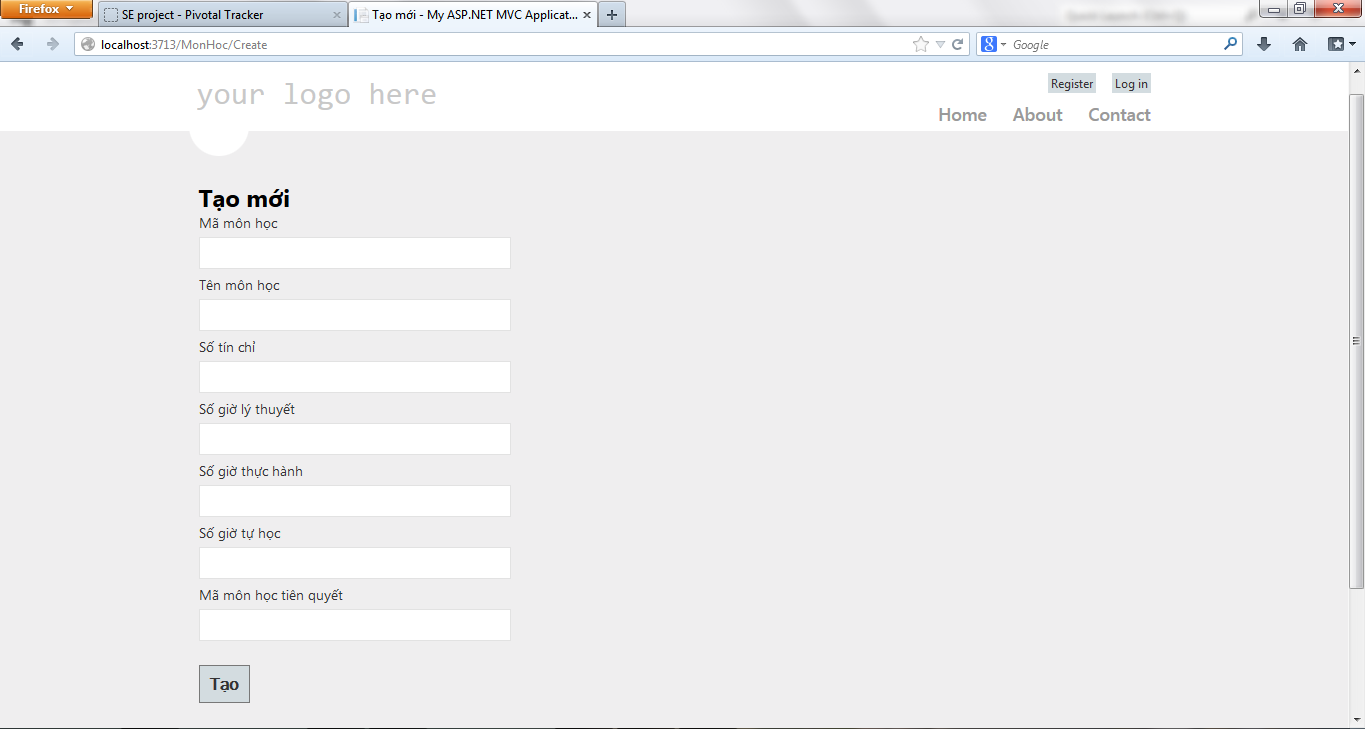
</div>

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

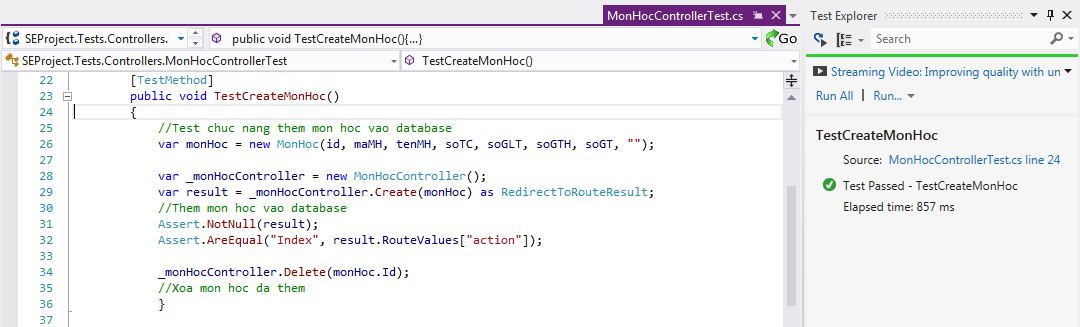
}

**Screenshots:**



Hình 6 – Giao diện chức năng tạo môn học mới

Test chức năng tạo mới môn học:



Hình 7 – Test chức năng tạo môn học mới

1. *Chức năng xem bảng thông tin môn học*

**User story:** Xem các môn học

**Tác giả:** Đào Văn Tiến

**Mô tả:**

Là User tôi muốn xem danh sách các môn học

**Task:**

- In bảng danh sách các môn học

**Code:**

**MonHocControler.cs**

//

// GET: /MonHoc/

public ActionResult Index()

{

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

}

**Index.cshtml**( Giao diện chức năng in danh sách môn học )

@model IEnumerable<SEProject.Models.MonHoc>

@{

ViewBag.Title = "Môn học";

}

<h2>Danh sách môn học</h2>

<p>

@Html.ActionLink("Tạo mới", "Create") |

@Html.ActionLink("Tìm kiếm", "SearchMonHoc")

</p>

<table >

<tr align="center">

<td>

<h4>Mã môn học</h4>

</td>

<td>

<h4>Tên môn học</h4>

</td>

<td>

<h4>Số tín chỉ</h4>

</td>

<td>

<h4>Môn tiên quyết</h4>

</td>

<th></th>

</tr>

@foreach (var item in Model) {

<tr align="center">

<td>

@Html.DisplayFor(modelItem => item.maMonHoc)

</td>

<td>

@Html.DisplayFor(modelItem => item.tenMonHoc)

</td>

<td>

@Html.DisplayFor(modelItem => item.soTinChi)

</td>

<td>

@Html.DisplayFor(modelItem => item.maMonKienQuyet)

</td>

<td>

@Html.ActionLink("Chỉnh sửa", "Edit", new { id=item.Id }) |

@Html.ActionLink("Chi tiết", "Details", new { id=item.Id }) |

@Html.ActionLink("Xóa", "Delete", new { id=item.Id })

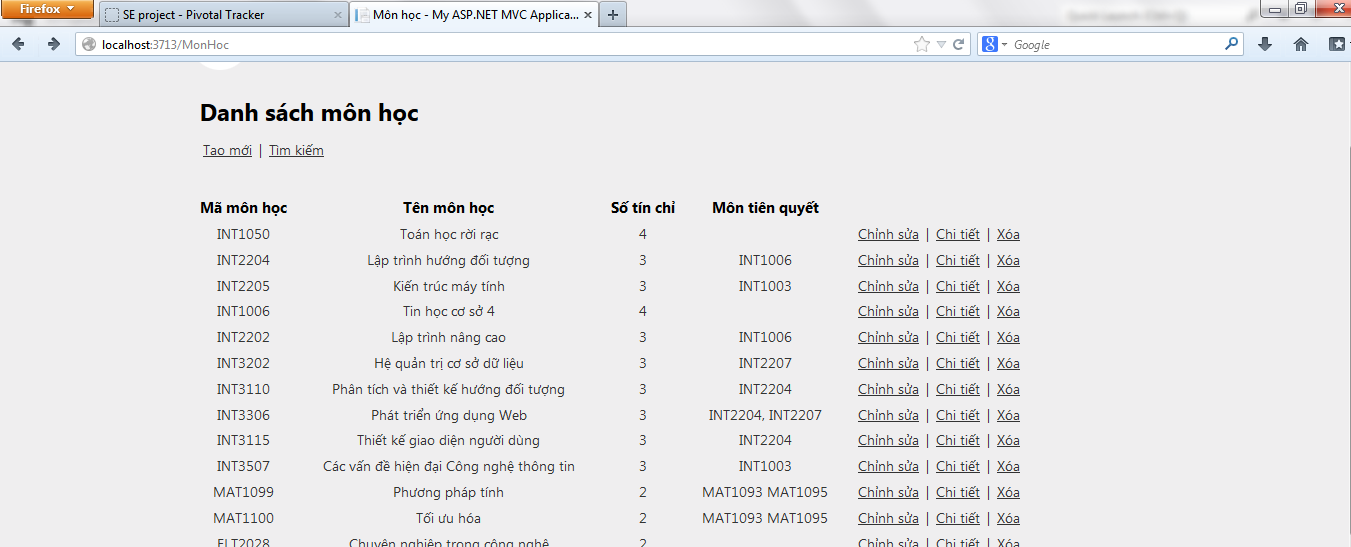
</td>

</tr>

}

</table>

**Screenshots:**



Hình 8 – Danh sách môn học

1. *Chức năng tìm kiếm thông tin môn học*

**User story:** Tìm kiếm môn học

**Tác giả:** Thái Mạnh Tuấn

Mô tả nội dung: Là User tôi muốn tìm kiếm môn học theo tên, theo mã môn học

**Task:**

- Viết chức năng tìm kiếm môn học theo tên môn học hoặc mã môn học

- In danh sách tìm được

**Code:**

**MonHocControler.cs:**

//Tim kiem mon hoc theo ten mon hoc va ma mon hoc

public ActionResult SearchMonHoc(string searchmh)

{

var monhocs = from m in db.monHocs

select m;

if (!String.IsNullOrEmpty(searchmh))

{

monhocs = monhocs.Where(s => s.tenMonHoc.Contains(searchmh)

|| s.maMonHoc.Contains(searchmh));

}

return View(monhocs);

}

**SearchMonHoc.cshtml**( Giao diện chức năng tìm kiếm môn học )

@model IEnumerable<SEProject.Models.MonHoc>

@{

ViewBag.Title = "Search";

}

<h2>Tìm kiếm môn học</h2>

<p>

@using (Html.BeginForm("SearchMonHoc", "MonHoc", FormMethod.Get))

{

<p> Nhập tên môn: @Html.TextBox("searchmh")

<input type="submit" value="Tìm kiếm" /></p>

}

</p>

<table>

<tr align ="center">

<td>

<h4>Mã môn học</h4>

</td>

<td>

<h4>Tên môn học</h4>

</td>

<td>

<h4>Số tín chỉ</h4>

</td>

<td>

<h4>Môn tiên quyết</h4>

</td>

<th></th>

</tr>

@foreach (var item in Model) {

<tr align ="center">

<td>

@Html.DisplayFor(modelItem => item.maMonHoc)

</td>

<td>

@Html.DisplayFor(modelItem => item.tenMonHoc)

</td>

<td>

@Html.DisplayFor(modelItem => item.soTinChi)

</td>

<td>

@Html.DisplayFor(modelItem => item.maMonKienQuyet)

</td>

<td>

@Html.ActionLink("Chi tiết", "Details", new { id=item.Id })

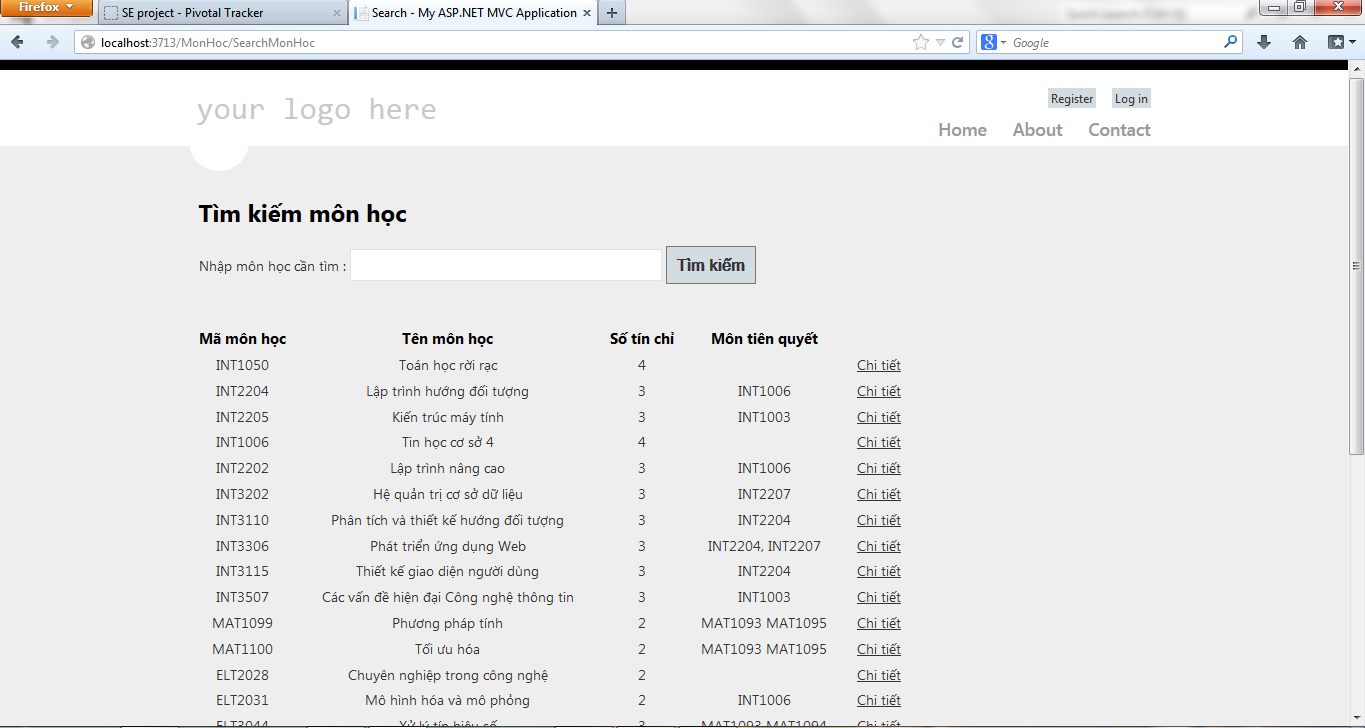
</td>

</tr>

}

</table>

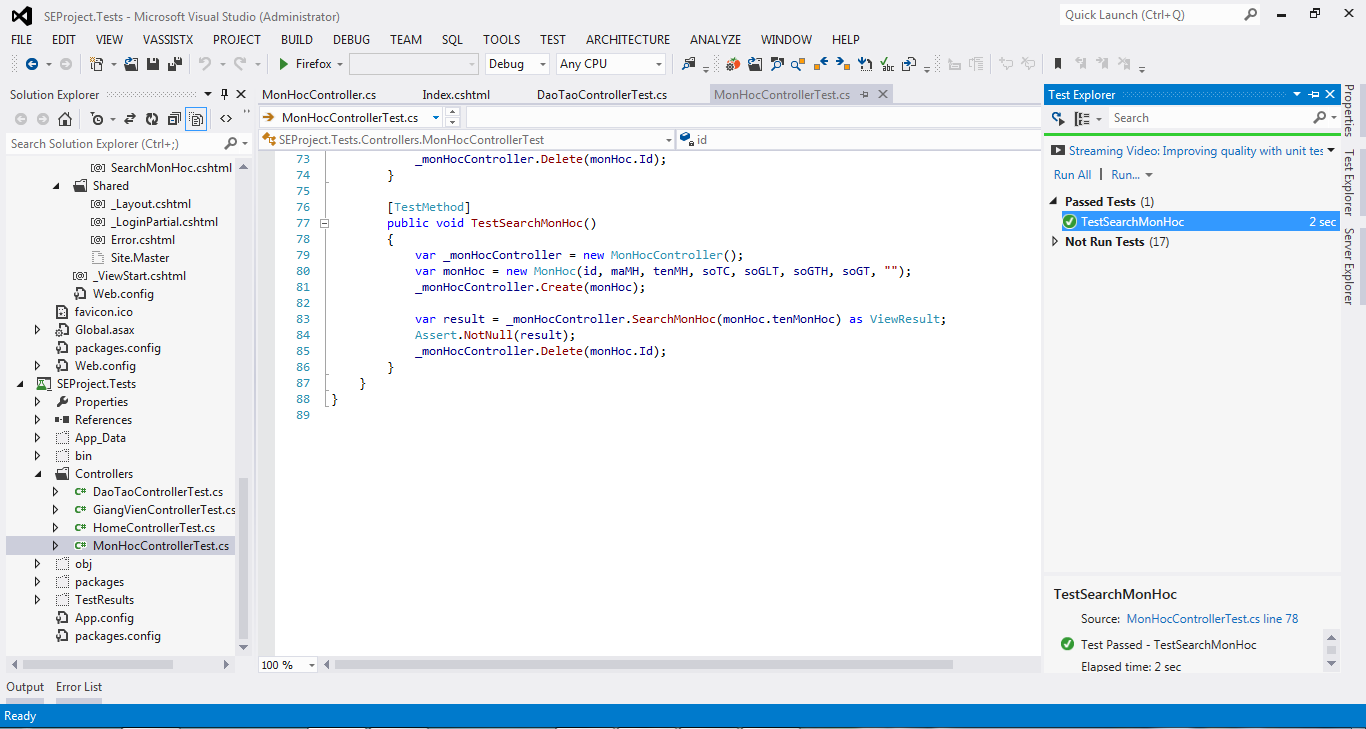
**Screenshots:**



Hình 9 – Giao diện chức năng tìm kiếm môn học theo tên và

mã môn học

Test chức năng tìm kiếm môn học:



Hình 10 – Test chức năng tìm kiếm môn học

1. *Chức năng xây dựng chương trình đào tạo*

**User story:** Xây dựng chương trình đào tạo

**Tác giả:** Tạ Đức Việt

**Mô tả:**

Là User tôi muốn xây dựng chương trình đào tạo theo từng năm và ngành học.Trên trang xây dựng chương trình đào tạo có các mục như môn học, giảng viên, năm, ngành để người sử dụng thêm vào cơ sở dữ liệu mới.

**Task:**

- Xây dựng giao diện chức năng tạo mới 1 môn học với thông tin về giảng viên, năm, ngành

- Tạo droplist để lựa chọn thông tin khi tạo mới

**Code:**

**DaoTaoControler.cs:**

// GET: /DaoTao/Create

public ActionResult Create()

{

var dsMonHoc = new List<string>();

var monHocQry = from dM in dbMH.monHocs

orderby dM.tenMonHoc

select dM.tenMonHoc;

dsMonHoc.AddRange(monHocQry.Distinct());

ViewBag.listMonHoc = new SelectList(dsMonHoc);

var dsGiangVien = new List<string>();

var giangVienQry = from dG in dbGV.giangViens

orderby dG.hoTen

select dG.hoTen;

dsGiangVien.AddRange(giangVienQry.Distinct());

ViewBag.listGiangVien = new SelectList(dsGiangVien);

var dsNganhHoc = new List<string>();

dsNganhHoc.Add("Công nghệ thông tin");

dsNganhHoc.Add("Khoa học máy tính");

dsNganhHoc.Add("Hệ thống thông tin");

dsNganhHoc.Add("Mạng máy tính");

ViewBag.listNganhHoc = new SelectList(dsNganhHoc);

return View();

}

//

// POST: /DaoTao/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(string tenMonHoc, string tenGiangVien,

int khoaHoc,string nganhHoc)

{

var daotao = new DaoTao(tenMonHoc, tenGiangVien, khoaHoc, nganhHoc);

if (ModelState.IsValid)

{

db.daoTaos.Add(daotao);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(daotao);

}

public ActionResult \_Create(DaoTao dT)

{

if (ModelState.IsValid)

{

db.daoTaos.Add(dT);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(dT);

}

**Create.cshtml**( Giao diện chức năng tạo mới )

@model SEProject.Models.DaoTao

@{

ViewBag.Title = "Tạo mới";

}

<h2>Tạo mới</h2>

@using (Html.BeginForm()) {

@Html.AntiForgeryToken()

@Html.ValidationSummary(true)

<fieldset>

<legend>DaoTao</legend>

<div class="editor-label">

<h4>Môn học</h4>

</div>

<p>@Html.DropDownList("tenMonHoc", (SelectList)ViewBag.listMonHoc)</p>

<div class="editor-label">

<h4>Giảng viên</h4>

</div>

<p>@Html.DropDownList("tenGiangVien",(SelectList)ViewBag.listGiangVien)</p>

<div class="editor-label">

<h4>Khóa</h4>

</div>

<div class="editor-field">

@Html.EditorFor(model => model.khoaHoc)

@Html.ValidationMessageFor(model => model.khoaHoc)

</div>

<div class="editor-label">

<h4>Ngành</h4>

</div>

<p>@Html.DropDownList("nganhHoc",(SelectList)ViewBag.listNganhHoc)</p>

<p>

<input type="submit" value="Tạo" />

</p>

</fieldset>

}

<div>

@Html.ActionLink("Quay lại danh sách", "Index")

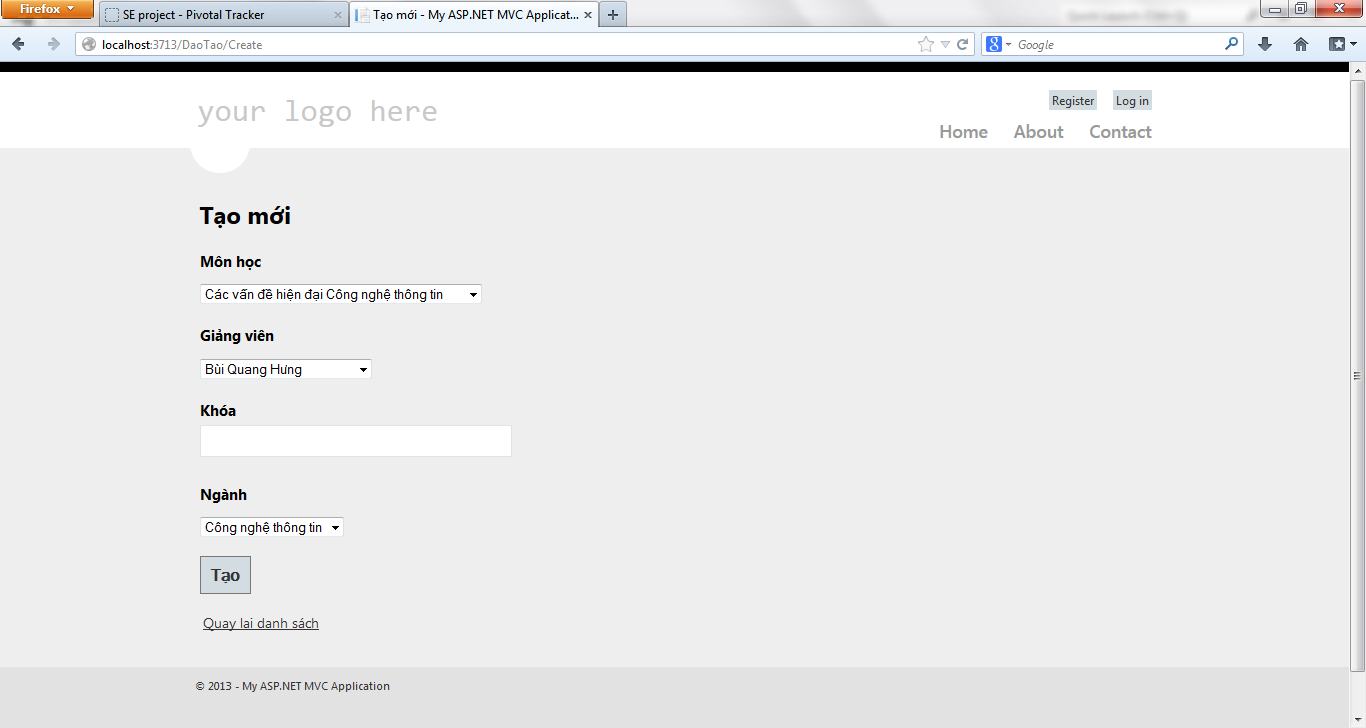
</div>

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

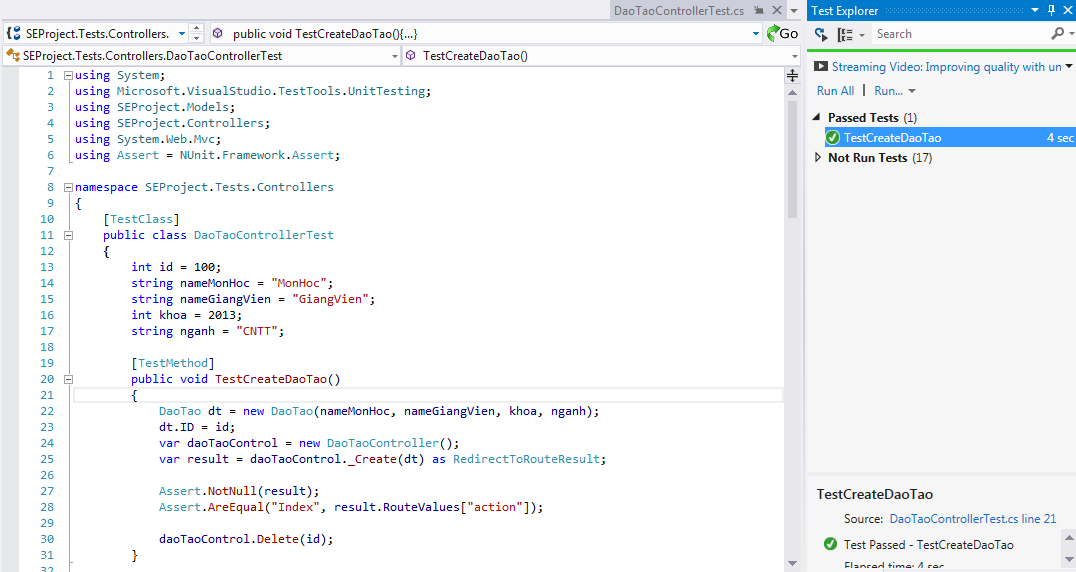
}

**Screenshots:**



Hình 11 – Giao diện chức năng tạo mới

Test chức năng xây dựng chương trình đào tạo:



Hình 12 – Test các chức năng

1. *Chức năng xem bảng chương trình đào tạo theo từng năm học, ngành học*

**User story:** In bảng chương trình đào tạo

**Tác giả:** Thái Mạnh Tuấn

Mô tả nội dung:

Là User tôi muốn xem bảng chương trình đào tạo theo năm học và ngành học

**Task:**

- In bảng chương trình đào tạo

- Tạo chức năng in bảng theo năm học, ngành học

**Code:**

- In bảng chương trình đào tạo

DaoTaoControler.cs:

public ActionResult Index()

{

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

}

**Index.cshtml**( Giao diện bảng chương trình đào tạo )

@model IEnumerable<SEProject.Models.DaoTao>

@{

ViewBag.Title = "Index";

}

<h2>Index</h2>

<p>

@Html.ActionLink("Tạo mới", "Create")

</p>

<table>

<tr align = "center">

<th>

<h4>Môn học</h4>

</th>

<th>

<h4>Giảng viên</h4>

</th>

<th>

<h4>Khóa</h4>

</th>

<th>

<h4>Ngành</h4>

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.tenMonHoc)

</td>

<td>

@Html.DisplayFor(modelItem => item.tenGiangVien)

</td>

<td>

@Html.DisplayFor(modelItem => item.khoaHoc)

</td>

<td>

@Html.DisplayFor(modelItem => item.nganhHoc)

</td>

<td>

@Html.ActionLink("Chỉnh sửa", "Edit", new { id=item.ID }) |

@Html.ActionLink("Chi tiết", "Details", new { id=item.ID }) |

@Html.ActionLink("Xóa", "Delete", new { id=item.ID })

</td>

</tr>

}

</table>

- Tạo chức năng in bảng theo năm học, ngành học:

**SearchDaoTao.cshtml**( Giao diện chức năng in )

@model IEnumerable<SEProject.Models.DaoTao>

@{

ViewBag.Title = "Chương trình đào tạo";

}

<h2>Chương trình đào tạo</h2>

<p>

@using (Html.BeginForm()){

<p> Ngành: @Html.DropDownList("nganh", (SelectList)ViewBag.listChuyenNganh)

Khóa: @Html.DropDownList("nam", (SelectList)ViewBag.listKhoaHoc)

<input type="submit" value="Xem" /></p>

}

</p>

<table>

<tr align = "center">

<th>

<h4>Môn học</h4>

</th>

<th>

<h4>Giảng viên</h4>

</th>

<th>

<h4>Khóa</h4>

</th>

<th>

<h4>Ngành</h4>

</th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.tenMonHoc)

</td>

<td>

@Html.DisplayFor(modelItem => item.tenGiangVien)

</td>

<td>

@Html.DisplayFor(modelItem => item.khoaHoc)

</td>

<td>

@Html.DisplayFor(modelItem => item.nganhHoc)

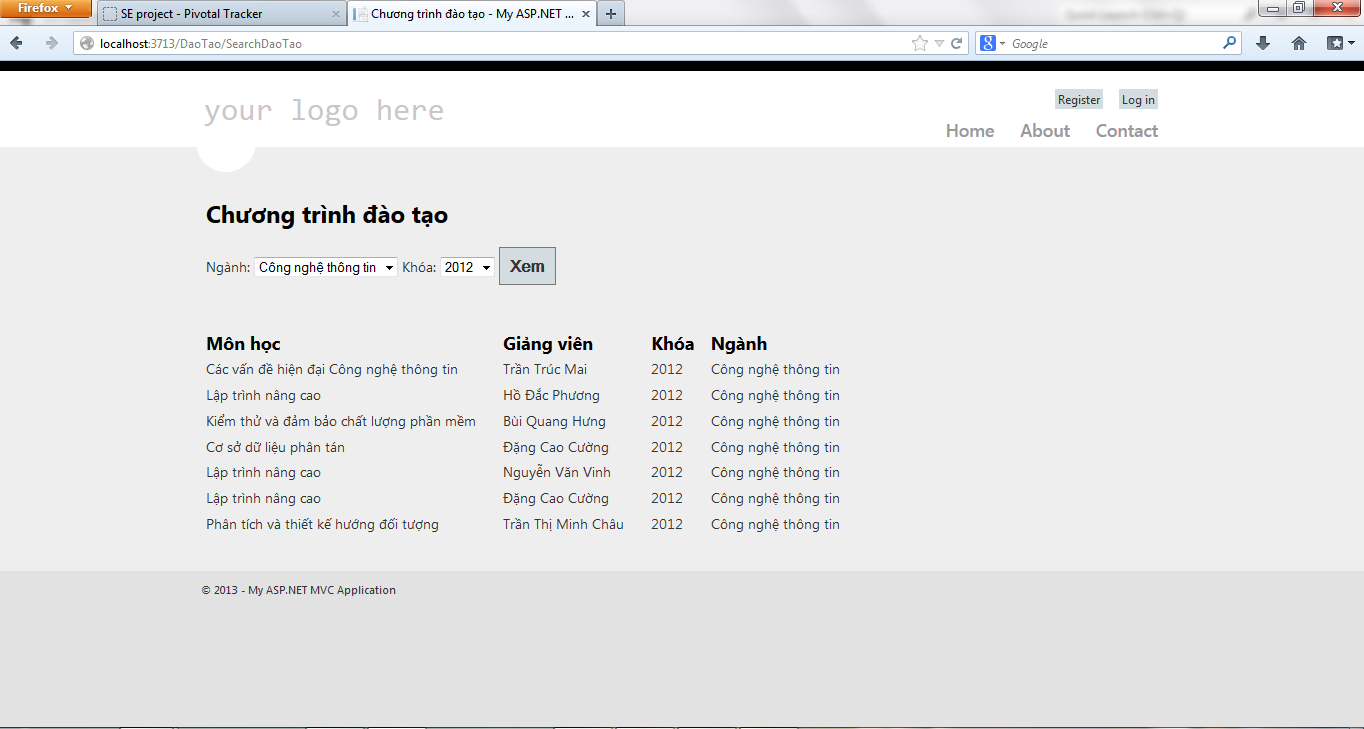
</td>

</tr>

}

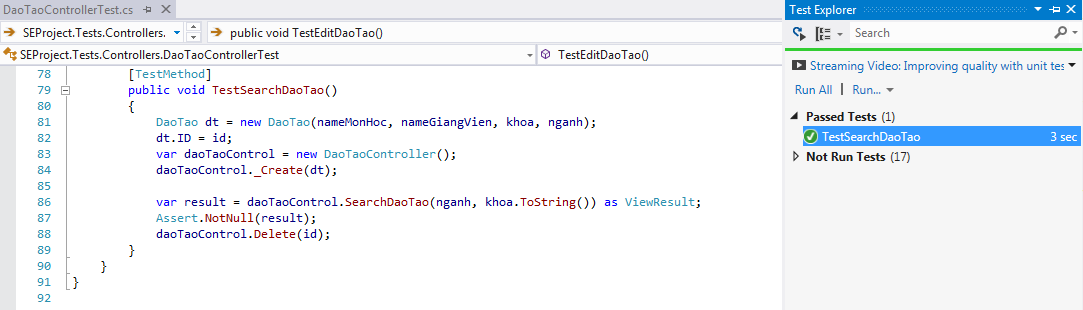
</table>

**Screenshots:**

****

Hình 13 – Bảng chương trình đào tạo theo ngành và chức năng filter theo ngành và khóa

Test chức năng in theo năm học, ngành học:



Hình 14 – Test chức năng in theo năm học, ngành học

**Nhận xét về phương pháp Agile:**

Phương pháp Agile là một phương pháp hay trong việc phát triển phần mềm:

- Nó rất linh hoạt, đa dạng và có hiệu quả cao.

- Phương pháp Agile thích hợp cho các dự án luôn phải thay đổi theo yêu cầu khách hàng. Qua mỗi lần lặp, ta có các bản mẫu đưa cho khách hàng kiểm tra, nếu chức năng mới thêm vào đó chưa đạt yêu cầu của khách hàng ta có thể kịp thời sửa đổi.

- Nhóm phát triển đối thoại trực tiếp với khách hàng để hiểu rõ hơn cái khách hàng cần.

- Bằng việc phát triển phần mềm qua phương pháp Agile ta có thể thấy rõ tốc độ phát triển của phần mềm, ước lượng được thời gian hoàn thành dự án.

- Các cá nhân trong đội ngũ phát triển có thể tự do lựa chọn công việc phù hợp với mình chứ không cần phải phân công công việc. Các nhóm tự quyết định, theo dõi tiến độ, giải quyết các vấn đề mà không chờ mệnh lệnh của các cấp quản lý.

- Luôn có sự trao đổi giữa các cá nhân, các nhóm thông qua các buổi họp tập trung hàng ngày. Tại đây, tất cả các thành viên được yêu cầu nói rõ cho nhóm của mình biết mình đã làm gì, đang làm gì, sắp làm gì và đang gặp phải khó khăn nào trong quá trình làm việc. Từ đó nhóm luôn luôn nắm được tình hình công việc của mình, có các hành động thích hợp để vượt qua các trở lực để thực hiện thành công mục tiêu của dự án.

- Tập trung vào sản phẩm, loại bỏ đi các công việc dư thừa không trực tiếp mang lại giá trị cho sản phẩm.