

Lab 05:

# LẬP TRÌNH WEB MVC5

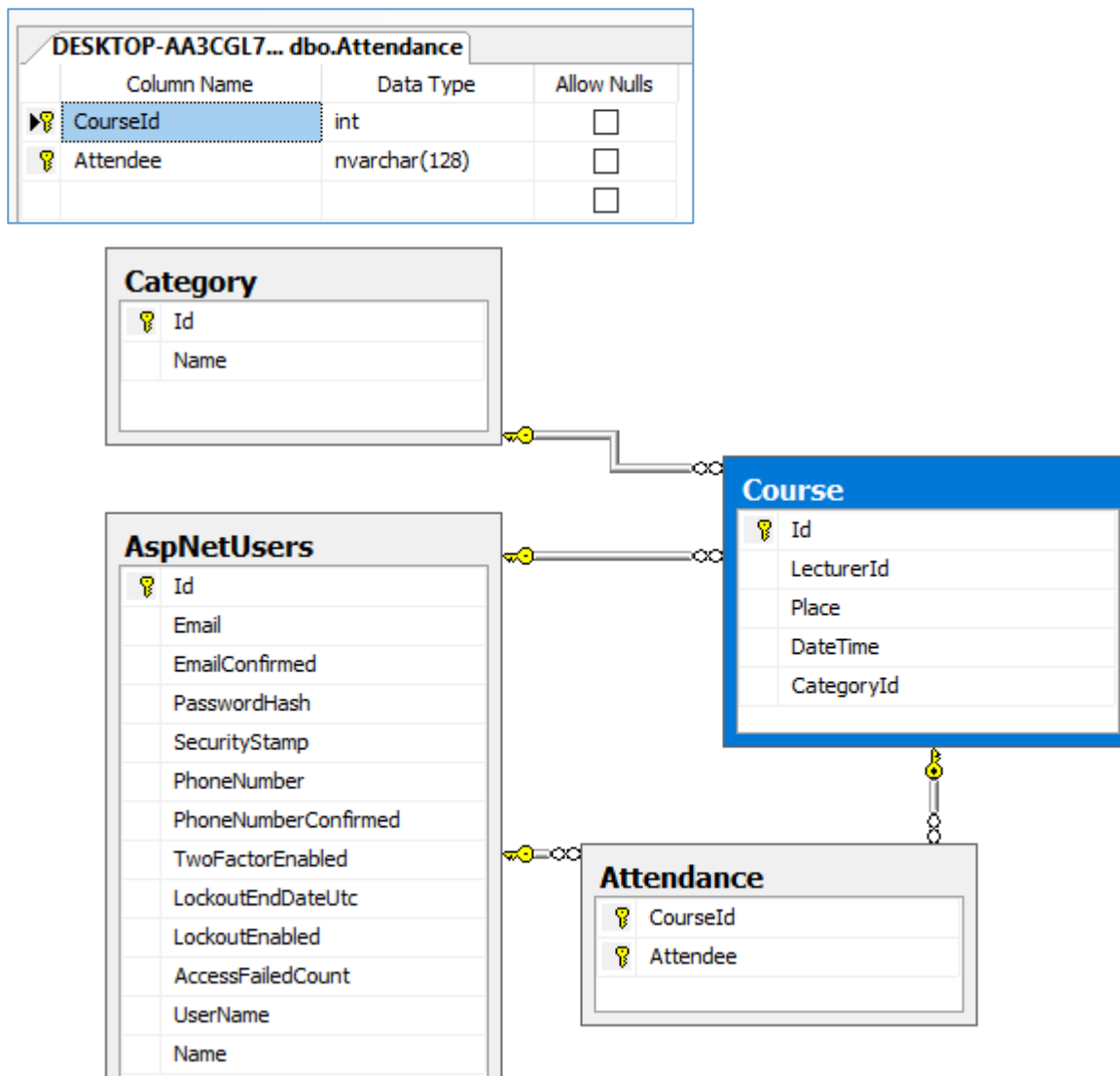
## Xây Dựng Project BigSchool Phần 2

### 1. MỤC TIÊU: Hoàn thành các chức năng BigSchool (tiếp theo)

- ✓ Override bootstrap default CSS, Chính sửa trang Home theo yêu cầu
- ✓ Thêm Menu để xem khóa học của người dùng, khóa học tham dự...
- ✓ Thêm bảng tham dự khóa học (**Attendance**). Xây dựng chức năng học viên đăng kí khóa học
- ✓ Đăng kí tham gia khóa học và sử dụng Web API
- ✓ Quản lý khóa học ( Xóa, Cập nhật Course) của người tạo khóa học

### 2. Thêm bảng tham gia vào CSDL:

Quan hệ nhiều – nhiều giữa hai bảng User và Course, Tạo bảng mới Attendance



### 3. Các yêu cầu cần thực hiện

**Yêu cầu 1: Hiển thị các khóa học sắp tới (Home) theo mẫu** (Trang 52 giáo trình thực hành)

Home	Add Course
10 24	NGUYEN HUY CUONG Business
09 18	NGUYEN HUY CUONG Marketing

**Yêu cầu 2: Thay đổi giao diện, Thêm *menu*** (Kết quả như Trang 45)

**Course I'm Going:** Các khóa học mà user đã đăng ký tham gia

**Lecture I'm Following:** Các giảng viên mà user đã theo dõi (Y/c này phần 3 tuy nhiên ở phần này các bạn thực hiện giao diện)

Hiện tại:

Home	Add Course	Hello huycuong719@gmail.com! Log off
10 24	NGUYEN HUY CUONG Business	
09 18	NGUYEN HUY CUONG Marketing	

Mong muốn:

Home	Add Course	huycuong719@gmail.com ▾
10 24	NGUYEN HUY CUONG Business	Course I am going Lecture I am Following Log off
09 18	NGUYEN HUY CUONG Marketing	

**Yêu cầu 3: Thêm Going?** để đăng kí tham gia khóa học (cho user đã đăng nhập và chưa tham gia khóa học)

Home	Add Course	user1@gmail.com ▾
10 24	NGUYEN HUY CUONG Business	Going?
09 18	NGUYEN HUY CUONG Marketing	Going?

Nếu chưa đăng nhập không hiển thị Going?

**Yêu cầu 4: Khi click vào going? cho khóa học mà chưa đăng ký**

Home	Add Course	user1@gmail.com ▾
10 24	NGUYEN HUY CUONG Business	Going?
09 18	NGUYEN HUY CUONG Marketing	Going?

Sau khi user1 click vào Going? (Đăng ký tham gia khóa học) cần thực hiện

- Thêm khóa học được đăng kí vào CSDL ( bảng **Attendance** )

**Ví dụ: user1** (có AspNetUsers.id = dd109436-818d-4244-ba06-79d8881ace68) đăng ký lớp “Business” ngày 24/10 của NGUYEN HUY CUONG (course.Id = 1) thì sẽ tạo THÊM dòng trong Attendance

	CourseId	Attendee
▶	1	dd109436-818d-4244-ba06-79d8881ace68

- Thay đổi màu và Going? để thể hiện đã đăng ký khóa học

**Yêu cầu 5:** Giữ đúng trạng thái đăng ký khóa học với CSDL (Với khóa học đã đăng ký sẽ thể hiện Going và màu xanh , chưa đăng ký Going? và không có màu)

**Yêu cầu 6:** Xem danh sách khóa học đăng ký tham dự [Trang 65 giáo trình]

Ở menu khi click vào Course I'm going -> Sẽ hiện thị ra các khóa học mà login user đã đăng ký.

Khi click vào Course I am going: sẽ hiện thị các khóa học đã và đang đăng ký tham dự

**Yêu cầu 7:** Khóa học của tôi [My Upcoming Courses – Trang 67 giáo trình ]

Khi click vào **My Upcoming Courses** sẽ xem danh sách khóa học được tạo bởi tài khoản người dùng đăng nhập

**Yêu cầu 8:** Quản lý khóa học (Cập nhật, Xóa– Giáo trình trang 69, phải hoàn thành yêu cầu 8).

# Mine

10 NGUYEN HUY CUONG  
24 Business

[Edit](#) [Delete](#)

09 NGUYEN HUY CUONG  
18 Marketing

Sử dụng Xóa tạm thay vì xóa luôn từ cơ sở dữ liệu (**Trang 74**, Thêm **IsCanceled** vào Course, và khi xóa = True, mặc định là False)

## Yêu cầu 9:

Khi 1 user click vào khóa học đã đăng ký rồi [Going], hiển thị thông báo “Khóa học này đã đăng ký rồi, vui lòng chọn khóa học khác!”

Đối với các khóa học do chính mình tạo ra, không hiển thị Going? [Không thể đăng ký tham gia chính khóa học mình tạo ra]

## 4. Một số hướng dẫn thực hiện yêu cầu ☺

### 4.1 Hiển thị các khóa học sắp tới

- Chỉnh sửa lại phân giao diện ở View ( sử dụng class css như “course“, “date” “details”.. )

```
Index.cshtml*
1 @model IEnumerable<BigSchool.Models.Course>
2
3 @{
4     ViewBag.Title = "Index";
5     Layout = "~/Views/Shared/_Layout.cshtml";
6 }
7
8 <div class="row">
9     <ul class="course">
10         @foreach (var course in Model)
11         {
12             <li>
13                 <div class="date">
14                     <div class="month">
15                         @course.DateTime.ToString("MM")
16                     </div>
17                     <div class="day">
18                         @course.DateTime.ToString("dd")
19                     </div>
20                 </div>
21                 <div class="details">
22                     <span class="lecture">
23                         @course.Name
24                     </span>
25                     <span class="category">
26                         @course.Category.Name
27                     </span>
28                 </div>
29             </li>
30         }
31     </ul>
32 </div>
33
34
```

- Thêm css cho class “course”, “date”, “details” ... ở Site.css

```
/* course layout*/
.course
{
    list-style:none;
}
```

```
.course > li
{
    position:relative;
    margin-bottom:30px;
}

.course > li .date {
    text-align : center;
    background-color: #d41d50;
    color: white;
    width:60px;
    border-radius: 8px;
}

.course > li .details
{
    position:absolute;
    top:0;
    left:70px;
}
```

#### 4.2 Thêm menu ( Kết quả như trang 45)

- Tạo menu trong phần login ( thay đổi ở **Views/Shared/\_\_\_LoginPartial.cshtml**) bằng cách thẻ <ul> <li>

```
<ul class="nav navbar-nav navbar-right">
    <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-
expanded="false">@User.Identity.GetUserName() <span class="caret"></span></a>
        <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton">
            <li> <a href ="#">Course I am going</a></li>
            <li> <a href ="#">Lecture I am Following</a></li>
            <li role="separator" class="divider"> </li>
            <li><a
href="javascript:document.getElementById('logoutForm').submit()">Log off</a></li>
        </ul>
    </li>
</ul>
```

Kết quả thay đổi



Chỉnh lại màu cho navbar, thêm css

```
.navbar-inverse {
    background-color: #d41d50;
    border-color: #fbf6f6;
}

.navbar-inverse .navbar-nav > li > a {
    color: white;
}

.navbar-inverse .navbar-brand {
    color: white;
}
```

}

## Kết quả



### 4.3 Giao diện đăng kí tham gia khóa học khi chưa login

Khi chưa login sẽ không hiện thị phần Going? nên chúng ta sẽ cần kiểm tra việc login và truyền qua View. Ở đây chúng ta có thể sử dụng ViewBag

- Viết code để xác định việc đăng nhập này ở Home controller trước khi gọi View

```
0 references
public ActionResult Index()
{
    BigSchoolContext context = new BigSchoolContext();
    var upcomingCourse = context.Courses.Where(p => p.DateTime > DateTime.Now).OrderBy(p => p.DateTime).ToList();
    //lấy user login hiện tại đưa vào ViewBag -> để truyền qua view
    //Nếu giá trị = null -> nghĩa là chưa đăng nhập
    ViewBag.LoginUser = User.Identity.GetUserId();
    foreach (Course i in upcomingCourse)
    {
        ApplicationUser user = System.Web.HttpContext.Current.GetOwinContext().GetUserManager<ApplicationUserManager>().
            FindById(i.LecturerId);
        i.Name = user.UserName;
    }
    return View(upcomingCourse);
}
```

- Thay đổi lại view có thêm phần kiểm tra điều kiện và đưa Going? vào

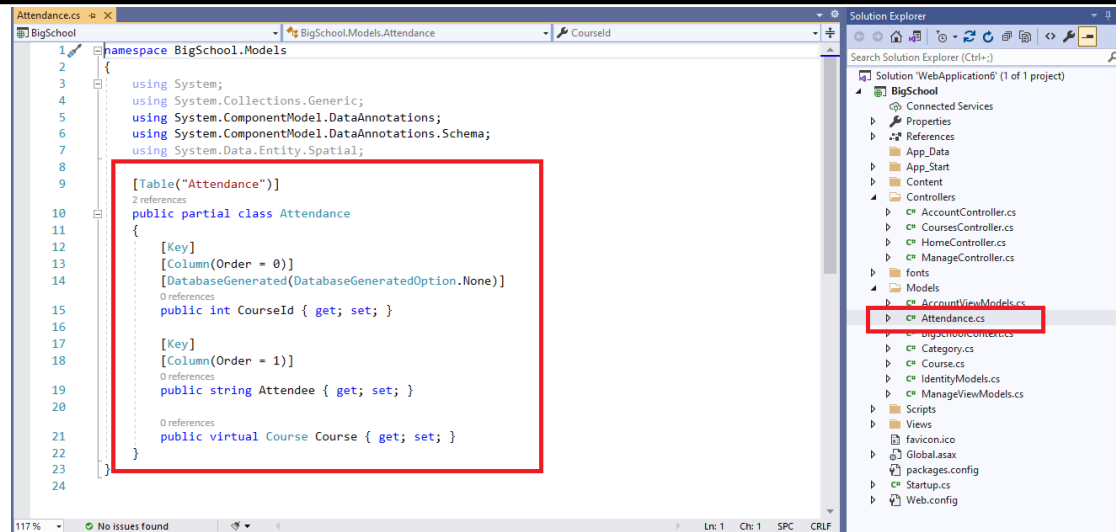
```
<div class="details">
    <span class="lecture">
        @course.Name
    </span>
    <br />
    <span class="category">
        @course.Category.Name
        @if (ViewBag.LoginUser != null)
        {
            <button data-course-id="@course.Id" class="btn btn-default btn-sm js-tongle-attendance">Going?</button>
        }
    </span>
</div>
```

Giải thích: nếu đăng nhập thì mới cho phép Render thẻ button

Kiểm tra lại kết quả khi chưa login/ và sau khi login

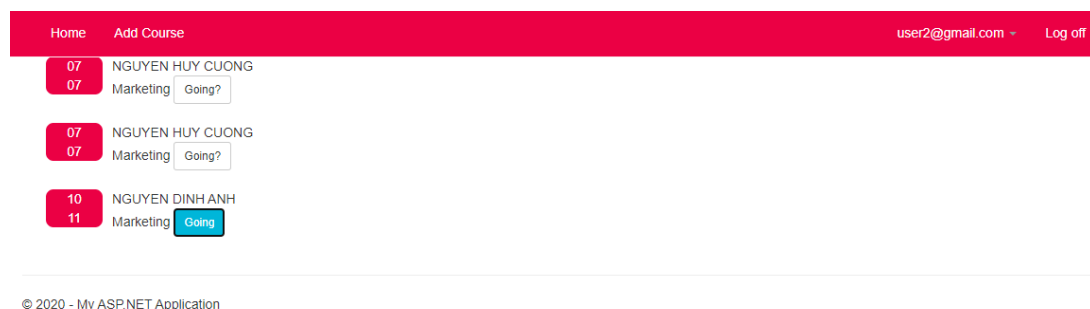
### 4.4 Đăng kí tham gia khóa học (Going?)

- Việc đăng kí khóa học sẽ liên quan đến bảng Attendance, như vậy cần phải điều chỉnh models phù hợp với cơ sở dữ liệu mới (có thể code thêm, xóa tạo lại nếu như không thực hiện code gì thêm ở model, hoặc quan sát một số thay đổi khi tạo từ đầu để đưa sang Models cũ cho phù hợp) (Đoạn này nhờ GV trên lớp làm 1 lần là hiểu, chỉ kéo thả 1 xíu 😊)



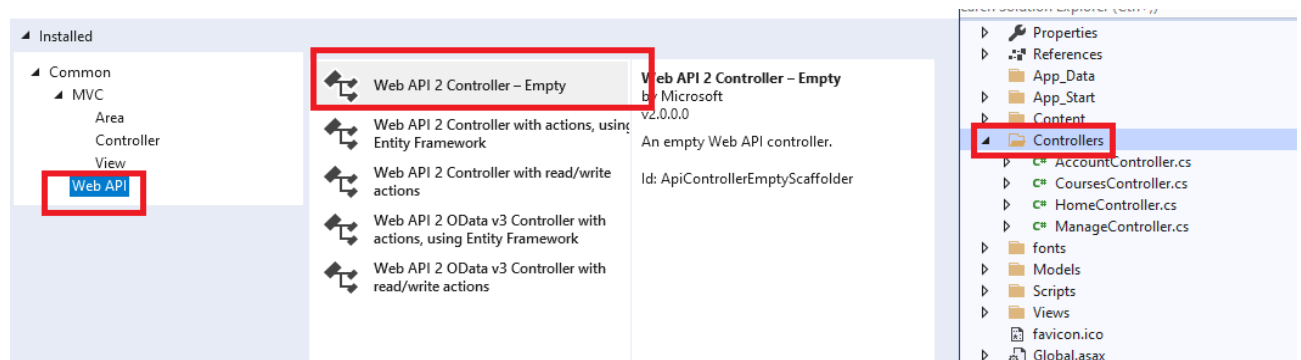
Lúc này: Models của chúng ta đã phù hợp với sự thay đổi của CSDL. Tiến hành thực hiện tiếp

- Khi click vào going? Sẽ tô màu tham gia khóa học (màu xanh), và Text được chuyển thành Going,

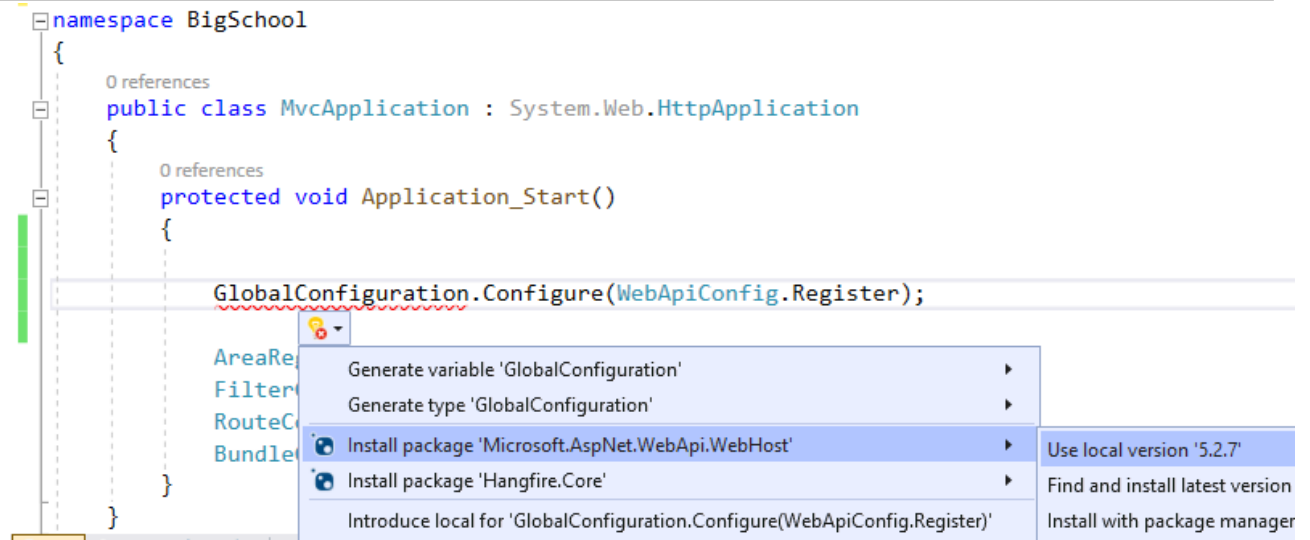


Ở đây, khi click vào Going? chúng ta sẽ gọi tới Web API để thực hiện (thực hiện việc thêm vào cơ sở dữ liệu). Khi thành công sẽ thay đổi css của Going? [Ngoài cách gọi Web API vẫn có thể có 1 số cách khác]

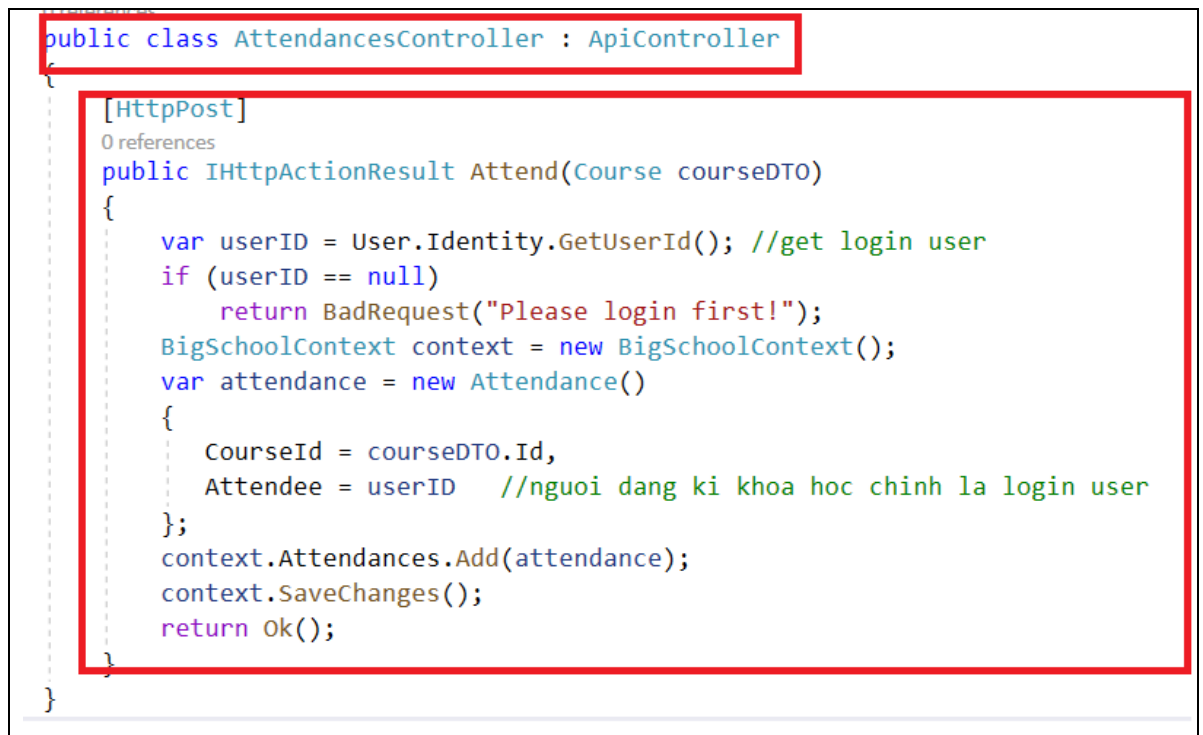
- Chọn Web API 2 Controller – Empty, đặt tên AttendancesController [Trang 55 giáo trình]



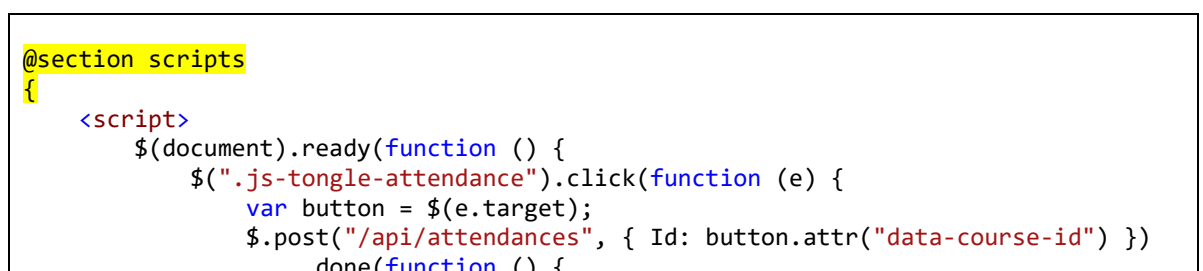
- Bổ sung đoạn code khởi tạo Web API vào Global.asax ( Do ban đầu template của chúng ta không chọn Web API). Có thể phải cài đặt



- Viết Web API : Hàm Attend ( nhận đối tượng Course truyền lên, và có chức năng sẽ thực hiện thêm mới vào CSDL - Thực chất chỉ sử dụng mỗi Id của khóa học, nên các bạn cũng có thể không nhất thiết truyền cả Course)



- Như vậy cần viết Script từ View để gọi hàm vừa viết trên ( Attend từ WebAPI có tham số truyền vào )
- Ở View của Home sử dụng Script để gọi web api ( Trang 63 giáo trình)





```

        alert("Đăng ký khóa học thành công!");
        button.removeClass("btn-default");
        button.addClass("btn-info");
        button.text("Going");
    }).fail(function () {
        alert("Something fail");
    });
    });
</script>
}

```

Giải thích: Gọi Web API **attendances** khi đối tượng có lớp **"js-tongle-attendance"** được click (ở đây là nút Going?), **trong đó có truyền** Id được lấy giá trị từ `button.attr("data-course-id")`

Khi thực hiện thành công: sẽ vào done:

Khi không thành công, hoặc lỗi sẽ vào fail

- Kết quả sau khi đăng ký khóa học



DESKTOP-AA3CGL7... dbo.Attendance		
	CourseId	Attendee
	1	8d156c4a-6a00-4066-8961-73752c...
▶*	NULL	NULL

#### 4.5 Giữ đúng trạng thái đăng ký khóa học với CSDL ( Với khóa học đã đăng ký sẽ thể hiện

**Going và màu xanh , chưa đăng ký Going? và không có màu)**

Ý tưởng thực hiện: Để xác định khóa học đã được đăng ký hay chưa chúng ta cần phải biết loginUser và cũng như kiểm tra từ Attendance

- Thêm 1 biến **isShowGoing** dạng False/ True giúp việc kiểm tra khóa học này đã được (người đăng nhập) đăng ký hay chưa. Mặc định **isShowGoing** = False
- Duyệt tất cả các khóa học để kiểm tra liệu đã có Course.Id này ở Attendance chưa? Nếu có **isShowGoing** = True
- Ở View lại kiểm tra 1 lần nữa để Render

Thực hiện

## Models/Course.cs

```
public bool isShowGoing = false;
```

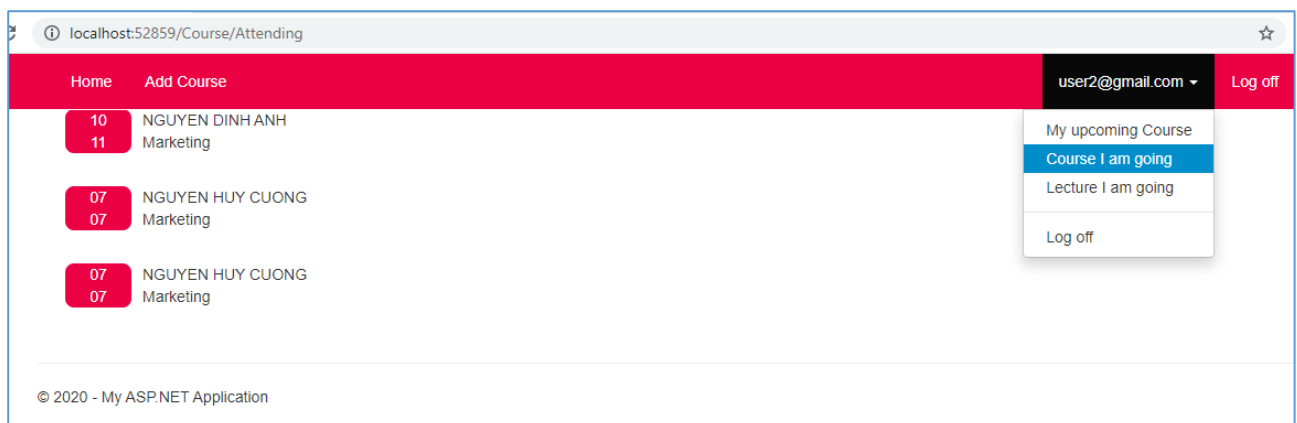
## HomeController/ Index

```
0 references
public ActionResult Index()
{
    BigSchoolContext context = new BigSchoolContext();
    var upcomingCourse = context.Courses.Where(p => p.DateTime > DateTime.Now).OrderBy(p => p.DateTime).ToList();
    //lấy user login hiện tại đưa vào ViewBag để truyền qua view //Nếu giá trị = null -> nghĩa là chưa đăng nhập
    var loginUser = User.Identity.GetUserId();
    ViewBag.LoginUser = loginUser;
    foreach (Course i in upcomingCourse)
    {
        ApplicationUser user = System.Web.HttpContext.Current.GetOwinContext().GetUserManager<ApplicationUserManager>().
            FindById(i.LecturerId);
        i.Name = user.UserName;
        //ktra user đã chưa tham gia khóa học
        Attendance find = context.Attendances.FirstOrDefault(p => p.CourseId == i.Id && p.Attendee == loginUser);
        if (find == null)
            i.isShowGoing = true;
    }
    return View(upcomingCourse);
}
```

Và ở View nếu isShowGoing = True thì sẽ Render Going?

```
<span class="category">
    @course.Category.Name
    @if (ViewBag.LoginUser != null)
    {
        if (course.isShowGoing)
        {
            <button data-course-id="@course.Id" class="btn btn-default btn-sm js-tongle-attendance">Going</button>
        }
        else
        {
            <button data-course-id="@course.Id" class="btn btn-info btn-sm js-tongle-attendance">Going</button>
        }
    }
</span>
```

## 4.6 Xem các khóa học tham dự ( Click vào Course I am going)



Hướng Dẫn: Tạo Action “Attending” trong “Course” controller

```

public ActionResult Attending()
{
    BigSchoolContext context = new BigSchoolContext();
    var userID = User.Identity.GetUserId(); //get login user
    //lấy danh sách khóa học mà userlogin đó tham dự (ở đây mỗi chỉ lấy được id)
    var listAttendances = context.Attendances.Where(p => p.Attendee == userID).ToList();
    var courses = new List<Course>();
    //Tìm chi tiết khóa học từ listAttendances ( mã khóa học, tên GV phải truy cập từ AspNetUser.Name )
    foreach (Attendance temp in listAttendances)
    {
        Course objCourse = temp.Course;
        objCourse.Name = System.Web.HttpContext.Current.GetOwinContext().GetUserManager<ApplicationUserManager>().FindById(objCourse.LecturerId).Name;
        courses.Add(objCourse);
    }
    return View(courses);
}

```

Tạo View (Template List khóa học) sau đó điều chỉnh lại View theo yêu cầu ( giống View của Home trước đây, nên copy chỉnh sửa cho nhanh)

```

Attending.cshtml
6
7 <h2>Attending</h2>
8 <div class="row">
9     <ul class="course">
10         @foreach (var course in Model)
11         {
12             <li>
13                 <div class="date">
14                     <div class="month">
15                         @course.DateTime.ToString("MM")
16                     </div>
17                     <div class="day">
18                         @course.DateTime.ToString("dd")
19                     </div>
20                 </div>
21                 <div class="details">
22                     <span class="lecture">
23                         @course.Name
24                     </span>
25                     <br />
26                     <span class="category">
27                         @course.Category.Name
28                     </span>
29                 </div>
30             </li>
31         }
32     </ul>
33 </div>

```

Điều chỉnh css phù hợp

Ở LoginPartial.cshtml: thêm Url.Action để gọi vào đúng controller/action (Course / Attending) khi click vào menu

```

LoginPartial.cshtml
1 using Microsoft.AspNet.Identity
2 @if (Request.IsAuthenticated)
3 {
4     using (Html.BeginForm("LogOff", "Account", FormMethod.Post, new { id = "logoutForm", @class = "navbar-right" }))
5     {
6         @Html.AntiForgeryToken()
7
8         <ul class="nav navbar-nav navbar-right">
9             <li class="nav navbar-nav navbar-right">
10                 <li class="dropdown">
11                     <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="true">Course I am going</a></li>
12                     <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton">
13                         <li><a href="@Url.Action("Attending","Course")">Course I am going</a></li>
14                         <li><a href="#">Lecture I am Following</a></li>
15                         <li role="separator" class="divider"></li>
16                         <li><a href="javascript:document.getElementById('logoutForm').submit()">Log off</a></li>
17                     </ul>
18                 </li>
19             </ul>
20

```

#### 4.7 Xem danh sách khóa học được tạo bởi người dùng hiện tại( Click vào My upcoming

Course)

Hướng Dẫn: Tạo Action “Mine” trong “Course” controller (Tương tự chỉ khác câu truy vấn một chút).

[Home](#)
[Add Course](#)

10

24

NGUYEN HUY CUONG

Business

09

18

NGUYEN HUY CUONG

Marketing

nhcuong@gmail.com

My Upcoming courses

Course I am going

Lecture I am Following

Log off

#### 4.8 Quản lý khóa học (Xóa, cập Nhật)

- Sửa giao diện Mine.cshtml và css tương ứng (nếu có css rồi thì thôi)

```

<h2>Mine</h2>
<div class="row">
    <ul class="course">
        @foreach (var course in Model)
        {
            <li>
                <div class="date">
                    <div class="month">
                        @course.DateTime.ToString("MM")
                    </div>
                    <div class="day">
                        @course.DateTime.ToString("dd")
                    </div>
                </div>
                <div class="details">
                    <span class="lecture">
                        @course.Name
                    </span>
                    <br />
                    <span class="category">
                        @course.Category.Name
                    </span>
                </div>
                <div class="action">
                    <a href="">Edit</a>
                    <a href="">Delete</a>
                </div>
            </li>
        }
    </ul>
</div>
    
```

```

.course > li .lecture{
}
.course > li .category{
}
.course > li .action {
    display:none;
    left:170px;
    position:absolute;
}
.course > li:hover .action {
    display:block;
}
    
```

Kết quả giao diện khi đưa chuột hover vào

## Mine

10	NGUYEN HUY CUONG
24	Business
<a href="#">Edit</a> <a href="#">Delete</a>	
09	NGUYEN HUY CUONG
18	Marketing

- Thêm Action Edit ( khi có thông tin id là mã khóa học) trong CoursesController

```
[Authorize]
0 references
public ActionResult Edit(int id)
{
    BigSchoolContext context = new BigSchoolContext();
    var loginUser = User.Identity.GetUserId(); //get login user
    var course = context.Courses.FirstOrDefault(c => c.LecturerId == loginUser && c.Id == id);
    if (course == null)
        return HttpNotFound("Không tìm thấy khóa học");
    course.ListCategory = context.Categories.ToList(); //lấy danh sách loại khóa học
    return View("Create", course);
}
```

- Gọi Action vừa thực hiện

```
<div class="action">
    <a href="@Url.Action("Edit","Courses", new { id = course.Id})">Edit</a>
    <a href="">Delete</a>
</div>
```

Thực hiện việc xóa tạm thời thông tin khóa học:

- Thêm vào *IsCanceled* vào Course ( Ở cả CSDL và Models)

```
[Table("Course")]
12 references
public partial class Course
{
    4 references
    public int Id { get; set; }

    [Required]
    [StringLength(128)]
    5 references
    public string LecturerId { get; set; }

    [Required]
    [StringLength(255)]
    0 references
    public string Place { get; set; }

    2 references
    public DateTime DateTime { get; set; }

    0 references
    public int CategoryId { get; set; }

    0 references
    public bool? IsCanceled { get; set; } //tương ứng với sự thay đổi DB
```

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
LecturerId	nvarchar(128)	<input type="checkbox"/>
Place	nvarchar(255)	<input type="checkbox"/>
DateTime	datetime	<input type="checkbox"/>
CategoryId	int	<input type="checkbox"/>
IsCanceled	bit	<input checked="" type="checkbox"/>

- Viết tiếp hàm để cập nhật ( IsCanceled = True)

```
[Authorize]
0 references
public ActionResult Mine()
{
    var loginUser = User.Identity.GetUserId(); //get login user
    ApplicationUser user = System.Web.HttpContext.Current.GetOwinContext().GetUserManager<ApplicationUserManager>().FindById(loginUser)
    BigSchoolContext context = new BigSchoolContext();
    var courses = context.Courses.Where(c => c.LecturerId == loginUser && c.IsCanceled != true).ToList();
    foreach (Course i in courses)
    {
        i.Name = user.Name; //lấy thông tin Name từAspNetUsers
    }
    return View(courses);
}

0 references
public ActionResult Delete(int id)
{
    var userID = User.Identity.GetUserId(); //get login user
    BigSchoolContext context = new BigSchoolContext();
    var findCourse = context.Courses.FirstOrDefault(p => p.Id == id);
    findCourse.IsCanceled = true; // mang ý nghĩa là xóa khóa học
    context.SaveChanges();
    return RedirectToAction("Mine");
}
```

Khi hiện thị sẽ không hiện thị những khóa học đã bị Deleted.

**Yêu cầu nâng cao:** Ở sách giáo trình chức năng xóa từ trang 74, và không gọi trực tiếp ( Refresh lại trang, gọi Mine), mà sử dụng Web API để cập nhật dữ liệu và dùng script css để xóa <li> ra khỏi view. Các bạn có thể thực hiện được không ? ☺

4.9 Chưa biết cách hướng dẫn thế nào nữa ☺

----- Hết Lab05 -----