

Bài tập 3

Lab 07 – Truyền tham số trong MVC

- Tạo một project đặt tên là BaiTap7, chọn mẫu **Empty**, kích chọn **MVC** phần Add folders & core references.
- Tạo một Controller chọn mẫu Empty Controller, đặt tên controller là Home.
- Tạo view không sử dụng Layout tương ứng với action Index trong controller. Trong view viết code hiển thị một form đăng ký như sau:

Register là action method trong controller Home

```
<body>
<div>
  <h1>Đăng ký</h1>
  <form action="/Home/Register" method="post">
    <div>Mã</div><input name="Id" />
    <div>Họ tên</div><input name="Name" />
    <div>Điểm</div><input name="Mark" />
    <hr/>
    <input type="submit" value="Đăng ký" />
  </form>
  <br/>
  <a href="/Home/Register?Id=15&Name=Hoàng&Mark=8.5">Đùng Query string</a>
</div>
</body>
```

- Trong HomeController tạo một action method có tên là **Register** và một view tương ứng.

Quy trình hoạt động người dùng sẽ nhập dữ liệu cho các tham số Id, Name và Mark từ view Index.cshtml. Controller sẽ lấy dữ liệu từ view Index và chuyển qua view Register.

1. Nhận tham số bằng Request

- Trong HomeController viết code cho action Register như sau:

```
public ActionResult Register()
{
    //Lấy dữ liệu từ view bằng Request ["<tham số>"]
    int Ma = Convert.ToInt32(Request["Id"]);
    string Ten = Request["Name"];
    decimal Diem = Convert.ToDecimal(Request["Mark"]);

    ViewBag.ThongTin = "Mã sinh viên: " + Ma + " - Tên sinh viên: " + Ten + " - Điểm: " + Diem;
    return View();
}
```

- Trong Register.cshtml viết code để hiển thị giá trị được truyền qua ViewBag

```
<body>
  <div>
    <p>@ViewBag.ThongTin</p>
  </div>
</body>
```

- Mở HomeController, ấn phím F5 (hoặc Ctrl+F5) để chạy thử. Kích chuột vào “Đùng Query string”

Đăng ký

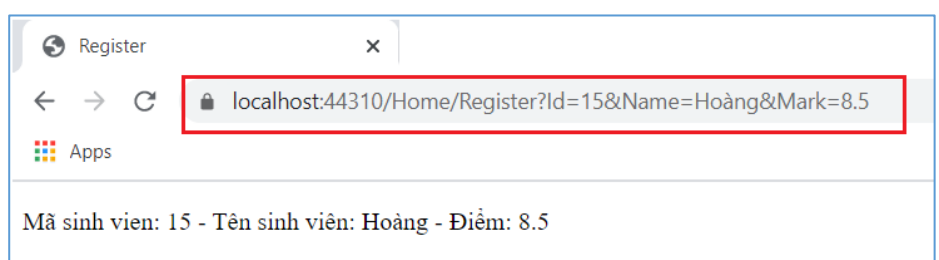
Mã

Họ tên

Điểm

Đăng ký

[Đùng Query string](#)



2. Nhận tham số bằng **FormCollection**

- Trong HomeController sửa code của action Register như sau:

```
public ActionResult Register(FormCollection data)
{
    int Ma = Convert.ToInt32(data["Id"]);
    string Ten = data["Name"];
    decimal Diem = Convert.ToDecimal(data["Mark"]);

    ViewBag.ThongTin = "Mã sinh viên: " + Ma + " - Tên sinh viên: " + Ten + " - Điểm: " + Diem;
    return View();
}
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử. Nhập Mã, Họ tên và Điểm vào các trường tương ứng sau đó kích vào nút “Đăng ký”

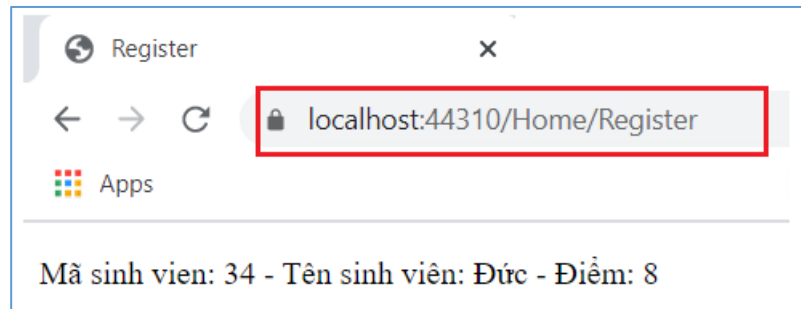
Đăng ký

Mã

Họ tên

Điểm

[Dùng_Query_string](#)



3. Nhận tham số bằng **Action Arguments**

- Trong HomeController sửa code của action Register như sau:

```
public ActionResult Register(int Id=0, string Name="", decimal Mark=0)
{
    ViewBag.ThongTin = "Mã sinh viên: " + Id + " - Tên sinh viên: " + Name + " - Điểm: " + Mark;
    return View();
}
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử. Nhập Mã, Họ tên và Điểm vào các trường tương ứng sau đó kích vào nút “Đăng ký”

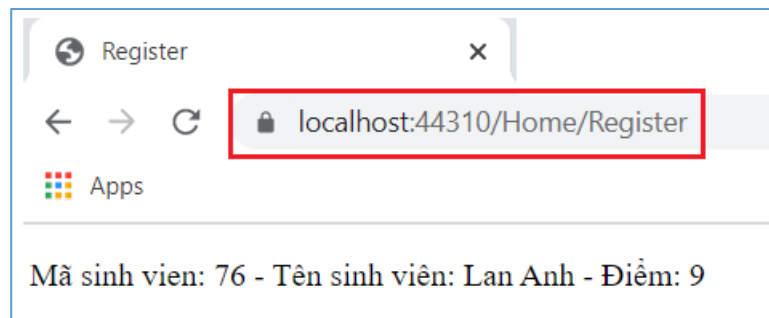
Đăng ký

Mã

Họ tên

Điểm

[Dùng_Query_string](#)



4. Nhận tham số sử dụng Model

- Tạo một class trong folder Model và đặt tên là Student. Mở file Student.cs khai báo các thuộc tính **Id**, **Name**, **Mark** cho class này.

```
public class Student
{
    0 references
    public int Id { get; set; }
    0 references
    public string Name { get; set; }
    0 references
    public decimal Mark { get; set; }
}
```

- Trong HomeController sửa code của action Register như sau:

```
public ActionResult Register(Student sv)
{
    ViewBag.ThongTin = "Mã sinh viên: " + sv.Id + " - Tên sinh viên: " + sv.Name + " - Điểm: " + sv.Mark;
    return View();
}
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử. Nhập Mã, Họ tên và Điểm vào các trường tương ứng sau đó kích vào nút “Đăng ký”

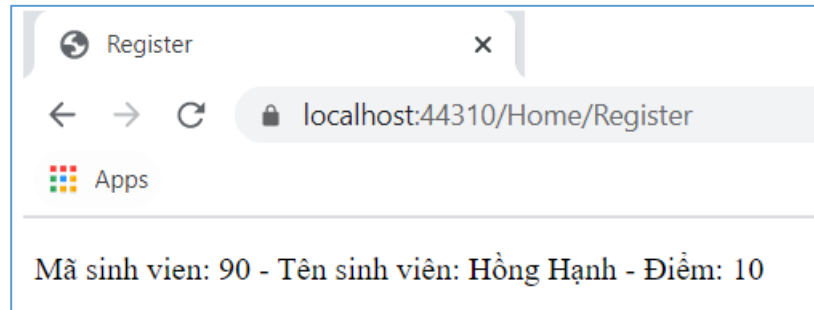
Đăng ký

Mã

Họ tên

Điểm

[Dùng Query string](#)



5. Đọc ghi file văn bản

- Trong HomeController thêm một action **Save** để ghi vào file và một action **Open** để đọc từ file như sau:

```
public ActionResult Save(Student sv)
{
    //Đường dẫn đến file lưu dữ liệu
    string path = Server.MapPath("~/StudentInfo.txt");
    //Khái báo một mảng kiểu chuỗi ký tự
    string[] lines = { sv.Id.ToString(), sv.Name, sv.Mark.ToString() };
    //Ghi thông tin vào file
    System.IO.File.WriteAllLines(path, lines);

    ViewBag.HanhDong = "Đã ghi vào file!";
    //Trả về view Index
    return View("Index");
}

public ActionResult Open(Student sv)
{
    //Đường dẫn đến file lưu dữ liệu
    string path = Server.MapPath("~/StudentInfo.txt");
    //Đọc file vào một mảng kiểu chuỗi ký tự
    string[] lines = System.IO.File.ReadAllLines(path);
    //Gán giá trị vào Model
    sv.Id = Convert.ToInt32(lines[0]);
    sv.Name = lines[1];
    sv.Mark = Convert.ToDecimal(lines[2]);

    //Thông tin đọc được
    ViewBag.ThongTin = "Mã sinh viên: " + sv.Id + " - Tên sinh viên: " + sv.Name +
    " - Điểm: " + sv.Mark;
    ViewBag.HanhDong = "Đã đọc từ file!";
    //Trả về view Index
    return View("Index", sv);
}
```

**** Chú ý:** Hai action method trên không tạo view riêng mà cùng sử dụng view Index vì vậy khi return View("Tên view") cần chỉ rõ tên view là Index.

- File Index.cshtml sửa lại code như sau:

```
<body>
    <div>
        <h2> Đọc/Ghi file</h2>
        <form action="" method="post">
            <div>Mã</div><input name="Id" />
            <div>Họ tên</div><input name="Name" />
```

```

        <div>Điểm</div><input name="Mark" />
        <hr />
        <input type="submit" value="Lưu" formaction="/Home/Save" />
        <input type="submit" value="Mở" formaction="/Home/Open" />
    </form>
    <br />
    <p>@ViewBag.HanhDong</p>
    <p>@ViewBag.ThongTin</p>
</div>
</body>

```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.
 - Nhập thông tin vào các trường rồi kích nút “Lưu”.
 - Tìm file có tên `StudentInfo.txt` bên trong project và mở ra xem.
 - Kích vào nút “Mở”.

6. Upload file

- Tạo một Controller chọn mẫu Empty Controller, đặt tên controller là UploadFile.
- Tạo view **sử dụng Layout** tương ứng với action Index trong controller. Trong view viết code như sau:

```

<h2>Upload file</h2>
<form action="" method="post" enctype="multipart/form-data">
    <p><label for="ufile" style="cursor: pointer;">Chọn file</label></p>
    <p><input name="document" type="file" id="ufile"/></p>
    <hr />
    <input type="submit" value="Upload" formaction="/UploadFile/Upload" />
</form>
<br />
<p>@ViewBag.FileName</p>
<p>@ViewBag.FileType</p>
<p>@ViewBag.FileSize</p>

```

- Trong controller UploadFile tạo một action method đặt tên là **Upload** như sau:

```

public ActionResult Upload()
{
    //Lấy file được upload bởi client có name được chỉ rõ
    var f = Request.Files["document"];
    if (f != null && f.ContentLength > 0)
    {
        //Gắn tên file với đường dẫn của nơi lưu file
        var path = Server.MapPath("~/FileUpload/" + f.FileName);
        f.SaveAs(path);

        ViewBag.FileName = f.FileName;
        ViewBag.FileType = f.ContentType;
        ViewBag.FileSize = f.ContentLength;
    }
    return View("Index");
}

```

- Kích chuột phải vào BaiTap7 chọn Add sau đó chọn New Folder tạo folder có tên là `FileUpload`.
- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử. Kiểm tra xem file đã được lưu vào đó chưa.
- Sửa view Index.cshtml của controller UploadFile để có thể xem ảnh trước khi upload

```

<h2>Upload file</h2>
<form action="" method="post" enctype="multipart/form-data">
    <p><img id="output" width="200" /></p>
    <p><label for="ufile" style="cursor: pointer;">Chọn file</label></p>
    <p><input name="document" type="file" id="ufile" style="display: none;"
    onchange="loadFile(event)" /></p>
    <hr />
    <input type="submit" value="Upload" formaction="/UploadFile/Upload" />
</form>
<br />
<p>@ViewBag.FileName</p>
<p>@ViewBag.FileType</p>

```

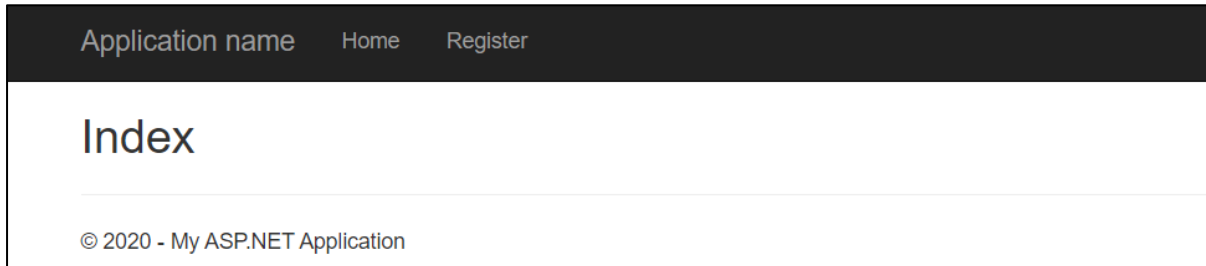
```
<p>@ViewBag.FileSize</p>
```

```
<script>
    var loadFile = function (event) {
        var image = document.getElementById('output');
        image.src = URL.createObjectURL(event.target.files[0]);
    };
</script>
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử. Chọn 1 file ảnh để load.

Lab 08 – Helper

- Tạo một project đặt tên là BaiTap8, chọn mẫu **MVC**.
7. Nhập xuất sử dụng Helper
- Sửa Layout để có menu như sau



- Tạo view RegisterForm.cshtml để nhập thông tin đăng ký như sau:

```
@{
    ViewBag.Title = "RegisterForm";
}

<h2>Đăng ký thành viên</h2>
@{ Html.BeginForm("Register", "Home", FormMethod.Post, new { enctype = "multipart/form-data" }); }
<p>
    <div>Họ và tên</div>
    @Html.TextBox("FullName")
</p>
<p>
    <div>Mật khẩu</div>
    @Html.Password("Password")
</p>
<p>
    <div>Ảnh</div>
    <input name="ImageFile" type="file" />
</p>
<p>
    <div>Tình trạng hôn nhân</div>
    <label>@Html.CheckBox("Single", true) Độc thân</label>
</p>
<p>
    <div>Giới tính</div>
    <label>@Html.RadioButton("Gender", "Nam", true) Nam</label>
    <label>@Html.RadioButton("Gender", "Nữ", false) Nữ</label>
</p>
<p>
    <div>Tỉnh/Thành phố</div>
    @Html.DropDownList("City", new SelectList(ViewBag.Cities), "- Chọn tỉnh -")
</p>
<p>
    <div>Mô tả</div>
    @Html.TextArea("Description")
</p>
<p>
    @Html.Hidden("Active", "Hoạt động")
</p>
```

```

<hr />
<input type="submit" value="Gửi đăng ký" />
@{ Html.EndForm();}

```

****Chú ý:** `Html.BeginForm(Tên_Action, Tên_Controller, FormMethod.Post, new { enctype="multipart/form-data" })` được dùng khi trong form có upload ảnh.

- Tạo action method RegisterForm trong HomeController sử dụng [HttpGet] như sau:

```

[HttpGet]
public ActionResult RegisterForm()
{
    //Tạo danh sách mục chọn cho DropDownList "City" trong view RegisterForm
    List<string> Cities = new List<string>
    {
        "Hà Nội", "Hải Phòng", "Nam Định", "Thái Bình"
    };
    ViewBag.Cities = Cities;
    return View();
}

```

- Tạo một model có tên là Member như sau:

```

public class Member
{
    public string FullName { get; set; }
    public string Password { get; set; }
    public string ImagePath { get; set; }
    public HttpPostedFileBase ImageFile { get; set; }
    public bool Single { get; set; }
    public string Gender { get; set; }
    public string City { get; set; }
    public string Active { get; set; }
    public string Description { get; set; }
}

```

- Tạo một view có tên là Register sử dụng model Member: kích chuột phải vào Folder View và chọn Add sau đó chọn View... chọn như hình sau:

The screenshot shows the 'Add View' dialog box. The 'View name' field contains 'Register'. The 'Template' dropdown is set to 'Details'. The 'Model class' dropdown is set to 'Member (BaiTap8.Models)'. In the 'Options' section, the 'Use a layout page' checkbox is checked. At the bottom right, the 'Add' button is highlighted with a red rectangular box.

- Tạo một action method có tên là **Register** trong HomeController sử dụng [HttpPost] để nhận dữ liệu được nhập từ RegisterForm như sau:

```

[HttpPost]
public ActionResult Register(Member mem)
{
    var f = mem.ImageFile;
    if (f != null && f.ContentLength > 0)
    {

```

```
//Use Namespace called : System.IO
string FileName = System.IO.Path.GetFileName(f.FileName);
//Lấy tên file upload
string UploadPath = Server.MapPath("~/UserImage/" + FileName);
//Copy Và lưu file vào server.
f.SaveAs(UploadPath);
//Lấy đường dẫn của file upload
mem.ImagePath = "~/UserImage/" + FileName;
}
return View(mem);
}
```

- Tạo một folder có tên là UserImage để lưu các file ảnh upload.
- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử. Kích vào menu Register, nhập dữ liệu vào form Đăng ký thành viên và xem kết quả

- Sửa view Register.cshtml để hiển thị ảnh upload: Thêm đoạn mã
``

Vào trước dòng mã hiển thị ImagePath như sau:

```
<dt>
    @Html.DisplayNameFor(model => model.ImagePath)
</dt>

<dd>
    
    <br/>
    @Html.DisplayFor(model => model.ImagePath)
</dd>
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử. Kích vào menu Register, nhập dữ liệu vào form Đăng ký thành viên và xem kết quả.

8. Sinh giao diện theo model

- Tạo một model có tên là Student như sau:

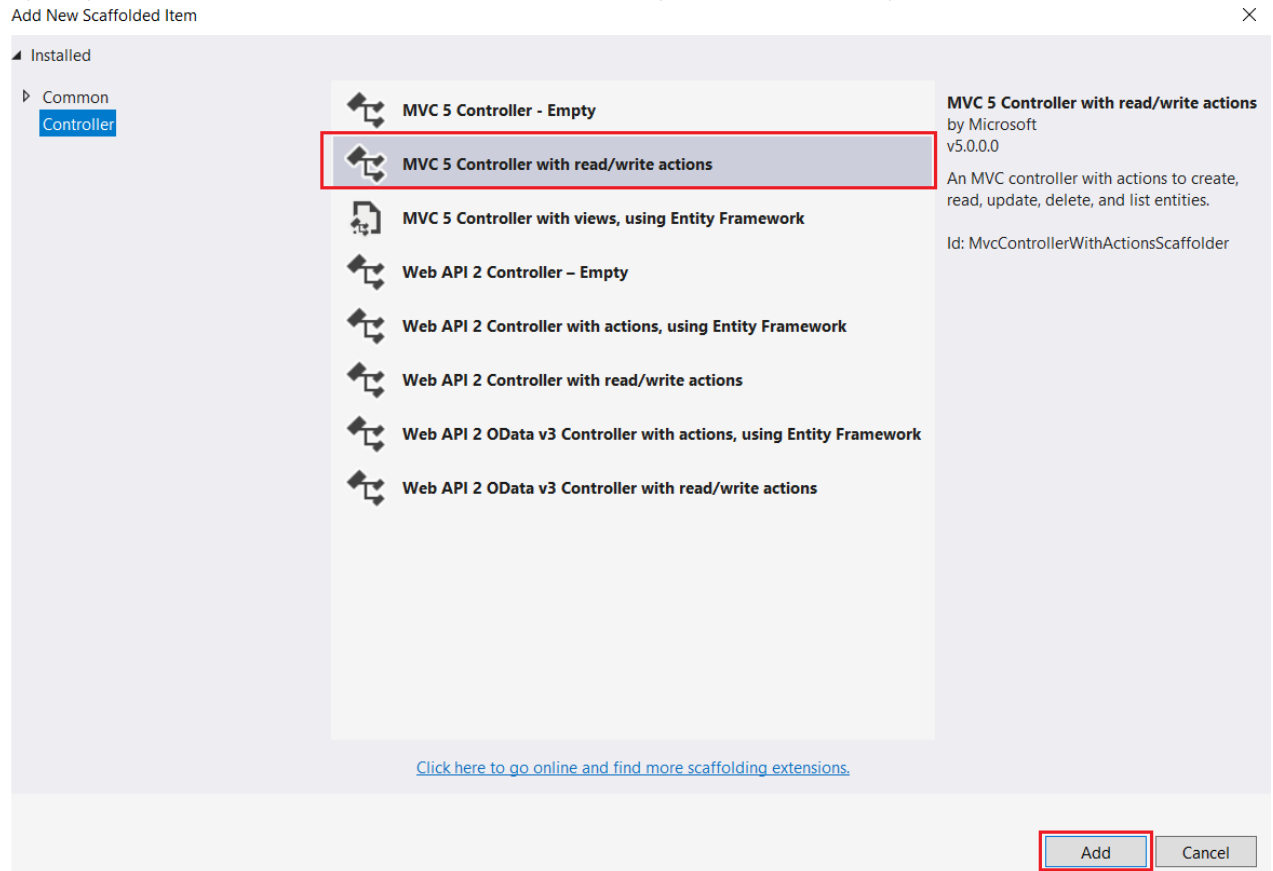
```
public class Student
{
    [DisplayName("Mã sinh viên")]
    public string Id { get; set; }
    [DisplayName("Mật khẩu")]
    public string Password { get; set; }
    [DisplayName("Họ và tên")]
}
```

```

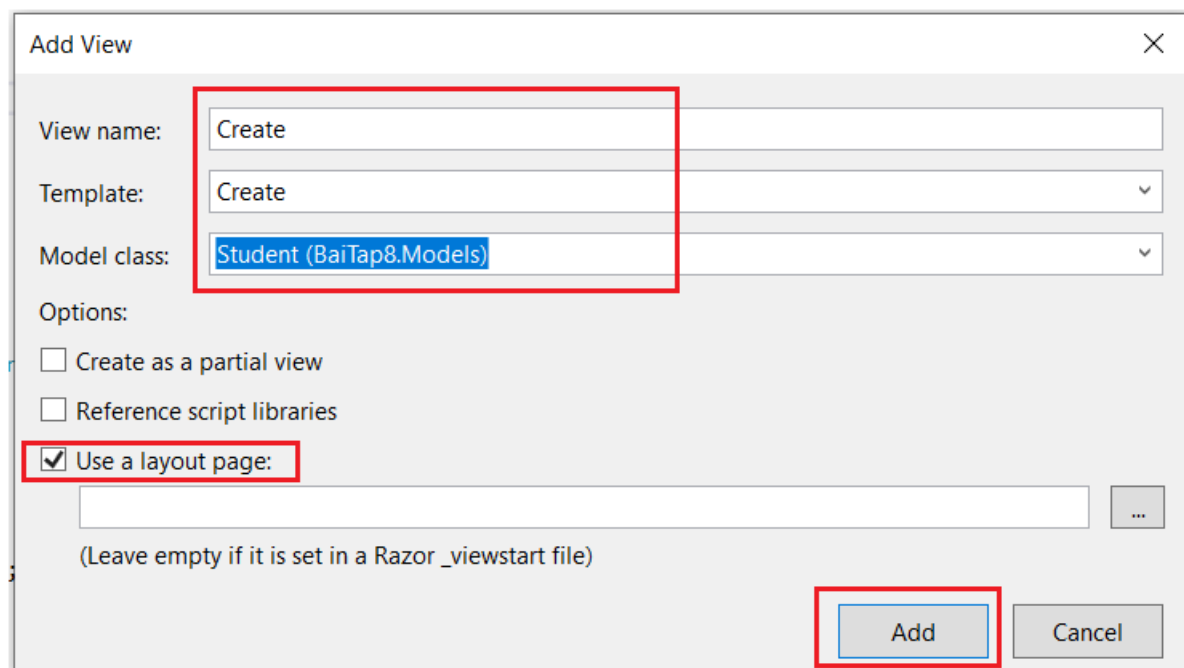
public string FullName { get; set; }
[DisplayName("Giới tính")]
public bool Gender { get; set; }
[DisplayName("Ngày sinh")]
public DateTime Birthday { get; set; }
[DisplayName("Ghi chú")]
public string Notes { get; set; }
}

```

- Tạo một controller với mẫu MVC5 Controller with read/write actions và đặt tên là Student



- Kích chuột phải vào action Create trong StudentController và chọn Add View... Nhập như trong hình và kích vào nút Add.



- Thêm liên kết `@Html.ActionLink("Create student", "Create", "Student")` vào `_Layout.cshtml`
- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử. Kích vào "Create student"

Create

Student

Mật khẩu

Họ và tên

Giới tính

☐

Ngày sinh

Ghi chú

Create

Back to List

- Sửa model Student.cs như sau:

```
public class Student
{
    [DisplayName("Mã sinh viên")]
    public string Id { get; set; }
    [DisplayName("Mật khẩu"),DataType(DataType.Password)]
    public string Password { get; set; }
    [DisplayName("Họ và tên")]
    public string FullName { get; set; }
    [DisplayName("Giới tính")]
    public bool Gender { get; set; }
    [DisplayName("Ngày sinh"),DataType(DataType.Date)]
    public DateTime Birthday { get; set; }
    [DisplayName("Ghi chú"),DataType(DataType.MultilineText)]
    public string Notes { get; set; }
}
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử. Kích vào "Create student"

Create

Student

Mật khẩu

Họ và tên

Giới tính

☐

Ngày sinh

Ghi chú

Create

Back to List

- Tạo một folder có tên là **EditorTemplates** trong folder View/Shared.
- Tạo một Partial View trong folder **EditorTemplates** và đặt tên là Boolean. Nhập nội dung sau vào **Boolean.cshtml**:

```
@model Nullable<bool>
<label>@Html.RadioButtonFor(model => model, true, @Model == true) Nam</label>
<label>@Html.RadioButtonFor(model => model, false, @Model == false) Nữ</label>
```

Đây là tạo một Template cho các Property có kiểu bool hiển thị giới tính.

- Sửa lại model Student.cs để sử dụng Template giới tính trên:

```
public class Student
{
    [DisplayName("Mã sinh viên")]
    public string Id { get; set; }
    [DisplayName("Mật khẩu"),DataType(DataType.Password)]
    public string Password { get; set; }
    [DisplayName("Họ và tên")]
    public string FullName { get; set; }
    [DisplayName("Giới tính")]
    public bool Gender { get; set; }
    [DisplayName("Ngày sinh"),DataType(DataType.Date)]
    public DateTime Birthday { get; set; }
    [DisplayName("Ghi chú"),DataType(DataType.MultilineText)]
    public string Notes { get; set; }
}
```

```

public string FullName { get; set; }
[DisplayName("Giới tính")]
[UIHint("Boolean")]
public bool Gender { get; set; }
[DisplayName("Ngày sinh"),DataType(DataType.Date)]
public DateTime Birthday { get; set; }
[DisplayName("Ghi chú"),DataType(DataType.MultilineText)]
public string Notes { get; set; }
}

```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử. Kích vào “Create student”

- Sửa action method [HttpPost] Create() trong StudentController như sau:

```

// POST: Student/Create
[HttpPost]
public ActionResult Create(Student sv)
{
    try
    {
        return RedirectToAction("Details",sv);
    }
    catch
    {
        return View();
    }
}

```

- Sửa action method Details() trong StudentController như sau:

```

public ActionResult Details(Student sv)
{
    return View(sv);
}

```

- Kích chuột phải vào action Details trong StudentController và chọn Add View... Nhập như trong hình và kích vào nút Add.

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.

- Kích vào "Create student"
- Nhập thông tin vào các trường đăng ký sau đó kích vào nút Create

Create

Student

Mật khẩu

...

Họ và tên

Hoàng Văn Việt

Giới tính

☒ Nam
 ☐ Nữ

Ngày sinh

16/02/2005

Ghi chú

Test thử chương trình

Create

Details

Student

Mật khẩu

abc

Họ và tên

Hoàng Văn Việt

Giới tính

☒

Ngày sinh

16/02/2005

Ghi chú

Test thử chương trình

Edit | Back to List

- Tạo một folder có tên là **DisplayTemplates** trong folder View/Shared.
- Tạo một Partial View trong folder **DisplayTemplates** và đặt tên là Boolean. Nhập nội dung sau vào **Boolean.cshtml** của folder **DisplayTemplates**:

```
@model bool
<label>@(Model ? "Nam" : "Nữ")</label>
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.
 - Kích vào "Create student"
 - Nhập thông tin vào các trường đăng ký sau đó kích vào nút Create.
 - Xem hiển thị của trường **Giới tính**

9. Kiểm tra dữ liệu nhập (Validation)

- Tạo một model có tên là NhanVien với nội dung như sau:

```
public class NhanVien
{
    [DisplayName("Họ và tên")]
    [MinLength(5, ErrorMessage = "Tên phải có ít nhất 5 ký tự")]
    public string HoTen { get; set; }
    [DisplayName("Tuổi")]
    [Required(ErrorMessage = "Không được để trống!")]
    [Range(16, 65, ErrorMessage = "Tuổi phải từ 16 đến 65")]
    public int Tuoi { get; set; }
}
```

- Thêm liên kết `@Html.ActionLink("Validation", "Validation", "Home")` vào `_Layout.cshtml`
- Thêm một action **Validation** trong HomeController.
- Tạo view sử dụng Layout tương ứng với action **Validation** trong HomeController với Template là Create, theo Model class là NhanVien. Kích chọn Reference script libraries để tham chiếu tới script của JQuery hỗ trợ kiểm tra nhập dữ liệu phía client.

Add View

View name:

Validation

Template:

Create

Model class:

NhanVien (BaiTap8.Models)

Options:

☐ Create as a partial view

☒ Reference script libraries

☒ Use a layout page:

~/Views/Shared/_Layout.cshtml

(Leave empty if it is set in a Razor _viewstart file)

Add

Cancel

- Sửa code để khi chạy thử hiển thị như sau:

Kiểm tra lỗi

Thông tin nhân viên

Họ và tên

Tuổi

[Quay lại](#)

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử. Kích vào “**Validation**”
- Thử bộ kiểm tra lỗi phía client:
 - Nhập họ tên có 1 ký tự và không nhập tuổi
 - Nhập họ tên có ít nhất 5 ký tự và nhập tuổi là 5
 - Nhập họ tên có ít nhất 5 ký tự và nhập tuổi là 70
 - Nhập họ tên có ít nhất 5 ký tự và nhập tuổi trong khoảng từ 16 đến 65 vào nút Nhập
- Trong HomeController thêm một action method có tên là Validate như sau:

```
public ActionResult Validate(NhanVien nv)
{
    if (ModelState.IsValid)
    {
        ModelState.AddModelError("", "Chúc mừng bạn đã nhập đúng!");
    }

    return View("Validation");
}
```

- Sửa code của Validation phân khai báo Form để hướng đến action “Validate”:
@using (Html.BeginForm("Validation", "Home"))
- Chạy thử và nhập họ tên có ít nhất 5 ký tự và nhập tuổi trong khoảng từ 16 đến 65 vào nút Nhập.
- Chạy thử và nhập đoạn văn bản sau vào mục Họ và Tên <script>destroyWebsite();</script>
- Trong HomeController thêm một annotation là [ValidateInput(false)] vào trước action Validate để vô hiệu hóa kiểm tra như sau:

```
[ValidateInput(false)]
public ActionResult Validate(NhanVien nv)
{
    if (ModelState.IsValid)
    {
        ModelState.AddModelError("", "Chúc mừng bạn đã nhập đúng!");
    }

    return View("Validation");
}
```

- Chạy thử và nhập đoạn văn bản sau vào mục Họ và Tên <script>destroyWebsite();</script>