

## BÀI 9. THỰC HÀNH SỬ DỤNG ENTITY FRAMEWORK TRONG ASP.NET MVC-PHẦN 1

- Mục đích: Hướng dẫn sinh viên cài thành thạo chức năng CRUD trên một bảng.

- Yêu cầu: Sinh viên biết cách sử dụng Entity Framework để kết nối dữ liệu

- Hình thức tổ chức dạy học: Thực hành

- Thời gian: Lý thuyết( trên lớp: 3; online: 3) Tự học, tự nghiên cứu: 2

- **Nội dung chính:**

1. Các bước cài đặt chức năng CRUD trên một bảng .....	2
2. Hướng dẫn thực hành.....	2
2.1. Tạo controller thực hiện các chức năng CRUD cho một bảng .....	2
2.2. Tạo controller thực hiện các chức năng CRUD cho một bảng có cột chứa ảnh .....	3
2.2.1. Sửa hiển thị tên các property và đưa vào các thông báo lỗi tiếng Việt vào lớp product .....	3
2.2.2. Tạo controller CRUD.....	3
2.2.3. Hiển thị ảnh của sản phẩm trong danh sách sản phẩm.....	5
2.2.4. Upload ảnh trong phần thêm sản phẩm mới .....	6
2.2.5. Upload ảnh trong phần sửa sản phẩm .....	7

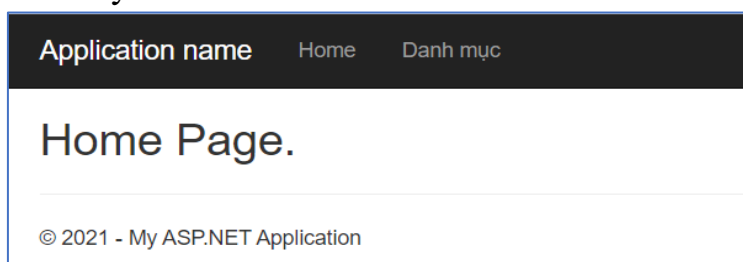
## 1. Các bước cài đặt chức năng CRUD trên một bảng

- Bước 1. Tạo một cơ sở dữ liệu (nếu chưa có)
  - Cần có file script sql, folder chứa ảnh nếu dữ liệu có cột ảnh.
- Bước 2. Tạo project MVC
- Bước 3. Tạo kết nối với Database
  - Kích chuột phải vào folder Model và chọn: Add => New item => Data => ADO.NET Entity Model => Code First from data
  - Tạo một connection tới cơ sở dữ liệu
- Bước 4. Mở file webconfig.cs để kiểm tra connection string
- Bước 5. Tạo controller
  - Sửa hiển thị tên cột trong các lớp model
  - Chọn mẫu **MVC 5 Controller with view, using Entity Framework** để tự sinh ra controller và view ứng với chức năng CRUD

## 2. Hướng dẫn thực hành

### 2.1. Tạo controller thực hiện các chức năng CRUD cho một bảng

- Chạy file script WineDB.sql trong SQLServer để tạo cơ sở dữ liệu **WineStore**
- Tạo một project MVC và tạo kết nối với Database **WineStore**. Kết nối với hai bảng Catalogy và Product.
- Sửa Layout để có màn hình với menu như sau:



- Sửa hiển thị tên các property trong các lớp Model thành tiếng Việt và đưa vào các thông báo lỗi bằng tiếng Việt.
- Tạo controller để thực hiện CRUD (thêm, xem, sửa, xóa) cho bảng Catalogy. Liên kết controller với link “Danh mục” trên menu.
- Thêm Try ... Catch để bắt lỗi
- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.

## 2.2. Tạo controller thực hiện các chức năng CRUD cho một bảng có cột chứa ảnh

### 2.2.1. Sửa hiển thị tên các property và đưa vào các thông báo lỗi tiếng Việt vào lớp product

```
public partial class Product
{
    [DatabaseGenerated(DatabaseGeneratedOption.None)]
    [Key]
    [DisplayName("Mã rượu")]
    public int ProductID { get; set; }

    [Required(ErrorMessage = "Tên rượu không được để trống!")]
    [StringLength(50)]
    [DisplayName("Tên rượu")]
    public string ProductName { get; set; }

    [Column(TypeName = "text")]
    [DisplayName("Mô tả")]
    public string Description { get; set; }

    [Column(TypeName = "numeric")]
    [DisplayName("Giá nhập")]
    public decimal PurchasePrice { get; set; }

    [Column(TypeName = "numeric")]
    [DisplayName("Giá bán")]
    public decimal Price { get; set; }
    [DisplayName("Số lượng")]
    public int Quantity { get; set; }

    [StringLength(20)]
    [DisplayName("Năm sản xuất")]
    public string Vintage { get; set; }

    [Required(ErrorMessage = "Danh mục không được để trống!")]
    [StringLength(10)]
    public string CatalogyID { get; set; }

    [Column(TypeName = "text")]
    [DisplayName("Hình ảnh")]
    public string Image { get; set; }

    [Required(ErrorMessage = "Vùng không được để trống!")]
    [StringLength(100)]
    [DisplayName("Vùng")]
    public string Region { get; set; }

    public virtual Catalogy Catalogy { get; set; }
}
```

### 2.2.2. Tạo controller CRUD

- Thêm Controllers có mẫu gắn với model class là Product
- Sửa lại phần action link của \_Layout.cshtml thêm một Actionlink:  
`<li>@Html.ActionLink("Sản phẩm", "Index", "Products")</li>`
- Sửa các nút lệnh thành tiếng Việt
- Tùy chỉnh các chức năng để xử lý lỗi:
  - [HttpPost]Create

```

public ActionResult Create([Bind(Include =
"ProductID,ProductName,Description,PurchasePrice,Price,Quantity,Vintage,CatalogyID,Image,Re
gion")] Product product)
{
    try
    {
        if (ModelState.IsValid)
        {
            db.Products.Add(product);
            db.SaveChanges();
        }
        return RedirectToAction("Index");
    }
    catch (Exception ex)
    {
        ViewBag.Error = "Lỗi nhập dữ liệu! " + ex.Message;
        ViewBag.CatalogyID = new SelectList(db.Catalogies, "CatalogyID",
"CatalogyName", product.CatalogyID);
        return View(product);
    }
}

```

và Create.cshtml ...

- [HttpPost]Edit Và Edit.cshtml ...
- DeleteConfirmed Và Delete.cshtml
- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.
- Chạy chức năng của Thêm sản phẩm.
  - Không nhập thông tin gì mà kích luôn vào nút Tạo xem hiển thị lỗi.
  - Sửa lại model Product.cs thêm các thông báo lỗi vào các trường số không được NULL

```

public partial class Product
{
    [DatabaseGenerated(DatabaseGeneratedOption.None)]
    [Key]
    [DisplayName("Mã rượu")]
    public int ProductID { get; set; }

    [Required(ErrorMessage = "Tên rượu không được để trống!")]
    [StringLength(50)]
    [DisplayName("Tên rượu")]
    public string ProductName { get; set; }

    [Column(TypeName = "text")]
    [DisplayName("Mô tả")]
    public string Description { get; set; }

    [Required(ErrorMessage = "Giá nhập không được để trống!")]
    [Column(TypeName = "numeric")]
    [DisplayName("Giá nhập")]
    public decimal PurchasePrice { get; set; }

    [Required(ErrorMessage = "Giá bán không được để trống!")]
    [Column(TypeName = "numeric")]
    [DisplayName("Giá bán")]
    public decimal Price { get; set; }

    [Required(ErrorMessage = "Số lượng không được để trống!")]
    [DisplayName("Số lượng")]
    public int? Quantity { get; set; }
}

```

```

[StringLength(20)]
[DisplayName("Năm sản xuất")]
public string Vintage { get; set; }

[Required(ErrorMessage = "Danh mục không được để trống!")]
[StringLength(10)]
public string CatalogyID { get; set; }

[Column(TypeName = "text")]
[DisplayName("Hình ảnh")]
public string Image { get; set; }

[Required(ErrorMessage = "Vùng không được để trống!")]
[StringLength(100)]
[DisplayName("Vùng")]
public string Region { get; set; }

public virtual Catalogy Catalogy { get; set; }
}

```

- Xóa bớt phần hiển thị của cột Mô tả, Giá nhập, Chất lượng, Vùng trên view Index.cshtml
- Sửa lại create.cshtml và Edit.cshtml phần nhãn hiển thị từ “CatalogyID” thành “Danh mục”

```

<div class="form-group">
    @Html.LabelFor(model => model.CatalogyID, "Danh mục", htmlAttributes: new {
        @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("CatalogyID", null, htmlAttributes: new { @class =
            "form-control" })
    </div>
</div>

```

```

        @Html.ValidationMessageFor(model => model.CatalogyID, "", new { @class =
            "text-danger" })
    </div>
</div>

```

### 2.2.3. Hiển thị ảnh của sản phẩm trong danh sách sản phẩm

- Tạo folder wwwroot và copy folder chứa ảnh sản phẩm vào folder này.
- Trong view **Index.cshtml** của folder Products sửa đoạn code hiển thị tên file ảnh

```

<td>
    @Html.DisplayFor(modelItem => item.Image)
</td>

```

Thành

```

<td>
    @{
        //Lấy đường dẫn file ảnh
        string ImagePath = "~/wwwroot/WineImages/" + item.Image;
    }
    
    <br/>
    @Html.DisplayFor(modelItem => item.Image)
</td>

```

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

- Trong view **Details.cshtml** của folder Products sửa đoạn code hiển thị tên file ảnh

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

Thành

```
<dd>
    @{
        //Lấy đường dẫn file ảnh
        string ImagePath = "~/wwwroot/WineImages/" + Model.Image;
    }
    
    <br />
    @Html.DisplayFor(model => model.Image)
</dd>
```

#### 2.2.4. Upload ảnh trong phần thêm sản phẩm mới

- Trong view Create.cshtml của folder Products
  - Sửa `@using` (`Html.BeginForm()`) thành

```
@using (Html.BeginForm("Create", "Products", FormMethod.Post, new { enctype =
    "multipart/form-data" }))
```

- sửa đoạn code lấy tên file ảnh

```
<div class="form-group">
    @Html.LabelFor(model => model.Image, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Image, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Image, "", new { @class = "text-danger" })
    </div>
</div>
```

Thành

```
<div class="form-group">
    @Html.LabelFor(model => model.Image, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        
        <p><label for="ufile" style="cursor: pointer;">Chọn file ảnh</label></p>
        <input name="ImageFile" id="ufile" type="file" style="display: none;"
        onchange="loadFile(event)" />
    </div>
</div>
```

- Thêm đoạn code script sau vào cuối file:

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

- Trong ProductsController sửa action `[HttpPost]Create()` như sau:

```

...
try
{
    if (ModelState.IsValid)
    {
        product.Image = "";
        var f = Request.Files["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("~/wwwroot/WineImages/" + FileName);
            //Copy Và lưu file vào server.
            f.SaveAs(UploadPath);
            //Lưu tên file vào trường Image
            product.Image = FileName;
        }
        db.Products.Add(product);
        db.SaveChanges();
    }
    return RedirectToAction("Index");
}
...

```

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

#### 2.2.5. Upload ảnh trong phần sửa sản phẩm

- Trong view Create.cshtml của folder Products
  - Sửa `@using` (`Html.BeginForm()`) thành

```

@using (Html.BeginForm("Edit", "Products", FormMethod.Post, new { enctype = "multipart/form-data" }))

```

- Khai báo biến lấy đường dẫn đến file ảnh

```

@{
    ViewBag.Title = "Edit";
    var ImagePath = "~/wwwroot/WineImages/" + Model.Image;
}

```

- sửa đoạn code lấy hiển thị tên file ảnh

```

<div class="form-group">
    @Html.LabelFor(model => model.Image, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Image, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Image, "", new { @class = "text-danger" })
    </div>
</div>

```

Thành

```

<div class="form-group">
    @Html.LabelFor(model => model.Image, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        
        <p><label for="ufile" style="cursor: pointer;">Chọn file ảnh</label></p>
        <input name="ImageFile" id="ufile" type="file" style="display: none;"
        onchange="loadFile(event)" />
    </div>
</div>

```

- Thêm đoạn code script sau vào cuối file:

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

- Trong ProductsController sửa action [HttpPost]Edit() như sau:

```
...
try
{
    if (ModelState.IsValid)
    {
        var f = Request.Files["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("~/wwwroot/WineImages/" + FileName);
            //Copy Và lưu file vào server.
            f.SaveAs(UploadPath);
            //Lưu tên file vào trường Image
            product.Image = FileName;
        }
        db.Entry(product).State = EntityState.Modified;
        db.SaveChanges();
    }
    return RedirectToAction("Index");
}
...

```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.
- \*\*\*Sửa code sử dụng **EditorTemplates** và **DisplayTemplates** trong việc hiển thị và upload ảnh

**Yêu cầu sinh viên chuẩn bị:**

Đọc slides xem hướng dẫn. Xem đề cương chi tiết bài giảng, xem video.