

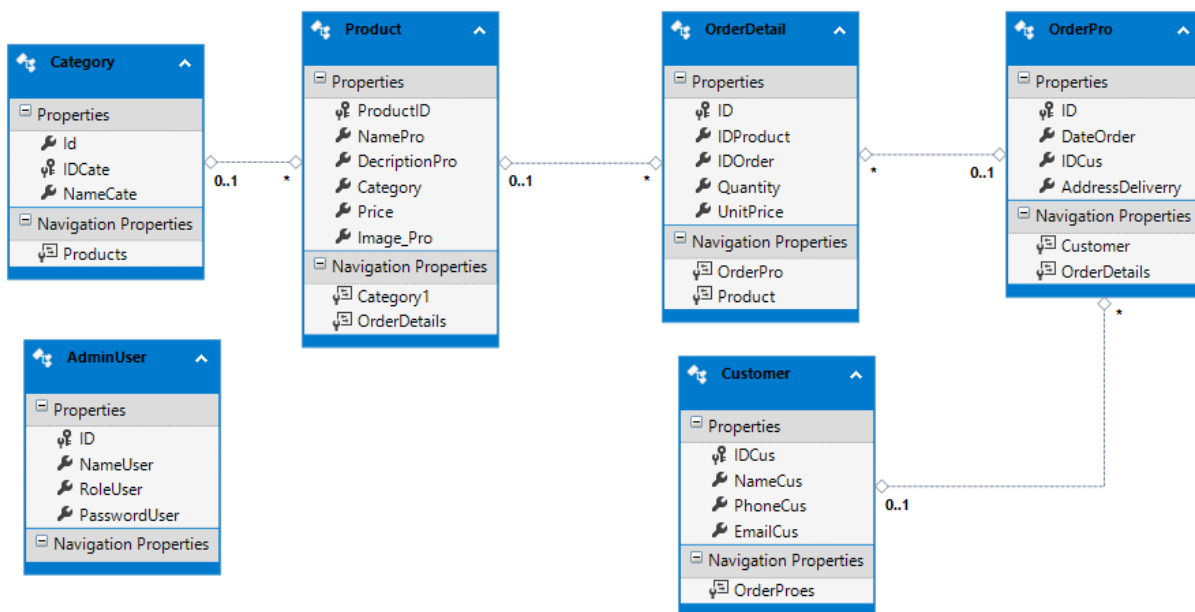
THỰC HÀNH LẬP TRÌNH WEB NÂNG CAO

BÀI THỰC HÀNH 02: Controller

Kết nối CSDL bằng Database First

- Cơ sở dữ liệu được sử dụng trong bài thực hành: **DBSportStore**

[Link tải database](#)



- Tạo mới Controller trong thư mục Controller đặt tên là: Product.cs
- Sử dụng cấu trúc Razor Html Helper để hiển thị:

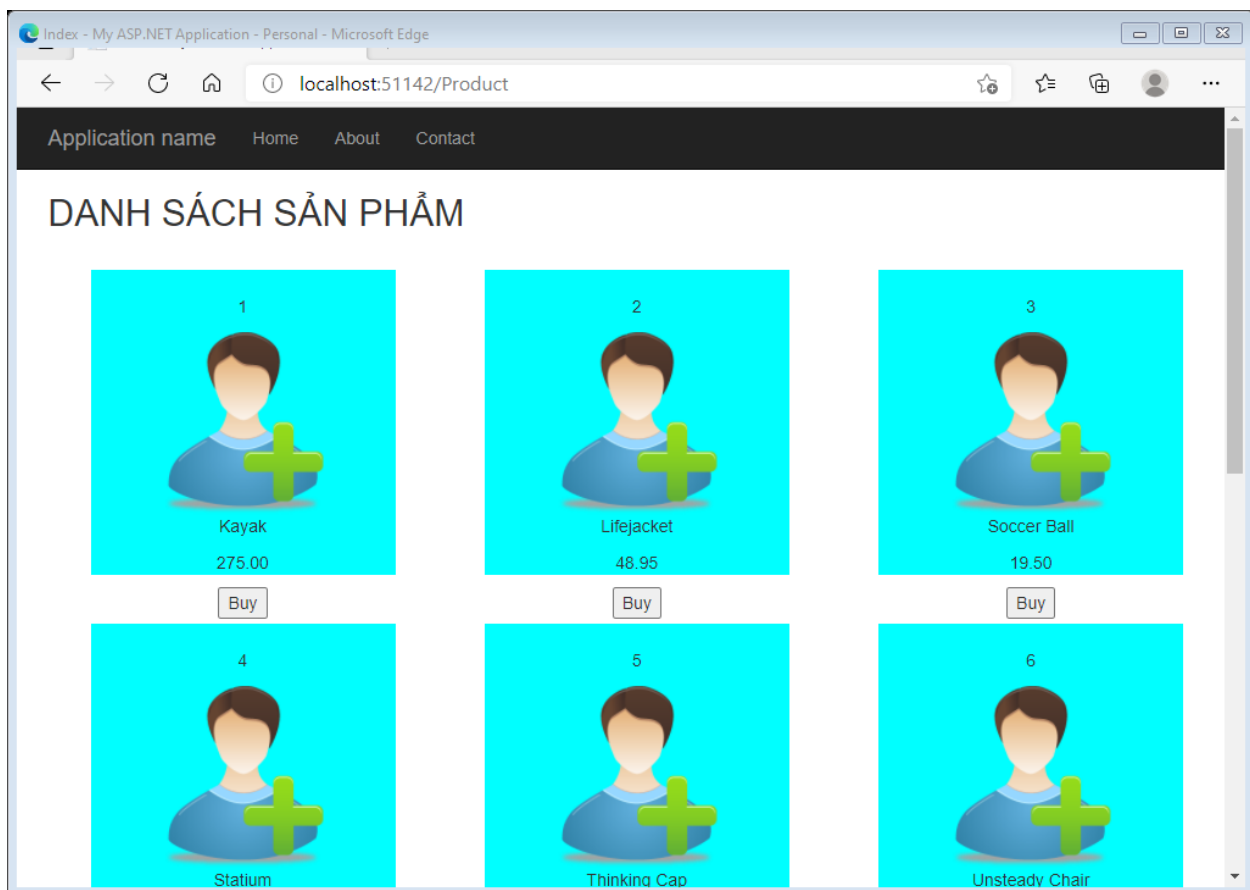
1. Danh sách Product: sử dụng `_Layout`, style định dạng tùy ý

Trong Controller Product, viết hàm Index hiển thị danh sách Product, sau đó Add đến View sử dụng Template Empty và chọn Model Product:

```

public class ProductController : Controller
{
    DBSportStoreEntities database = new DBSportStoreEntities();
    // GET: Product
    0 references
    public ActionResult Index()
    {
        return View(database.Products.ToList());
    }
}

```



```

1  @model IEnumerable<DemoConnectDB.Models.Product>
2
3  @{
4      ViewBag.Title = "Index";
5  }
6  <h2>DANH SÁCH SẢN PHẨM</h2>
7  <div class="container">
8      <div class="row">
9          @foreach (var pro in Model)
10         {
11             <div class="col-sm-12 col-sm-6 col-md-4 col-lg-3">
12                 <div class="card-pro">
13                     <p>@Html.DisplayFor(model => pro.ProductID)</p>
14                     
15                     <p>@Html.DisplayFor(model => pro.NamePro)</p>
16                     <p>@Html.DisplayFor(model => pro.Price)</p>
17                     <p><input type="submit" value="Buy" /></p>
18                 </div>
19             </div>
20         }
21     </div>
22 </div>
23
24 </div>

```

2. Form thêm mới Product

- Trong controller Product:
 - o Viết action Create() để hiển thị form


```

public ActionResult Create()
{
    Product pro = new Product();
    return View(pro);
}

```
- Render action Create đến View

Add View

View name:

Template:

Model class:

Data context class:

Options:

☐ Create as a partial view

☒ Reference script libraries

☒ Use a layout page:

...

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


Add Cancel

- Dùng Html Helper thiết kế form trong **Create.cshtml** với định dạng, sắp xếp các element tùy ý, dưới đây là Form gợi ý:

Create - Personal - Microsoft Edge

localhost:51142/Product/Create

THÊM MỚI SẢN PHẨM

NamePro	<input type="text" value="Sport abc"/>
Price	<input type="text" value="2"/>
Category	<input type="text" value="Soccer"/>
DecriptionPro	<input type="text" value=""/>
Image_Pro	
Choose File	<input type="text" value="img_fruit.png"/>
Tạo mới	<input type="button" value="Tạo mới"/>


Nội dung trong tập tin Create.cshtml:

```

1  @model DemoContextDB.Models.Product
2
3  @{
4      ViewBag.Title = "Create";
5      // Layout = "~/Views/Shared/_Layout.cshtml";
6      Layout = "~/Views/Shared/_LayoutCreate.cshtml";
7  }
8
9  <h2>THÊM MỚI SẢN PHẨM</h2>
10 @using (Html.BeginForm("Create", "Product", FormMethod.Post, new { enctype="multipart/form-data"}))
11 {
12     <table class="table-bordered">
13     <tr>
14         <td> @Html.LabelFor(model => model.NamePro)</td>
15         <td> @Html.EditorFor(model => model.NamePro)</td>
16     </tr>
17     <tr>
18         <td> @Html.LabelFor(model => model.Price)</td>
19         <td> @Html.EditorFor(model => model.Price)</td>
20     </tr>
21     <tr>
22         <td> @Html.LabelFor(model => model.Category)</td>
23         <td>
24             @Html.RenderAction("SelectCate", "Product");
25         </td>
26     </tr>
27     <tr>
28     </tr>
29     <tr>
30     </tr>
31     <td> @Html.LabelFor(model => model.DescriptionPro)</td>
32     <td> @Html.EditorFor(model => model.DescriptionPro)</td>
33     </tr>
34     <tr>
35     </tr>
36     <td> @Html.LabelFor(model => model.Image_Pro)</td>
37     <td>
38         
39     </td>
40     </tr>
41     <tr>
42     </tr>
43     <td colspan="2">
44         <input type="file" name="UploadImage" onchange="ChangeImage(this, document.getElementById('previewImg'))" />
45         <input type="submit" value="Tạo mới" />
46     </td>
47     </tr>
48 </table>
49 }

```

THÊM MỚI SẢN PHẨM

NamePro	Sport abc
Price	2
Category	Soccer ▼
DecriptionPro	--Select Cate-- Soccer Sport 10 Sport 11 Sport 11 sport Cate 10 Sport Chees Watersports Watersports
Image_Pro	
Choose File	img_fruit.png
Tạo mới	

Hướng dẫn hiển thị Dropdown List Category trong Form Create:

Bước 1:

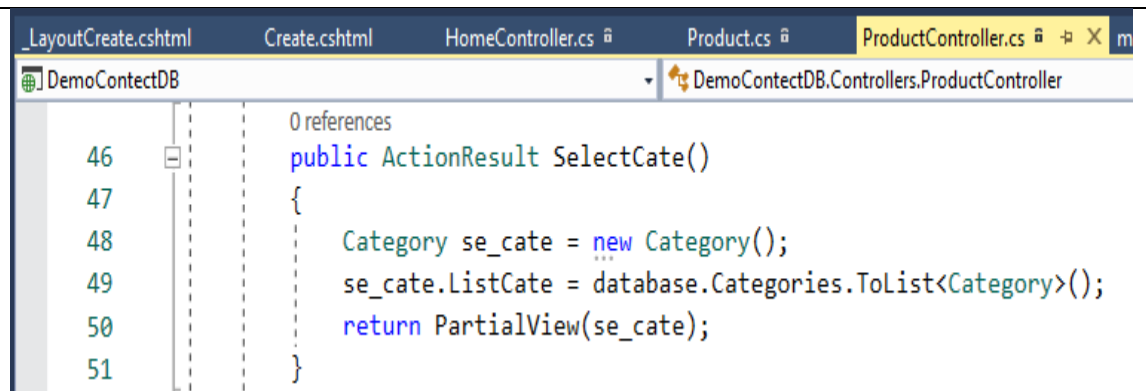
- Vào thư mục Model mở Domain class **Category.cs** thêm thuộc tính Listcate kiểu List.
- Trong đó: **[NotMapped]**: đánh dấu thuộc tính ListCate chỉ sử dụng xử lý dữ liệu không ánh xạ lưu vào trong bảng Category trong Database

[NotMapped]

0 references

```
public List<Category> ListCate { get; set; }
```

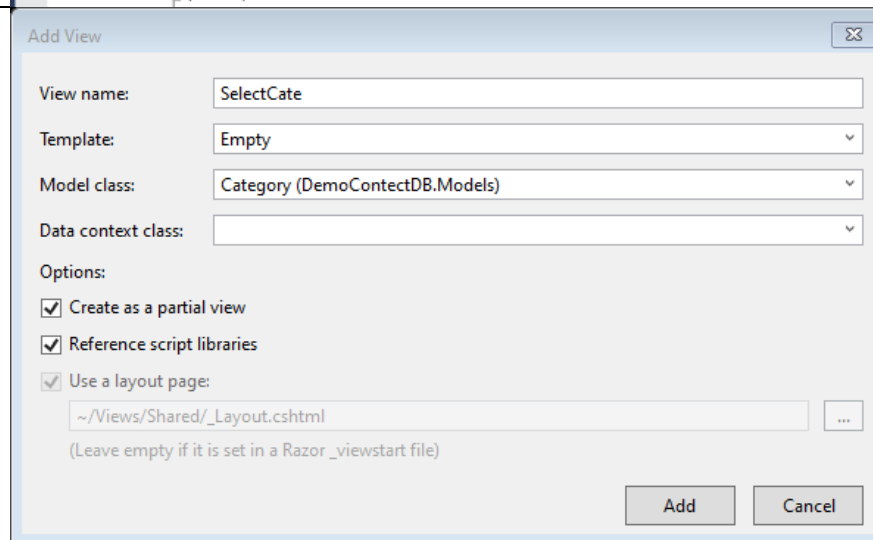
Bước 2: Vào ProductController viết action để render đến Partial View



```
46 public ActionResult SelectCate()  
47 {  
48     Category se_cate = new Category();  
49     se_cate.ListCate = database.Categories.ToList<Category>();  
50     return PartialView(se_cate);  
51 }
```

Bước 3:

- Add action SelectCate() đến PartialView
- Trong tập tin **SelectCate.cshtml**, sử dụng DropDownListFor() để lấy dữ liệu trong bảng Category đổ vào DropDownList



Add View

View name:

Template:

Model class:

Data context class:

Options:

☒ Create as a partial view

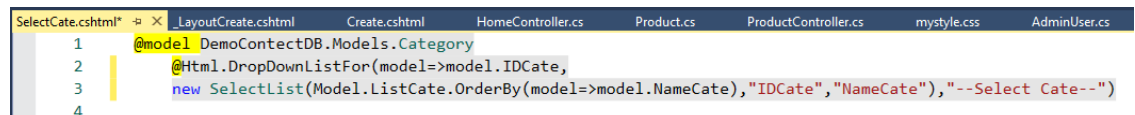
☒ Reference script libraries

☒ Use a layout page:

...

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

Add Cancel



```
1 @model DemoContextDB.Models.Category  
2 @Html.DropDownListFor(model=>model.IDCate,  
3     new SelectList(Model.ListCate.OrderBy(model=>model.NameCate), "IDCate", "NameCate"), "--Select Cate--")  
4
```

Bước 4: Sử dụng Partial View SelectCate vào trong Form Create.cshtml

Được hướng dẫn tại đánh dấu số 1

```
<tr>
  <td>@Html.LabelFor(model => model.Category)</td>
  <td>
    @Html.RenderAction("SelectCate", "Product");
  </td>
</tr>
```

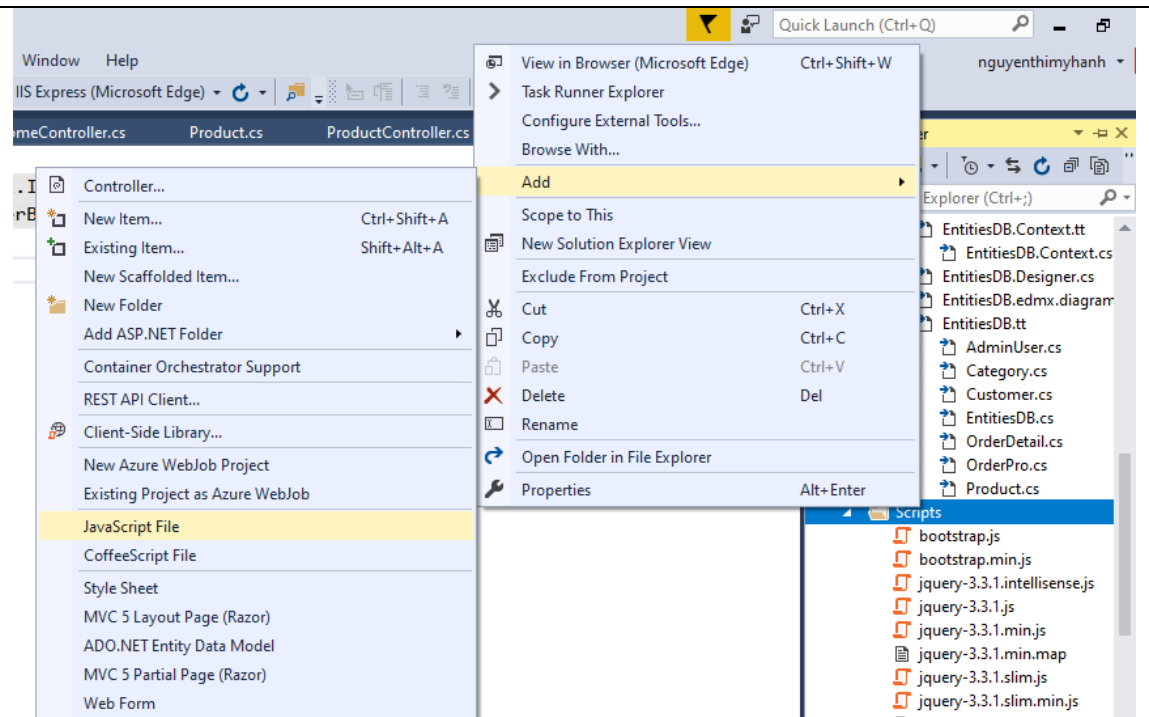
1

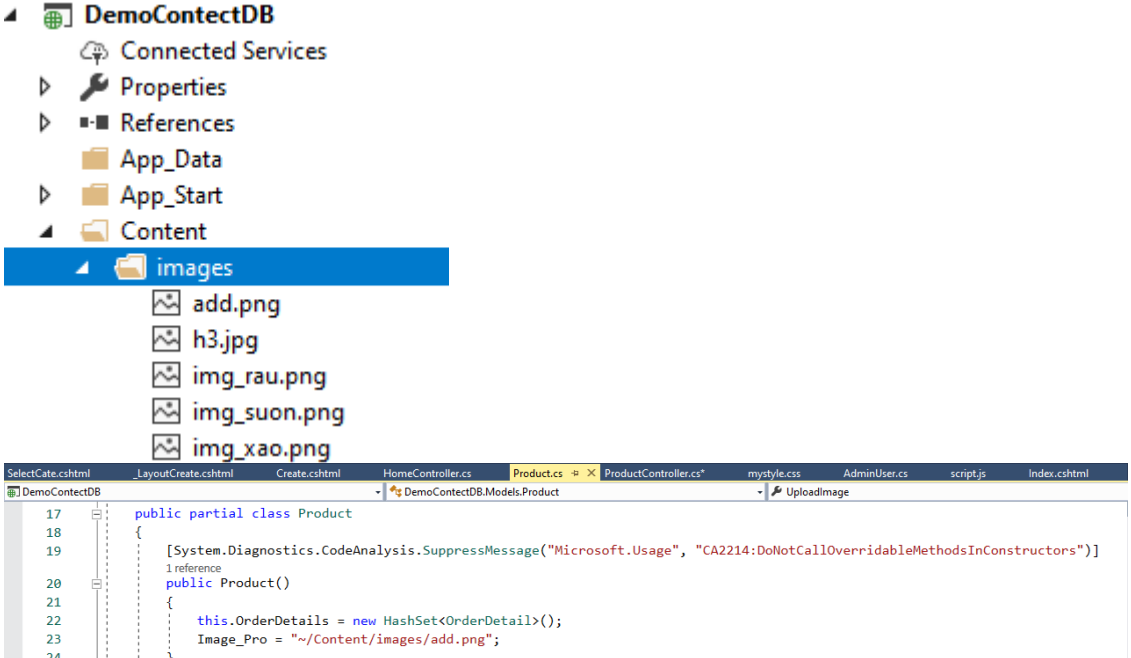
```
@{Html.RenderAction("action_name","controller_name"); }
```

Đổi hình mới khi nhấp vào nút Choose File

Bước 1: Tạo mới JavaScript

-Trong thư mục Scripts -> Nhấp phải chuột -> Add -> JavaScript File -> đặt tên là **script**



<p>Nội dung trong file script mới tạo</p>	<pre> 1 function ChangeImage(UploadImage, previewImg) { 2 if (UploadImage.files && UploadImage.files[0]) { 3 var reader = new FileReader(); 4 reader.onload = function (e) { 5 \$(previewImg).attr('src', e.target.result); 6 } 7 reader.readAsDataURL(UploadImage.files[0]); 8 } 9 }</pre>
<p>Bước 2: Add link script vào Layout</p>	<pre><script src="~/Scripts/script.js"></script></pre>
<p>Trong ProductController, viết action xử lý việc lưu dòng mới vào bảng Product, đồng thời lưu đường dẫn hình cho thuộc tính ImagePro trong bảng Product</p>	
<p>Bước 1:</p> <ul style="list-style-type: none"> -Trong thư mục Content -> Add -> New Folder: đặt tên là: images -Trong thư mục images: Add -> Existing Item -> chọn hình có trong máy tính -Mở Domain class Product.cs thêm thuộc tính UploadImage -Trong constructor Product: thêm dòng khởi tạo giá trị mặc định cho ImagePro gán link hình mặc định khi load Form Create, chú ý hình mặc định phải được add vào thư mục image trước đó 	 <p>The screenshot shows the Visual Studio interface. On the left, the 'Solution Explorer' displays the project structure for 'DemoContextDB', with the 'Content' folder expanded to show the 'images' subfolder. Inside 'images', several image files are listed: 'add.png', 'h3.jpg', 'img_rau.png', 'img_suon.png', and 'img_xao.png'. On the right, the 'Code Editor' shows the 'Product' class in 'ProductController.cs'. The constructor 'Product()' is defined, and the 'Image_Pro' property is initialized with the path '~/Content/images/add.png'.</p>

	<pre> [NotMapped] 4 references public HttpPostedFileBase UploadImage { get; set; } </pre>
<p>Bước 2:</p> <p>-Viết hàm Create(..) trong ProductController. Chú ý phía trên hàm khai báo thêm thuộc tính [HttpPost] đánh dấu action Create có sự thay đổi, tạo mới dữ liệu</p>	<pre> [HttpPost] 0 references public ActionResult Create(Product pro) { try { if(pro.UploadImage!=null) { string filename = Path.GetFileNameWithoutExtension(pro.UploadImage.FileName); string extent = Path.GetExtension(pro.UploadImage.FileName); filename = filename + extent; pro.Image_Pro = "~/Content/images/" + filename; pro.UploadImage.SaveAs(Path.Combine(Server.MapPath("~/Content/images/"), filename)); } database.Products.Add(pro); database.SaveChanges(); return RedirectToAction("Index"); } catch { return View(); } } </pre>