

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Thiết kế giao diện với razor
- ✓ Thiết kế giao diện với TagHelper
- ✓ Kết hợp razor, taghelper

PHẦN I

Bài 1 (3điểm) Viết ứng dụng Asp.net core MVC để khi chạy được giao diện hiển thị các chuyên ngành CNTT như hình bên dưới

CÔNG NGHỆ THÔNG TIN



THIẾT KẾ ĐỒ HỌA



LẬP TRÌNH MÁY TÍNH - THIẾT BỊ DI ĐỘNG



THIẾT KẾ WEBSITE



CÔNG NGHỆ THÔNG TIN - ỨNG DỤNG PHẦN MỀM

Gợi ý:

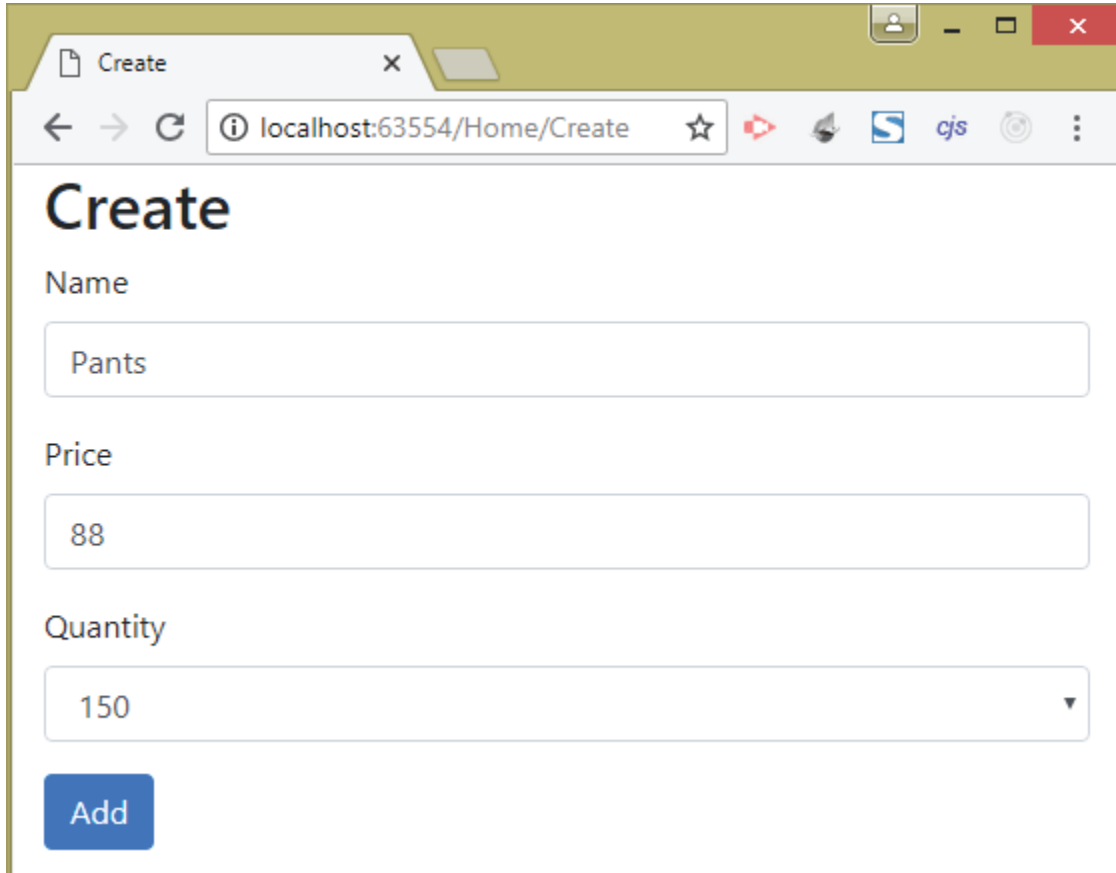
- Thiết kế lớp model với
 - ✓ các field: Image_string, Subname_string
 - ✓ method hiển thị danh sách các chuyên ngành CNTT
 - ✓ Khởi tạo dữ liệu mẫu Subname: thiết kế đồ họa, lập trình máy tính-thiết bị di động, thiết kế website, CNTT-ứng dụng phần mềm.
 - ✓ Lưu hình ảnh vào wwwroot và dùng middleware app.UseStaticFiles()
- Thiết kế giao diện kết hợp Css, html và @Model
- Sử dụng vòng lặp trong razor để hiển thị dữ liệu
- Tham khảo lab3

Bài 2 (2điểm): Làm lại bài 1, các thành phần liên quan Html tag phải dùng Tag Helper

Phần2

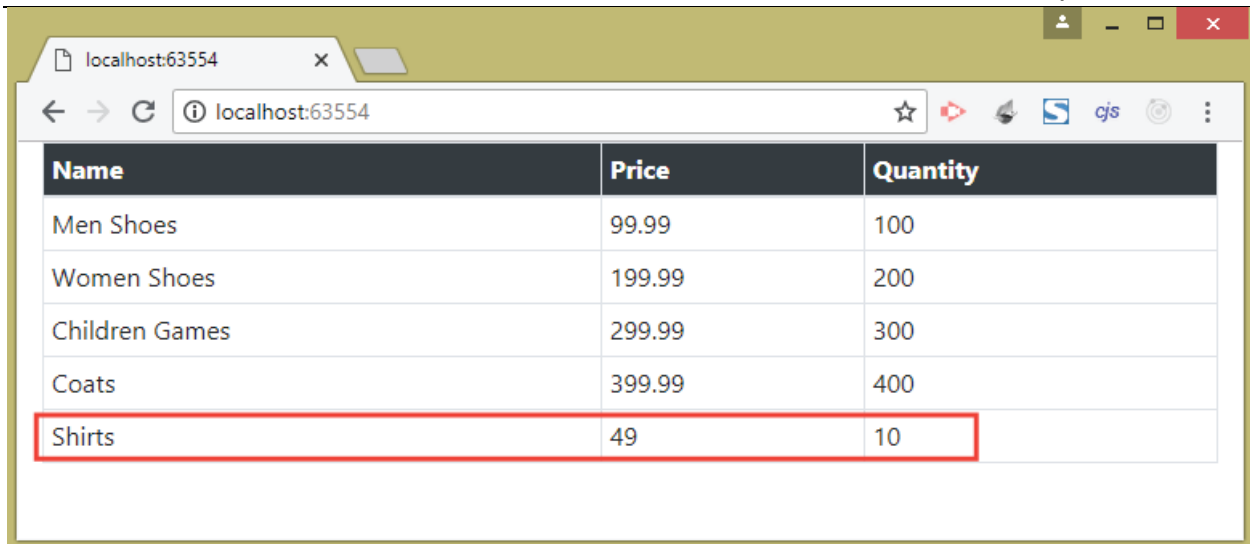
Bài 3 (5điểm): Tạo ứng dụng Asp.Net Core Mvc hiển thị:

Màn hình thêm Product:



The screenshot shows a web browser window with the address bar displaying 'localhost:63554/Home/Create'. The page title is 'Create'. The form contains three input fields: 'Name' with the text 'Pants', 'Price' with the text '88', and 'Quantity' with the text '150'. A blue button labeled 'Add' is located at the bottom left of the form.

Sử dụng Taghelper cho các thành phần Form, Html tag. Người dùng click button “Add” thì chuyển sang màn hình hiển thị danh sách Product



Name	Price	Quantity
Men Shoes	99.99	100
Women Shoes	199.99	200
Children Games	299.99	300
Coats	399.99	400
Shirts	49	10

Hướng dẫn:

- Tạo model class đáp ứng đủ các thông tin cần lưu trữ:

```
public class Product
{
    public string Name { get; set; }
    public float Price { get; set; }
    public int Quantity { get; set; }
}
```

- Tạo controller class gồm ít nhất 3 action method

- ✓ Gọi hiển thị màn hình thêm Product

```
[HttpGet]
public ActionResult Create()
{
    .....|
    return View();
}
```

- ✓ Nhận dữ liệu từ màn hình thêm product, khởi tạo đối tượng Product lưu trữ dữ liệu và truyền dữ liệu này sang action method (Index) chịu trách nhiệm gọi view hiển thị product (có thể dùng ViewBag, ViewData...tham khảo lab5)

```
[HttpPost]
public IActionResult Create(Product product)
{
    .....|
    return RedirectToAction("Index");
}
```

- ✓ Dùng action method Index gọi màn hình hiển thị product

```
public IActionResult Index()  
{  
    return View(.....);  
}
```