

Lab 04

Model in ASP.NET MVC

Mục tiêu

- Xây dựng các thành phần truy xuất dữ liệu trong model.
- Đưa dữ liệu Model từ Controller ra View theo nhiều cách và hiển thị.
- Sử dụng tính năng Scaffolding của Visual Studio để sinh View.

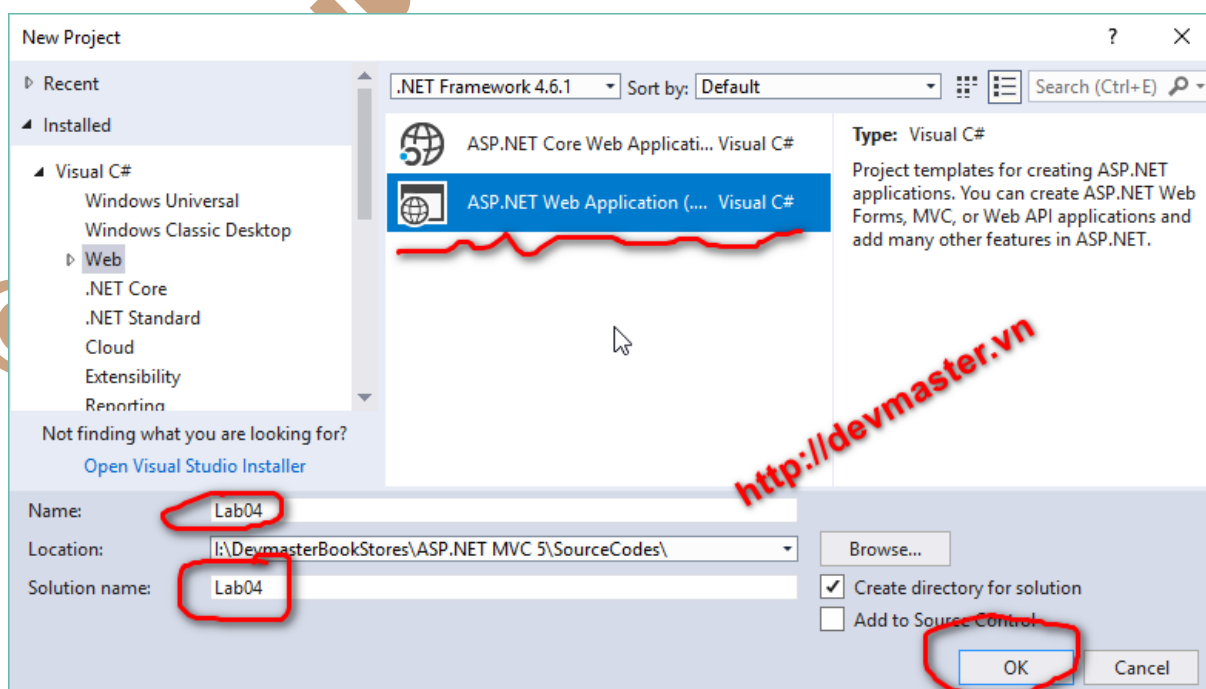
Phần I Bài tập step by step

Lab 04.1

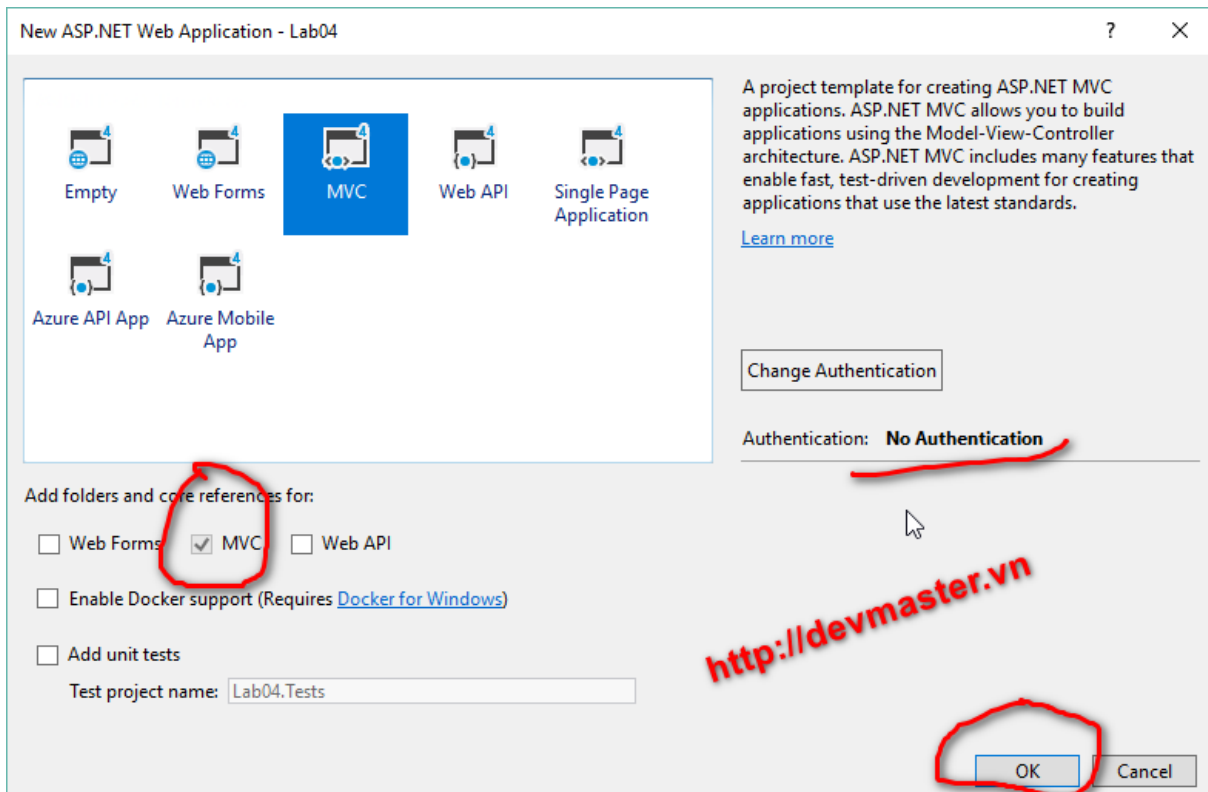
Tạo ứng dụng ASP.NET MVC5 thực hiện những công việc sau:

- Hiển thị thông tin về đối tượng Customer, thông tin bao gồm: CustomerId, FullName, Address, Phone, Email, Balance (sử dụng ViewBag, Model...)
- Hiển thị 1 danh sách các Customer.

Bước 1: Mở Visual Studio 2017 -> vào File -> New -> Projects -> Chọn Web -> ASP.NET Web Application -> Nhập tên project -> Chọn đường dẫn lưu trữ -> Nhập tên solution -> Nhấn OK (xem hình dưới).



Nhấn OK -> màn hình tiếp theo xuất hiện bạn chọn template là MVC sau đó chọn Change Authentication -> No Authentication -> OK -> OK (Xem hình bên dưới)



Bước 2: Kích chuột phải vào thư mục **Models** chọn **Add -> Class -> Nhập tên "Customer"** -> Add và code như sau:

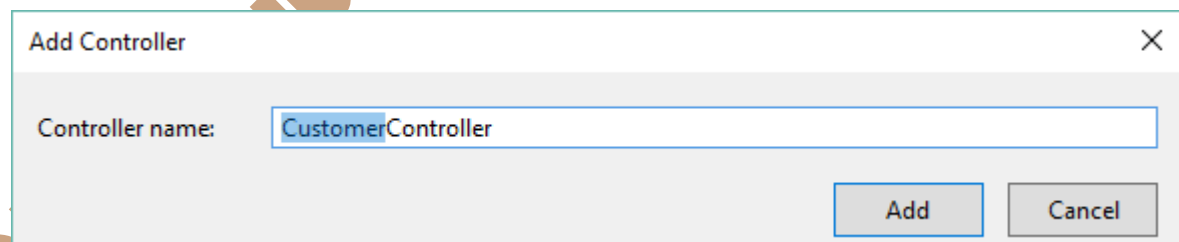
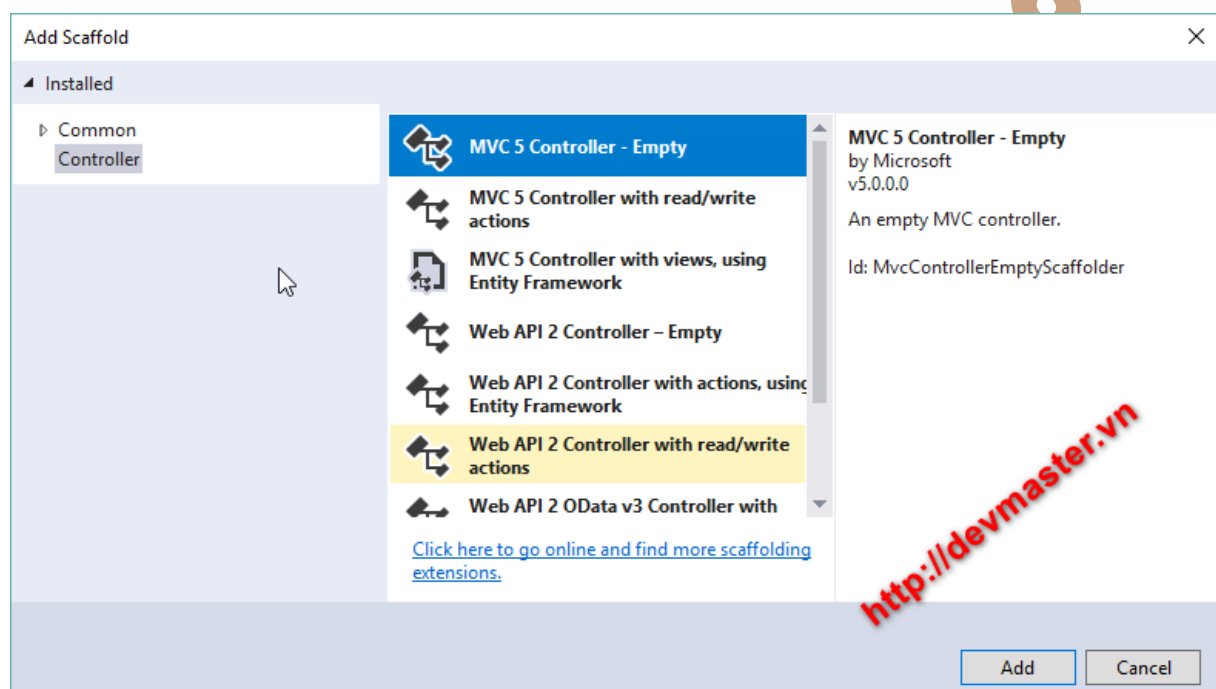
```
namespace Lab04.Models
{
    public class Customer
    {
        public string CustomerId { get; set; }
        public string FullName { get; set; }
        public string Address { get; set; }
        public string Email { get; set; }
        public string Phone { get; set; }
        public int Balance { get; set; }
    }
}
```

Bước 3: Mở trang **_Layout.cshtml** trong đường dẫn **/View/Shared/_Layout.cshtml** và tìm đoạn code sau và thêm vào đúng vị trí

```
<div class="navbar-collapse collapse">
```

```
<ul class="nav navbar-nav">
  <li>@Html.ActionLink("Khách hàng", "CustomerDetail",
    "Customer")</li>
  <li>@Html.ActionLink("Danh sách khách hàng", "CustomerList",
    "Customer")</li>
</ul>
</div>
```

Bước 4: Kích chuột phải vào thư mục Controllers chọn Add -> Controller -> Nhập tên controller -> Add

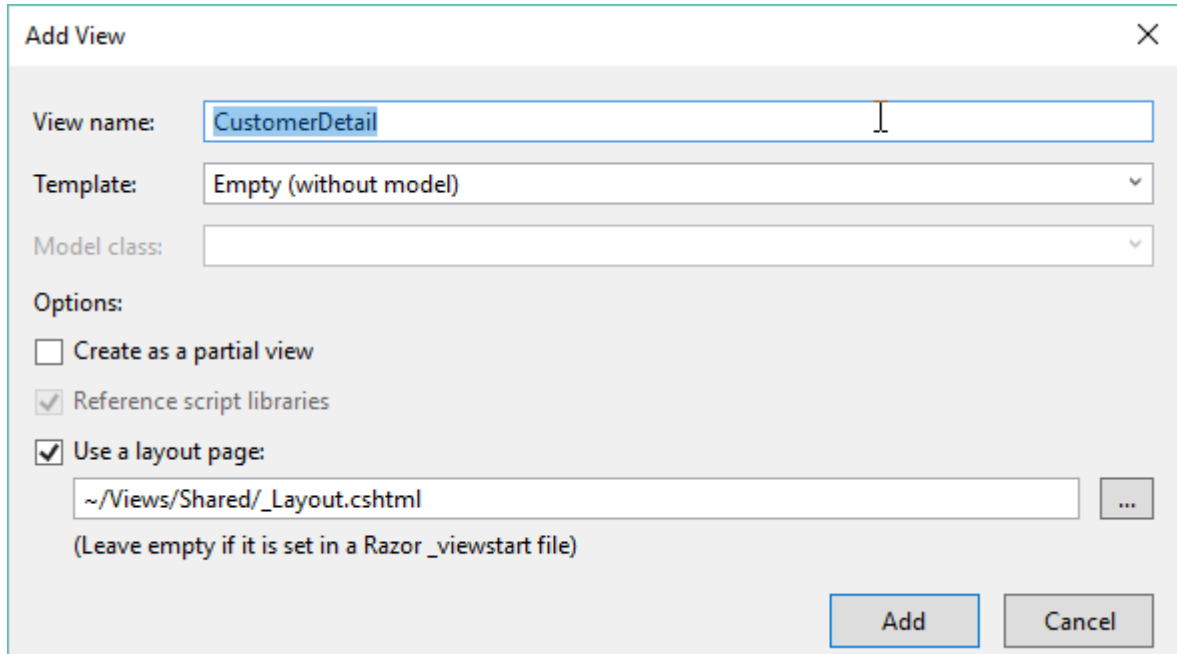


Bước 5: Code cho CustomerController như sau

```
public class CustomerController : Controller
{
    // GET: /Customer/CustomerDetail (action trả về thông tin chi tiết 1
    khách hàng cho view CustomerDetail
    public ActionResult CustomerDetail()
```

```
{
    //tạo một đối tượng Customer ( nhớ using Lab04.Models vào nhé)
    Customer cus = new Customer()
    {
        CustomerId = "KH001",
        FullName = "Trịnh Văn Chung",
        Address = "Hà Nội",
        Email = "devmaster.founder@gmail.com",
        Phone = "0978.611.889",
        Balance = 15200000
    };
    //cách 1 gán dữ liệu vào ViewBag để chuyển tới View
    //ViewBag.customer = cus;
    //return View();
    //có thể truyền dữ liệu qua đối tượng model để chuyển tới View
    return View(cus);
}
//GET /Product/CustomerList (action trả về danh sách khách hàng cho view
CustomerList)
public ActionResult CustomerList()
{
    //tạo một danh sách khách hàng
    List<Customer> listcustomer = new List<Customer>(){
        new Customer(){ CustomerId = "KH001",
            FullName = "Trịnh Văn Chung",
            Address = "Hà Nội",Email = "devmaster.founder@gmail.com",
            Phone = "0978.611.889",Balance = 15200000},
        new Customer(){ CustomerId = "KH002", FullName = "Đỗ Thị Cúc",
            Address = "Hà Nội",Email = "cucdt@gmail.com",
            Phone = "0986.767.444",Balance = 2200000},
        new Customer(){ CustomerId = "KH003",
            FullName = "Nguyễn Tuấn Thắng",
            Address = "Nam Định",Email = "thangnt@gmail.com",
            Phone = "0924.656.542",Balance = 1200000},
        new Customer(){ CustomerId = "KH004", FullName = "Lê Ngọc Hải",
            Address = "Hà Nội",Email = "hailn@gmail.com",
            Phone = "0996.555.267",Balance = 6200000 }
    };
    //gán dữ liệu vào ViewBag để chuyển qua View
    ViewBag.listcustomer = listcustomer;
    return View();
}
}
```

Bước 6: Kích chuột phải vào tên phương thức **CustomerDetail** chọn Add View -> nhấn Add



Bước 7: Code cho View **CustomerDetail** như sau: (**CustomerDetail.cshtml**)

@*lấy dữ liệu truyền qua đối tượng model theo cách 1*@

@model Lab04.Models.Customer

@{

ViewBag.Title = "Chi tiết khách hàng";

//lấy dữ liệu truyền qua đối tượng theo cách 2

//var customer = Model as Lab04.Models.Customer;

}

<h2>Chi tiết khách hàng lấy theo cách 1 </h2>

<p>Mã số: @Model.CustomerId </p>

<p>Họ và tên: @Model.FullName </p>

<p>Địa chỉ: @Model.Address </p>

<p>Điện thoại: @Model.Phone </p>

<p>Email: @Model.Email </p>

<p>Số tiền dư: @Model.Balance </p>

@*<h2>Chi tiết khách hàng lấy theo cách 1</h2>

<p>Mã số: @customer.CustomerId </p>

<p>Họ và tên: @customer.FullName </p>

<p>Địa chỉ: @customer.Address </p>

```
<p>Điện thoại: @customer.Phone </p>  
<p>Email: @customer.Email </p>  
<p>Số tiền dư: @customer.Balance </p>*@
```

```
@* <h2>Chi tiết khách hàng lấy qua ViewBag</h2>  
    <p>Mã số: @ViewBag.customer.CustomerId </p>  
    <p>Họ và tên: @ViewBag.customer.FullName </p>  
    <p>Địa chỉ: @ViewBag.customer.Address </p>  
    <p>Điện thoại: @ViewBag.customer.Phone </p>  
    <p>Email: @ViewBag.customer.Email </p>  
    <p>Số tiền dư: @ViewBag.customer.Balance </p>*@
```

Bước 8: Mở CustomerController kích chuột phải vào tên phương thức CustomerList chọn Add View -> nhấn Add

Add View

View name: CustomerList

Template: Empty (without model)

Model class:

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

Bước 9: Code cho View **CustomerList** như sau: (CustomerList.cshtml)

```
@{  
    ViewBag.Title="Danh sách khách hàng";  
    //lấy danh sách khách hàng và convert data  
    var listcustomer = ViewBag.listcustomer as List<Lab04.Models.Customer>;  
}  
  
<h2>Danh sách khách hàng</h2>  
<table class="table-bordered">
```

```
<thead>
    <tr>
        <th>Mã KH</th>
        <th>Họ và tên</th>
        <th>Địa chỉ</th>
        <th>Email</th>
        <th>Điện thoại</th>
        <th>Số dư</th>
    </tr>
</thead>
<tbody>
    @*Hiển thị danh sách khách hàng*@
    @foreach (var c in listcustomer)
    {
        <tr>
            <td>@c.CustomerId</td>
            <td>@c.FullName</td>
            <td>@c.Address</td>
            <td>@c.Email</td>
            <td>@c.Phone</td>
            <td>@c.Balance</td>
        </tr>
    }
</tbody>
</table>
```

Bước 10: F5 để chạy và kích vào link “Khách hàng”, “Danh sách khách hàng” kiểm tra kết quả

Chi tiết khách hàng lấy theo cách 1

Mã số: KH001

Họ và tên: Trịnh Văn Chung

Địa chỉ: Hà Nội

Điện thoại: 0978.611.889

Email: devmaster.founder@gmail.com

Số tiền dư: 15200000

© 2018 - ITRD- <http://www.devmaster.edu.vn>

Danh sách khách hàng

Mã KH	Họ và tên	Địa chỉ	Email	Điện thoại	Số dư
KH001	Trịnh Văn Chung	Hà Nội	devmaster.founder@gmail.com	0978.611.889	15200000
KH002	Đỗ Thị Cúc	Hà Nội	cucdt@gmail.com	0986.767.444	2200000
KH003	Nguyễn Tuấn Thắng	Nam Định	thangnt@gmail.com	0924.656.542	1200000
KH004	Lê Ngọc Hải	Hà Nội	hailn@gmail.com	0996.555.267	6200000

© 2018 - ITRD- <http://www.devmaster.edu.vn>

Lab 04.2

Tạo ứng dụng ASP.NET MVC 5 và sử dụng tính năng Scaffolding của Visual Studio để sinh View dựa trên model class “Customer”, thông tin Customer như Lab 04.1. Các thao tác cần thực hiện là hiển thị, tìm kiếm theo tên, thêm, xóa, sửa, xem chi tiết.

Bước 1: Kích chuột phải vào Solution “Lab04” -> chọn Add -> Chọn New Project -> Chọn ASP.NET Web Application và đặt tên project “Lab04_2” -> OK -> Chọn MVC và No Authentication -> OK.

Bước 2: Mở trang _Layout.cshtml trong đường dẫn /View/Shared/_Layout.cshtml và tìm đoạn code sau và thêm vào đúng vị trí

```
<div class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
```



```
<li>@Html.ActionLink("Home", "Index", "Home")</li>
<li>@Html.ActionLink("About", "About", "Home")</li>
<li>@Html.ActionLink("Contact", "Contact", "Home")</li>
<li>@Html.ActionLink("Quản lý khách hàng", "GetCustomers",
"Customer")</li>
</ul>
</div>
```

Bước 3: Tạo lớp “Customer” trong thư mục Models với code gợi ý như sau:

```
//định nghĩa lớp Customer
public class Customer
{
    public string CustomerId { get; set; }
    public string FullName { get; set; }
    public string Address { get; set; }
    public string Email { get; set; }
    public string Phone { get; set; }
    public int Balance { get; set; }
}
```

Bước 4: Tạo giao diện “ICustomerRepository” trong thư mục Modes với code gợi ý sau:

```
//định nghĩa giao diện khách hàng
public interface ICustomerRepository
{
    //khai báo phương thức lấy danh sách khách hàng
    IList<Customer> GetCustomers();
    //khai báo phương thức lấy khách hàng
    Customer GetCustomer(string customerId);
    //khai báo phương thức thêm khách hàng
    void AddCustomer(Customer cus);
    //khai báo phương thức cập nhật khách hàng
    void UpdateCustomer(Customer cus);
    //khai báo phương thức tìm kiếm khách hàng
    IList<Customer> SearchCustomer(string name);
    //khai báo phương thức xóa khách hàng
    void DeleteCustomer(Customer cus);
}
```

Bước 5: Tạo lớp “CustomerRepository” trong thư mục **Modes** thực thi từ giao diện **ICustomerRepository** với code gợi ý sau:

```
//định nghĩa lớp CustomerRepository thực thi từ giao diện
ICustomerRepository

public class CustomerRepository : ICustomerRepository
{
    //khởi tạo danh sách khách hàng
    static readonly List<Customer> data = new List<Customer>()
    {
        new Customer(){ CustomerId = "KH001",
            FullName = "Trịnh Văn Chung",Address = "Hà Nội",
            Email = "devmaster.founder@gmail.com",
            Phone = "0978.611.889",Balance = 15200000},
        new Customer(){ CustomerId = "KH002",
            FullName = "Đỗ Thị Cúc",Address = "Hà Nội",
            Email = "cucdt@gmail.com",Phone = "0986.767.444",
            Balance = 2200000},
        new Customer(){ CustomerId = "KH003",
            FullName = "Nguyễn Tuấn Thắng",Address = "Nam Định",
            Email = "thangnt@gmail.com",Phone = "0924.656.542",
            Balance = 1200000},
        new Customer(){ CustomerId = "KH004",
            FullName = "Lê Ngọc Hải",Address = "Hà Nội",
            Email = "hailn@gmail.com",Phone = "0996.555.267",
            Balance = 6200000}
    };
    //thực thi phương thức lấy danh sách khách hàng
    public IList<Customer> GetCustomers()
    {
        return data;
    }
    //thực thi phương thức tìm khách hàng theo tên
    public IList<Customer> SearchCustomer(string name)
    {
        return data.Where(c=>c.FullName.EndsWith(name)).ToList();
    }
    //thực thi phương thức lấy khách hàng theo Id
    public Customer GetCustomer(string customerId)
    {
        return data.FirstOrDefault(c => c.CustomerId.Equals(customerId));
    }
}
```

```
//thực thi phương thức thêm khách hàng
public void AddCustomer(Customer cus)
{
    data.Add(cus);
}

//thực thi phương thức cập nhật khách hàng
public void UpdateCustomer(Customer cus)
{
    //lấy khách hàng theo id
    var customer = data.FirstOrDefault(c =>
        c.CustomerId.Equals(cus.CustomerId));
    //nếu có thì sửa thông tin
    if(customer!=null)
    {
        customer.FullName = cus.FullName;
        customer.Address = cus.Address;
        customer.Email = cus.Email;
        customer.Phone = cus.Phone;
        customer.Balance = cus.Balance;
    }
}

//thực thi phương thức xóa khách hàng
public void DeleteCustomer(Customer cus)
{
    data.Remove(cus);
}
}
```

Bước 6: Tạo Controller “CustomerController” trong thư mục Controllers

```
public class CustomerController : Controller
{
    //khai báo biến CustomerRepository
    static CustomerRepository listCustomer;
    public CustomerController()
    {
        //khởi tạo CustomerRepository
    }
}
```

```
listCustomer = new CustomerRepository();
}
// GET: /Customer/GetCustomers
public ActionResult GetCustomers()
{
    return View(listCustomer.GetCustomers());
}
//POST: /Customer/GetCustomers
[HttpPost]
public ActionResult GetCustomers(string name)
{
    return View(listCustomer.SearchCustomer(name));
}
// GET: /Customer/Details/5
public ActionResult Details(string id)
{
    return View(listCustomer.GetCustomer(id));
}
// GET: /Customer/Create
public ActionResult Create()
{
    return View();
}
// POST: /Customer/Create
[HttpPost]
public ActionResult Create(Customer cus)
{
    listCustomer.AddCustomer(cus);
    return RedirectToAction("GetCustomers");
}
// GET: /Customer/Edit/5
public ActionResult Edit(string id)
{
    return View(listCustomer.GetCustomer(id));
}
// POST: /Customer/Edit
[HttpPost]
```

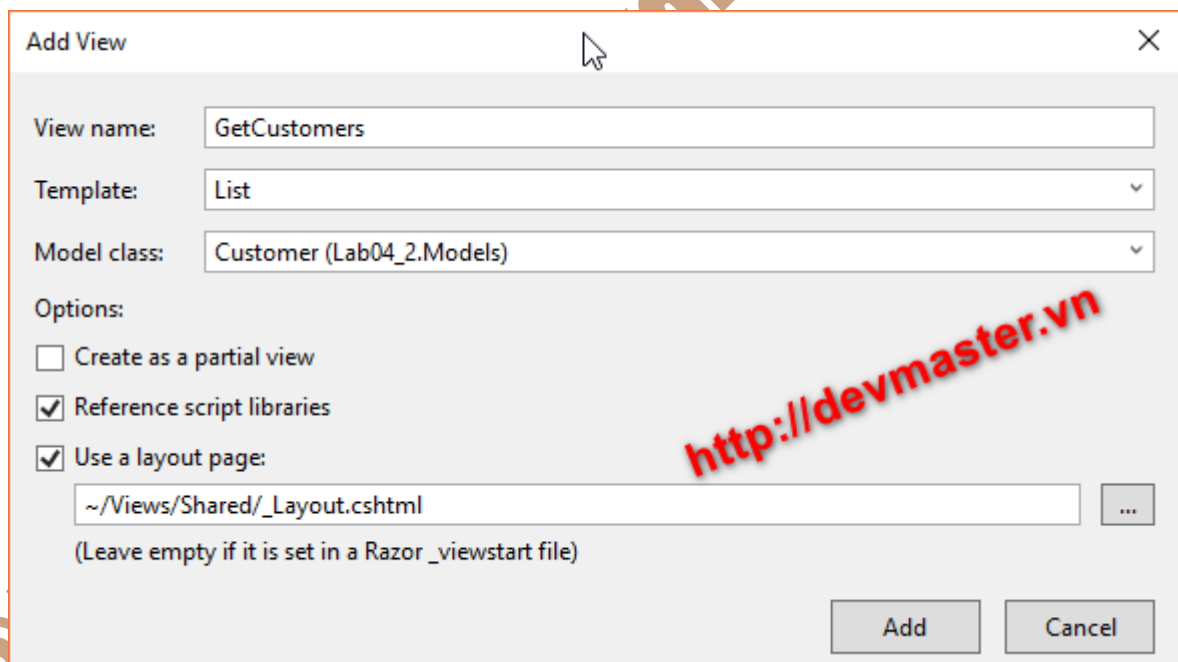
```

public ActionResult Edit(Customer cus)
{
    listCustomer.UpdateCustomer(cus);
    return RedirectToAction("GetCustomers");
}
// GET: /Customer/Delete/5
public ActionResult Delete(string id)
{
    listCustomer.DeleteCustomer(listCustomer.GetCustomer(id));
    return RedirectToAction("GetCustomers");
}
}

```

Bước 7: Tạo các View ứng với các Action trong Controller

- Kích chuột phải vào Action “GetCustomers” và chọn Add View.. -> Chọn template và Model class như hình.



Sau đó chỉnh sửa lại code Razor theo gợi ý sau:

```
@model IEnumerable<Lab04_2.Models.Customer>
```

```
@{
```

```
    ViewBag.Title = "Danh sách khách hàng";
```

```
    Layout = "~/Views/Shared/_Layout.cshtml";
```

```
}
```

```
<h2>Danh sách khách hàng</h2>
```

```
@using (Html.BeginForm())
```

```
{
```

```
    <b>Nhập tên cần tìm </b> @Html.TextBox("name")
```

```
    <input type="submit" value=" Tìm " />
```

```
}
```

```
<p>
```

```
    @Html.ActionLink("Thêm mới", "Create")
```

```
</p>
```

```
<table class="table table-bordered">
```

```
    <tr>
```

```
        <th>
```

```
            Mã khách hàng
```

```
        </th>
```

```
        <th>
```

```
            Họ và tên
```

```
        </th>
```

```
        <th>
```

```
            Địa chỉ
```

```
        </th>
```

```
        <th>
```

```
            Email
```

```
        </th>
```

```
        <th>
```

```
            Điện thoại
```

```
        </th>
```

```
        <th>
```

```
            Số dư
```

```
        </th>
```

```
        <th>Thao tác</th>
```

```
    </tr>
```

```
    @foreach (var item in Model)
```

```
    {
```

```
        <tr>
```

```
            <td>
```

```
                @Html.DisplayFor(modelItem => item.CustomerId)
```

```
            </td>
```

```
<td>
    @Html.DisplayFor(modelItem => item.FullName)
</td>
<td>
    @Html.DisplayFor(modelItem => item.Address)
</td>
<td>
    @Html.DisplayFor(modelItem => item.Email)
</td>
<td>
    @Html.DisplayFor(modelItem => item.Phone)
</td>
<td>
    @Html.DisplayFor(modelItem => item.Balance)
</td>
<td>
    @Html.ActionLink("Sửa", "Edit", new { id = item.CustomerId
    }) |
    @Html.ActionLink("Chi tiết", "Details", new { id =
    item.CustomerId }) |
    @Html.ActionLink("Xóa", "Delete", new { id =
    item.CustomerId }, new { onclick = "return confirm('Bạn có muốn xóa khách
    hàng " + item.CustomerId + " không') " })
</td>
</tr>
}
</table>
```

- Kích chuột phải vào Action “**Details**” và chọn Add View.. -> Chọn template và Model class như hình.

Add View

View name:

Template:

Model 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

Sau đó chỉnh lại code Razor theo gợi ý sau:

```

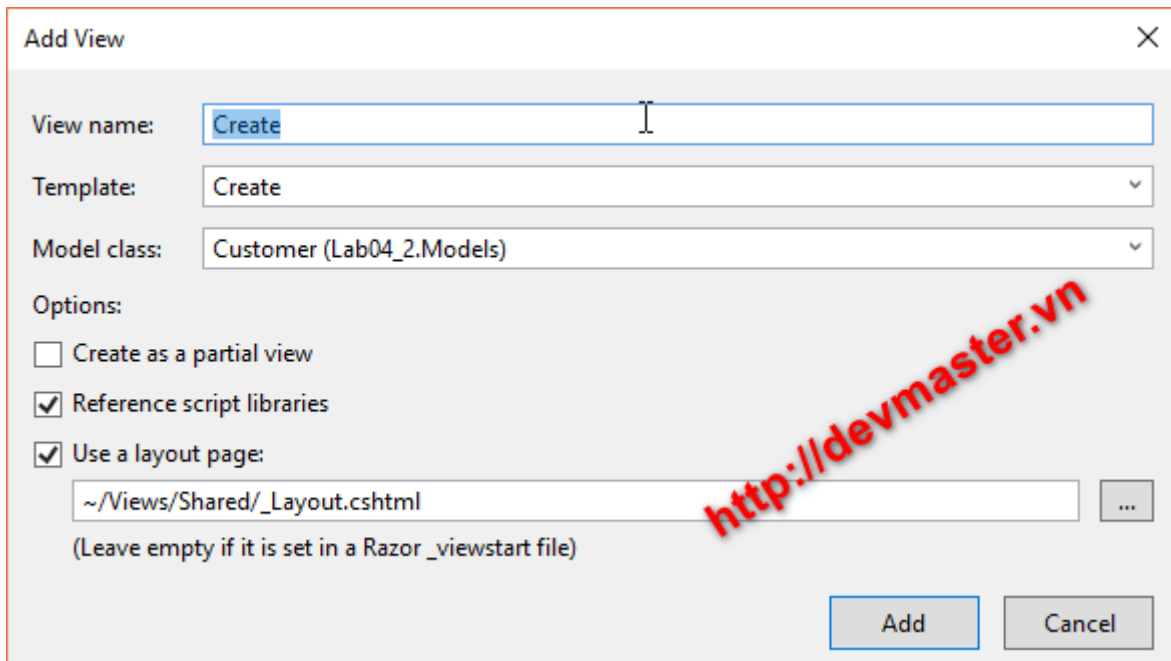
@model Lab04_2.Models.Customer
@{
    ViewBag.Title = "Thông tin chi tiết khách hàng";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<div>
    <h3>Thông tin chi tiết</h3>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            Mã khách hàng
        </dt>
        <dd>
            @Html.DisplayFor(model => model.CustomerId)
        </dd>
        <dt>
            Họ và tên
        </dt>
        <dd>
            @Html.DisplayFor(model => model.FullName)
        </dd>
    </dl>

```



```
        Địa chỉ
    </dt>
    <dd>
        @Html.DisplayFor(model => model.Address)
    </dd>
    <dt>
        Email
    </dt>
    <dd>
        @Html.DisplayFor(model => model.Email)
    </dd>
    <dt>
        Điện thoại
    </dt>
    <dd>
        @Html.DisplayFor(model => model.Phone)
    </dd>
    <dt>
        Số dư
    </dt>
    <dd>
        @Html.DisplayFor(model => model.Balance)
    </dd>
</dl>
</div>
<p>
    @Html.ActionLink("Sửa", "Edit", new { id = Model.CustomerId }) |
    @Html.ActionLink("Quay lại", "GetCustomers")
</p>
```

- Kích chuột phải vào Action **"Create"** và chọn Add View.. -> Chọn template và Model class như hình.



Sau đó chỉnh sửa lại code Razor theo gợi ý sau:

```
@model Lab04_2.Models.Customer
@{
    ViewBag.Title = "Thêm mới khách hàng";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h3>Thêm mới khách hàng</h3>
        <hr />
        @Html.ValidationSummary(true)
        <div class="form-group">
            @Html.Label("Mã khách hàng", new { @class = "control-label col-
md-2" })

            <div class="col-md-10">
                @Html.EditorFor(model => model.CustomerId)
                @Html.ValidationMessageFor(model => model.CustomerId)
            </div>
        </div>
        <div class="form-group">
```

```
@Html.Label("Họ và tên", new { @class = "control-label col-md-2" })

<div class="col-md-10">
    @Html.EditorFor(model => model.FullName)
    @Html.ValidationMessageFor(model => model.FullName)
</div>
</div>

<div class="form-group">
    @Html.Label("Địa chỉ", new { @class = "control-label col-md-2" })

    <div class="col-md-10">
        @Html.EditorFor(model => model.Address)
        @Html.ValidationMessageFor(model => model.Address)
    </div>
</div>

<div class="form-group">
    @Html.Label("Email", new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Email)
        @Html.ValidationMessageFor(model => model.Email)
    </div>
</div>

<div class="form-group">
    @Html.Label("Điện thoại", new { @class = "control-label col-md-2" })

    <div class="col-md-10">
        @Html.EditorFor(model => model.Phone)
        @Html.ValidationMessageFor(model => model.Phone)
    </div>
</div>

<div class="form-group">
    @Html.Label("Số dư", new { @class = "control-label col-md-2" })
    <div class="col-md-10">
```

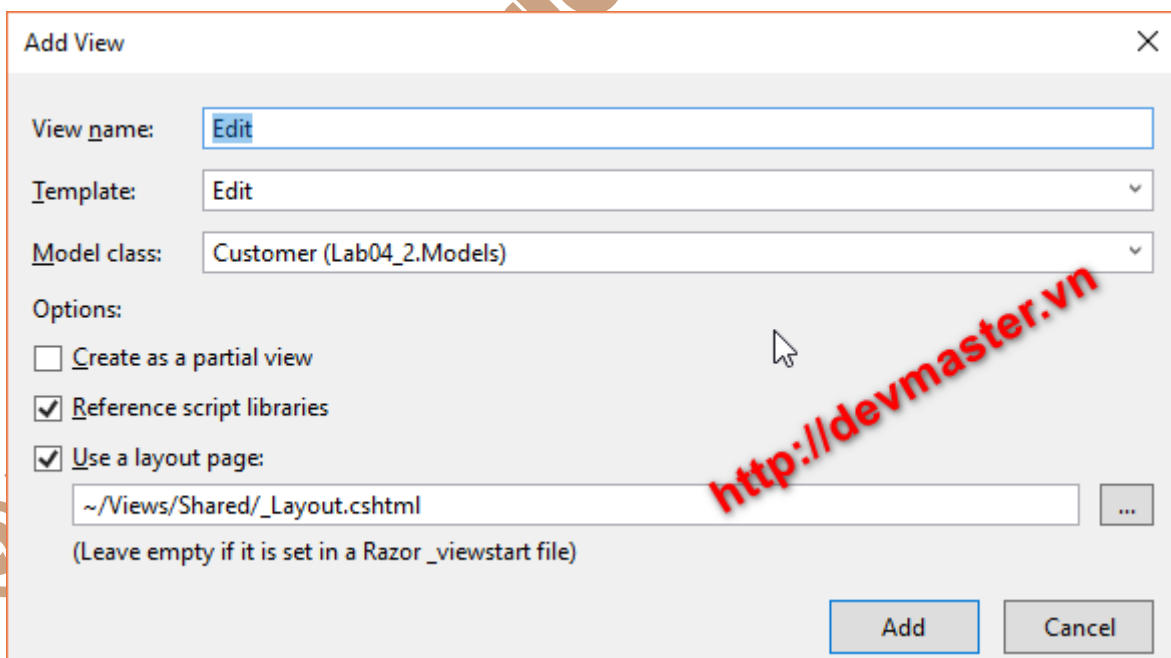
```

@Html.EditorFor(model => model.Balance)
@Html.ValidationMessageFor(model => model.Balance)
</div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value=" Lưu " class="btn btn-default"
/>
    </div>
</div>
</div>
}
<div>
    @Html.ActionLink("Quay lại", "GetCustomers")
</div>

```

- Kích chuột phải vào Action “**Edit**” và chọn Add View.. -> Chọn template và Model class như hình.



Sau đó chỉnh sửa code Razor theo gợi ý sau:

```

@model Lab04_2.Models.Customer
@{

```

```
        ViewBag.Title = "Chỉnh sửa khách hàng";
        Layout = "~/Views/Shared/_Layout.cshtml";
    }
    @using (Html.BeginForm())
    {
        @Html.AntiForgeryToken()

        <div class="form-horizontal">
            <h3>Chỉnh sửa khách hàng</h3>
            <hr />
            @Html.ValidationSummary(true)
            @Html.HiddenFor(model => model.CustomerId)

            <div class="form-group">
                @Html.Label("Họ và tên", new { @class = "control-label col-md-2" })

                <div class="col-md-10">
                    @Html.EditorFor(model => model.FullName)
                    @Html.ValidationMessageFor(model => model.FullName)
                </div>
            </div>

            <div class="form-group">
                @Html.Label("Địa chỉ", new { @class = "control-label col-md-2" })

                <div class="col-md-10">
                    @Html.EditorFor(model => model.Address)
                    @Html.ValidationMessageFor(model => model.Address)
                </div>
            </div>

            <div class="form-group">
                @Html.Label("Email", new { @class = "control-label col-md-2" })

                <div class="col-md-10">
                    @Html.EditorFor(model => model.Email)
                    @Html.ValidationMessageFor(model => model.Email)
                </div>
            </div>
        </div>
    }
```

```
<div class="form-group">
    @Html.Label("Điện thoại", new { @class = "control-label col-md-
2" })

    <div class="col-md-10">
        @Html.EditorFor(model => model.Phone)
        @Html.ValidationMessageFor(model => model.Phone)
    </div>
</div>

<div class="form-group">
    @Html.Label("Số dư", new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Balance)
        @Html.ValidationMessageFor(model => model.Balance)
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value=" Lưu " class="btn btn-default"
/>
    </div>
</div>
</div>
}
<div>
    @Html.ActionLink("Quay lại", "GetCustomers")
</div>
```

Bài 8: Browse trang **Index.cshtml** và kích vào các link để kiểm tra kết quả

- Danh sách khách hàng

localhost:39220/Customer/GetCustomers

DEVMaster.vn Home About Contact **Quản lý khách hàng**

Danh sách khách hàng

Nhập tên cần tìm

[Thêm mới](#)

Mã khách hàng	Họ và tên	Địa chỉ	Email	Điện thoại	Số dư	Thao tác
KH001	Trình Văn Chung	Hà Nội	devmaster.founder@gmail.com	0978.611.889	15200000	Sửa Chi tiết Xóa
KH002	Đỗ Thị Cúc	Hà Nội	cucdt@gmail.com	0986.767.444	2200000	Sửa Chi tiết Xóa
KH003	Nguyễn Tuấn Thắng	Nam Định	thangnt@gmail.com	0924.656.542	1200000	Sửa Chi tiết Xóa
KH004	Lê Ngọc Hải	Hà Nội	hailn@gmail.com	0996.555.267	6200000	Sửa Chi tiết Xóa

© 2018 - ITRD- <http://www.devmaster.edu.vn>

- Tìm kiếm theo tên

DEVMaster.vn Home About Contact **Quản lý khách hàng**

Danh sách khách hàng

Nhập tên cần tìm

[Thêm mới](#)

Mã khách hàng	Họ và tên	Địa chỉ	Email	Điện thoại	Số dư	Thao tác
KH001	Trình Văn Chung	Hà Nội	devmaster.founder@gmail.com	0978.611.889	15200000	Sửa Chi tiết Xóa

© 2018 - ITRD- <http://www.devmaster.edu.vn>

- Thêm mới

Thêm mới khách hàng

Mã khách hàng

Họ và tên

Địa chỉ

Email

Điện thoại

Số dư

[Quay lại](#)© 2018 - ITRD- <http://www.devmaster.edu.vn>

- Chức năng Sửa

@DEV - www.edu.vn



Chỉnh sửa khách hàng

Họ và tên

Địa chỉ

Email

Điện thoại

Số dư

[Quay lại](#)© 2018 - ITRD- <http://www.devmaster.edu.vn>[- Xem Chi tiết](#)

Thông tin chi tiết

Mã khách hàng

KH001

Họ và tên

Trịnh Văn Chung

Địa chỉ

Hà Nội

Email

devmaster.founder@gmail.com

Điện thoại

0978.611.889

Số dư

15200000

[Sửa](#) | [Quay lại](#)© 2018 - ITRD- <http://www.devmaster.edu.vn>[- Xóa có hỏi](#)

DEVMASTER.vn
Home
About

Danh sách khách hàng

Thêm mới

Mã khách hàng	Họ và tên	Địa chỉ	Email	Điện thoại	Số dư	Thao tác
KH001	Trịnh Văn Chung	Hà Nội	devmaster.founder@gmail.com	0978.611.889	15200000	Sửa Chi tiết Xóa

Phần II Bài tập tự làm

Bài 1: Tạo ứng dụng ASP.NET MVC 5 với các yêu cầu sau

- Tạo tệp dữ liệu XML có cấu trúc sau

```
<?xml version="1.0" encoding="utf-8" ?>
<Products>
  <Product>
    <ProductId>P001</ProductId>
    <ProductName>Lẩu vịt</ProductName>
    <Unit>Nồi</Unit>
    <Price>150000</Price>
  </Product>

  <Product>
    <ProductId>P002</ProductId>
    <ProductName>Lẩu cá kèo</ProductName>
    <Unit>Nồi</Unit>
    <Price>250000</Price>
  </Product>

  <Product>
    <ProductId>P003</ProductId>
    <ProductName>Lẩu thập cẩm</ProductName>
    <Unit>Nồi</Unit>
    <Price>350000</Price>
  </Product>
</Products>
```

- Tạo các lớp và giao diện trong Model để thao tác với dữ liệu trong tệp XML trên

- Tạo ProductController cung cấp các Action để thực hiện các thao tác thêm, xóa, sửa, tìm kiếm, hiển thị các thực đơn.
- Tạo View để quan sát và tương tác
- Tạo các lớp và giao diện trong Model để thao tác với dữ liệu trong tệp XML trên
- Tạo ProductController cung cấp các Action để thực hiện các thao tác thêm, xóa, sửa, tìm kiếm, hiển thị các thực đơn.
- Tạo View để quan sát và tương tác

Product ID	Product Name	Unit	Price	Action
P001	Lẩu vịt	Nồi	150000	+ ✎ 🗑
P002	Lẩu cá kèo	Nồi	250000	+ ✎ 🗑
P003	Lẩu thập cẩm	Nồi	350000	+ ✎ 🗑

© 2018 - ITRD- <http://www.devmaster.edu.vn>

Bài 1: Tạo ứng dụng ASP.NET MVC 5 với các yêu cầu sau










- Tạo tệp dữ liệu XML có cấu trúc sau

```
<NhaHang>
  <Category CategoryId="L01" CategoryName="Lẩu">
    <Product>
      <ProductId>P001</ProductId>
      <ProductName>Lẩu vịt</ProductName>
      <Unit>Nồi</Unit>
      <Price>150000</Price>
    </Product>
    <Product>
      <ProductId>P002</ProductId>
      <ProductName>Lẩu cá kèo</ProductName>
      <Unit>Nồi</Unit>
      <Price>250000</Price>
    </Product>
  </Category>
</NhaHang>
```

```
</Category>
<Category CategoryId="L01" CategoryName="Tôm">
.....
</Category>
.....
</ NhaHang>
```

- Tạo các lớp và giao diện trong Model để thao tác với dữ liệu trong tệp XML trên
- Tạo NhaHangController cung cấp các Action để thực hiện các thao tác thêm, xóa, sửa, tìm kiếm, hiển thị các thực đơn.
- Tạo View để quan sát và tương tác
- Tạo các lớp và giao diện trong Model để thao tác với dữ liệu trong tệp XML trên
- Tạo ProductController cung cấp các Action để thực hiện các thao tác thêm, xóa, sửa, tìm kiếm, hiển thị các thực đơn.
- Tạo View để quan sát và tương tác
- Trang hiển thị sản phẩm sẽ cho phép chọn loại thực đơn trong Combobox sau đó hiển thị các thực đơn tương ứng

Chọn thực đơn

Tên món	Đơn vị tính	Giá	Xóa	Sửa	Chi tiết
Lẩu thập cẩm	Nồi	200000			
Lẩu hải sản	Nồi	160000			
Lẩu cá kèo	Nồi	230000			

Bài 2: Tạo ứng dụng ASP.NET MVC cho phép quản lý album ảnh với các yêu cầu sau

- Tạo thư mục Images trong thư mục content của project
- Copy vào đó 1 số ảnh
- Tạo View cho phép hiển thị các ảnh trong thư mục lên màn hình (xem hình minh họa)
- Upload ảnh vào thư mục
- Xóa ảnh

Quản lý ảnh slide show



Upload ảnh No file chosen

Khuyến cáo: Nên chọn kích thước ảnh 990x320- Số lượng ảnh tối đa là 15

HẾT