

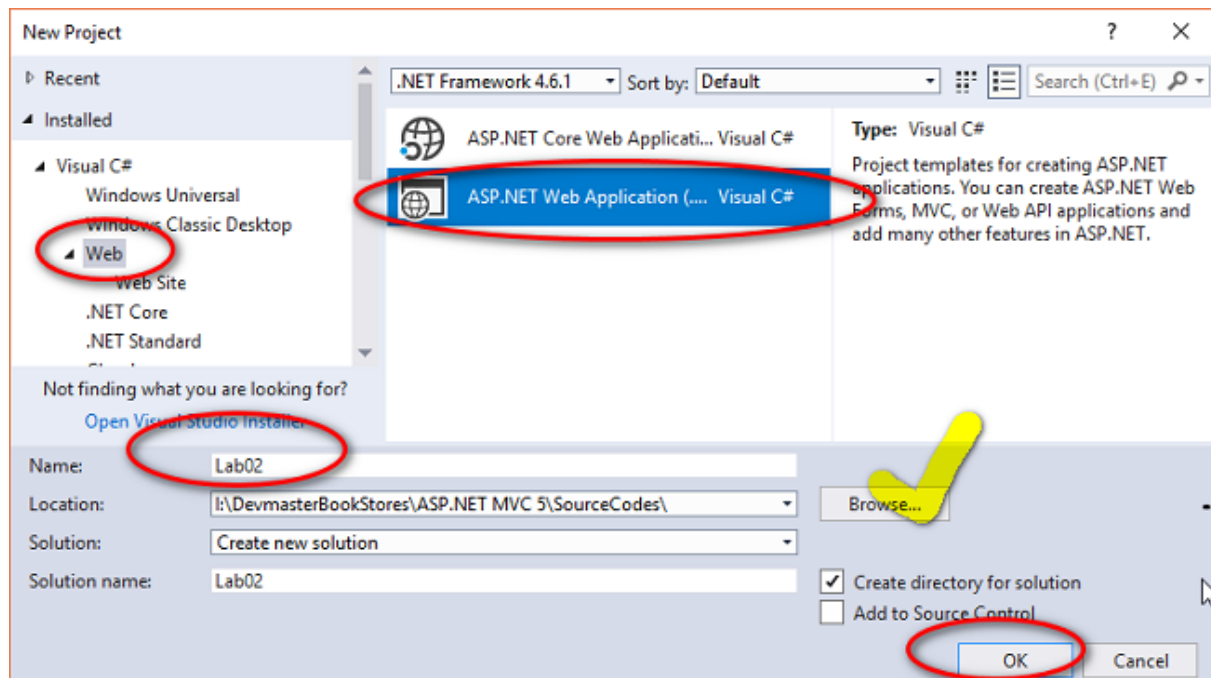
Lab02 - Controller in ASP.NET MVC

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

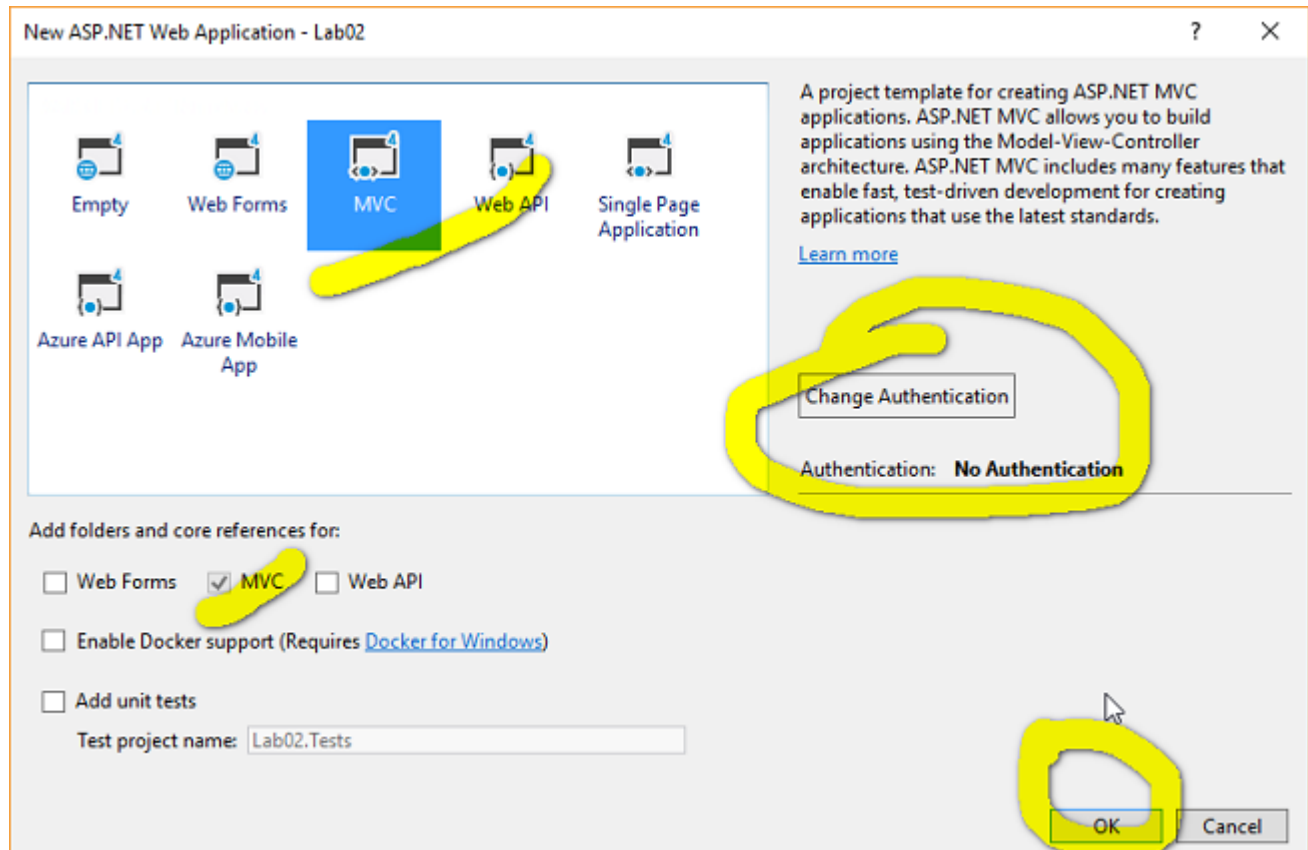
- Tạo các **Action** với các kiểu trả về **ActionResult** khác nhau trong **Controller**.

ActionResult	Helper Method	Description
ViewResult	View	Renders a view as a web page
PartialViewResult	PartialView	Section of a view, that can be rendered inside another view
RedirectResult	Redirect	Redirect to another action method
RedirectToRouteResult	RedirectToRoute	Redirect to another action method
ContentResult	Content	Returns a user-defined content type
JsonResult	Json	Returns a serialized JSON object
JavaScriptResult	JavaScript	Returns a script that can be executed on the client
FileResult	File	Returns a binary output to write to the response
EmptyResult	(None)	returns a null result

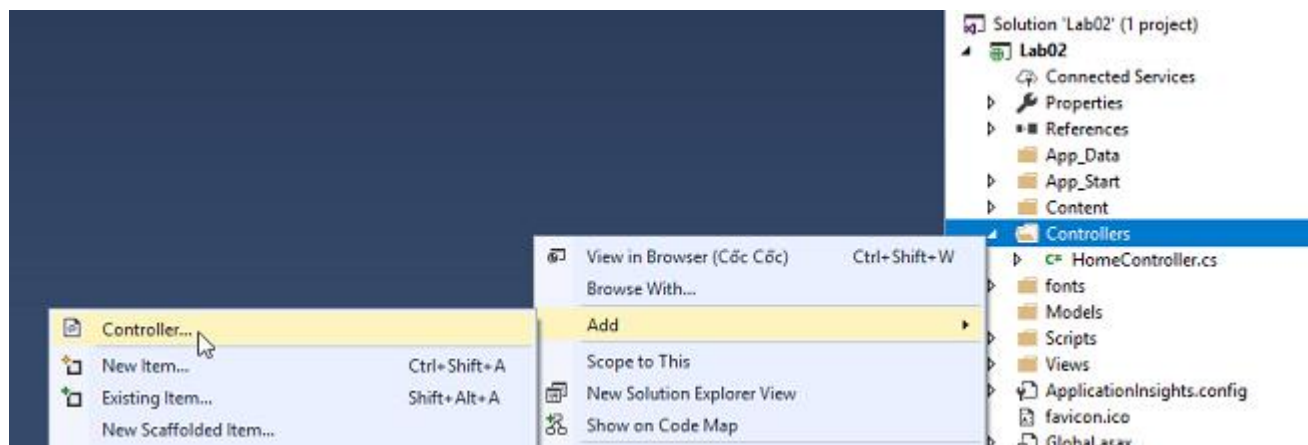
Bước 1: Mở Visual Studio 2017/2019 -> tạo project vào File -> New -> Projects -> Chọn Web -> ASP.NET Web Application -> Nhập tên project -> Chọn đường dẫn lưu trữ -> Nhập tên solution -> Nhấn OK (xem hình dưới).



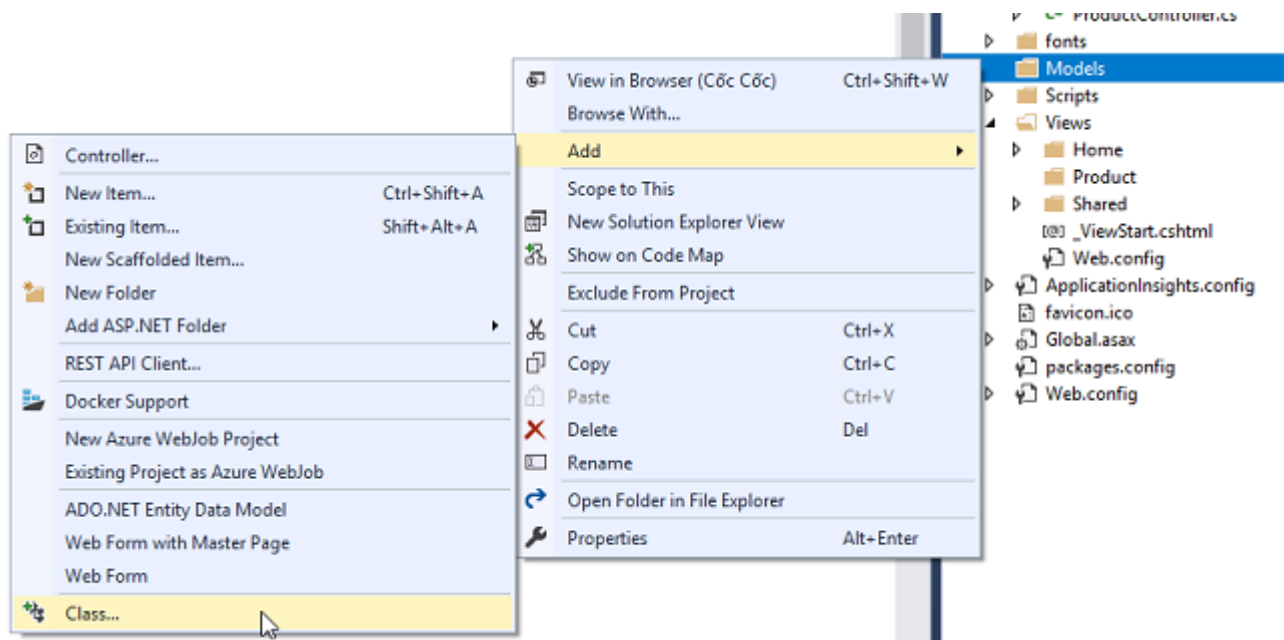
Nhấn OK -> màn hình tiếp theo xuất hiện bạn chọn template là MVC sau đó chọn Change Authentication -> No Authentication -> OK -> OK (Xem hình bên dưới)



Bước 2: Kích chuột phải vào thư mục Controller -> chọn Add -> chọn Controller -> Nhập tên Controller -> Add (Tên controller : **ProductController**)



Bước 3: kích chuột phải vào thư mục Model Chọn Add-> Chọn Class-> nhập tên Student -> thêm code theo gợi ý sau:



```
namespace Lab02_WebApplication1.Models
{
    public class Student
    {
        public int ID { get; set; }
        public string Name { get; set; }
        public string ClassName { get; set; }
    }
}
```

Bước 4: Copy các tệp tin tài nguyên vào thư mục Content (demovideo.mp4, vonsong.docx). Sinh viên có thể thay thế các file tài nguyên khác để minh họa cho chương trình.

Bước 5: Mở ProductController và tạo các Action ứng với từng loại ActionResult theo gợi ý sau:

```
using System.Text;
using System.IO;
using System.Xml.Linq;
using Lab02_WebApplication1.Models;

namespace Lab02_WebApplication1.Controllers
{
    public class ProductController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }
        //Action trả về một View là một trang html và nó có thể truyền tham số hóa model
        public ViewResult TestViewResult()
        {
        }
```

```

        return View();
    }
    //Action trả về một PartialViewResult
    public PartialViewResult TestPartialViewResult()
    {
        return PartialView();
    }
    //Action trả về một View trống (null)
    public EmptyResult TestEmptyResult()
    {
        return new EmptyResult();
    }
    // Action đáp ứng việc chuyển trực tiếp tới một view khác
    public RedirectResult TestRedirectResult()
    {
        return Redirect("Index");
    }
    //Action trả về một kết quả dạng Json
    public JsonResult TestJsonResult()
    {
        List<Student> listStudent = new List<Student>();
        listStudent.Add(new Student() { ID = 001,
            Name = "Nguyễn Quang Huy", ClassName = "C1311L" });
        listStudent.Add(new Student() { ID = 001,
            Name = "Nguyễn Quang Huy", ClassName = "C1311J" });
        listStudent.Add(new Student() { ID = 001,
            Name = "Nguyễn Quang Hiên", ClassName = "C1311H" });
        listStudent.Add(new Student() { ID = 001,
            Name = "Nguyễn Duy Huân", ClassName = "C1311T" });
        listStudent.Add(new Student() { ID = 001,
            Name = "Vũ Quang Huy", ClassName = "C1311C" });
        listStudent.Add(new Student() { ID = 001,
            Name = "Trần Quang Huy", ClassName = "C1311L" });
        listStudent.Add(new Student() { ID = 001,
            Name = "Phạm Quang Huy", ClassName = "C1311L" });
        listStudent.Add(new Student() { ID = 001,
            Name = "Trịnh Quang Huy", ClassName = "C1311B" });
        listStudent.Add(new Student() { ID = 001,
            Name = "Vũ Quang Huy", ClassName = "C1311L" });
        listStudent.Add(new Student() { ID = 001,
            Name = "Vũ Minh Trịnh", ClassName = "C1311M" });
        return Json(listStudent, JsonRequestBehavior.AllowGet);
    }
    //Action trả về một view là JavaScript
    public JavaScriptResult TestJavaScriptResult()
    {
        string js = "function checkEmail(){var btloc=/^([w-]+(?:.[w-]+)*)@((?:[w-]+.)*w[w-]{0,66}).([a-z]{2,6}(?:.[a-z]{2})?)$/i if(btloc.test(mail)) {kq=true ;} else { alert("Email address invalid"); kq=false; } return kq;}";
        return JavaScript(js);
    }
    //Action trả về một ContentResult dữ liệu là dạng văn bản
    public ContentResult TestContentResult()
    {
        XElement contentXML = new XElement("Students",
            new XElement("Student",
                new XElement("ID", "001"),
                new XElement("FullName", "Nguyễn Viết Nam"),
                new XElement("ClassName", "C1308H"),
                new XElement("Student",

```

```

        new XElement("ID", "002"),
        new XElement("FullName", "Nguyễn Hoàng Anh"),
        new XElement("ClassName", "C1411P"),
        new XElement("Student",
            new XElement("ID", "003"),
            new XElement("FullName", "Phạm Ngọc Anh"),
            new XElement("ClassName", "C1411L"),
            new XElement("Student",
                new XElement("ID", "004"),
                new XElement("FullName", "Trần Ngọc Linh"),
                new XElement("ClassName", "C1411H"),
                new XElement("Student",
                    new XElement("ID", "005"),
                    new XElement("FullName", "Nguyễn Hồng Anh"),
                    new XElement("ClassName", "C1411L"))));
        return Content(contentXML.ToString(), "text/xml", Encoding.UTF8);
    }
    // Cả ba kiểu FileContentResult, FileStreamResult, FilePathResult đều cho phép trình
    duyệt mở hộp thoại lưu file và tải file về
    // phương thức trả về có 3 tham số
    // tham số thứ nhất đối với kiểu FileContentResult là một mảng byte của file
    // tham số thứ nhất đối với kiểu FileStreamResult là một FileStream
    // tham số thứ nhất đối với kiểu PathFileResult là một đường dẫn file
    // tham số thứ hai chỉ ra loại định dạng của file
    // tham số thứ ba tên file mà trình duyệt sẽ tải về
    public FileContentResult TestFileContentResult()
    {
        byte[] fileBytes = File.ReadAllBytes(Server.MapPath("~/Content/demovideo.mp4"));
        string fileName = "demovideo.mp4";
        //return File(fileBytes, System.Net.Mime.MediaTypeNames.Application.Octet, fileName);
        return File(fileBytes, "video/mp4", fileName);
    }
    public FileStreamResult TestFileStreamResult()
    {
        string pathFile = Server.MapPath("~/Content/vonsong.docx");
        string fileName = "vonsong.docx";
        return File(new FileStream(pathFile, FileMode.Open), "text/doc", fileName);
    }
    public FilePathResult TestFilePathResult()
    {
        string pathFile = Server.MapPath("~/Content/vonsong.docx");
        string fileName = "vonsong.docx";
        return File(pathFile, "text/doc", fileName);
    }
}
}

```

Bước 6: Tạo các View cho các Action trong **ProductController** với code gợi ý như sau:

- **Views/index.cshtml**

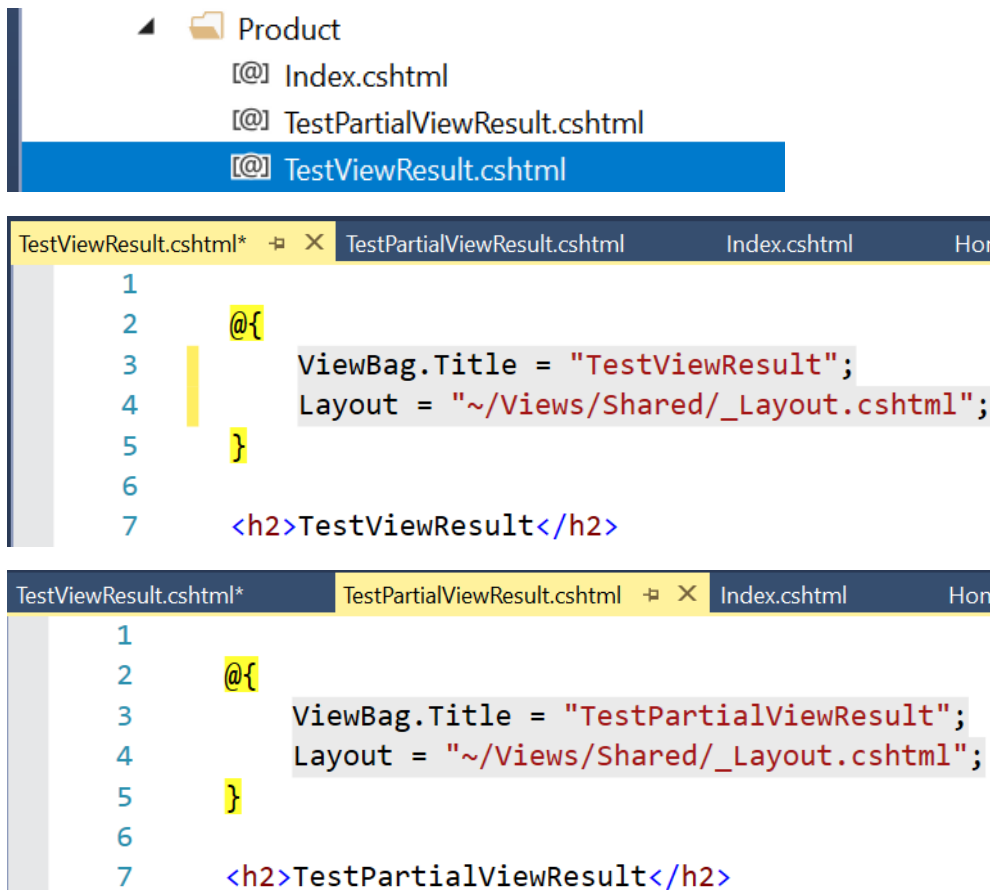
```

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<br />
<br />
<br />

```

```
<h2>Trang chủ</h2>
Views/TestPartialViewResult.cshtml
@{
    ViewBag.Title = "TestPartialViewResult";
}
<h3>PartialViewResult</h3>
<ol>
    <li>item 1</li>
    <li>item 2</li>
    <li>item 3</li>
    <li>item 4</li>
    <li>item 5</li>
</ol>
Views/TestViewResult.cshtml
@{
    ViewBag.Title = "TestViewResult";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<br />
<br />
<h2>ViewResult </h2>
```

- Tương tự sinh viên tạo **Views TestViewResult và TestPartialViewResult:**



Bước 7: Mở `Views/Shared/_Layout.cshtml` tìm và chỉnh lại code như sau:

```
<div class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <li>@Html.ActionLink("ViewResult", "TestViewResult", "Product")</li>

    <li>@Html.ActionLink("PartialViewResult", "TestPartialViewResult", "Product")</li>

    <li>@Html.ActionLink("EmptyResult", "TestEmptyResult", "Product")</li>

    <li>@Html.ActionLink("RedirectResult ", "TestRedirectResult", "Product")</li>

    <li>@Html.ActionLink("JsonResult", "TestJsonResult", "Product")</li>

    <li>@Html.ActionLink("JavaScriptResult", "TestJavaScriptResult", "Product")</li>

    <li>@Html.ActionLink("ContentResult", "TestContentResult", "Product")</li>

    <li>@Html.ActionLink("FileContentResult ", "TestFileContentResult", "Product")</li>

    <li>@Html.ActionLink("FileStreamResult", "TestFileStreamResult", "Product")</li>

    <li>@Html.ActionLink("FilePathResult", "TestFilePathResult", "Product")</li>
  </ul>
</div>
```

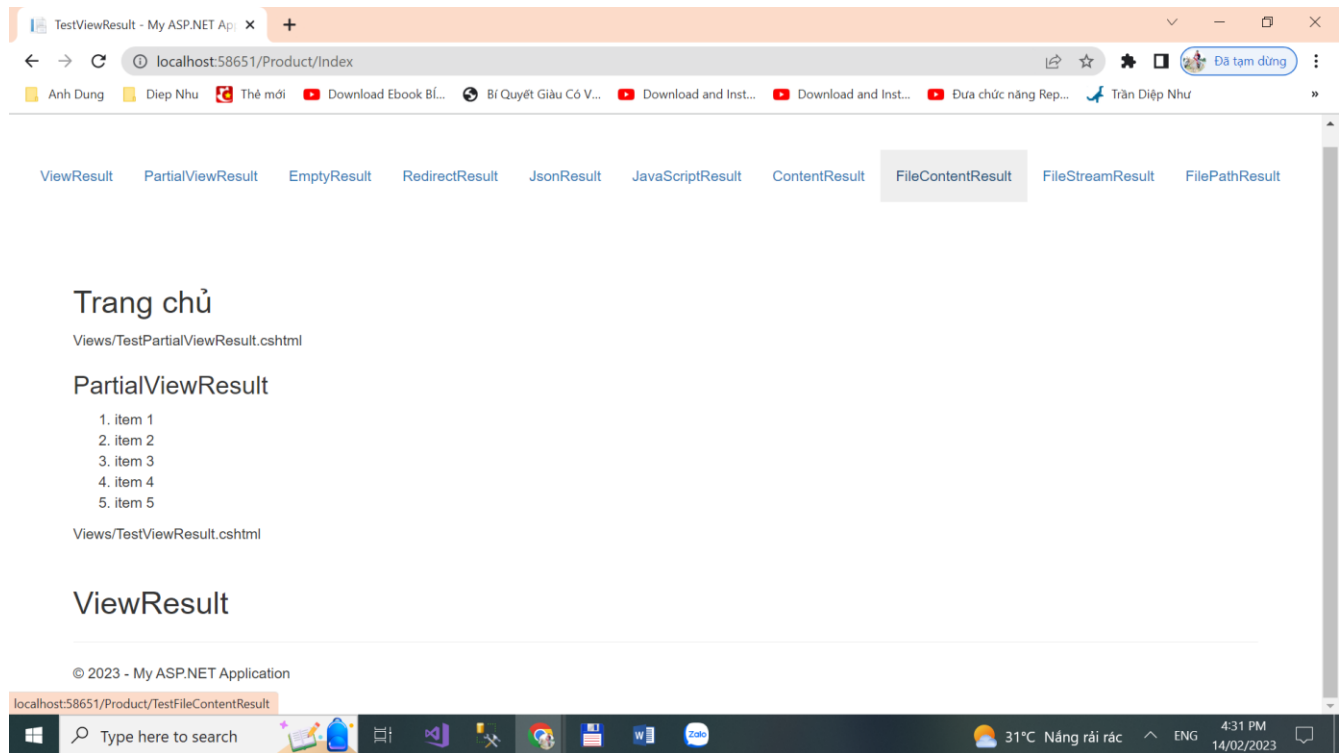
Bước 8 : Cấu hình lại **RouteConfig** cho ứng dụng:

App_Start
 ▶ BundleConfig.cs
 ▶ FilterConfig.cs
 ▶ RouteConfig.cs

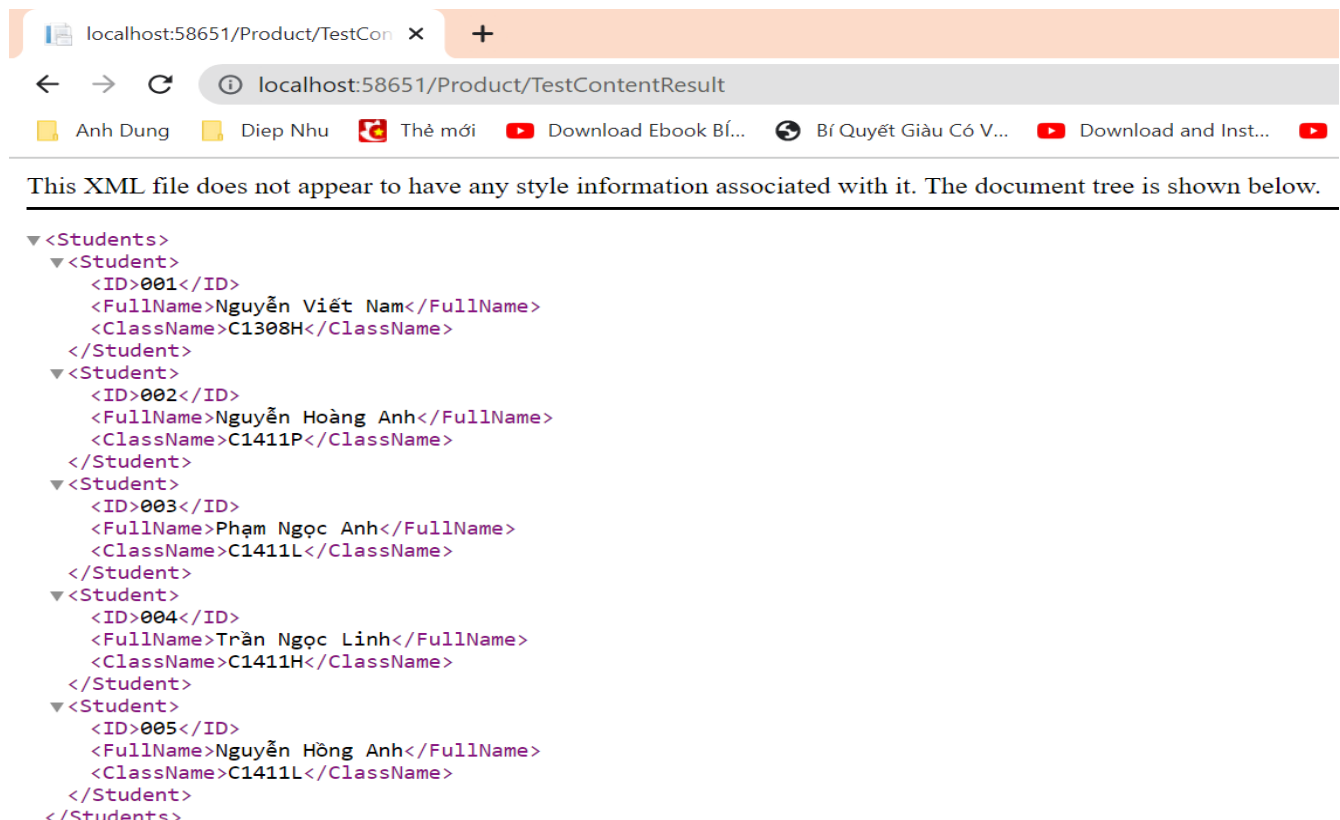
```
public class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

        routes.MapRoute(
            name: "Default",
            url: "{controller}/{action}/{id}",
            defaults: new { controller = "Product",
                           action = "Index", id = UrlParameter.Optional }
        );
    }
}
```

Bước 9 : F5 để chạy và xem kết quả



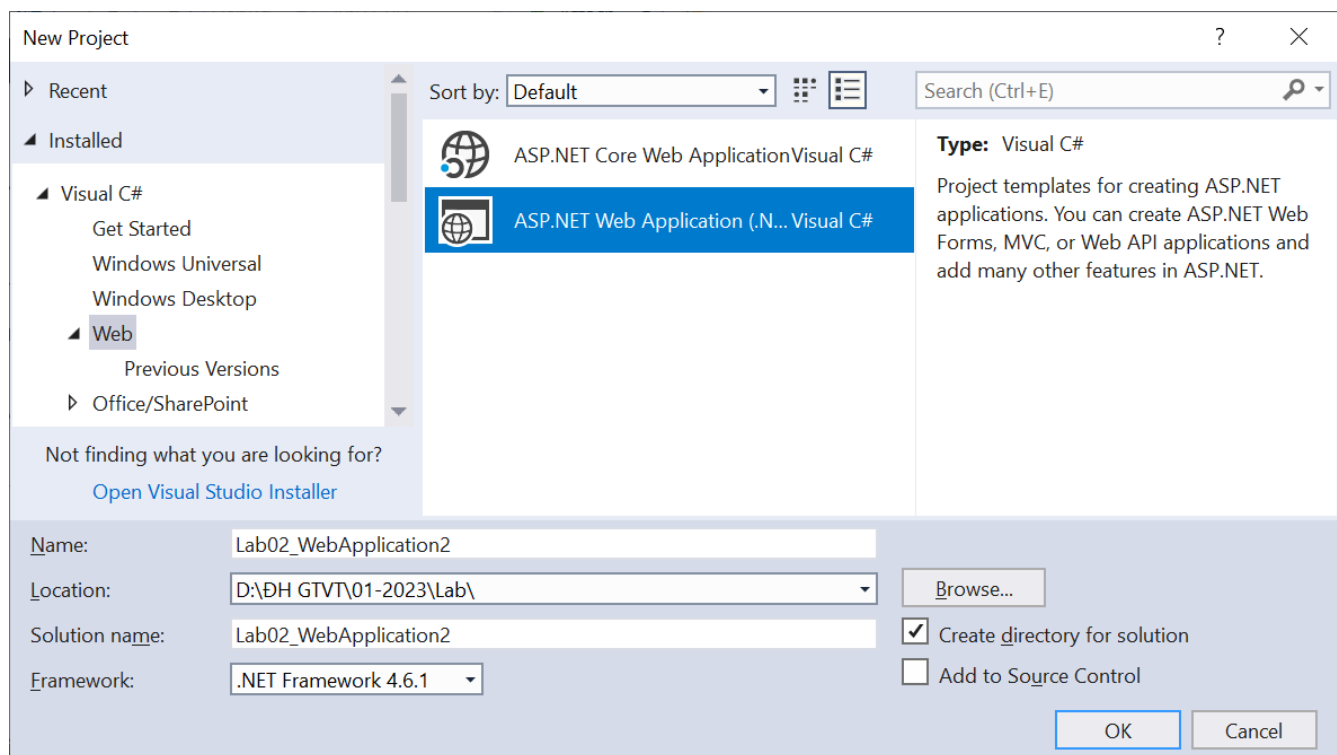
Lần lượt click chuột vào từng link để xem kết quả mỗi một link kết quả của một kiểu trả về...
Hình bên dưới là kết quả của kiểu ContentResult



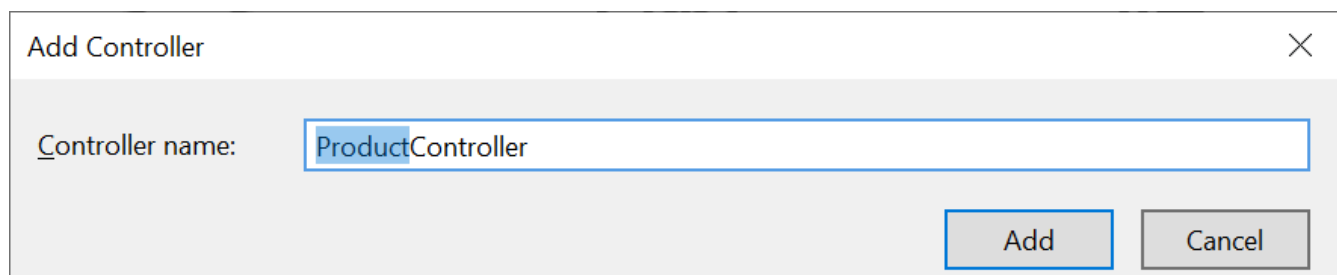
Bài 2: Tạo ứng dụng ASP.NET MVC 5, tạo các Url thân thiện với google sử dụng MapRoute và các Url Pattern:

- Trang chủ hiển thị link “Sản phẩm” với mẫu: <http://domain//San-Pham/Danh-Sach-San-Pham>
- Trang sản phẩm hiển thị danh sách sản phẩm với các link “Xem chi tiết” với mẫu: <http://domain/San-Pham/Ten-san-pham/ma-so>, Link “Sửa sản phẩm” với mẫu: <http://domain/San-Pham/Sua/ma-so>

Bước 1: Kích chuột phải vào Solution “Lab02” -> Chọn Add -> Chọn New Project Nhập tên project -> Chọn đường dẫn lưu trữ -> Nhấn OK (xem hình dưới).



Bước 2: kích chuột phải vào thư mục Controller -> chọn Add -> chọn Controller -> Nhập tên Controller -> Add



Bước 3: Mở `ProductController.cs`, tạo các action theo gợi ý sau:

```
public class ProductController : Controller
{
    // GET: Product
    public ActionResult ShowProduct()
    {
        return View();
    }
    // GET: Action sửa sản phẩm
    public ActionResult EditProduct(int? productId)
    {
        ViewBag.id = productId;
        return View();
    }
    //GET: Action chi tiết sản phẩm
    public ActionResult DetailsProduct(string productName, int? productId)
    {
        ViewBag.name = productName;
        ViewBag.id = productId;
        return View();
    }
}
```

Bước 4: Tạo các View cho các Action trong `ProductController`.

- View `ShowProduct.cshtml`

```
@{
    ViewBag.Title = "ShowProduct";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Danh Sách Sản Phẩm</h2>

<table class="table table-bordered">
    <thead>
        <tr>
            <th>Mã Sản Phẩm</th>
            <th>Tên Sản Phẩm</th>
            <th>Lựa Chọn</th>
        </tr>
    </thead>
```

```
<tbody>
  @for (int i = 1; i < 10; i++)
  {
    <tr>
      <td>@i</td>
      <td>Sản Phẩm Thứ @i</td>
      <td>
        @Html.ActionLink("Chi Tiết", "DetailsProduct",
          new { productName = "Sản Phẩm Thứ " + @i, productId = @i }) |
        @Html.ActionLink("Sửa", "EditProduct", new { productId = @i })
      </td>
    </tr>
  }
</tbody>
</table>
```

- View **EditProduct**.cshtml

```
@{
    ViewBag.Title = "EditProduct";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Sửa thông tin chi tiết sản phẩm </h2><br />

<h3> Mã Sản Phẩm : @ViewBag.id</h3>
```

- View **DetailsProduct**.cshtml

```
@{
    ViewBag.Title = "DetailsProduct";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Thông Tin Sản Phẩm</h2>

<h3> Mã Sản Phẩm : @ViewBag.id</h3>

<h3> Tên sản phẩm : @ViewBag.name</h3>
```

Bước 5: Mở [Views/Shared/_Layout.cshtml](#) tìm và chỉnh lại như sau:

```
<div class="navbar-collapse collapse">
    <ul class="nav navbar-nav">
        <li>@Html.ActionLink("Sản Phẩm", "ShowProduct", "Product")</li>
    </ul>
</div>
```

Bước 6: Mở tệp tin [App_Start/RouteConfig.cs](#) và chèn đoạn [MapRoute](#) sau vào trước [MapRoute default](#)

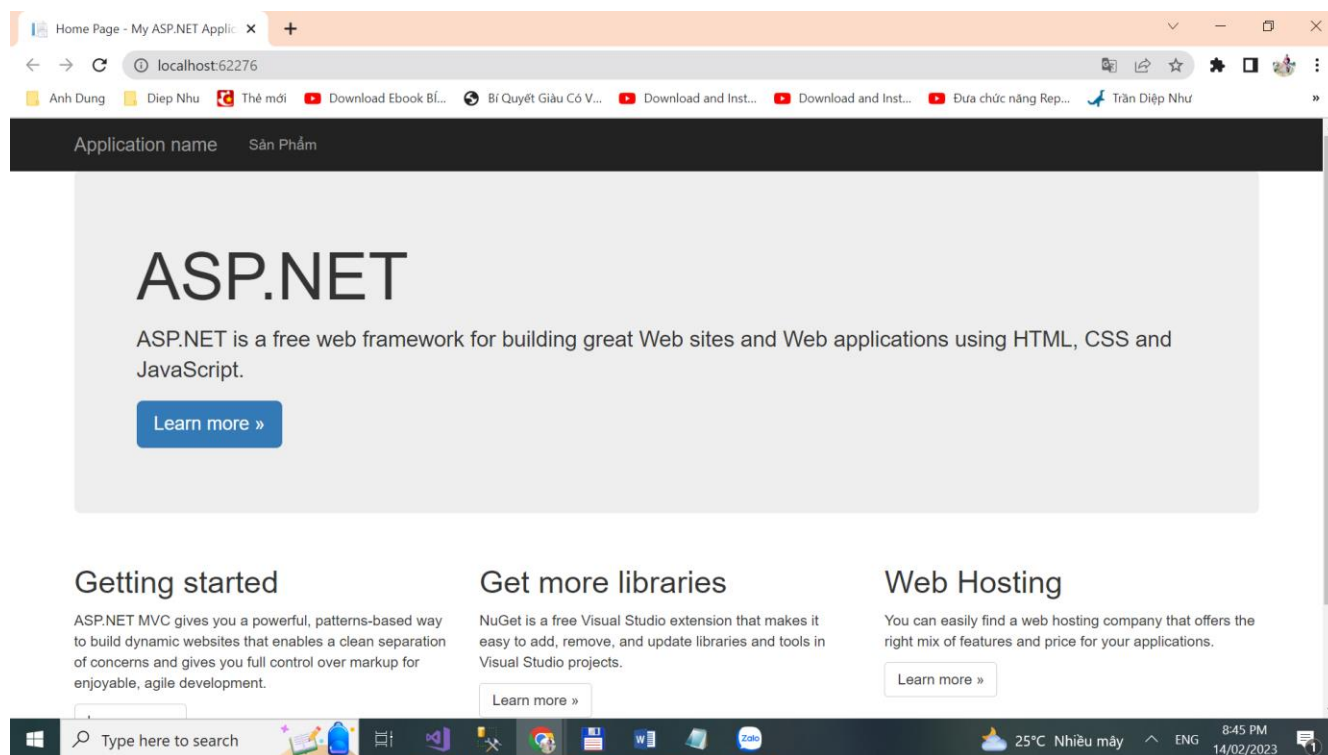
```
//http://domain-name/San-Pham/Danh-Sach-San-Pham
routes.MapRoute(
    name: "Show Product",
    url: "San-Pham/Danh-Sach-San-Pham",
    defaults: new { controller = "Product", action = "ShowProduct" }
);

//http://domain-name/San-Pham
routes.MapRoute(
    name: "Product",
    url: "San-Pham",
    defaults: new { controller = "Product", action = "ShowProduct" }
);

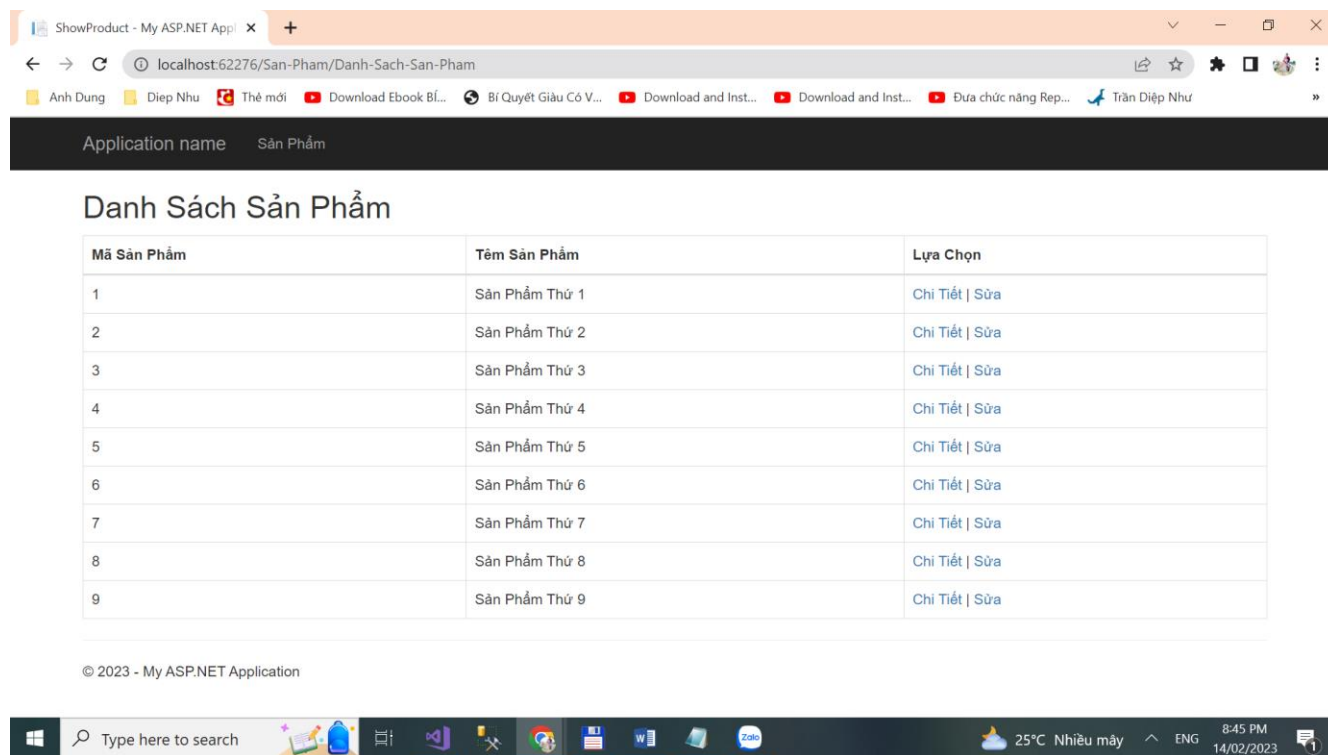
//http://domain-name/San-Pham/Sua/productId
routes.MapRoute(
    name: "Edit Product",
    url: "San-Pham/Sua/{productId}",
    defaults: new { controller="Product", action="EditProduct", productId=@"\d{1,4}" }
);

//http://domain-name/San-Pham/productName/productId
routes.MapRoute(
    name: "Details Product",
    url: "San-Pham/{productName}/{productId}",
    defaults: new { controller = "Product", action="DetailsProduct", productName =
(string)null, productId = @"\d{1,4}" }
);
```

Bước 7: Chạy và quan sát kết quả



Kích vào link Sản phẩm



Kích vào link **Chi tiết**

Application name Sản Phẩm

Danh Sách Sản Phẩm

Mã Sản Phẩm	Tên Sản Phẩm	Lựa Chọn
1	Sản Phẩm Thứ 1	Chi Tiết Sửa
2	Sản Phẩm Thứ 2	Chi Tiết Sửa
3	Sản Phẩm Thứ 3	Chi Tiết Sửa
4	Sản Phẩm Thứ 4	Chi Tiết Sửa
5	Sản Phẩm Thứ 5	Chi Tiết Sửa
6	Sản Phẩm Thứ 6	Chi Tiết Sửa
7	Sản Phẩm Thứ 7	Chi Tiết Sửa
8	Sản Phẩm Thứ 8	Chi Tiết Sửa
9	Sản Phẩm Thứ 9	Chi Tiết Sửa

© 2023 - My ASP.NET Application

Kích vào link **Sửa**

Application name Sản Phẩm

Sửa thông tin chi tiết sản phẩm

Mã Sản Phẩm : 1

© 2023 - My ASP.NET Application

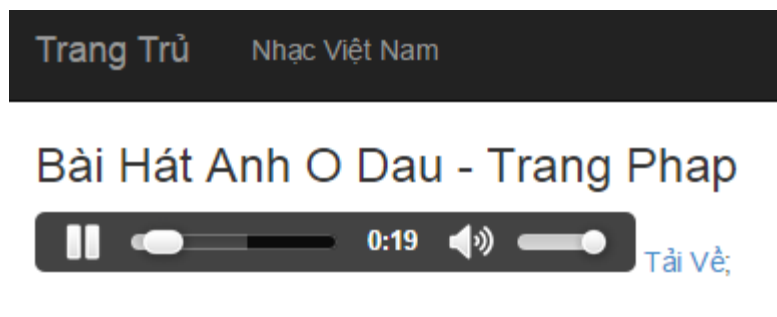
Bài 3: Bài tập tự làm

Tạo ứng dụng MVC thực hiện các yêu cầu:

- Xây dựng trang hiển thị một danh sách các bài hát tương ứng với mỗi bài hát là một nút tải về với giao diện dạng như sau:



- Trang chi tiết hiển thị thông tin bài hát như tên bài hát và cho chạy phát bài hát đó



© 2015 - My ASP.NET Application

- Khi kích chuột vào nút tải về thì cho phép trình duyệt mở hộp thoại save và tải bài hát về
- Tạo MapRoute cho Url thân thiện với Google