

Bài thực hành: Wingtip Toys

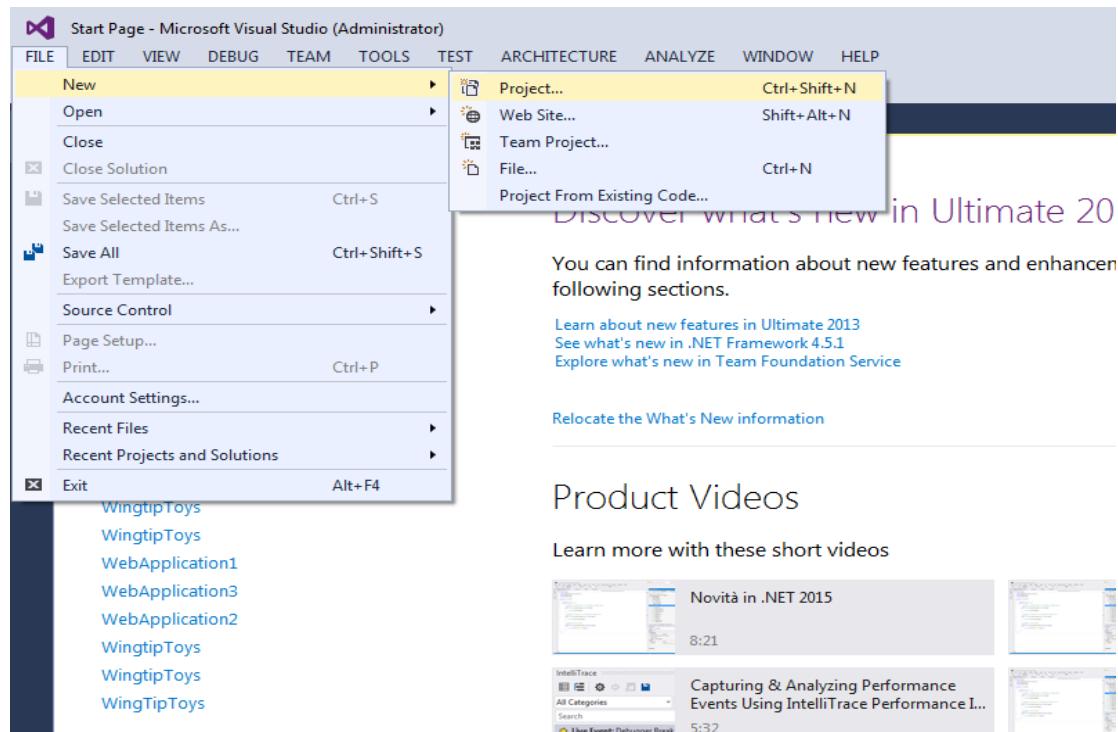
Người thực hiện: Nguyễn Viết Chinh

(Sinh viên ngành CNTT khóa 9 trường CĐ KTCNNT)

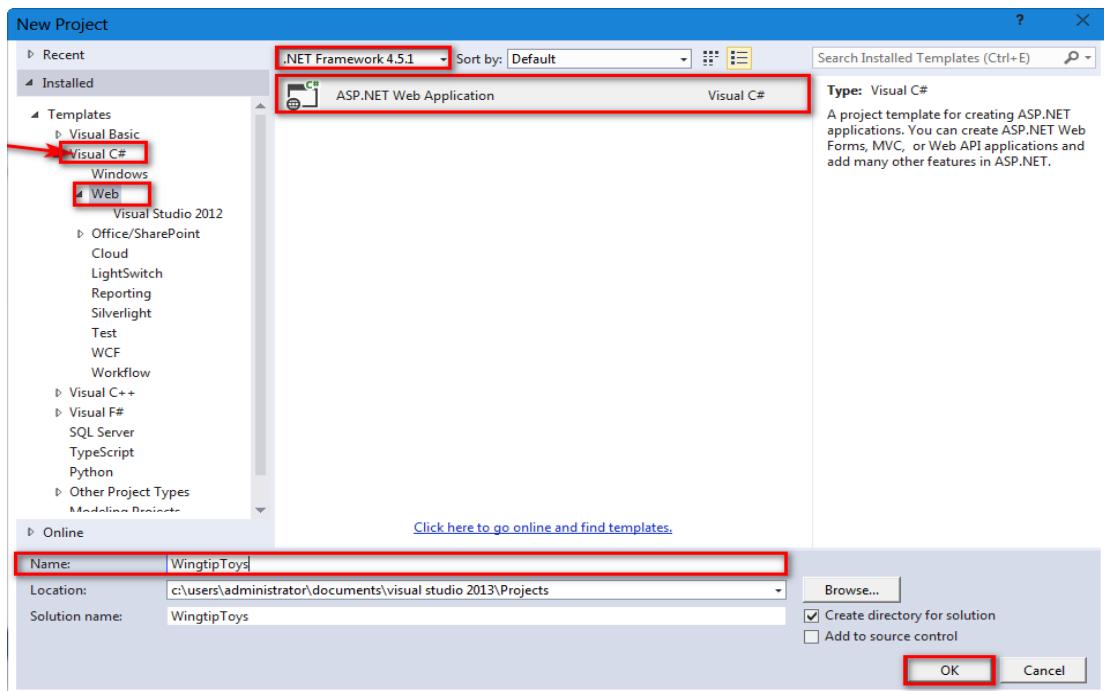
Các bước thực hiện:

Bước 1: Tạo dự án

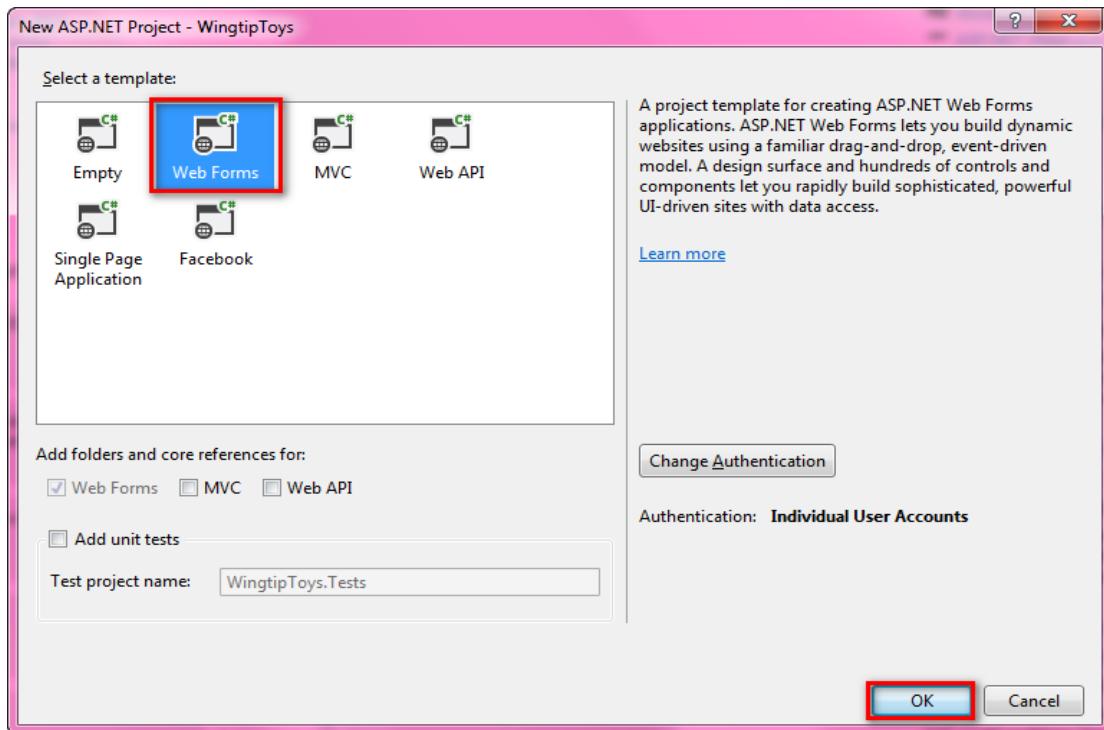
- Mở Visual Studio 2013 : File -> New -> Project



- Chọn Templates -> Visual C# -> Web -> ASP.NET Web Application -> Name : WingtipToys -> OK.

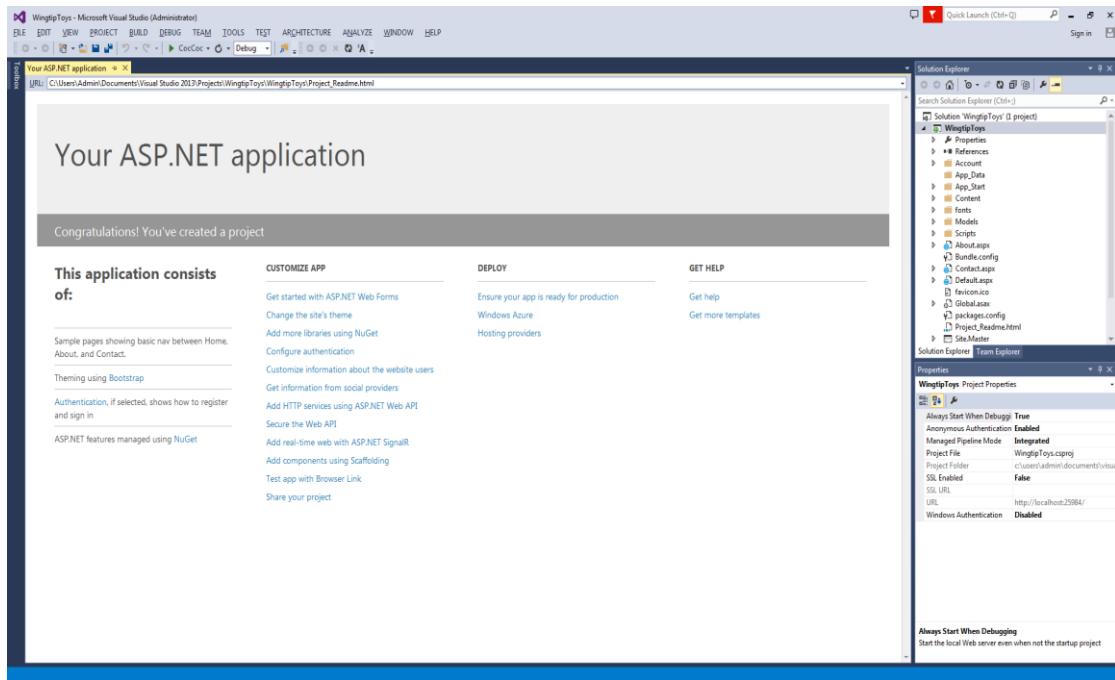


- Trong bảng New ASP.NET Project – WingtipToys chọn Web Forms -> Ok

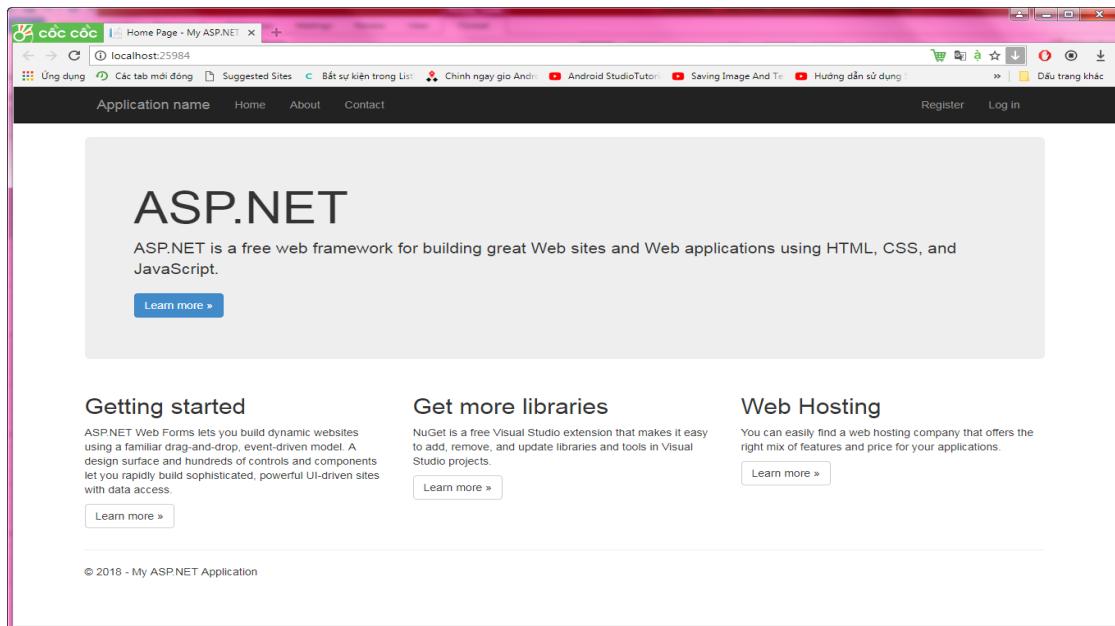


Đợi vài phút cho Visual Studio 2013 tạo dự án WingtipToys

Kết quả :

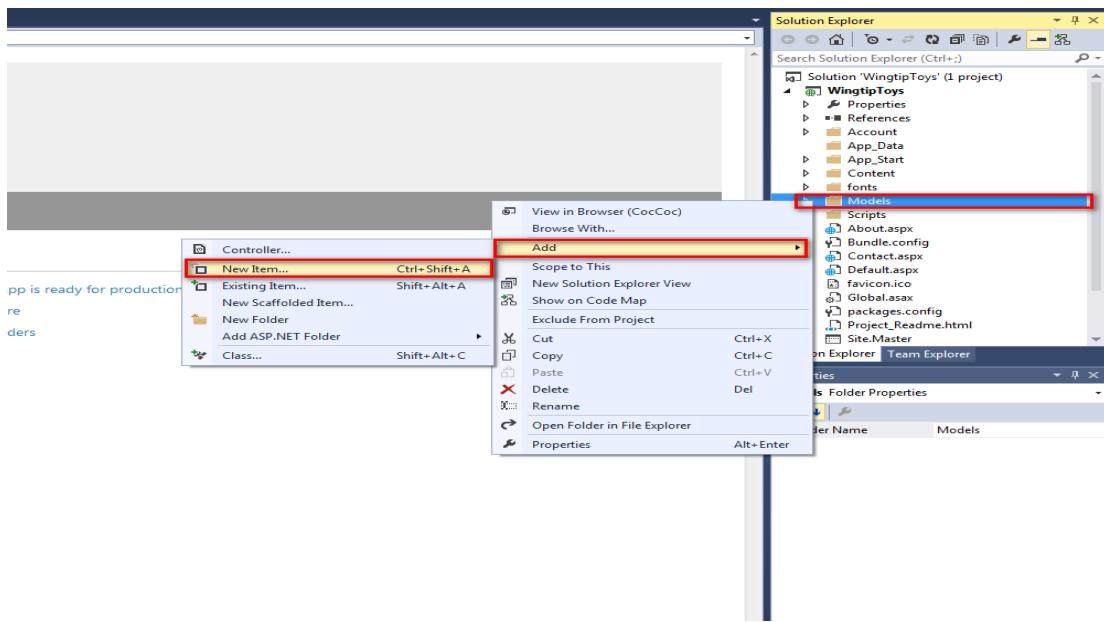


Nhấn F5 để chạy thử và kết quả :

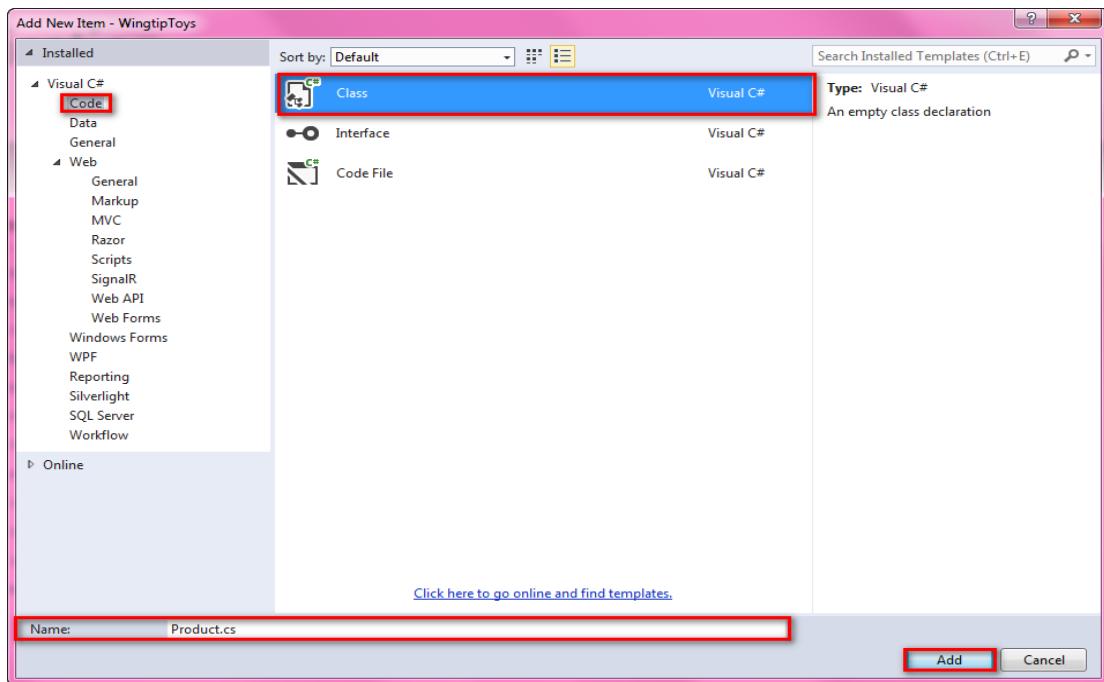


Đóng trình duyệt và Stop Debugging (Shift + F5)

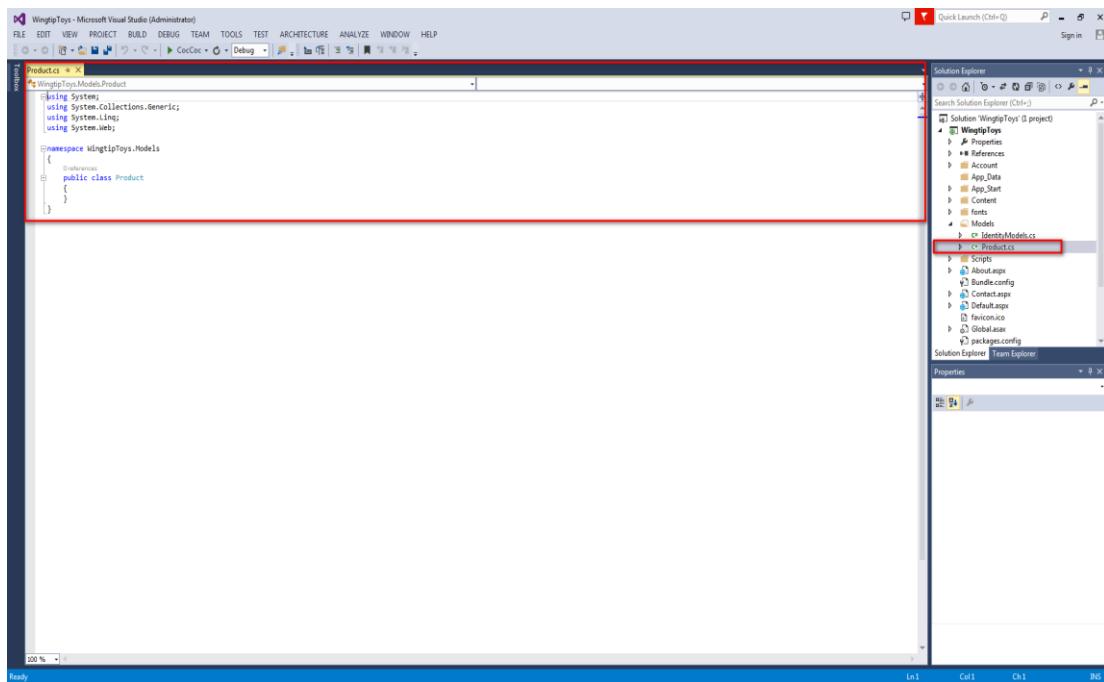
Bước 2: Tại Solution Explorer -> Click chuột phải thư mục Models -> Add -> New Item :



Chọn Visual C# -> Code -> Class -> Name : Product -> Add :



Một lớp mới được tạo ra như sau :



Tiếp theo xóa tất cả code mặc định trong lớp Product.cs thay bằng code sau (Ctr + A để copy hết code) :

```
using System.ComponentModel.DataAnnotations;

namespace WingtipToys.Models
{
    public class Product
    {
        [ScaffoldColumn(false)]
        public int ProductID { get; set; }
```

Tạo một lớp mới tên : Category.cs như các bước trên và thay thế code mặc định bằng đoạn code sau:

```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace WingtipToys.Models
{
    public class Category
    {
        [ScaffoldColumn(false)]
        public int CategoryID { get; set; }

        [Required, StringLength(100), Display(Name = "Name")]
        public string CategoryName { get; set; }

        [Display(Name = "Product Description")]
        public string Description { get; set; }

        public virtual ICollection<Product> Products { get; set; }
    }
}
```

Tạo thêm một lớp tên : ProductContext.cs và cũng thay code mặc định bằng đoạn code sau:

```
using System.Data.Entity;
namespace WingtipToys.Models
{
    public class ProductContext : DbContext
    {
        public ProductContext()
            : base("WingtipToys")
        {
        }
        public DbSet<Category> Categories { get; set; }
        public DbSet<Product> Products { get; set; }
    }
}
```

Tạo một lớp khác trong folder Models và đặt tên : ProductDatabaseInitializer.cs. Làm như trên thay code mặc định bằng đoạn code sau :

```
using System.Collections.Generic;
using System.Data.Entity;

namespace WingtipToys.Models
{
    public class ProductDatabaseInitializer : DropCreateDatabaseIfModelChanges<ProductContext>
    {
        protected override void Seed(ProductContext context)
        {
            GetCategories().ForEach(c => context.Categories.Add(c));
            GetProducts().ForEach(p => context.Products.Add(p));
        }

        private static List<Category> GetCategories()
        {
            var categories = new List<Category> {
                new Category
                {
                    CategoryID = 1,
                    CategoryName = "Cars"
                },
                new Category
                {
                    CategoryID = 2,
                    CategoryName = "Planes"
                },
                new Category
                {
                    CategoryID = 3,
                    CategoryName = "Trucks"
                },
                new Category
                {
                    CategoryID = 4,
                    CategoryName = "Boats"
                },
                new Category
                {
                    CategoryID = 5,
                    CategoryName = "Rockets"
                },
            };

            return categories;
        }
    }
}
```

```
private static List<Product> GetProducts()
{
    var products = new List<Product> {
        new Product
        {
            ProductID = 1,
            ProductName = "Convertible Car",
            Description = "This convertible car is fast! The engine is
powered by a neutrino based battery (not included)." +
                           "Power it up and let it go!",
            ImagePath="carconvert.png",
            UnitPrice = 22.50,
            CategoryID = 1
        },
        new Product
        {
            ProductID = 2,
            ProductName = "Old-time Car",
            Description = "There's nothing old about this toy car, except
it's looks. Compatible with other old toy cars.",
            ImagePath="carearly.png",
            UnitPrice = 15.95,
            CategoryID = 1
        },
        new Product
        {
            ProductID = 3,
            ProductName = "Fast Car",
            Description = "Yes this car is fast, but it also floats in
water.",
            ImagePath="carfast.png",
            UnitPrice = 32.99,
            CategoryID = 1
        },
        new Product
        {
            ProductID = 4,
            ProductName = "Super Fast Car",
            Description = "Use this super fast car to entertain guests.
Lights and doors work!",
            ImagePath="carfaster.png",
            UnitPrice = 8.95,
            CategoryID = 1
        },
        new Product
        {
            ProductID = 5,
            ProductName = "Old Style Racer",
            Description = "This old style racer can fly (with user
assistance). Gravity controls flight duration." +
                           "No batteries required.",
            ImagePath="carracer.png",
            UnitPrice = 34.95,
            CategoryID = 1
        },
        new Product
        {
            ProductID = 6,
            ProductName = "Ace Plane",
            Description = "Authentic airplane toy. Features realistic color
and details.",
            ImagePath="planeace.png",
            UnitPrice = 95.00,
            CategoryID = 2
        },
    };
}
```

```

new Product
{
    ProductID = 7,
    ProductName = "Glider",
    Description = "This fun glider is made from real balsa wood.
Some assembly required.",
    ImagePath="planeglider.png",
    UnitPrice = 4.95,
    CategoryID = 2
},
new Product
{
    ProductID = 8,
    ProductName = "Paper Plane",
    Description = "This paper plane is like no other paper plane.
Some folding required.",
    ImagePath="planepaper.png",
    UnitPrice = 2.95,
    CategoryID = 2
},
new Product
{
    ProductID = 9,
    ProductName = "Propeller Plane",
    Description = "Rubber band powered plane features two wheels.",
    ImagePath="planeprop.png",
    UnitPrice = 32.95,
    CategoryID = 2
},
new Product
{
    ProductID = 10,
    ProductName = "Early Truck",
    Description = "This toy truck has a real gas powered engine.
Requires regular tune ups.",
    ImagePath="truckearly.png",
    UnitPrice = 15.00,
    CategoryID = 3
},
new Product
{
    ProductID = 11,
    ProductName = "Fire Truck",
    Description = "You will have endless fun with this one quarter
sized fire truck.",
    ImagePath="truckfire.png",
    UnitPrice = 26.00,
    CategoryID = 3
},
new Product
{
    ProductID = 12,
    ProductName = "Big Truck",
    Description = "This fun toy truck can be used to tow other
trucks that are not as big.",
    ImagePath="truckbig.png",
    UnitPrice = 29.00,
    CategoryID = 3
},

```

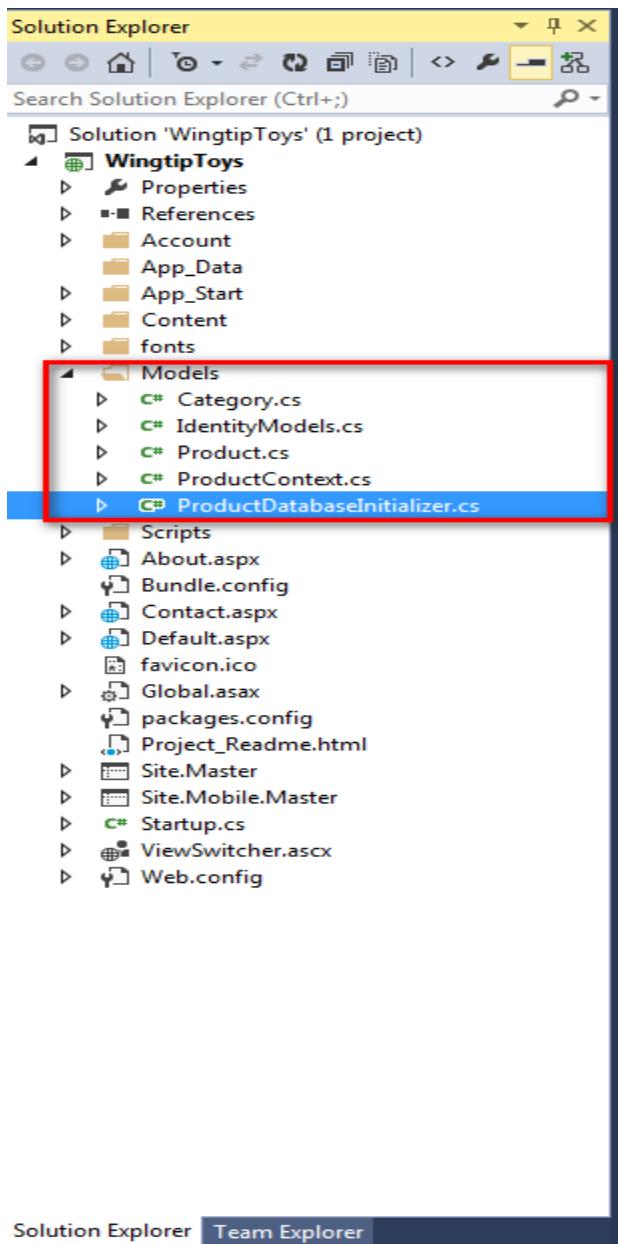
```

        new Product
    {
        ProductID = 13,
        ProductName = "Big Ship",
        Description = "Is it a boat or a ship. Let this floating vehicle
decide by using its " +
                        "artificially intelligent computer brain!",
        ImagePath="boatbig.png",
        UnitPrice = 95.00,
        CategoryID = 4
    },
    new Product
    {
        ProductID = 14,
        ProductName = "Paper Boat",
        Description = "Floating fun for all! This toy boat can be
assembled in seconds. Floats for minutes!" +
                        "Some folding required.",
        ImagePath="boatpaper.png",
        UnitPrice = 4.95,
        CategoryID = 4
    },
    new Product
    {
        ProductID = 15,
        ProductName = "Sail Boat",
        Description = "Put this fun toy sail boat in the water and let
it go!",
        ImagePath="boatsail.png",
        UnitPrice = 42.95,
        CategoryID = 4
    },
    new Product
    {
        ProductID = 16,
        ProductName = "Rocket",
        Description = "This fun rocket will travel up to a height of 200
feet.",
        ImagePath="rocket.png",
        UnitPrice = 122.95,
        CategoryID = 5
    }
};

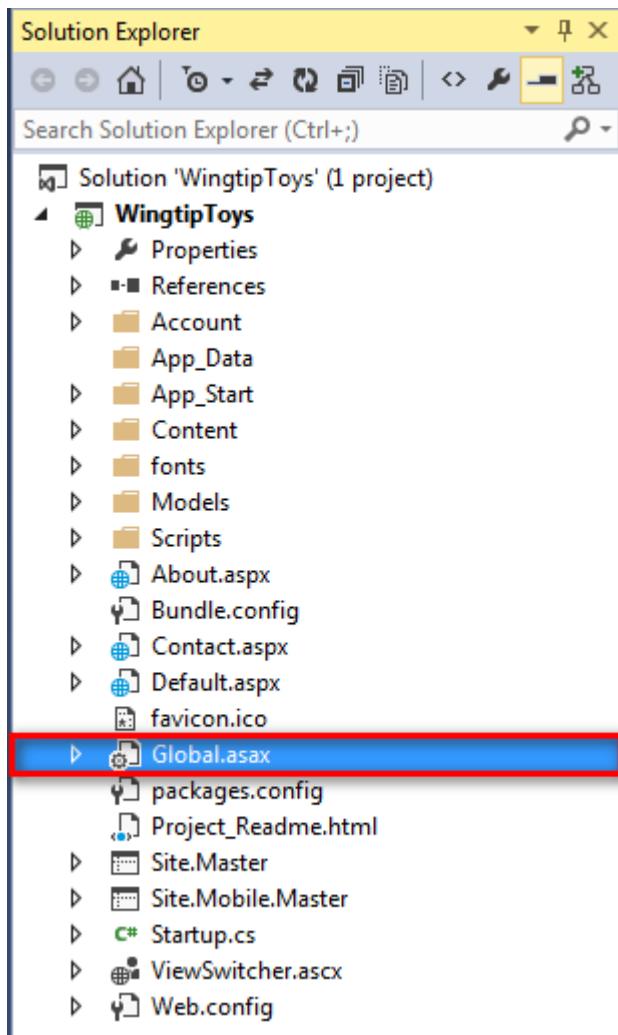
        return products;
    }
}

```

Lúc này folder Models ngoài 1 class mặc định sẽ có thêm 4 class chúng ta vừa tạo :



Bước 3: Trong Solution Explorer mở file Global.asax :



Thêm những đoạn code được bôi màu vàng vào file Global.asax :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Optimization;
using System.Web.Routing;
using System.Web.Security;
using System.Web.SessionState;
using System.Data.Entity;
using WingtipToys.Models;

namespace WingtipToys
{
    public class Global : HttpApplication
    {
        void Application_Start(object sender, EventArgs e)
        {
            // Code that runs on application startup
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);

            // Initialize the product database.
            Database.SetInitializer(new ProductDatabaseInitializer());
        }
    }
}
```

Tiếp theo trong Solution Explorer mở file Web.config. Thêm chuỗi kết nối được bôi vàng vào giữa <connectionStrings> ... </connectionStrings> như sau :

Chuỗi kết nối:

```
<add name="WingtipToys" connectionString="Data  
Source=(LocalDB)\v11.0;AttachDbFilename=|DataDirectory|\wingtiptoys.mdf;Integrated  
Security=True" providerName="System.Data.SqlClient" />
```

Sau khi thêm vào :

```
<connectionStrings>  
  <add name="DefaultConnection" connectionString="Data  
Source=(LocalDb)\v11.0;AttachDbFilename=|DataDirectory|\aspnet-WingtipToys-  
20131119102907.mdf;Initial Catalog=aspnet-WingtipToys-20131119102907;Integrated  
Security=True"  
    providerName="System.Data.SqlClient" />  
  <add name="WingtipToys"  
    connectionString="Data  
Source=(LocalDB)\v11.0;AttachDbFilename=|DataDirectory|\wingtiptoys.mdf;Integra  
ted Security=True" providerName="System.Data.SqlClient" />  
</connectionStrings>
```

Bước 4: Mở trang Default.aspx thay đổi Title như sau :

```
<%@ Page Title="Welcome" Language="C#" MasterPageFile="~/Site.Master"  
AutoEventWireup="true" CodeBehind="Default.aspx.cs"  
Inherits="WingtipToys._Default" %>
```

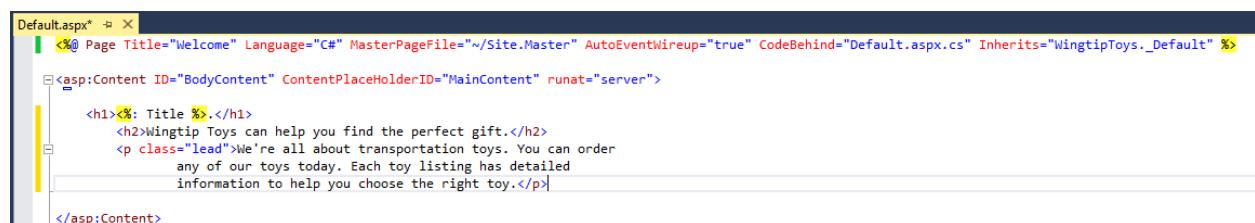
Tại trang Default.aspx xóa hết nội dung giữa :

```
<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">  
.....  
</asp:Content>
```

và thay thế bằng đoạn code :

```
<h1><%: Title %></h1>  
<h2>Wingtip Toys can help you find the perfect gift.</h2>  
<p class="lead">We're all about transportation toys. You can order  
any of our toys today. Each toy listing has detailed  
information to help you choose the right toy.</p>
```

Kết quả trang Default.aspx trông như thế này :



Bước 5: Trong Solution Explorer mở Site.Master. Cập nhật trang master bằng cách sửa đổi và thêm vào những đoạn code được bôi vàng như sau:

Thay thế : “My ASP.NET Application” -> “Wingtip Toys”:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs" Inherits="WingtipToys.SiteMaster" %>

<!DOCTYPE html>

<html lang="en">
    <head runat="server">
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title><%: Page.Title %> - Wingtip Toys</title>
```

và:

```
<div class="container body-content">
    <asp:ContentPlaceHolder ID="MainContent" runat="server">
    </asp:ContentPlaceHolder>
    <hr />
    <footer>
        <p>&copy; <%: DateTime.Now.Year %> - Wingtip Toys</p>
    </footer>
</div>
```

Thay thế : “Application name” -> “Wingtip Toys” :

```
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" runat="server" href="/">Wingtip Toys</a>
        </div>
```

Thêm đoạn code sau trước “`<div class="container body-content">`” :

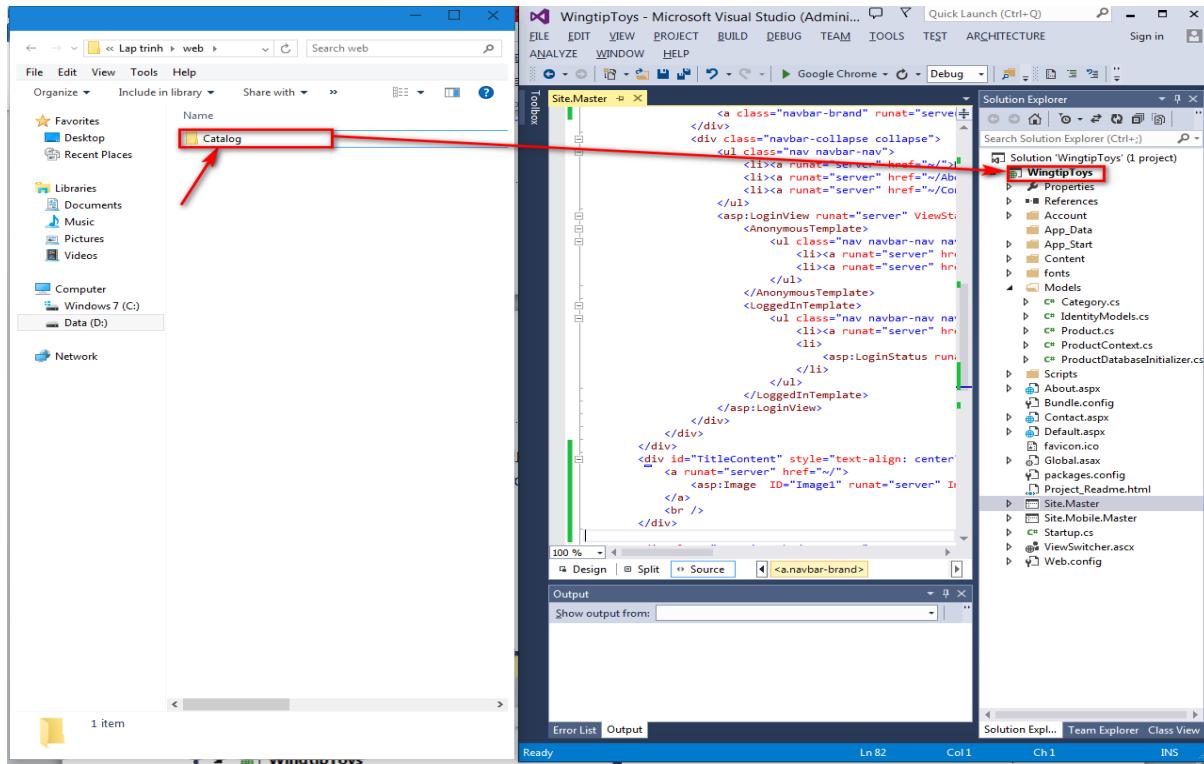
Code :

```
<div id="TitleContent" style="text-align: center">
    <a runat="server" href="/">
        <asp:Image ID="Image1" runat="server" ImageUrl="~/Images/logo.jpg" BorderStyle="None" />
    </a>
    <br />
</div>
```

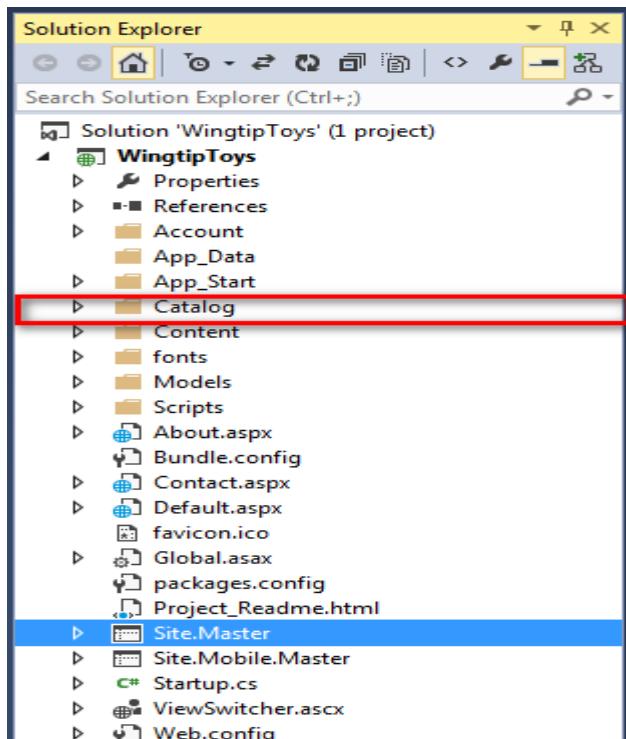
Kết quả :

```
        <!--
    </LoggedInTemplate>
    </asp:LoginView>
</div>
</div>
<div id="TitleContent" style="text-align: center">
    <a runat="server" href="/">
        <asp:Image ID="Image1" runat="server" ImageUrl="~/Images/logo.jpg" BorderStyle="None" />
    </a>
    <br />
</div>
<div class="container body-content">
```

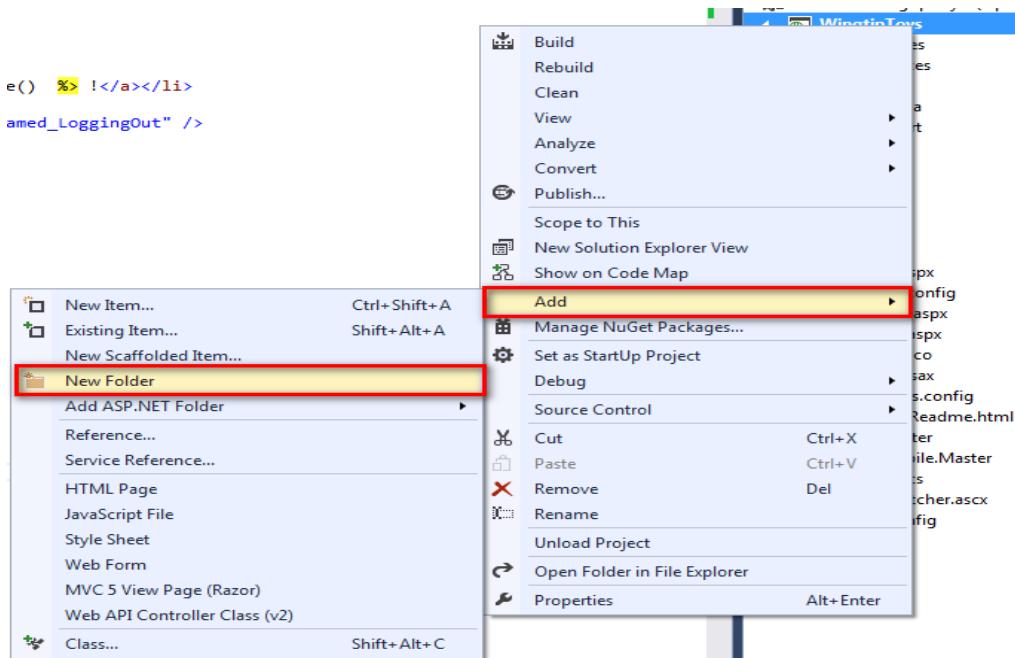
Bước 6: Truy cập : <https://drive.google.com/open?id=1Q32SZJ6AT9ZhzQvAaobOHnaMrte0z6Cu> và tải tập tin “Catalog.rar” sau đó giải nén. Kéo thư mục “Catalog” sau khi giải nén vào dự án Wingtip Toys như sau :



Kết quả :

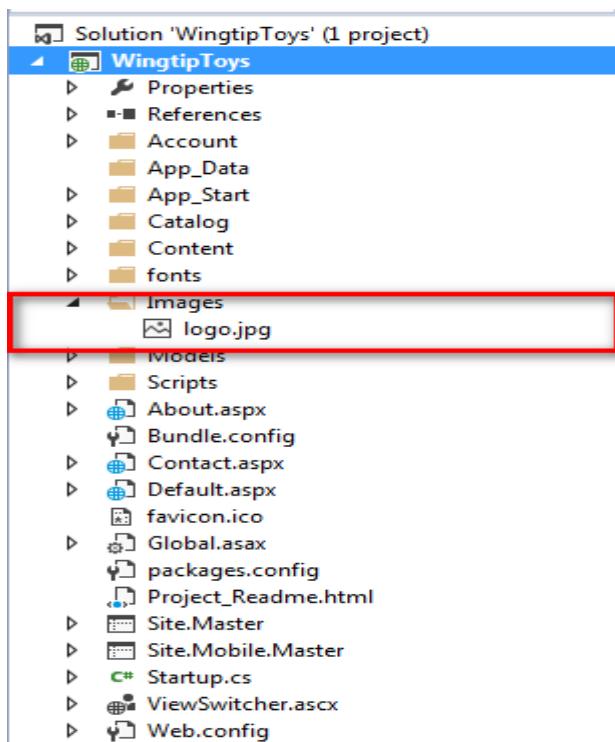


Tiếp theo tạo một folder mới tên “**Images**” bằng cách chọn chuột phải vào dự án “Wingtip Toys”-> Add-> New Folder.



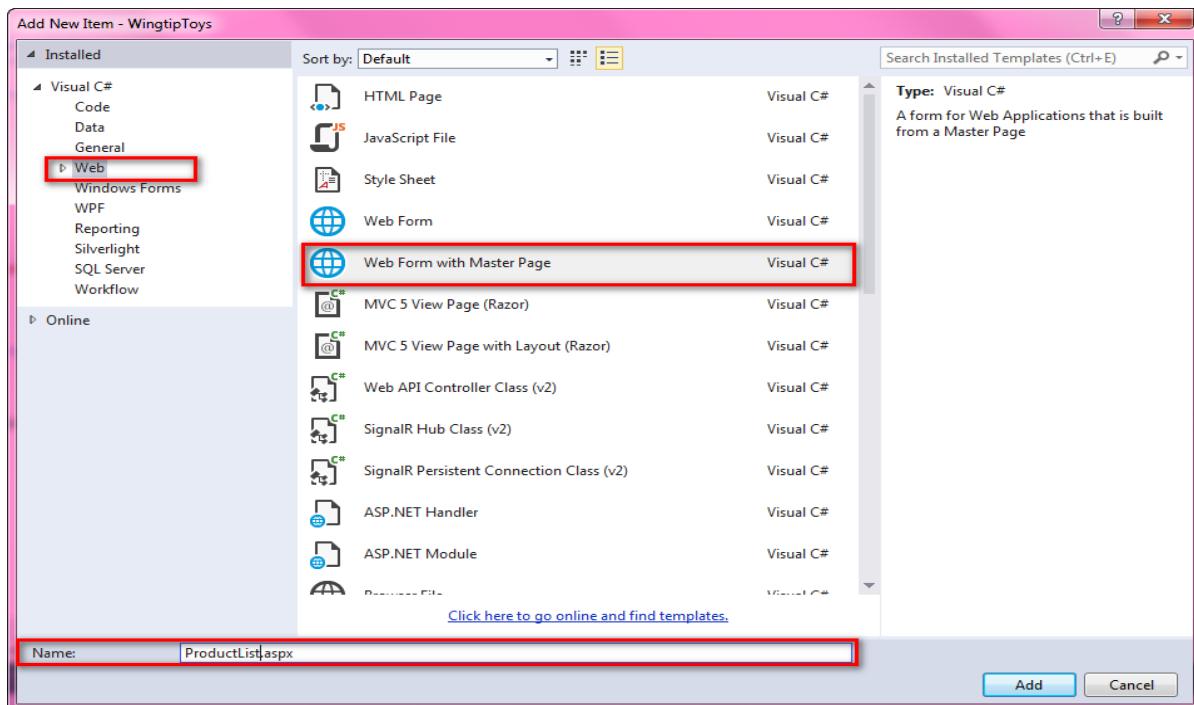
Truy cập : https://drive.google.com/open?id=1Nb_vd3IpXuiJW1vyO-1LRf77YKmfuo-h . Tải về file “logo.jpg” sau đó kéo bỏ vào thư mục Images vừa mới tạo.

Kết quả:

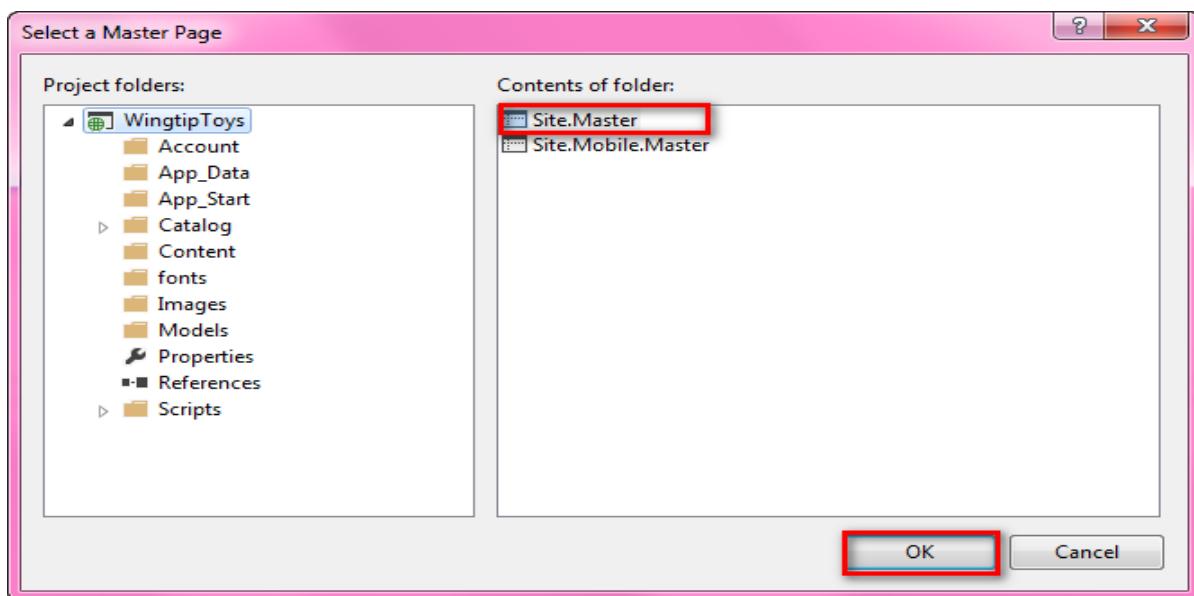


Bước 7: Tại Solution Explorer -> chuột phải WingtipToys -> Add -> New Item :

Chọn : Web -> Web Form with Master Page và đặt tên : “ProductList.aspx” -> Add:



Tiếp theo chọn : Site.Master -> OK :



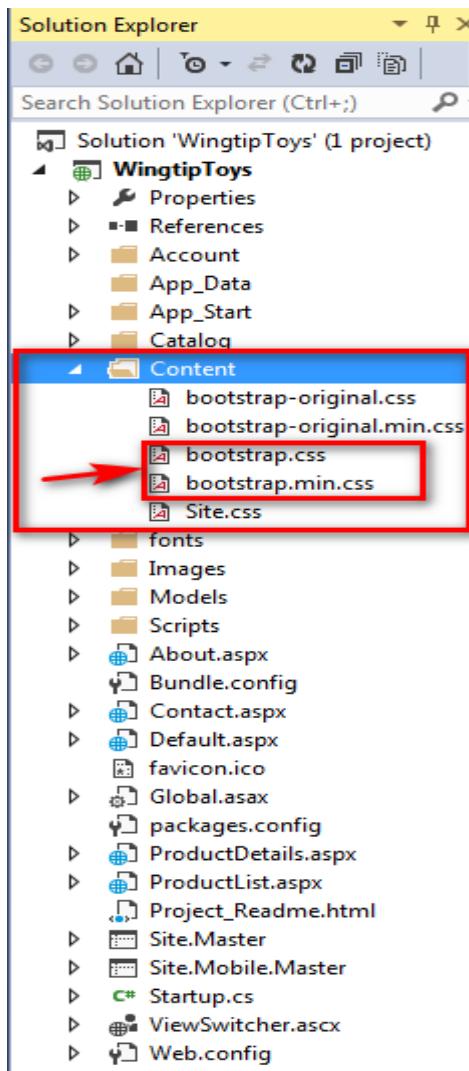
Làm tương tự các bước trên tạo ra một “Web Form with Master Page” và đặt tên “ProductDetails.aspx”.

Bước 8:

- Trong Solution Explorer, mở thư mục “Content”.
- Chuột phải vào file “bootstrap.css” và đổi tên nó thành “bootstrap-original.css”
- Đổi tên “bootstrap.min.css” thành “bootstrap-original.min.css”

-Truy cập : https://drive.google.com/open?id=1Xd3kTChprF2-Y5s2rnjHWUMWK7Z5_maJ . Tải về và giải nén sau đó copy 2 file vừa giải nén bỏ vào thư mục “ Content”

Kết quả :



Bước 9: Trong Solution Explorer mở file Site.Master. Tìm và thêm đoạn code được bôi vàng vào như hình :

```
<ul class="nav navbar-nav">
    <li><a runat="server" href="/">Home</a></li>
    <li><a runat="server" href="/About">About</a></li>
    <li><a runat="server" href="/Contact">Contact</a></li>
    <li><a runat="server" href="/ProductList">Products</a></li>
</ul>
```

- Tiếp tục thêm đoạn code được bôi vàng sau vào Site.Master như hình :

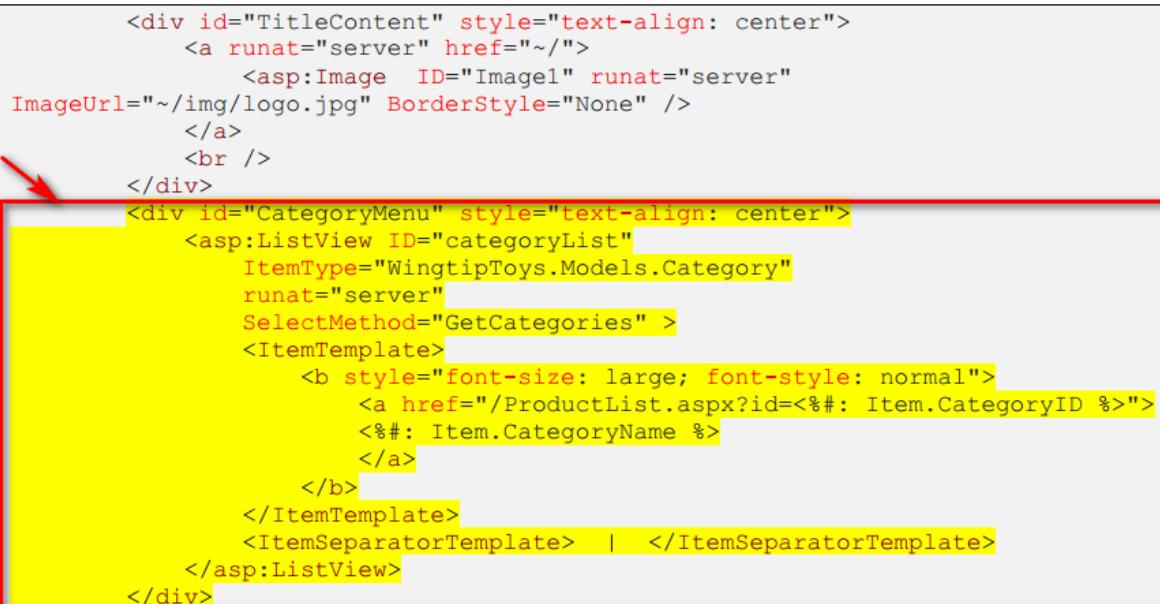
Code :

```

<div id="CategoryMenu" style="text-align: center">
    <asp:ListView ID="categoryList"
        ItemType="WingtipToys.Models.Category"
        runat="server"
        SelectMethod="GetCategories" >
        <ItemTemplate>
            <b style="font-size: large; font-style: normal">
                <a href="/ProductList.aspx?id=<%#: Item.CategoryID %>">
                    <%#: Item.CategoryName %>
                </a>
            </b>
        </ItemTemplate>
        <ItemSeparatorTemplate> | </ItemSeparatorTemplate>
    </asp:ListView>
</div>

```

Kết quả :



```

<div id="TitleContent" style="text-align: center">
    <a runat="server" href="~/">
        <asp:Image ID="Image1" runat="server"
        ImageUrl="~/img/logo.jpg" BorderStyle="None" />
    </a>
    <br />
</div>
<div id="CategoryMenu" style="text-align: center">
    <asp:ListView ID="categoryList"
        ItemType="WingtipToys.Models.Category"
        runat="server"
        SelectMethod="GetCategories" >
        <ItemTemplate>
            <b style="font-size: large; font-style: normal">
                <a href="/ProductList.aspx?id=<%#: Item.CategoryID %>">
                    <%#: Item.CategoryName %>
                </a>
            </b>
        </ItemTemplate>
        <ItemSeparatorTemplate> | </ItemSeparatorTemplate>
    </asp:ListView>
</div>

```

-Chuột phải vào Site.Master -> View Code . Thêm đoạn code được bôi vàng vào hình như sau :

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Linq;
using WingtipToys.Models;

```

-Thêm đoạn code sau vào sau sự kiện “Page Load” :

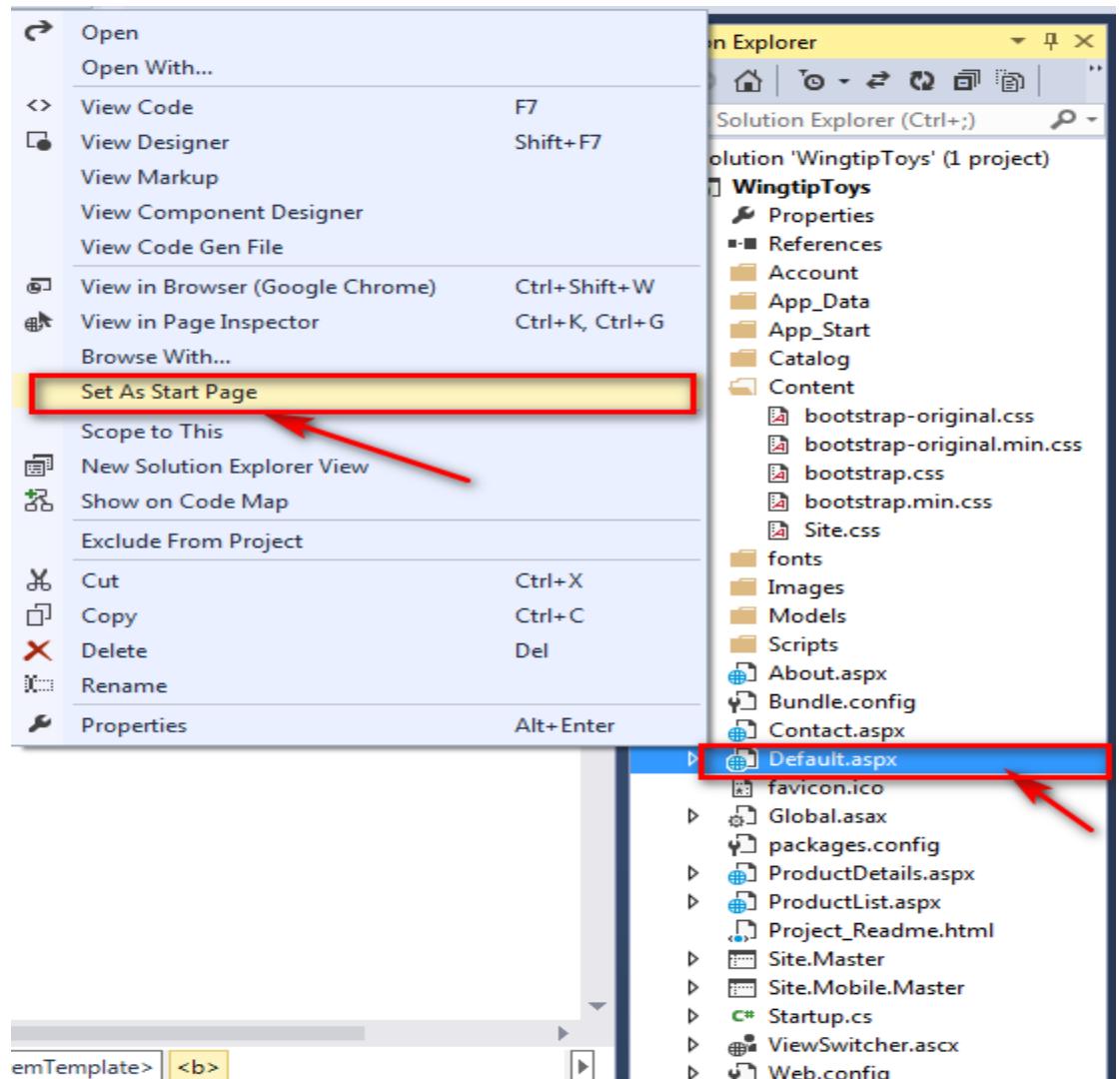
```

protected void Page_Load(object sender, EventArgs e)
{
}

public IQueryable<Category> GetCategories()
{
    var _db = new WingtipToys.Models.ProductContext();
    IQueryable<Category> query = _db.Categories;
    return query;
}

```

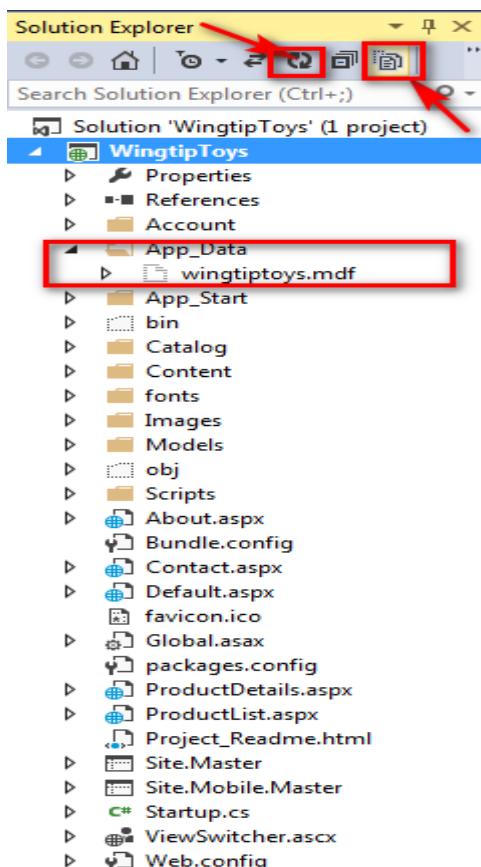
Trong Solution Explorer chuột phải Default.aspx -> Set As Start Page :



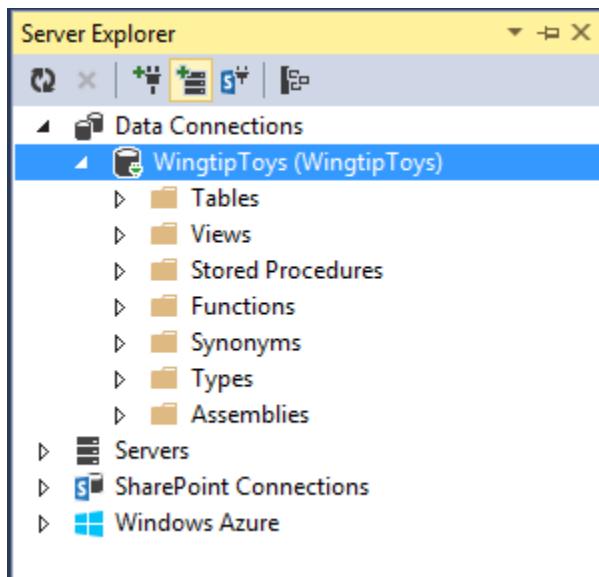
Nhấn F5 chạy thử và xem kết quả :

The screenshot shows a web browser window with the title "Welcome - Wingtip Toys". The address bar displays "localhost:1352". The page header includes a "Person 1" icon, a star icon, a refresh button, and a close button. The main navigation menu at the top has links for "Wingtip Toys", "Home", "About", "Contact", "Products", "Register", and "Log In". Below the menu, the "Wingtip Toys" logo is displayed in red script. A horizontal menu bar below the logo includes "Cars | Planes | Trucks | Boats | Rockets". The main content area features a "Welcome" message and a sub-message "Wingtip Toys can help you find the perfect gift.". A descriptive paragraph follows, stating: "We're all about transportation toys. You can order any of our toys today. Each toy listing has detailed information to help you choose the right toy." At the bottom of the page, a copyright notice reads "© 2018 - Wingtip Toys".

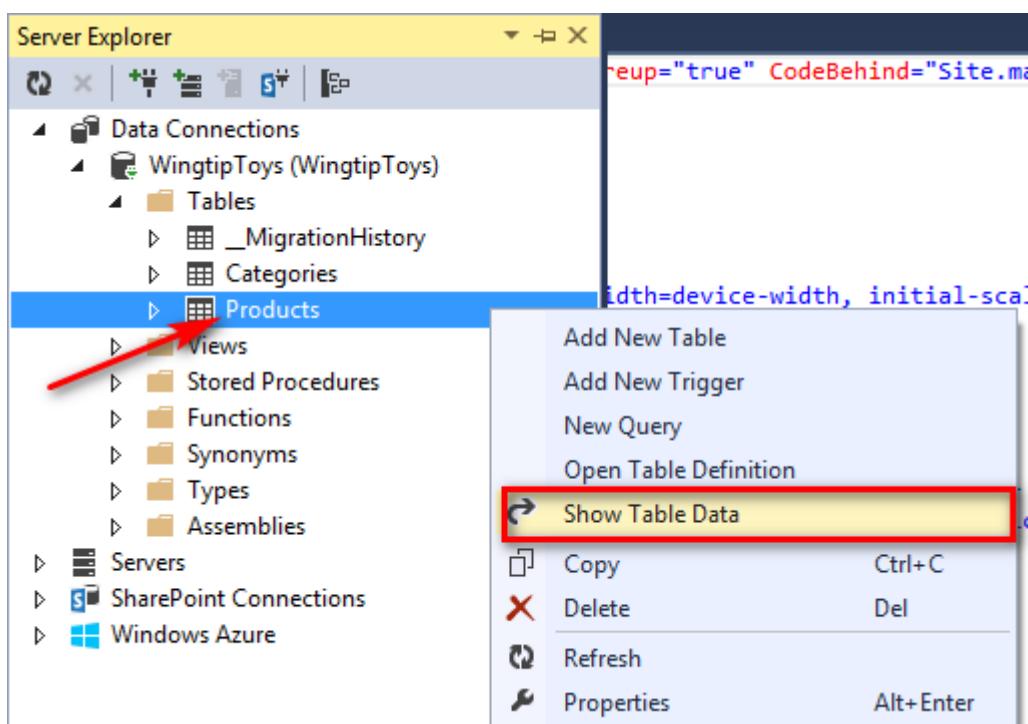
Trong Solution Explorer, chọn Refresh -> Show All Files. Khi đó trong thư mục App_Data xuất hiện tập tin wingtiptoys.mdf như thế này :



Tiếp tục chuột phải vào tập tin wingtiptoys.mdf -> Open. Một Server Explorer sẽ xuất hiện như sau :



Mở rộng thư mục Tables -> chuột phải vào Products và chọn Show Table Data :



Kết quả ta sẽ thấy được tất cả thông tin chi tiết của sản phẩm như sau :

dbo.Products [Data] X

Max Rows: 1000

	ProductID	ProductName	Description	ImagePath	UnitPrice	CategoryID
▶	1	Convertible Car	This convertible...	carconvert.png	22.5	1
	2	Old-time Car	There's nothing...	carearly.png	15.95	1
	3	Fast Car	Yes this car is fa...	carfast.png	32.99	1
	4	Super Fast Car	Use this super f...	carfaster.png	8.95	1
	5	Old Style Racer	This old style ra...	carracer.png	34.95	1
	6	Ace Plane	Authentic airpl...	planeace.png	95	2
	7	Glider	This fun glider i...	planeglider.png	4.95	2
	8	Paper Plane	This paper plan...	planepaper.png	2.95	2
	9	Propeller Plane	Rubber band p...	planeprop.png	32.95	2
	10	Early Truck	This toy truck h...	truckearly.png	15	3
	11	Fire Truck	You will have e...	truckfire.png	26	3
	12	Big Truck	This fun toy tru...	truckbig.png	29	3
	13	Big Ship	Is it a boat or a ...	boatbig.png	95	4
	14	Paper Boat	Floating fun for...	boatpaper.png	4.95	4
	15	Sail Boat	Put this fun toy...	boatsail.png	42.95	4
	16	Rocket	This fun rocket ...	rocket.png	122.95	5
*	NULL	NULL	NULL	NULL	NULL	NULL

Đóng bảng Products lại, mở lại Server Explorer tiếp tục chuột phải vào Products -> chọn Open Table Definition ta sẽ thấy được chi tiết thiết kế của bảng Products như sau :

dbo.Products [Design] X

Update | Script File: dbo.Products.sql

	Name	Data Type	Allow Nulls	Default	
PK	ProductID	int	<input type="checkbox"/>		
	ProductName	nvarchar(100)	<input type="checkbox"/>		
	Description	nvarchar(MAX)	<input type="checkbox"/>		
	ImagePath	nvarchar(MAX)	<input checked="" type="checkbox"/>		
	UnitPrice	float	<input checked="" type="checkbox"/>		
	CategoryID	int	<input checked="" type="checkbox"/>		

Keys (1)
PK_dbo.Products (Primary Key, Clustered: ProductID)
Check Constraints (0)

Indexes (1)
IX_CategoryID (CategoryID)

Foreign Keys (1)
FK_dbo.Products_dbo.Categories_CategoryID (CategoryID)
Triggers (0)

Design T-SQL

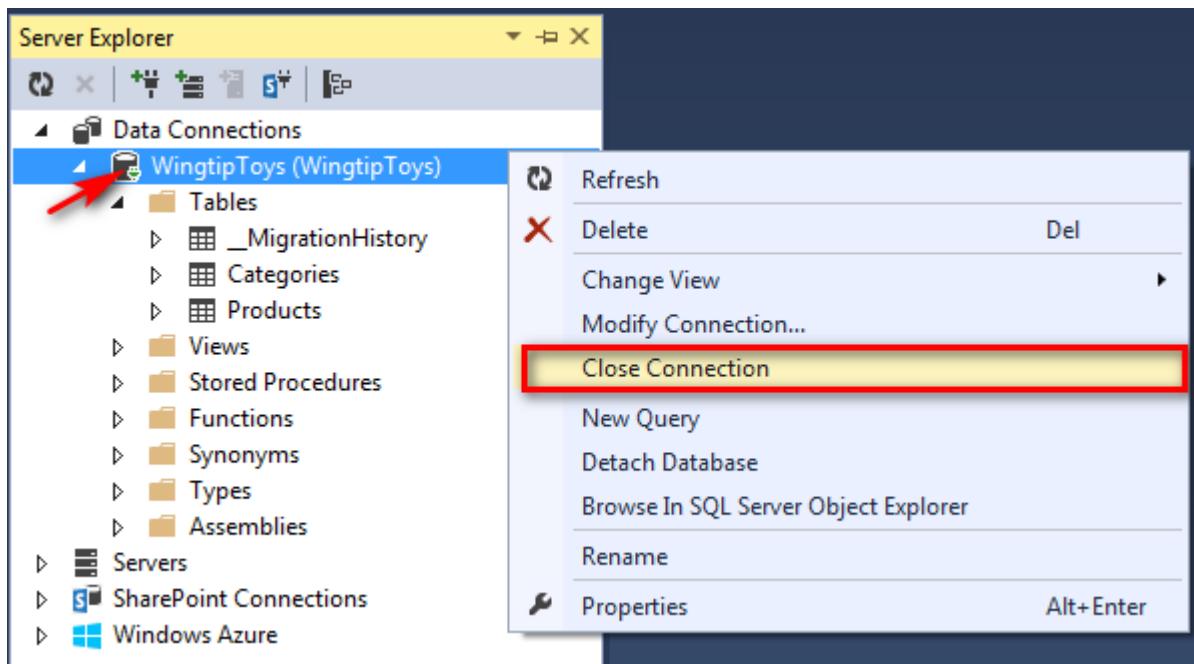
```

CREATE TABLE [dbo].[Products] (
    [ProductID] INT IDENTITY (1, 1) NOT NULL,
    [ProductName] NVARCHAR (100) NOT NULL,
    [Description] NVARCHAR (MAX) NOT NULL,
    [ImagePath] NVARCHAR (MAX) NULL,
    [UnitPrice] FLOAT (53) NULL,
    [CategoryID] INT NULL,
    CONSTRAINT [PK_dbo.Products] PRIMARY KEY CLUSTERED ([ProductID] ASC),
    CONSTRAINT [FK_dbo.Products_dbo.Categories_CategoryID] FOREIGN KEY ([CategoryID]) REFERENCES [dbo].[Categories] ([CategoryID])
);

```

100 %

Đóng cửa sổ dbo.Products[Design], mở lại Server Explorer -> chuột phải vào database “Wingtip Toys” -> Close Connection:



Bước 10: Trong Solution Explorer, mở trang ProductList.aspx xóa tất cả code mặc định và thay bằng đoạn code sau :

```
<%@ Page Title="Products" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true"
CodeBehind="ProductList.aspx.cs" Inherits="WingtipToys.ProductList" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
<section>
<div>
<hgroup>
<h2><%: Page.Title %></h2>
</hgroup>
<asp:ListView ID="productList" runat="server"
DataKeyNames="ProductID" GroupItemCount="4"
ItemType="WingtipToys.Models.Product" SelectMethod="GetProducts">
<EmptyDataTemplate>
<table >
<tr>
<td>No data was returned.</td>
</tr>
</table>
</EmptyDataTemplate>
<EmptyItemTemplate>
<td/>
</EmptyItemTemplate>
<GroupTemplate>
<tr id="itemPlaceholderContainer" runat="server">
<td id="itemPlaceholder" runat="server"></td>
</tr>
</GroupTemplate>
<ItemTemplate>
<td runat="server">
<table>
<tr>
<td>
<a href="ProductDetails.aspx?productID=<%#:Item.ProductID%>">
</a>
</td>
</tr>
<tr>
<td>
<a href="ProductDetails.aspx?productID=<%#:Item.ProductID%>">
<span>
<%#:Item.ProductName%>
</span>
</a>
<br />
<span>
<b>Price: </b><%:String.Format("{0:c}", Item.UnitPrice)%>
</span>
<br />
</td>
</tr>
<tr>
<td> </td>
</tr>
</table>
<p>
</td>
</ItemTemplate>
<LayoutTemplate>
<table style="width:100%;">
<tbody>
<tr>
<td>
<table id="groupPlaceholderContainer"
runat="server" style="width:100%">
<tr id="groupPlaceholder"></tr>
</table>
</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
</LayoutTemplate>
</asp:ListView>
</div>
</section>
</asp:Content>
```

Tiếp tục chuột phải ProductList.aspx -> View Code . Xóa tất cả code mặc định và thay bằng đoạn code sau :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using WingtipToys.Models;
using System.Web.ModelBinding;
using System.Web.Routing;

namespace WingtipToys
{
    public partial class ProductList : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        public IQueryable<Product> GetProducts([QueryString("id")] int?
categoryID,[RouteData] string categoryName)
        {
            var _db = new WingtipToys.Models.ProductContext();
            IQueryable<Product> query = _db.Products;
            if (categoryID.HasValue && categoryID > 0)
            {
                query = query.Where(p => p.CategoryID == categoryID);
            }
            if (!String.IsNullOrEmpty(categoryName))
            {
                query = query.Where(p =>
String.Compare(p.Category.CategoryName,
categoryName) == 0);
            }
            return query;
        }
    }
}
```

Chạy thử ứng dụng (F5) -> Chọn thử danh mục Car sẽ có kết quả :

The screenshot shows the Wingtip Toys website's Products page. At the top, there is a navigation bar with links for Home, About, Contact, Products, Register, and Log In. The main title "Wingtip Toys" is centered above a menu bar with links for Cars, Planes, Trucks, Boats, and Rockets. Below the menu, the heading "Products" is displayed. Five car items are listed in a grid:

- Convertible Car (Price: \$22.50)
- Old-time Car (Price: \$15.95)
- Fast Car (Price: \$32.99)
- Super Fast Car (Price: \$8.95)
- Old Style Racer (Price: \$34.95)

Each item has a small thumbnail image and its name and price below it.

Tiếp tục trong Solution Explorer -> mở trang ProductDetails.aspx thay thế code mặc định bằng đoạn code sau :

```
<%@ Page Title="Product Details" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true"
CodeBehind="ProductDetails.aspx.cs" Inherits="WingtipToys.ProductDetails"
%>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <asp:FormView ID="productDetail" runat="server"
    ItemType="WingtipToys.Models.Product" SelectMethod ="GetProduct"
    RenderOuterTable="false">
        <ItemTemplate>
            <div>
                <h1><%#:Item.ProductName %></h1>
            </div>
            <br />
            <table>
                <tr>
                    <td>
                        "/>
                    </td>
                    <td>&nbsp;</td>
                    <td style="vertical-align: top; text-align:left;">
                        <b>Description:</b><br /><%#:Item.Description %>
                        <br />
                        <span><b>Price:</b>&nbsp;<%#: String.Format("{0:c}", Item.UnitPrice) %></span>
                        <br />
                        <span><b>Product Number:</b>&nbsp;<%#:Item.ProductID %></span>
                    </td>
                </tr>
            </table>
        </ItemTemplate>
    </asp:FormView>
</asp:Content>
```

Tiếp tục chuột phải vào trang ProductDetails.aspx -> View Code và thay code mặc định bằng đoạn code sau :

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using WingtipToys.Models;
using System.Web.ModelBinding;

namespace WingtipToys
{
    public partial class ProductDetails : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        public IQueryable<Product> GetProduct([QueryString("ProductID")] int? productId, [RouteData] string productName)
        {
            var _db = new WingtipToys.Models.ProductContext();
            IQueryable<Product> query = _db.Products;
            if (productId.HasValue && productId > 0)
            {
                query = query.Where(p => p.ProductID == productId);
            }
            else if (!String.IsNullOrEmpty(productName))
            {
                query = query.Where(p =>
                    String.Compare(p.ProductName, productName) == 0);
            }
            else
            {
                query = null;
            }
            return query;
        }
    }
}

```

Chạy lại ứng dụng chọn danh mục Car -> Convertible Car . Lúc này sẽ xuất hiện chi tiết sản phẩm :

The screenshot shows a web browser displaying the Wingtip Toys website. The header includes links for Home, About, Contact, Products, Register, and Log In. The main content area features a red logo for "Wingtip Toys". Below the logo is a navigation bar with links for Cars | Planes | Trucks | Boats | Rockets. A heading "Convertible Car" is followed by a large image of a black convertible car. To the right of the image is a "Description" section containing the text: "This convertible car is fast! The engine is powered by a neutrino based battery (not included). Power it up and let it go!", "Price: \$22.50", and "Product Number: 1". At the bottom of the page, there is a copyright notice: "© 2018 - Wingtip Toys".

Bước 11: Tạo giỏ hàng.

- Chuột phải vào thư mục Models -> Add -> New Item -> Code -> Class. Đặt tên là : CartItem.cs.

- Thay thế code mặc định của class CartItem.cs bằng đoạn code sau :

```
using System.ComponentModel.DataAnnotations;

namespace WingtipToys.Models
{
    public class CartItem
    {
        [Key]
        public string ItemId { get; set; }

        public string CartId { get; set; }

        public int Quantity { get; set; }

        public System.DateTime DateCreated { get; set; }

        public int ProductId { get; set; }

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

- Trong Solution Explorer tìm và mở file ProductContext.cs (Nằm trong thư mục Models). Thêm đoạn code được đánh dấu màu vàng vào code mặc định như sau :

```
using System.Data.Entity;

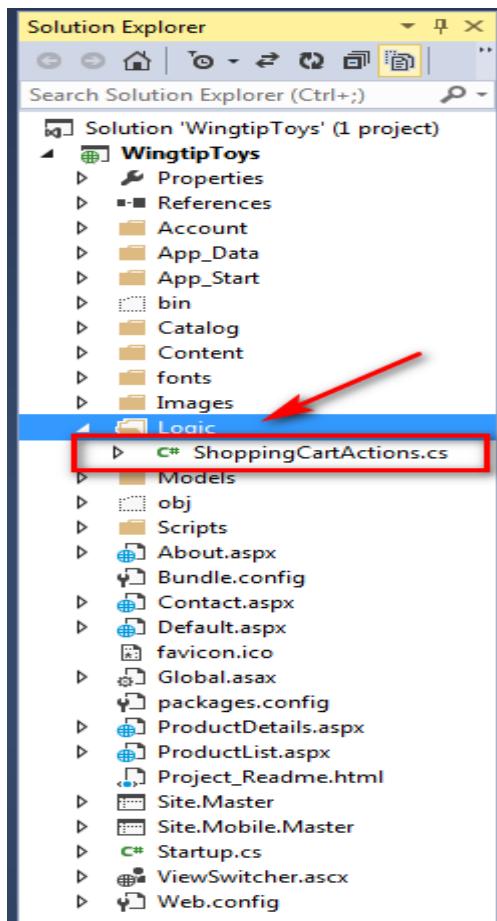
namespace WingtipToys.Models
{
    public class ProductContext : DbContext
    {
        public ProductContext()
            : base("WingtipToys")
        {

        }

        public DbSet<Category> Categories { get; set; }
        public DbSet<Product> Products { get; set; }
        public DbSet<CartItem> ShoppingCartItems { get; set; }
    }
}
```

- Tiếp theo trong Solution Explorer -> chuột phải Wingtip Toys -> Add -> New Folder và đặt tên là: Logic.

- Chuột phải vào folder Logic vừa tạo thêm một class mới tên là : ShoppingCartActions.cs . Kết quả :



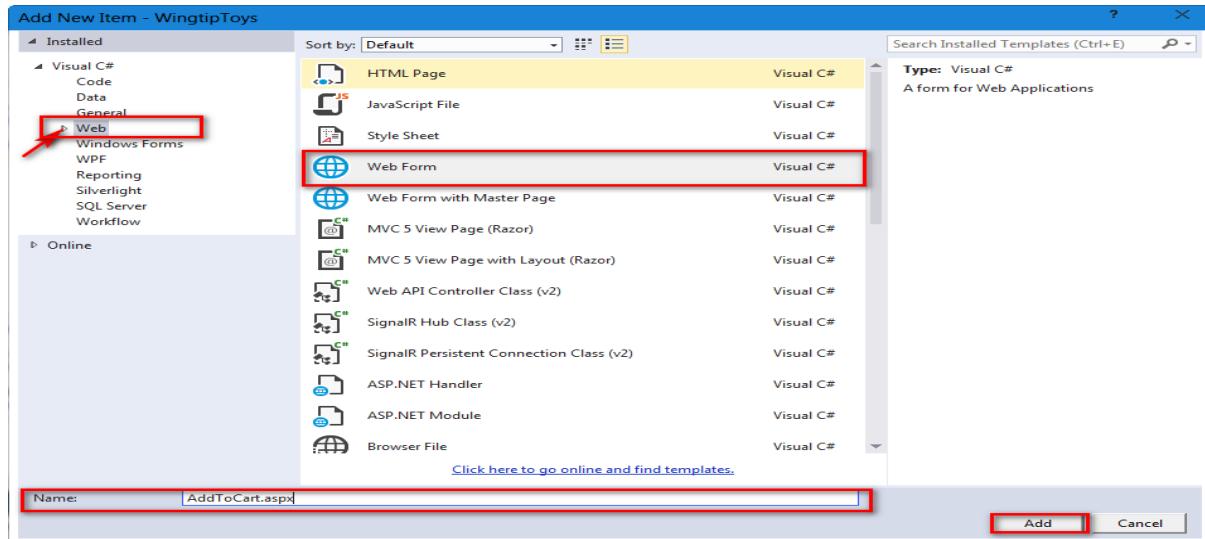
Thay thế code mặc định của ShoppingCartActions.cs bằng đoạn code sau :

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using WingtipToys.Models;
namespace WingtipToys.Logic
{
    public class ShoppingCartActions : IDisposable
    {
        public string ShoppingCartId { get; set; }
        private ProductContext _db = new ProductContext();
        public const string CartSessionKey = "CartId";
        public void AddToCart(int id)
        {
            // Retrieve the product from the database.
            ShoppingCartId = GetCartId();
            var cartItem = _db.ShoppingCartItems.SingleOrDefault(
                c => c.CartId == ShoppingCartId
                && c.ProductId == id);
            if (cartItem == null)
            {
                // Create a new cart item if no cart item exists.
                cartItem = new CartItem
                {
                    ItemId = Guid.NewGuid().ToString(),
                    ProductId = id,
                    CartId = ShoppingCartId,
                    Product = _db.Products.SingleOrDefault(
                        p => p.ProductID == id),
                    Quantity = 1,
                    DateCreated = DateTime.Now
                };
                _db.ShoppingCartItems.Add(cartItem);
            }
            else
            {
                // If the item does exist in the cart,
                // then add one to the quantity.
                cartItem.Quantity++;
            }
            _db.SaveChanges();
        }
        public void Dispose()
        {
            if (_db != null)
            {
                _db.Dispose();
                _db = null;
            }
        }
        public string GetCartId()
        {
            if (HttpContext.Current.Session[CartSessionKey] == null)
            {
                if (!string.IsNullOrWhiteSpace(HttpContext.Current.User.Identity.Name))
                {
                    HttpContext.Current.Session[CartSessionKey] =
                        HttpContext.Current.User.Identity.Name;
                }
                else
                {
                    // Generate a new random GUID using System.Guid class.
                    Guid tempCartId = Guid.NewGuid();
                    HttpContext.Current.Session[CartSessionKey] = tempCartId.ToString();
                }
            }
            return HttpContext.Current.Session[CartSessionKey].ToString();
        }
        public List<CartItem> GetCartItems()
        {
            ShoppingCartId = GetCartId();
            return _db.ShoppingCartItems.Where(
                c => c.CartId == ShoppingCartId).ToList();
        }
    }
}

```

- Trong Solution Explorer -> chuột phải Wingtip Toys -> Add -> New Item -> Web -> Web Form. Đặt tên AddToCart.aspx :



- Chuột phải vào AddToCart.aspx -> View Code . Thay code mặc định bằng đoạn code sau :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Diagnostics;
using WingtipToys.Logic;
namespace WingtipToys
{
    public partial class AddToCart : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            string rawId = Request.QueryString["ProductID"];
            int productId;
            if (!String.IsNullOrEmpty(rawId) && int.TryParse(rawId, out productId))
            {
                using (ShoppingCartActions usersShoppingCart = new ShoppingCartActions())
                {
                    usersShoppingCart.AddToCart(Convert.ToInt16(productId));
                }
            }
            else
            {
                Debug.Fail("ERROR : We should never get to AddToCart.aspx without a ProductId.");
                throw new Exception("ERROR : It is illegal to load AddToCart.aspx without setting a ProductId.");
            }
            Response.Redirect("ShoppingCart.aspx");
        }
    }
}
```

Tiếp theo chuột phải Wingtip Toys -> Add -> New Item -> Web -> Web Form with Master Page . Đặt tên là : ShoppingCart.aspx . Chọn Add -> Site.Master -> OK .

Thay tất cả code mặc định trong ShoppingCart.aspx bằng đoạn code sau :

```
<%@ Page Title="" Language="C#" MasterPageFile("~/Site.Master"
AutoEventWireup="true" CodeBehind="ShoppingCart.aspx.cs"
Inherits="WingtipToys.ShoppingCart" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
<div id="ShoppingCartTitle" runat="server" class="ContentHead"><h1>Shopping
Cart</h1></div>
<asp:GridView ID="CartList" runat="server" AutoGenerateColumns="False"
ShowFooter="True" GridLines="Vertical" CellPadding="4"
ItemType="WingtipToys.Models.CartItem"
SelectMethod="GetShoppingCartItems"
CssClass="table table-striped table-bordered" >
<Columns>
<asp:BoundField DataField="ProductID" HeaderText="ID"
SortExpression="ProductID" />
<asp:BoundField DataField="Product.ProductName" HeaderText="Name" />
<asp:BoundField DataField="Product.UnitPrice" HeaderText="Price (each)"
DataFormatString="{0:c}"/>
<asp:TemplateField HeaderText="Quantity">
<ItemTemplate>
<asp:TextBox ID="PurchaseQuantity" Width="40"
runat="server" Text="<%#: Item.Quantity %>"></asp:TextBox>
</ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Item Total">
<ItemTemplate>
<%#: String.Format("{0:c}",
((Convert.ToDouble(Item.Quantity)) *
Convert.ToDouble(Item.Product.UnitPrice)))%>
</ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Remove Item">
<ItemTemplate>
<asp:CheckBox id="Remove" runat="server"></asp:CheckBox>
</ItemTemplate>
</asp:TemplateField>
</Columns>
</asp:GridView>
<div>
<p></p>
<strong>
<asp:Label ID="LabelTotalText" runat="server" Text="Order Total:
"></asp:Label>
<asp:Label ID="lblTotal" runat="server"
EnableViewState="false"></asp:Label>
</strong>
</div>
<br />
</asp:Content>
```

-Chuột phải vào ShoppingCart.aspx -> View Code . Thay tất cả code mặc định bằng đoạn code sau :

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using WingtipToys.Models;
using WingtipToys.Logic;
namespace WingtipToys
{
    public partial class ShoppingCart : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        public List<CartItem> GetShoppingCartItems()
        {
            ShoppingCartActions actions = new ShoppingCartActions();
            return actions.GetCartItems();
        }
    }
}

```

-Trong Solution Explorer tìm và mở trang ProductList.aspx thêm đoạn code được bôi vàng như hình :

```



<%#:Item.ProductName%>


<br />

Price:

</b><%#String.Format("{0:c}", Item.UnitPrice)%>


<br />


Add To Cart



```

Nhấn F5 chạy ứng dụng. Chọn Cars -> Add To Cart một sản phẩm bất kỳ kết quả :

The screenshot shows the Wingtip Toys website with a shopping cart. The cart contains one item: a Convertible Car at \$22.50 quantity 1. The total is \$22.50. There is a 'Remove Item' link next to the item.

ID	Name	Price (each)	Quantity	Item Total	Remove Item
1	Convertible Car	\$22.50	1	\$22.50	<input type="button" value="Remove"/>

Order Total:

© 2018 - Wingtip Toys

Bước 12: Trong Solution Explorer mở class “ShoppingCartActions.cs” trong thư mục Logic. Thêm phương thức GetTotal được bôi vàng như hình sau :

```
    }
    return HttpContext.Current.Session[CartSessionKey].ToString();
}

public List<CartItem> GetCartItems()
{
    ShoppingCartId = GetCartId();

    return _db.ShoppingCartItems.Where(
        c => c.CartId == ShoppingCartId).ToList();
}

public decimal GetTotal()
{
    ShoppingCartId = GetCartId();
    // Multiply product price by quantity of that product to get
    // the current price for each of those products in the cart.
    // Sum all product price totals to get the cart total.
    decimal? total = decimal.Zero;
    total = (decimal?) (from cartItems in _db.ShoppingCartItems
        where cartItems.CartId == ShoppingCartId
        select (int?)cartItems.Quantity *
            cartItems.Product.UnitPrice).Sum();
    return total ?? decimal.Zero;
}
```

Tiếp theo chuột phải vào trang ShoppingCart.aspx trong Solution Explorer, chọn View Code. Thêm đoạn code được bôi vàng vào sự kiện Page_Load như hình :

```

{
    protected void Page_Load(object sender, EventArgs e)
    {
        using (ShoppingCartActions usersShoppingCart = new ShoppingCartActions())
        {
            decimal cartTotal = 0;
            cartTotal = usersShoppingCart.GetTotal();
            if (cartTotal > 0)
            {
                // Display Total.
                lblTotal.Text = String.Format("{0:c}", cartTotal);
            }
            else
            {
                LabelTotalText.Text = "";
                lblTotal.Text = "";
                ShoppingCartTitle.InnerText = "Shopping Cart is Empty";
            }
        }

        public List<CartItem> GetShoppingCartItems()
        {
            ShoppingCartActions actions = new ShoppingCartActions();
            return actions.GetCartItems();
        }
    }
}

```

Nhấn F5 chạy ứng dụng -> Chọn một sản phẩm bất kỳ và Add to Cart . Bây giờ ta sẽ thấy xuất hiện tổng tiền của đơn hàng như hình:

The screenshot shows the Wingtip Toys website with a blue header bar containing links for Home, About, Contact, Products, Register, and Log in. The main content area has a red title 'Wingtip Toys'. Below it are links for Cars | Planes | Trucks | Boats | Rockets. The 'Shopping Cart' section displays a table with two rows of items. The first row has ID 2, Name 'Old-time Car', Price '\$15.95', Quantity '1', Item Total '\$15.95', and a 'Remove Item' link. The second row has ID 1, Name 'Convertible Car', Price '\$22.50', Quantity '1', Item Total '\$22.50', and a 'Remove Item' link. At the bottom left of the page, the text 'Order Total: \$38.45' is highlighted with a red box and a red arrow pointing to it from the left.

ID	Name	Price (each)	Quantity	Item Total	Remove Item
2	Old-time Car	\$15.95	<input type="text" value="1"/>	\$15.95	<input type="checkbox"/>
1	Convertible Car	\$22.50	<input type="text" value="1"/>	\$22.50	<input type="checkbox"/>

Order Total: \$38.45

© 2018 - Wingtip Toys

Bước 13: Trong Solution Explorer mở trang ShoppingCart.aspx. Thêm đoạn code được bôi vàng vào phía trên </asp:Content> như hình :

```

<table>
<tr>
    <td>
        <asp:Button ID="UpdateBtn" runat="server" Text="Update"
        OnClick="UpdateBtn_Click" />
    </td>
    <td>
        <!--Checkout Placeholder -->
    </td>
</tr>
</table>

```

Tiếp theo trong Solution Explorer -> chuột phải ShoppingCart.aspx -> View Code . Thêm đoạn code được đánh dấu màu vàng vào như hình:

```
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using WingtipToys.Models;
using WingtipToys.Logic;
using System.Collections.Specialized;
using System.Collections;
using System.Web.ModelBinding;

namespace WingtipToys
{
    public partial class ShoppingCart : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            using (ShoppingCartActions usersShoppingCart = new ShoppingCartActions())
            {
                decimal cartTotal = 0;
                cartTotal = usersShoppingCart.GetTotal();
                if (cartTotal > 0)
                {
                    // Display Total.
                    lblTotal.Text = String.Format("{0:c}", cartTotal);
                }
                else
                {
                    LabelTotalText.Text = "";
                    lblTotal.Text = "";
                    ShoppingCartTitle.InnerText = "Shopping Cart is Empty";
                    UpdateBtn.Visible = false;
                }
            }
        }
    }
}
```

Và thêm vào sau :

```
public List<CartItem> GetShoppingCartItems()
{
    ShoppingCartActions actions = new ShoppingCartActions();
    return actions.GetCartItems();
}
```

Đoạn code :

```

public List<CartItem> UpdateCartItems()
{
    using (ShoppingCartActions usersShoppingCart = new ShoppingCartActions())
    {
        String cartId = usersShoppingCart.GetCartId();
        ShoppingCartActions.ShoppingCartUpdates[] cartUpdates = new ShoppingCartActions.ShoppingCartUpdates[CartList.Rows.Count];
        for (int i = 0; i < CartList.Rows.Count; i++)
        {
            IOrderedDictionary rowValues = new OrderedDictionary();
            rowValues = GetValues(CartList.Rows[i]);
            cartUpdates[i].ProductId = Convert.ToInt32(rowValues["ProductID"]);
            CheckBox cbRemove = new CheckBox();
            cbRemove = (CheckBox)CartList.Rows[i].FindControl("Remove");
            cartUpdates[i].RemoveItem = cbRemove.Checked;
            TextBox quantityTextBox = new TextBox();
            quantityTextBox =
                (TextBox)CartList.Rows[i].FindControl("PurchaseQuantity");
            cartUpdates[i].PurchaseQuantity =
                Convert.ToInt16(quantityTextBox.Text.ToString());
        }
        usersShoppingCart.UpdateShoppingCartDatabase(cartId, cartUpdates);
        CartList.DataBind();
        lblTotal.Text = String.Format("{0:c}", usersShoppingCart.GetTotal());
        return usersShoppingCart.GetCartItems();
    }
}
public static IOrderedDictionary GetValues(GridViewRow row)
{
    IOrderedDictionary values = new OrderedDictionary();
    foreach (DataControlFieldCell cell in row.Cells)
    {
        if (cell.Visible)
        {
            // Extract values from the cell.
            cell.ContainingField.ExtractValuesFromCell(values, cell,
                row.RowState, true);
        }
    }
    return values;
}
protected void UpdateBtn_Click(object sender, EventArgs e)
{
    UpdateCartItems();
}

```

Trong Solution Explorer, mở class ShoppingCartActions.cs trong thư mục Logic. Thêm vào sau:

```

public decimal GetTotal()
{
    ShoppingCartId = GetCartId();
    // Multiply product price by quantity of that product to get
    // the current price for each of those products in the cart.
    // Sum all product price totals to get the cart total.
    decimal? total = decimal.Zero;
    total = (decimal?) (from cartItems in _db.ShoppingCartItems
                        where cartItems.CartId == ShoppingCartId
                        select (int?)cartItems.Quantity *
                            cartItems.Product.UnitPrice).Sum();
    return total ?? decimal.Zero;
}

```

Đoạn code : (Nhớ Ctr+A để copy được hết code)

```
public ShoppingCartActions GetCart(HttpContext context)
{
    using (var cart = new ShoppingCartActions())
    {
        cart.ShoppingCartId = cart.GetCartId();
        return cart;
    }
}

public void UpdateShoppingCartDatabase(String cartId, ShoppingCartUpdates[]
CartItemUpdates)
{
    using (var db = new WingtipToys.Models.ProductContext())
    {
        try
        {
            int CartItemCount = CartItemUpdates.Count();
            List<CartItem> myCart = GetCartItems();
            foreach (var cartItem in myCart)
            {
                // Iterate through all rows within shopping cart list
                for (int i = 0; i < CartItemCount; i++)
                {
                    if (cartItem.Product.ProductID == CartItemUpdates[i].ProductId)
                    {
                        if (CartItemUpdates[i].PurchaseQuantity < 1 ||
                            CartItemUpdates[i].RemoveItem == true)
                        {
                            RemoveItem(cartId, cartItem.ProductId);
                        }
                        else
                        {
                            UpdateItem(cartId, cartItem.ProductId,
                            CartItemUpdates[i].PurchaseQuantity);
                        }
                    }
                }
            }
        }
        catch (Exception exp)
        {
            throw new Exception("ERROR: Unable to Update Cart Database - " +
exp.Message.ToString(), exp);
        }
    }
}

public void RemoveItem(string removeCartID, int removeProductID)
{
    using (var _db = new WingtipToys.Models.ProductContext())
    {
        try
        {
            var myItem = (from c in _db.ShoppingCartItems where c.CartId ==
removeCartID && c.Product.ProductID == removeProductID select
c).FirstOrDefault();
            if (myItem != null)
            {
                // Remove Item.
                _db.ShoppingCartItems.Remove(myItem);
                _db.SaveChanges();
            }
        }
        catch (Exception exp)
        {
            throw new Exception("ERROR: Unable to Remove Cart Item - " +
exp.Message.ToString(), exp);
        }
    }
}

public void UpdateItem(string updateCartID, int updateProductID, int
quantity)
{
    using (var _db = new WingtipToys.Models.ProductContext())
    {
        ....
```

Trong Solution Explorer mở Site.Master. Thêm đoạn code được bôi vàng như hình :

```
<ul class="nav navbar-nav">
    <li><a runat="server" href="/">Home</a></li>
    <li><a runat="server" href="/About">About</a></li>
    <li><a runat="server" href="/Contact">Contact</a></li>
    <li><a runat="server" href="/ProductList">Products</a></li>
    <li><a runat="server" href="/ShoppingCart"
        ID="cartCount">&nbsp;</a></li>
</ul>
```

Tiếp theo chuột phải Site.Master -> View Code . Thêm các đoạn code được bôi vàng như hình :

```
using System;
using System.Collections.Generic;
using System.Security.Claims;
using System.Security.Principal;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Linq;
using WingtipToys.Models;
using WingtipToys.Logic;

}

protected void Page_Load(object sender, EventArgs e)
{
}

protected void Page_PreRender(object sender, EventArgs e)
{
    using (ShoppingCartActions usersShoppingCart = new
    ShoppingCartActions())
    {
        string cartStr = string.Format("Cart ({0})",
usersShoppingCart.GetCount());
        cartCount.InnerText = cartStr;
    }
}

public IQueryable<Category> GetCategories()
{
    var _db = new WingtipToys.Models.ProductContext();
    IQueryable<Category> query = db.Categories;
    return query;
}
```

Nhấn F5 chạy ứng dụng. Chọn một vài sản phẩm. Kết quả như hình :



Wingtip Toys

Cars | Planes | Trucks | Boats | Rockets

Shopping Cart

ID	Name	Price (each)	Quantity	Item Total	Remove Item
1	Convertible Car	\$22.50	1	\$22.50	<input type="checkbox"/>

Order Total: \$22.50

© 2018 - Wingtip Toys

Thử thay số lượng ở mục “Quantity” từ 1-> 3 rồi nhấn Update sẽ có kết quả :



Wingtip Toys

Cars | Planes | Trucks | Boats | Rockets

Shopping Cart

ID	Name	Price (each)	Quantity	Item Total	Remove Item
1	Convertible Car	\$22.50	3	\$67.50	<input type="checkbox"/>

Order Total: \$67.50

© 2018 - Wingtip Toys

Bước 14: Trong thư mục Models tạo thêm 1 class mới đặt tên : Order.cs. Sau đó thay thế code mặc định bằng đoạn code sau :

```
using System.ComponentModel.DataAnnotations;
using System.Collections.Generic;
using System.ComponentModel;
namespace WingtipToys.Models
{
    public class Order
    {
        public int OrderId { get; set; }
        public System.DateTime OrderDate { get; set; }
        public string Username { get; set; }
        [Required(ErrorMessage = "First Name is required")]
        [DisplayName("First Name")]
        [StringLength(160)]
        public string FirstName { get; set; }
        [Required(ErrorMessage = "Last Name is required")]
        [DisplayName("Last Name")]
        [StringLength(160)]
        public string LastName { get; set; }
        [Required(ErrorMessage = "Address is required")]
        [StringLength(70)]
        public string Address { get; set; }
        [Required(ErrorMessage = "City is required")]
        [StringLength(40)]
        public string City { get; set; }
        [Required(ErrorMessage = "State is required")]
        [StringLength(40)]
        public string State { get; set; }
        [Required(ErrorMessage = "Postal Code is required")]
        [DisplayName("Postal Code")]
        [StringLength(10)]
        public string PostalCode { get; set; }
        [Required(ErrorMessage = "Country is required")]
        [StringLength(40)]
        public string Country { get; set; }
        [StringLength(24)]
        public string Phone { get; set; }
        [Required(ErrorMessage = "Email Address is required")]
        [DisplayName("Email Address")]
        [RegularExpression(@"[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}", ErrorMessage = "Email is not valid.")]
        [DataType(DataType.EmailAddress)]
        public string Email { get; set; }
        [ScaffoldColumn(false)]
        public decimal Total { get; set; }
        [ScaffoldColumn(false)]
        public string PaymentTransactionId { get; set; }
        [ScaffoldColumn(false)]
        public bool HasBeenShipped { get; set; }
        public List<OrderDetail> OrderDetails { get; set; }
    }
}
```

Tiếp theo thêm 1 class mới và đặt tên: OrderDetail.cs cũng trong thư mục Models. Tiếp tục thay code mặc định bằng đoạn code sau :

```
using System.ComponentModel.DataAnnotations;
namespace WingtipToys.Models
{
    public class OrderDetail
    {
        public int OrderDetailId { get; set; }
        public int OrderId { get; set; }
        public string Username { get; set; }
        public int ProductId { get; set; }
        public int Quantity { get; set; }
        public double? UnitPrice { get; set; }
    }
}
```

- Trong Solution Explorer, tìm và mở class ProductContext.cs . Thêm đoạn code được bôi vàng vào như hình :

```
{
}
public DbSet<Category> Categories { get; set; }
public DbSet<Product> Products { get; set; }
public DbSet<CartItem> ShoppingCartItems { get; set; }
public DbSet<Order> Orders { get; set; }
public DbSet<OrderDetail> OrderDetails { get; set; }
}
```

Bước 15: Chuột phải dự án Wingtip Toys trong Solution Explorer -> Add -> New Folder và đặt tên là Checkout.

-Chuột phải thư mục Checkout -> Add -> New Item -> Web -> Web Form with Master Page và đặt tên là : CheckoutStart.aspx -> Add -> Site.Master -> OK.

Thực hiện tương tự tạo ra thêm 4 trang và đặt tên lần lượt :

- CheckoutReview.aspx
- CheckoutComplete.aspx
- CheckoutCancel.aspx
- CheckoutError.aspx

Tiếp theo chuột phải thư mục Checkout -> Add -> New Item -> Web -> Web Configuration File (để tên mặc định là Web.config)-> Add. Thay thế code mặc định trong Web.config vừa mới tạo (nằm trong thư mục Checkout) bằng đoạn code sau:

```

<?xml version="1.0"?>
<configuration>
<system.web>
<authorization>
<deny users="?" />
</authorization>
</system.web>
</configuration>

```

Bước 16: Trong Solution Explorer, tìm và mở thư mục App_Start -> mở file Startup.Auth.cs. Bỏ dấu “//” trước app.UseGoogleAuthentication() như hình :

```

    // Uncomment the following lines to enable logging in with third
party login providers
    //app.UseMicrosoftAccountAuthentication(
    //    clientId: "",
    //    clientSecret: "");

    //app.UseTwitterAuthentication(
    //    consumerKey: "",
    //    consumerSecret: "");

    //app.UseFacebookAuthentication(
    //    appId: "",
    //    appSecret: "");

    app.UseGoogleAuthentication();

```

-Tiếp theo, mở thư mục Account trong Solution Explorer. Chuột phải trang Login.aspx -> View Code. Thêm đoạn code được bôi vàng như hình :

```

protected void LogIn(object sender, EventArgs e)
{
    if (IsValid)
    {
        // Validate the user password
        var manager = new UserManager();
        ApplicationUser user = manager.Find(UserName.Text,
Password.Text);
        if (user != null)
        {
            IdentityHelper.SignIn(manager, user, RememberMe.Checked);

            WingtipToys.Logic.ShoppingCartActions usersShoppingCart =
new WingtipToys.Logic.ShoppingCartActions();
            String cartId = usersShoppingCart.GetCartId();
            usersShoppingCart.MigrateCart(cartId, UserName.Text);

            IdentityHelper.RedirectToReturnUrl(Request.QueryString["ReturnUrl"], Response);
        }
        else
        {
            FailureText.Text = "Invalid username or password.";
            ErrorMessage.Visible = true;
        }
    }
}

```

-Trong thư mục Account, chuột phải Register.aspx -> View Code. Thêm đoạn code được bôi vàng như hình:

```

using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.AspNet.Identity.Owin;
using System;
using System.Linq;
using System.Web;
using System.Web.UI;
using WingtipToys.Models;

namespace WingtipToys.Account
{
    public partial class Register : Page
    {
        protected void CreateUser_Click(object sender, EventArgs e)
        {
            var manager = new UserManager();
            var user = new ApplicationUser() { UserName = UserName.Text };
            IdentityResult result = manager.Create(user, Password.Text);
            if (result.Succeeded)
            {
                IdentityHelper.SignIn(manager, user, isPersistent: false);

                using (WingtipToys.Logic.ShoppingCartActions usersShoppingCart
= new WingtipToys.Logic.ShoppingCartActions())
                {
                    String cartId = usersShoppingCart.GetCartId();
                    usersShoppingCart.MigrateCart(cartId, user.Id);
                }
            }

            IdentityHelper.RedirectToReturnUrl(Request.QueryString["ReturnUrl"], Response);
        }
        else
        {
            ErrorMessage.Text = result.Errors.FirstOrDefault();
        }
    }
}

```

-Mở ShoppingCartActions.cs trong thư mục Logic. Thêm đoạn code được bôi vàng như hình:

```

3 references
public struct ShoppingCartUpdates
{
    public int ProductId;
    public int PurchaseQuantity;
    public bool RemoveItem;
}

2 references
public void MigrateCart(string cartId, string userName)
{
    var shoppingCart = _db.ShoppingCartItems.Where(c => c.CartId == cartId);
    foreach (CartItem item in shoppingCart)
    {
        item.CartId = userName;
    }
    HttpContext.Current.Session[CartSessionKey] = userName;
    _db.SaveChanges();
}
}

```

Bước 17: Truy cập đường dẫn: <https://developer.paypal.com/developer/accounts/> . Kết quả như thế này :

The screenshot shows the PayPal Developer interface for managing sandbox accounts. The left sidebar has sections for Dashboard, Sandbox (Accounts, Notifications, API Calls, IPN Simulator, Webhooks Events), Mock (Webhooks Simulator, Credit Card Generator), and Live (API Calls, Webhooks Events). The main content area is titled "Sandbox Accounts". It features a box for "Existing sandbox users" with instructions to log in with a PayPal account. Below this, a note says to test code end-to-end using sandbox accounts and links to the Testing Guide. A separate box for non-U.S. developers links to the FAQ. At the bottom right of the main area is a "Create Account" button.

Sau đó tiến hành đăng nhập. Chú ý: nếu chưa có tài khoản thì nhớ đăng ký.

Sau khi đăng nhập màn hình sẽ hiện như thế này :

This screenshot shows the same PayPal Developer interface after logging in. The "Create Account" button is now visible in the top right of the main content area. A large red box highlights the central area where account details would normally be displayed, but it is currently empty. Other elements like the "Delete Accounts" button and the "Take a survey on sandbox negative testing" link are also visible.

Chúng ta tiến hành tạo 1 tài khoản để kiểm tra chức năng đặt hàng và thanh toán bằng Paypal bằng cách nhấn vào “Create Account”. Sau đó điền thông tin và Create Account :

Account Type

- Personal (Buyer Account)
- Business (Merchant Account)

Email Address

Điền Email

Password(8-20 characters)

Mật khẩu

First Name(optional)

Last Name(optional)

Payment Methods

PayPal Balance .00 USD

100 **Số tiền tài khoả**

Bank Verified Account

- Yes
- No

Select Payment Card

- Discover
- PayPal

Credit Cards

Visa ▾

Note(optional)

Create Account

Sau khi đăng ký xong tài khoản màn hình Sandbox sẽ như thế này :

The screenshot shows the 'Sandbox Accounts' page. At the top right is a 'Create Account' button. Below it is a message: 'Questions? Check out the [Testing Guide](#). Non-US developers should read our [FAQ](#).'. A note below says: 'To link your sandbox account to your developer account, [log in with PayPal](#) and provide your sandbox account credentials.' A table lists three accounts:

Email Address	Type	Country	Date Created	Status	Actions
[REDACTED]	PERSONAL	US	06 Mar 2018	complete	...
[REDACTED] facilitator@gmail.com	BUSINESS	US	06 Mar 2018	complete	...
[REDACTED] buyer@gmail.com	PERSONAL	US	06 Mar 2018	complete	...

Total records: 3

[Delete Accounts](#)

Click vào mũi tên ở email “facilitator” và chọn Profile như hình:

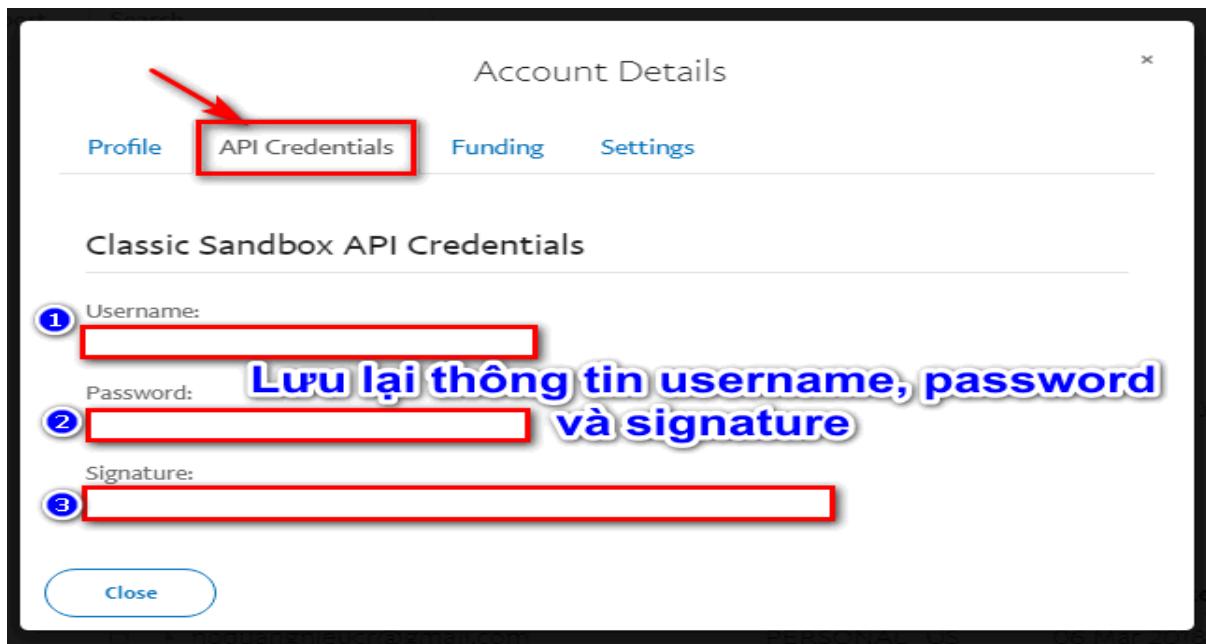
The screenshot shows the same 'Sandbox Accounts' page. A red arrow points to the dropdown menu next to the email address 'facilitator@gmail.com'. The 'Profile' tab is highlighted with a red box. The table data remains the same as the previous screenshot.

Email Address	Type	Country	Date Created	Status	Actions
[REDACTED] @gmail.com	PERSONAL	US	06 Mar 2018	complete	...
[REDACTED] @gmail.com	BUSINESS	US	06 Mar 2018	complete	...
[REDACTED]	PERSONAL	US	06 Mar 2018	complete	...

Total records: 3

[Delete Accounts](#)

Ở Account Details, chúng ta chuyển qua tab API Credentials và lưu lại 3 thông tin username, password, Signature để sau cần dùng:



-Tiếp theo chuột phải thư mục Logic -> Add -> New Item -> Code -> Class -> đặt tên : PayPalFunctions.cs. Sau đó thay thế code mặc định trong PayPalFunctions.cs bằng cách truy cập : https://drive.google.com/open?id=1zum-4tIP-Fc6prJnqEsCCrRoPiCNSWG_.

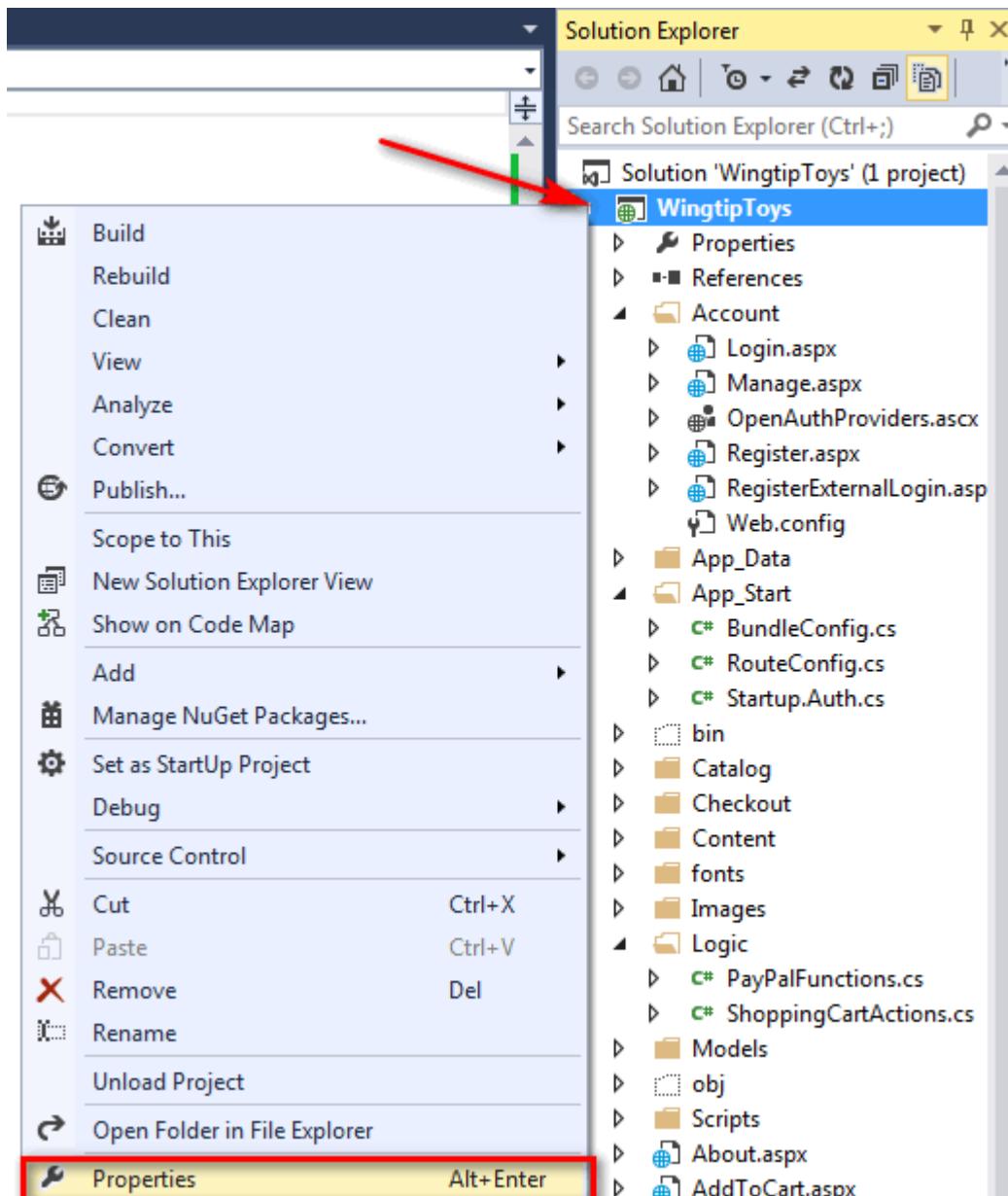
Tìm và điền các thông tin username, password, signature đã lưu như hình :

```

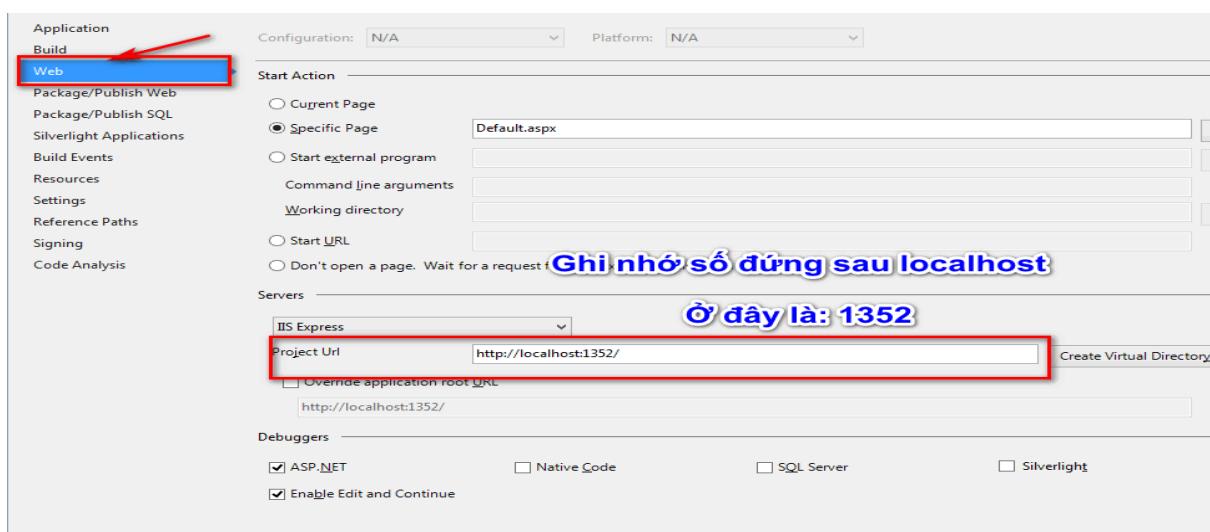
using System;
using System.Collections;
using System.Collections.Specialized;
using System.IO;
using System.Net;
using System.Text;
using System.Data;
using System.Configuration;
using System.Web;
using WingtipToys;
using WingtipToys.Models;
using System.Collections.Generic;
using System.Linq;
References
public class NVPAPICaller
{
    //Flag that determines the PayPal environment (live or sandbox)
    private const bool bSandbox = true;
    private const string CVV2 = "CVV2";
    // Live strings.
    private string pEndPointURL = "https://api-3t.paypal.com/nvp";
    private string host = "www.paypal.com";
    // Sandbox strings.
    private string pEndPointURL_SB = "https://api-3t.sandbox.paypal.com/nvp";
    private string host_SB = "www.sandbox.paypal.com";
    private const string SIGNATURE = "SIGNATURE";
    private const string PWD = "PWD";
    private const string ACCT = "ACCT";
    //Replace <Your API Username> with your API Username
    //Replace <Your API Password> with your API Password
    //Replace <Your Signature> with your Signature
    public string APIUsername = "<Your API Username>";
    private string APIPassword = "<Your API Password>";
    private string APISignature = "<Your Signature>";
}

```

-Tiếp theo chuột phải vào dự án Wingtip Toys -> Properties :



Chọn Web, nhìn xuống địa chỉ Project Url:



Quay trở lại PayPalFunctions.cs tìm và thay số port đã có ở trên vào như hình :

```
-----  
public bool ShortcutExpressCheckout(string amt, ref string token, ref string  
retMsg)  
{  
    if (bSandbox)  
    {  
        pEndPointURL = pEndPointURL_SB;  
        host = host_SB;  
    }  
    string returnUrl = "http://localhost 1234 Checkout/CheckoutReview.aspx";  
    string cancelURL = "http://localhost 1234 Checkout/CheckoutCancel.aspx";  
    NVPCodec encoder = new NVPCodec();  
    encoder["METHOD"] = "SetExpressCheckout";  
    encoder["RETURNURL"] = returnUrl;  
    encoder["CANCELURL"] = cancelURL;  
    encoder["BRANDNAME"] = "Wingtip Toys Sample Application";  
    encoder["PAYMENTREQUEST_0_AMT"] = amt;  
    encoder["PAYMENTREQUEST_0_ITEMAMT"] = amt;  
    encoder["PAYMENTREQUEST_0_PAYMENTACTION"] = "Sale";  
    encoder["PAYMENTREQUEST_0_CURRENCYCODE"] = "USD";  
    // Get the Shopping Cart Products  
    using (WingtipToys.Logic.ShoppingCartActions myCartOrders = new  
        WingtipToys.Logic.ShoppingCartActions())
```

Thay số : 1234 bằng số port
lúc này ta có

Trong Solution Explorer, mở trang ShoppingCart.aspx. Kéo xuống dưới và tìm dòng <!--Checkout Placeholder -->. Thêm đoạn code sau ở dưới <!--Checkout Placeholder --> :

```
<asp:ImageButton ID="CheckoutImageBtn" runat="server"  
    ImageUrl="https://www.paypal.com/en_US/i/btn/btn_xpressCheckout.gif"  
    Width="145" AlternateText="Check out with PayPal"  
    OnClick="CheckoutBtn_Click"  
    BackColor="Transparent" BorderWidth="0" />
```

Chuột phải ShoppingCart.aspx -> View Code. Kéo xuống dưới và thêm đoạn code được bôi vàng như hình :

```
-----  
protected void UpdateBtn_Click(object sender, EventArgs e)  
{  
    UpdateCartItems();  
}  
  
protected void CheckoutBtn_Click(object sender, ImageClickEventArgs e)  
{  
    using (ShoppingCartActions usersShoppingCart = new ShoppingCartActions())  
    {  
        Session["payment_amt"] = usersShoppingCart.GetTotal();  
    }  
    Response.Redirect("Checkout/CheckoutStart.aspx");  
}
```

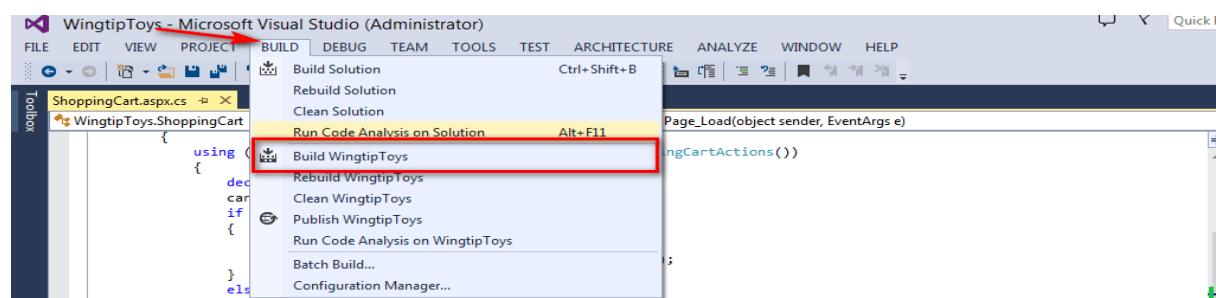
Cũng trong ShoppingCart.aspx.cs thêm đoạn code bôi vàng như hình:

```

protected void Page_Load(object sender, EventArgs e)
{
    using (ShoppingCartActions usersShoppingCart = new ShoppingCartActions())
    {
        decimal cartTotal = 0;
        cartTotal = usersShoppingCart.GetTotal();
        if (cartTotal > 0)
        {
            // Display Total.
            lblTotal.Text = String.Format("{0:c}", cartTotal);
        }
        else
        {
            LabelTotalText.Text = "";
            lblTotal.Text = "";
            ShoppingCartTitle.InnerText = "Shopping Cart is Empty";
            UpdateBtn.Visible = false;
            CheckoutImageBtn.Visible = false;
        }
    }
}

```

Từ thanh Menu -> BUILD -> Build Wingtip Toys.



Bước 18: Từ thư mục Checkout. Chuột phải CheckoutStart.aspx -> View Code. Thay thế code mặc định bằng đoạn code sau :

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace WingtipToys.Checkout
{
    public partial class CheckoutStart : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            NVPAPICaller payPalCaller = new NVPAPICaller();
            string retMsg = "";
            string token = "";
            if (Session["payment_amt"] != null)
            {
                string amt = Session["payment_amt"].ToString();
                bool ret = payPalCaller.ShortcutExpressCheckout(amt, ref token, ref
                retMsg);
                if (ret)
                {
                    Session["token"] = token;
                    Response.Redirect(retMsg);
                }
                else
                {
                    Response.Redirect("CheckoutError.aspx?" + retMsg);
                }
            }
            else
            {
                Response.Redirect("CheckoutError.aspx?ErrorCode=AmtMissing");
            }
        }
    }
}

```

- Tiếp tục mở trang CheckoutReview.aspx trong thư mục Checkout. Thay thế code mặc định bằng đoạn code sau:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="CheckoutReview.aspx.cs"
Inherits="WingtipToys.Checkout.CheckoutReview" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
<h1>Order Review</h1>
<p></p>
<h3 style="padding-left: 33px">Products:</h3>
<asp:GridView ID="OrderItemList" runat="server" AutoGenerateColumns="False"
GridLines="Both" CellPadding="10" Width="500" BorderColor="#efeeef"
BorderWidth="33">
<Columns>
<asp:BoundField DataField="ProductId" HeaderText=" Product ID" />
<asp:BoundField DataField="Product.ProductName" HeaderText="Product Name" />
<asp:BoundField DataField="Product.UnitPrice" HeaderText="Price (each)" DataFormatString="{0:c}"/>
<asp:BoundField DataField="Quantity" HeaderText="Quantity" />
</Columns>
</asp:GridView>
<asp:DetailsView ID="ShipInfo" runat="server" AutoGenerateRows="false"
GridLines="None" CellPadding="10" BorderStyle="None" CommandRowStyleBorderStyle="None">
<Fields>
<asp:TemplateField>
<ItemTemplate>
<h3>Shipping Address:</h3>
<br />
<asp:Label ID="FirstName" runat="server" Text='<%#: Eval("FirstName") %>'></asp:Label>
<asp:Label ID="LastName" runat="server" Text='<%#: Eval("LastName") %>'></asp:Label>
<br />
<asp:Label ID="Address" runat="server" Text='<%#: Eval("Address") %>'></asp:Label>
<br />
<asp:Label ID="City" runat="server" Text='<%#: Eval("City") %>'></asp:Label>
<asp:Label ID="State" runat="server" Text='<%#: Eval("State") %>'></asp:Label>
<asp:Label ID="PostalCode" runat="server" Text='<%#: Eval("PostalCode") %>'></asp:Label>
<p></p>
<h3>Order Total:</h3>
<br />
<asp:Label ID="Total" runat="server" Text='<%#: Eval("Total", "{0:C}") %>'></asp:Label>
</ItemTemplate>
<ItemStyle HorizontalAlign="Left" />
</asp:TemplateField>
</Fields>
</asp:DetailsView>
<p></p>
<hr />
<asp:Button ID="CheckoutConfirm" runat="server" Text="Complete Order"
OnClick="CheckoutConfirm_Click" />
</asp:Content>
```

- Chuột phải CheckoutReview.aspx -> View Code. Truy cập :

https://drive.google.com/open?id=1ARG1ydQS3-EhwKmHWLrVAKvH_JwINO-5 và thay code mặc định bằng code trong đường dẫn.

-Trong thư mục Checkout, mở trang CheckoutComplete.aspx và thay code mặc định bằng đoạn code sau:

```
<%@ Page Title="" Language="C#" MasterPageFile("~/Site.Master")
AutoEventWireup="true" CodeBehind="CheckoutComplete.aspx.cs"
Inherits="WingtipToys.Checkout.CheckoutComplete" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
<h1>Checkout Complete</h1>
<p></p>
<h3>Payment Transaction ID:</h3> <asp:Label ID="TransactionId"
runat="server"></asp:Label>
<p></p>
<h3>Thank You!</h3>
<p></p>
<hr />
<asp:Button ID="Continue" runat="server" Text="Continue Shopping"
OnClick="Continue_Click" />
</asp:Content>
```

-Chuột phải CheckoutComplete.aspx -> View Code. Thay code mặc định bằng đoạn code sau:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using WingtipToys.Models;
namespace WingtipToys.Checkout
{
    public partial class CheckoutComplete : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                // Verify user has completed the checkout process.
                if ((string)Session["userCheckoutCompleted"] != "true")
                {
                    Session["userCheckoutCompleted"] = string.Empty;
                    Response.Redirect("CheckoutError.aspx?" +
                        "Desc=Unvalidated%20Checkout.");
                }
                NVPAPICaller payPalCaller = new NVPAPICaller();
                string retMsg = "";
                string token = "";
                string finalPaymentAmount = "";
                string PayerID = "";
                NVPCodec decoder = new NVPCodec();
                token = Session["token"].ToString();
                PayerID = Session["payerId"].ToString();
                finalPaymentAmount = Session["payment_amt"].ToString();
                bool ret = payPalCaller.DoCheckoutPayment(finalPaymentAmount, token,
                    PayerID, ref decoder, ref retMsg);
                if (ret)
                {
                    // Retrieve PayPal confirmation value.
                    string PaymentConfirmation =
                        decoder["PAYMENTINFO_0_TRANSACTIONID"].ToString();
                    TransactionId.Text = PaymentConfirmation;
                    ProductContext _db = new ProductContext();
                    // Get the current order id.
                    int currentOrderId = -1;
                    if (Session["currentOrderId"] != string.Empty)
                    {
                        currentOrderId = Convert.ToInt32(Session["currentOrderID"]);
                    }
                    Order myCurrentOrder;
                    if (currentOrderId >= 0)
                    {
                        // Get the order based on order id.
                        myCurrentOrder = _db.Orders.Single(o => o.OrderId ==
                            currentOrderId);
                        // Update the order to reflect payment has been completed.
                        myCurrentOrder.PaymentTransactionId = PaymentConfirmation;
                        // Save to DB.
                        _db.SaveChanges();
                    }
                    // Clear shopping cart.
                    using (WingtipToys.Logic.ShoppingCartActions usersShoppingCart =
                        new WingtipToys.Logic.ShoppingCartActions())
                    {
                        usersShoppingCart.EmptyCart();
                    }
                    // Clear order id.
                    Session["currentOrderId"] = string.Empty;
                }
                else
                {
                    Response.Redirect("CheckoutError.aspx?" + retMsg);
                }
            }
        }

        protected void Continue_Click(object sender, EventArgs e)
        {
            Response.Redirect("~/Default.aspx");
        }
    }
}

```

- Mở trang CheckoutCancel.aspx. Thay code mặc định bằng đoạn code sau:

```
<%@ Page Title="" Language="C#" MasterPageFile "~/Site.Master"
AutoEventWireup="true" CodeBehind="CheckoutCancel.aspx.cs"
Inherits="WingtipToys.Checkout.CheckoutCancel" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
<h1>Checkout Cancelled</h1>
<p></p>
<h3>Your purchase has been cancelled.</h3>
</asp:Content>
```

-Tương tự mở CheckoutError.aspx và thay đoạn code sau:

```
<%@ Page Title="" Language="C#" MasterPageFile "~/Site.Master"
AutoEventWireup="true" CodeBehind="CheckoutError.aspx.cs"
Inherits="WingtipToys.Checkout.CheckoutError" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
<h1>Checkout Error</h1>
<p></p>
<table id="ErrorTable">
<tr>
<td class="field"></td>
<td><%=Request.QueryString.Get("ErrorCode")%></td>
</tr>
<tr>
<td class="field"></td>
<td><%=Request.QueryString.Get("Desc")%></td>
</tr>
<tr>
<td class="field"></td>
<td><%=Request.QueryString.Get("Desc2")%></td>
</tr>
</table>
<p></p>
</asp:Content>
```

-Trong Solution Explorer, mở file Global.asax thêm đoạn code được bôi vàng như hình:

```
public class Global : HttpApplication
{
    void Application_Start(object sender, EventArgs e)
    {
        // Code that runs on application startup
        RouteConfig.RegisterRoutes(RouteTable.Routes);
        BundleConfig.RegisterBundles(BundleTable.Bundles);

        // Initialize the product database.
        Database.SetInitializer(new ProductDatabaseInitializer());

        // This setting is now needed for PayPal in sandbox mode
        System.Net.ServicePointManager.SecurityProtocol = System.Net.SecurityProtocolType.Tls12;
    }
}
```

Chạy ứng dụng, đặt hàng sản phẩm bất kỳ lúc này sẽ xuất hiện thêm button thanh toán bằng PayPal:

Wingtip Toys

Cars | Planes | Trucks | Boats | Rockets

Shopping Cart

ID	Name	Price (each)	Quantity	Item Total	Remove Item
2	Old-time Car	\$15.95	1	\$15.95	<input type="checkbox"/>

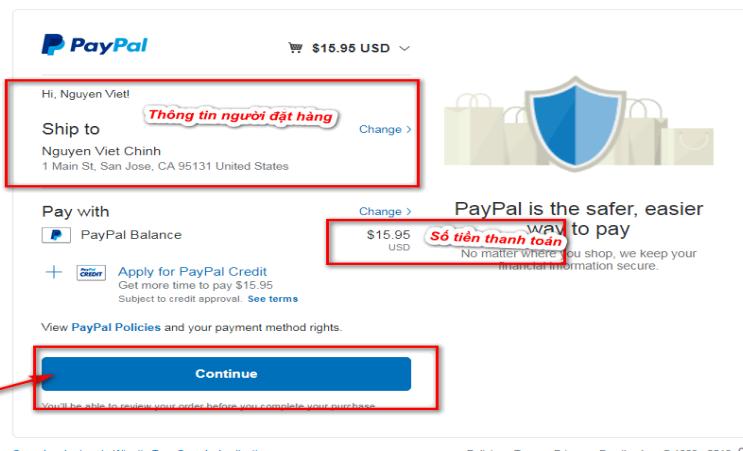
Order Total: \$15.95

The safer, easier way to pay

© 2018 - Wingtip Toys

Nhấn vào thanh toán, sẽ đưa tới trang thanh toán của PayPal như hình (nhớ đăng nhập PayPal) :

Wingtip Toys Sample Application



Tiếp tục click Continue :

Wingtip Toys

Cars | Planes | Trucks | Boats | Rockets

Order Review

Products:

Product ID	Product Name	Price(each)	Quantity
2	Old-time Car	\$15.95	1

Shipping Address:

Nguyen Viet Chinh
1 Main St
San Jose CA 95131

Order Total:

\$15.95

© 2018 - Wingtip Toys

Complete Order để hoàn thành thanh toán cho đơn hàng.

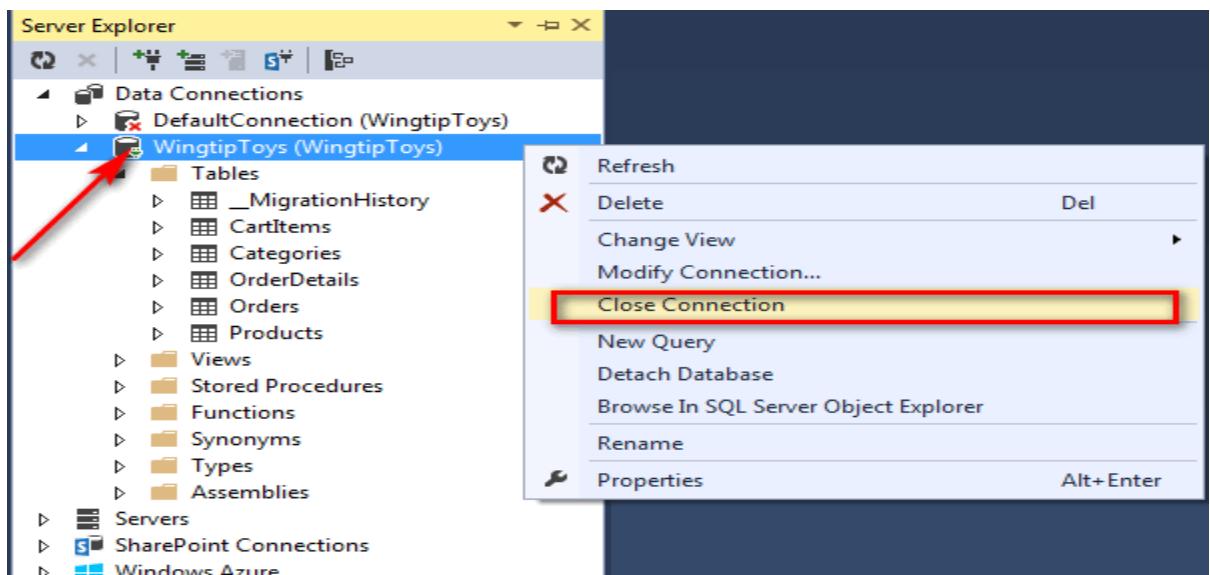
Bước 19: Trong Solution Explorer, Refresh -> Show All Files. Mở thư mục App_Data -> chuột phải wingtiptoys.mdf -> Open -> Table -> chuột phải Orders -> Show Table Data. Ta sẽ thấy được thông tin các đơn hàng đã Orders :

	OrderId	OrderDate	Username	FirstName	LastName	Address	City	State	PostalCode	Country
▶	1	3/8/2018 6:28:2...	chinh	Nguyen Viet	Chinh	1 Main St	San Jose	CA	95131	US
	2	3/9/2018 7:12:3...	chinh	Nguyen Viet	Chinh	1 Main St	San Jose	CA	95131	US
*	3	4/8/2018 8:00:1...	chinh	Nguyen Viet	Chinh	1 Main St	San Jose	CA	95131	US
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tương tự chọn OrderDetails -> Show Table Data sẽ hiện ra thông tin chi tiết sản phẩm từng Order:

	OrderDetailId	OrderId	Username	ProductId	Quantity	UnitPrice
▶	1	1	chinh	1	1	22.5
	2	2	chinh	1	1	22.5
*	3	3	chinh	2	1	15.95
	NULL	NULL	NULL	NULL	NULL	NULL

Tiếp theo đóng 2 cửa sổ Order và OrderDetails, mở lại Server Explorer -> chuột phải vào Database Wingtip Toys -> Close Connection :

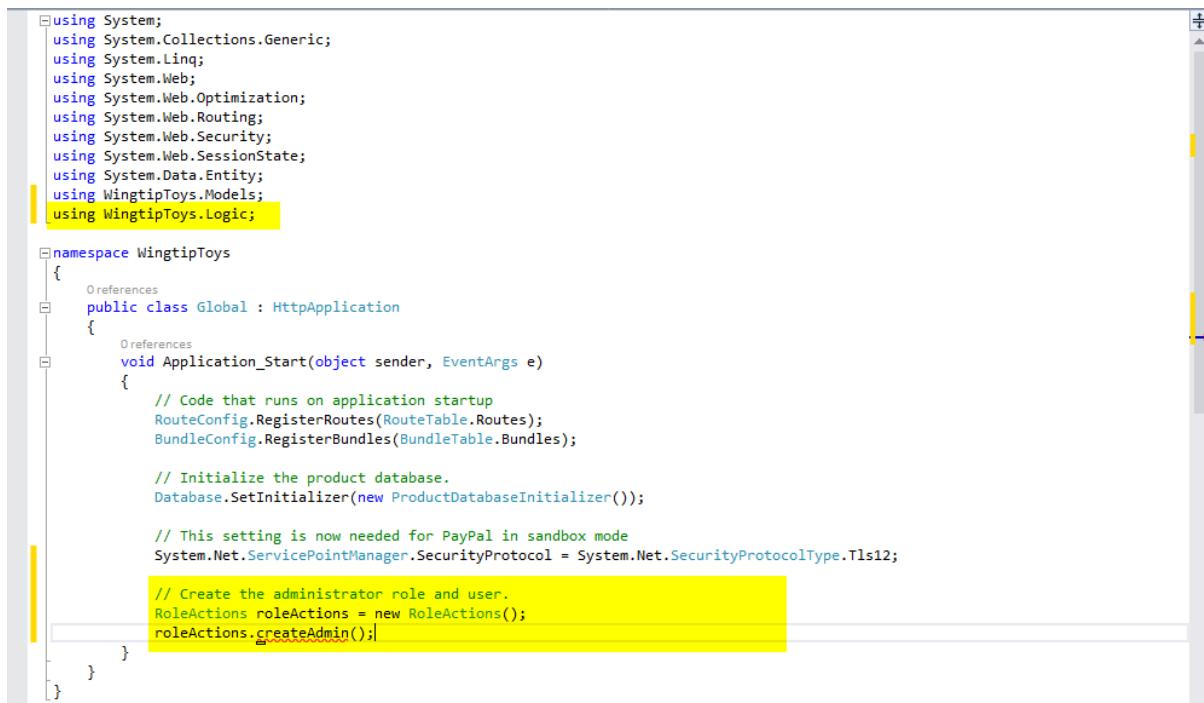


Bước 20: Trong Solution Explorer, tạo một class mới tên RoleActions.cs trong thư mục Logic. Thay đổi code mặc định bằng đoạn code bôi vàng như hình :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WingtipToys.Logic
{
    internal class RoleActions
    {
    }
}
```

Mở file Global.asax.cs và thêm các đoạn code được bôi vàng như hình vào :



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Optimization;
using System.Web.Routing;
using System.Web.Security;
using System.Web.SessionState;
using System.Data.Entity;
using WingtipToys.Models;
using WingtipToys.Logic;

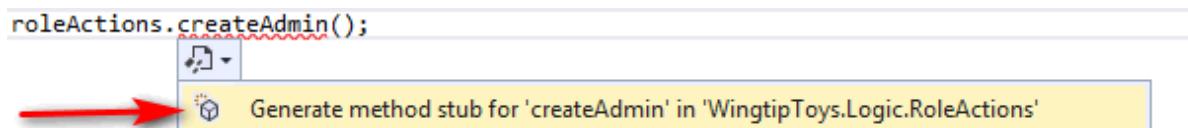
namespace WingtipToys
{
    public class Global : HttpApplication
    {
        void Application_Start(object sender, EventArgs e)
        {
            // Code that runs on application startup
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);

            // Initialize the product database.
            Database.SetInitializer(new ProductDatabaseInitializer());

            // This setting is now needed for PayPal in sandbox mode
            System.Net.ServicePointManager.SecurityProtocol = System.Net.SecurityProtocolType.Tls12;

            // Create the administrator role and user.
            RoleActions roleActions = new RoleActions();
            roleActions.createAdmin();
        }
    }
}
```

Trong hình ta thấy “ createAdmin() ” bị gạch chân màu đỏ khắc phục bằng cách di chuyển chuột ngay bên dưới chữ “c” và chọn như hình :



-Mở lại class RoleActions.cs trong thư mục Logic và thay tất cả code mặc định bằng đoạn code sau :

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using WingtipToys.Models;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;

namespace WingtipToys.Logic
{
    internal class RoleActions
    {
        internal void createAdmin()
        {
            // Access the application context and create result variables.
            Models.ApplicationDbContext context = new ApplicationDbContext();
            IdentityResult IdRoleResult;
            IdentityResult IdUserResult;

            // Create a RoleStore object by using the ApplicationDbContext object.
            // The RoleStore is only allowed to contain IdentityRole objects.
            var roleStore = new RoleStore<IdentityRole>(context);

            // Create a RoleManager object that is only allowed to contain
            IdentityRole objects.
            // When creating the RoleManager object, you pass in (as a parameter)
            a new RoleStore object.
            var roleMgr = new RoleManager<IdentityRole>(roleStore);

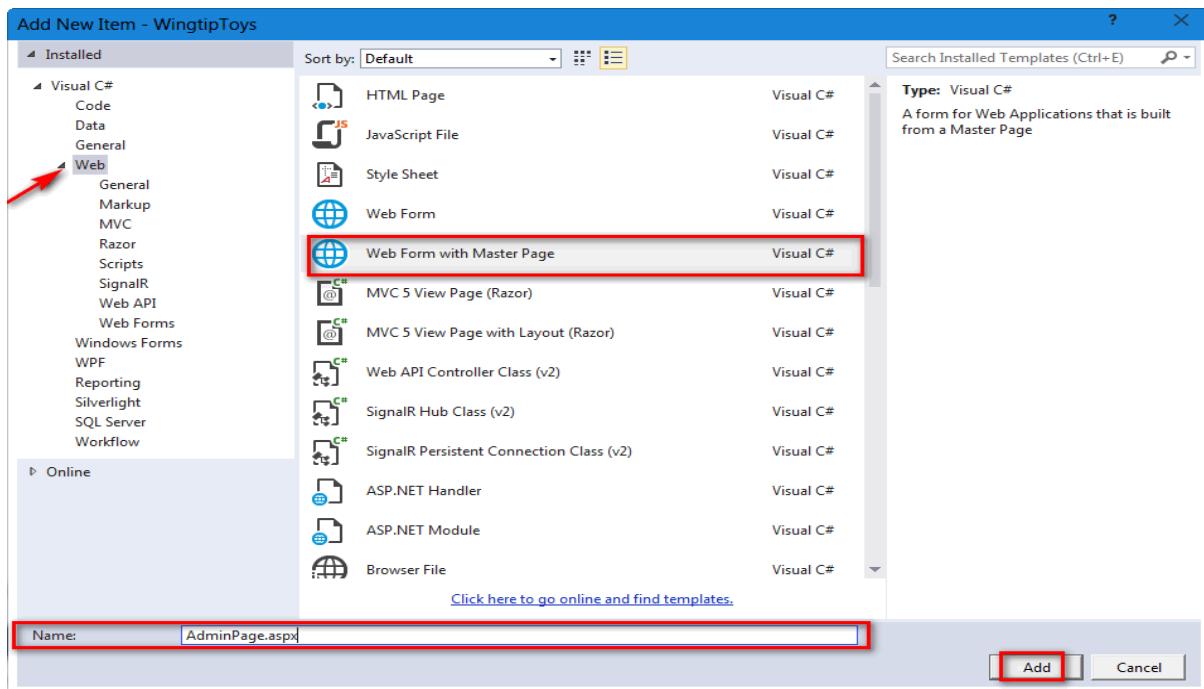
            // Then, you create the "Administrator" role if it doesn't already
            exist.
            if (!roleMgr.RoleExists("Administrator"))
            {
                IdRoleResult = roleMgr.Create(new
IdentityRole("Administrator"));
                if (!IdRoleResult.Succeeded)
                {
                    // Handle the error condition if there's a problem
                    creating the RoleManager object.
                }
            }

            // Create a UserManager object based on the UserStore object and the
            ApplicationDbContext
            // object. Note that you can create new objects and use them as
            parameters in
            // a single line of code, rather than using multiple lines of code, as
            you did
            // for the RoleManager object.

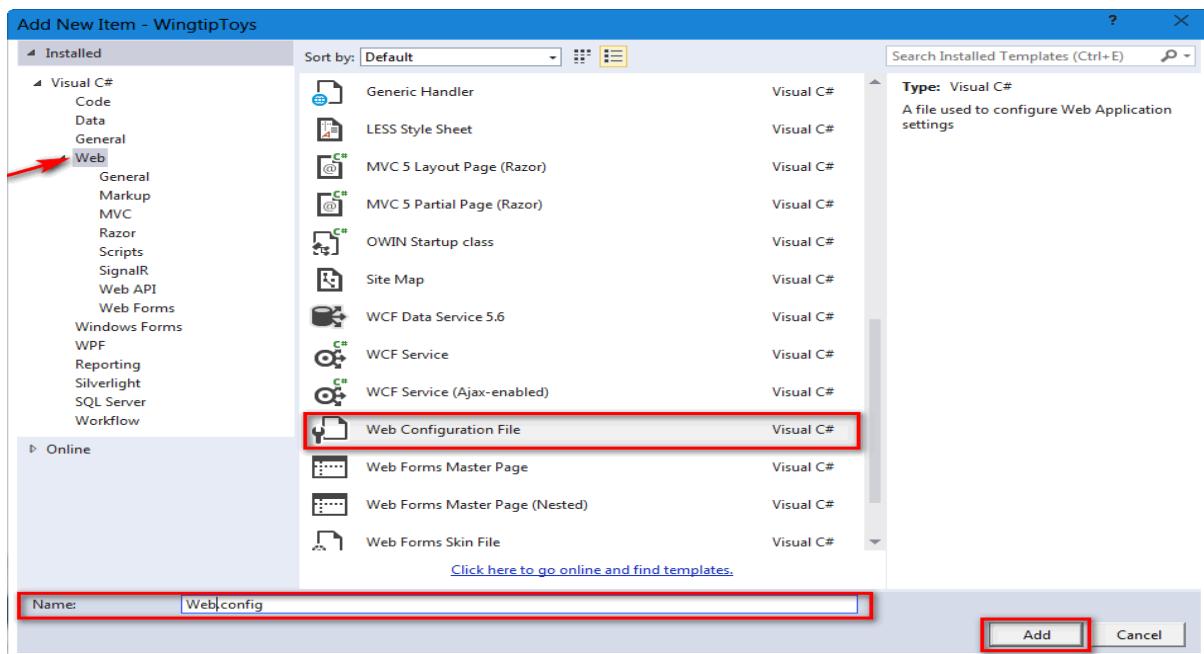
            var userMgr = new UserManager<ApplicationUser>(new
UserStore<ApplicationUser>(context));
            var appUser = new ApplicationUser()
            {
                UserName = "Admin",
            };
        }
    }
}

```

-Tiếp theo trong Solution Explorer -> chuột phải Wingtip Toys -> Add -> New Folder -> đặt tên Admin-> chuột phải thư mục Admin vừa tạo -> Add -> New Item -> Web -> Web Form with Master Page -> đặt tên AdminPage.aspx -> Add -> Site.Master -> OK :



-Tương tự chuột phải thư mục Admin -> Add -> New Item -> Web -> Web Configuration File -> đặt tên Web.config -> Add :



Thay code mặc định trong Web.config (của thư mục Admin) bằng đoạn code sau:

```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <authorization>
      <allow roles="Administrator"/>
      <deny users="*"/>
    </authorization>
  </system.web>
</configuration>
```

-Trong Solution Explorer, mở file Site.Master tìm và thêm đoạn code được bôi vàng như hình:

```
<ul class="nav navbar-nav">
    <li><a runat="server" id="adminLink" visible="false"
        href("~/Admin/AdminPage">Admin</a></li>
    <li><a runat="server" href="/">Home</a></li>
    <li><a runat="server" href "~/About">About</a></li>
    <li><a runat="server" href "~/Contact">Contact</a></li>
    <li><a runat="server" href "~/ProductList">Products</a></li>
    <li><a runat="server" href "~/ShoppingCart"
        ID="cartCount">&nbsp;</a></li>
</ul>
```

-Chuột phải Site.Master -> View Code . Thêm đoạn code được bôi vàng như hình vào sự kiện Page_Load :

```
protected void Page_Load(object sender, EventArgs e)
{
    if (HttpContext.Current.User.IsInRole("Administrator"))
    {
        adminLink.Visible = true;
    }
}
```

-Tiếp theo, mở lại trang AdminPage.aspx trong thư mục Admin và thay code mặc định bằng đoạn code theo đường dẫn: <https://drive.google.com/open?id=16ltmWAACuLr5x9QdUdnpX1SQEX5rk1o7>

-Chuột phải vào trang AdminPage.aspx -> View Code. Truy cập đường dẫn:

<https://drive.google.com/open?id=1x6bdfsBNBmxvd4VN80KXj8t25S4z1El> lấy code và thay code mặc định bằng code này.

-Trong Solution Explorer, chuột phải thư mục Logic -> Add -> New Item -> Code -> Class -> đặt tên: AddProducts.cs. Thay code mặc định bằng đoạn code sau :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using WingtipToys.Models;

namespace WingtipToys.Logic
{
    public class AddProducts
    {
        public bool AddProduct(string ProductName, string ProductDesc, string
ProductPrice, string ProductCategory, string ProductImagePath)
        {
            var myProduct = new Product();
            myProduct.ProductName = ProductName;
            myProduct.Description = ProductDesc;
            myProduct.UnitPrice = Convert.ToDouble(ProductPrice);
            myProduct.ImagePath = ProductImagePath;
            myProduct.CategoryID = Convert.ToInt32(ProductCategory);

            using (ProductContext _db = new ProductContext())
            {
                // Add product to DB.
                _db.Products.Add(myProduct);
                _db.SaveChanges();
            }
            // Success.
            return true;
        }
    }
}
```

Nhấn F5 chạy ứng dụng -> Login. Đăng nhập với User: Admin và Password: Pa\$\$word. Lúc này ta sẽ thấy xuất hiện như hình :



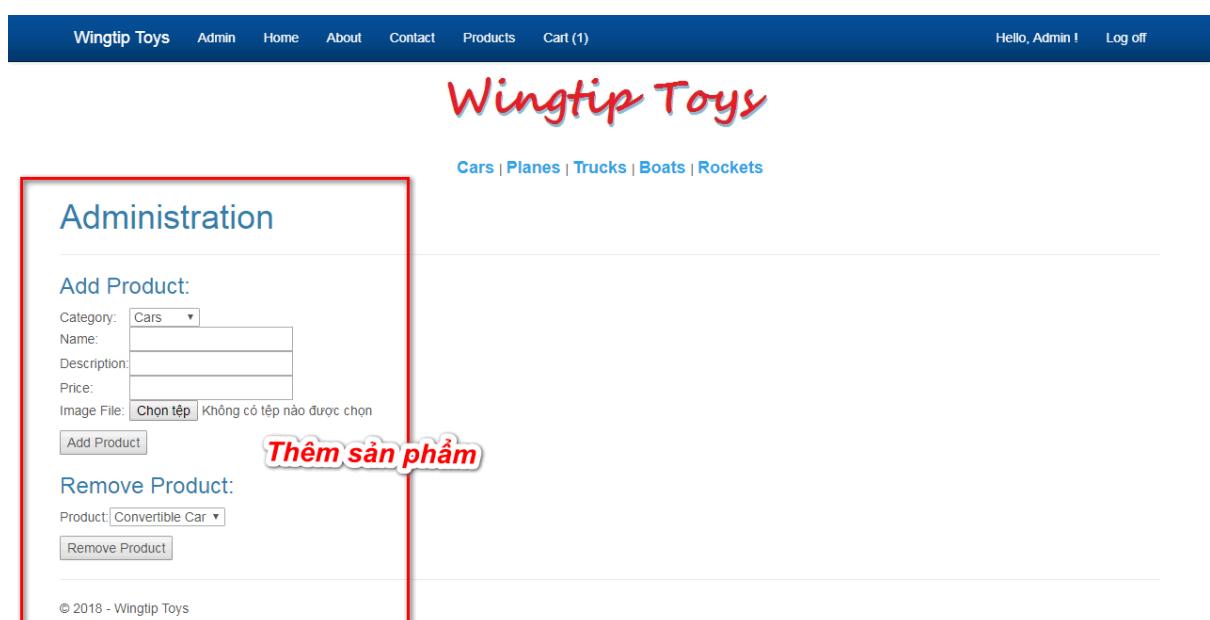
Welcome.

Wingtip Toys can help you find the perfect gift.

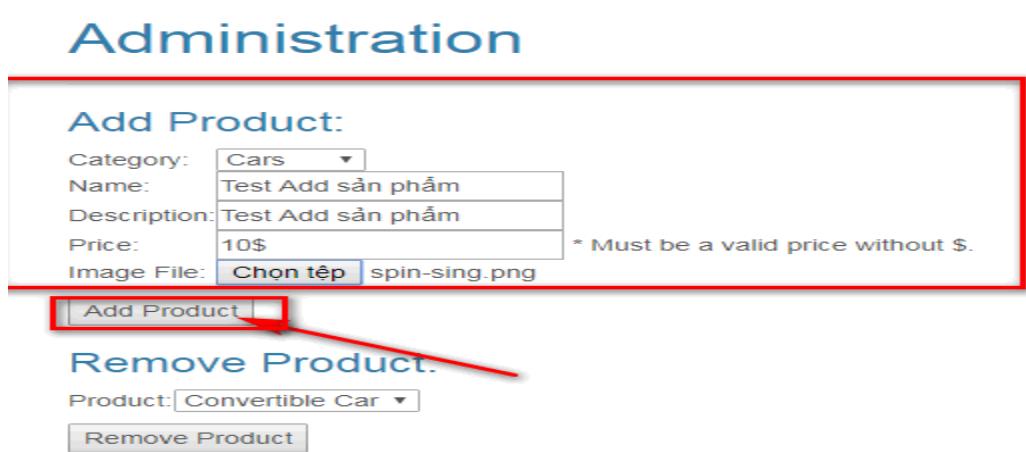
We're all about transportation toys. You can order any of our toys today. Each toy listing has detailed information to help you choose the right toy.

© 2018 - Wingtip Toys

Click vào Admin :



-Thử Add một sản phẩm :



Kết quả:

Screenshot of the Wingtip Toys website showing a list of products. A red box highlights the 'Old-time Car' product, which has been added to the cart. A red banner at the bottom of the page reads 'Sản phẩm thêm vào thành công' (Product added successfully).

Wingtip Toys

Cars | Planes | Trucks | Boats | Rockets

Products

Image	Name	Price	Action
	Convertible Car	\$22.50	Add To Cart
	Old-time Car	\$15.95	Add To Cart
	Fast Car	\$32.99	Add To Cart
	Super Fast Car	\$8.95	Add To Cart
	Old Style Racer	\$34.95	Add To Cart
Test Add sản phẩm Price: \$10.00 Add To Cart			

© 2018 - Wingtip Toys

-Thử xóa sản phẩm vừa thêm vào:

Administration

Add Product:

Category: Cars

Name:

Description:

Price:

Image File: Chọn tệp Không có tệp nào được chọn

Add Product

Chọn tên sản phẩm cần xóa

Remove Product:

Product: Test Add sản phẩm

Remove Product

Kết quả:

Screenshot of the Wingtip Toys website showing the product list after deletion. The 'Old-time Car' product is missing from the list, and a red box highlights the empty slot where it used to be. A red banner at the bottom of the page reads 'Sản phẩm đã xóa' (Product deleted).

Wingtip Toys

Cars | Planes | Trucks | Boats | Rockets

Products

Image	Name	Price	Action
	Convertible Car	\$22.50	Add To Cart
	Old-time Car	\$15.95	Add To Cart
	Fast Car	\$32.99	Add To Cart
	Super Fast Car	\$8.95	Add To Cart
	Old Style Racer	\$34.95	Add To Cart

© 2018 - Wingtip Toys

Bước 21: Trong Solution Explorer, mở file Global.asax.cs . Thêm đoạn code được bôi vàng như hình vào như sau:

```
{  
    // Code that runs on application startup  
    RouteConfig.RegisterRoutes(RouteTable.Routes);  
    BundleConfig.RegisterBundles(BundleTable.Bundles);  
  
    // Initialize the product database.  
    Database.SetInitializer(new ProductDatabaseInitializer());  
  
    // Create administrator role and user.  
    RoleActions roleActions = new RoleActions();  
    roleActions.createAdmin();  
  
    // Add Routes.  
    RegisterCustomRoutes(RouteTable.Routes);  
}  
  
void RegisterCustomRoutes(RouteCollection routes)  
{  
    routes.MapPageRoute(  
        "ProductsByCategoryRoute",  
        "Category/{categoryName}",  
        "~/ProductList.aspx"  
    );  
    routes.MapPageRoute(  
        "ProductByNameRoute",  
        "Product/{productName}",  
        "~/ProductDetails.aspx"  
    );  
}
```

-Tiếp tục mở Site.Master thay thế đoạn code mặc định bằng đoạn code được bôi vàng như hình :

```
<asp:ListView ID="categoryList"  
    ItemType="WingtipToys.Models.Category"  
    runat="server"  
    SelectMethod="GetCategories" >  
    <ItemTemplate>  
        <b style="font-size: large; font-style: normal">  
            <a href="<%#: GetRouteUrl("ProductsByCategoryRoute", new { categoryName  
= Item.CategoryName}) %>">  
                <%#: Item.CategoryName %>  
            </a>  
        </b>  
    </ItemTemplate>  
    <ItemSeparatorTemplate> | </ItemSeparatorTemplate>  
</asp:ListView>
```

-Mở trang ProductList.aspx thay thế đoạn code mặc định bằng đoạn code được bôi vàng như hình :

```

<ItemTemplate>
    <td runat="server">
        <table>
            <tr>
                <td>
                    <a href="<%#: GetRouteUrl("ProductNameRoute", new {productName = Item.ProductName}) %>">
                        <image src='/Catalog/Images/Thumbs/<%#: Item.ImagePath%>' width="100" height="75" border="1" />
                    </a>
                </td>
            </tr>
            <tr>
                <td>
                    <a href="<%#: GetRouteUrl("ProductNameRoute", new {productName = Item.ProductName}) %>">
                        <%#: Item.ProductName%>
                    </a>
                    <br />
                    <span>
                        <b>Price: </b><%: String.Format("{0:c}", Item.UnitPrice)%>
                    </span>
                    <br />
                    <a href="/AddToCart.aspx?productID=<%#: Item.ProductID %>">
                        <span class="ProductListItem">
                            <b>Add To Cart</b>
                        </span>
                    </a>
                </td>
            </tr>
            <tr>
                <td>&ampnbsp</td>
            </tr>
        </table>
    </td>
</ItemTemplate>

```

-Chuột phải ProductList.aspx -> View Code thêm đoạn code được bôi vàng như hình:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using WingtipToys.Models;
using System.Web.ModelBinding;
using System.Web.Routing;

```

-Cũng trong ProductList.aspx.cs thay thế đoạn code được bôi vàng như hình:

```

using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using WingtipToys.Models;
using System.Web.ModelBinding;
using System.Web.Routing;
}

namespace WingtipToys
{
    public partial class ProductList : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        public IQueryable<Product> GetProducts([QueryString("id")] int? categoryId, [RouteData] string categoryName)
        {
            var _db = new WingtipToys.Models.ProductContext();
            IQueryable<Product> query = _db.Products;
            if (categoryId.HasValue && categoryId > 0)
            {
                query = query.Where(p => p.CategoryID == categoryId);
            }
            if (!String.IsNullOrEmpty(categoryName))
            {
                query = query.Where(p =>
                    String.Compare(p.Category.CategoryName,
                    categoryName) == 0);
            }
            return query;
        }
    }
}

```

Bắt đầu đoạn code sau :

```

public IQueryable<Product> GetProducts(
[QueryString("id")] int? categoryId,
[RouteData] string categoryName)
{
    var _db = new WingtipToys.Models.ProductContext();
    IQueryable<Product> query = _db.Products;
    if (categoryId.HasValue && categoryId > 0)
    {
        query = query.Where(p => p.CategoryID == categoryId);
    }
    if (!String.IsNullOrEmpty(categoryName))
    {
        query = query.Where(p =>
            String.Compare(p.Category.CategoryName,
            categoryName) == 0);
    }
    return query;
}

```

-Tiếp theo chuột phải ProductDetails -> View Code. Thay thế đoạn code mặc định được bôi vàng như hình :

```

namespace WingtipToys
{
    public partial class ProductDetails : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        public IQueryable<Product> GetProduct([QueryString("ProductID")] int? productId, [RouteData] string productName)
        {
            var _db = new WingtipToys.Models.ProductContext();
            IQueryable<Product> query = _db.Products;
            if (productId.HasValue && productId > 0)
            {
                query = query.Where(p => p.ProductID == productId);
            }
            else if (!String.IsNullOrEmpty(productName))
            {
                query = query.Where(p =>
                    String.Compare(p.ProductName, productName) == 0);
            }
            else
            {
                query = null;
            }
            return query;
        }
    }
}

```

Bằng đoạn code sau :

```

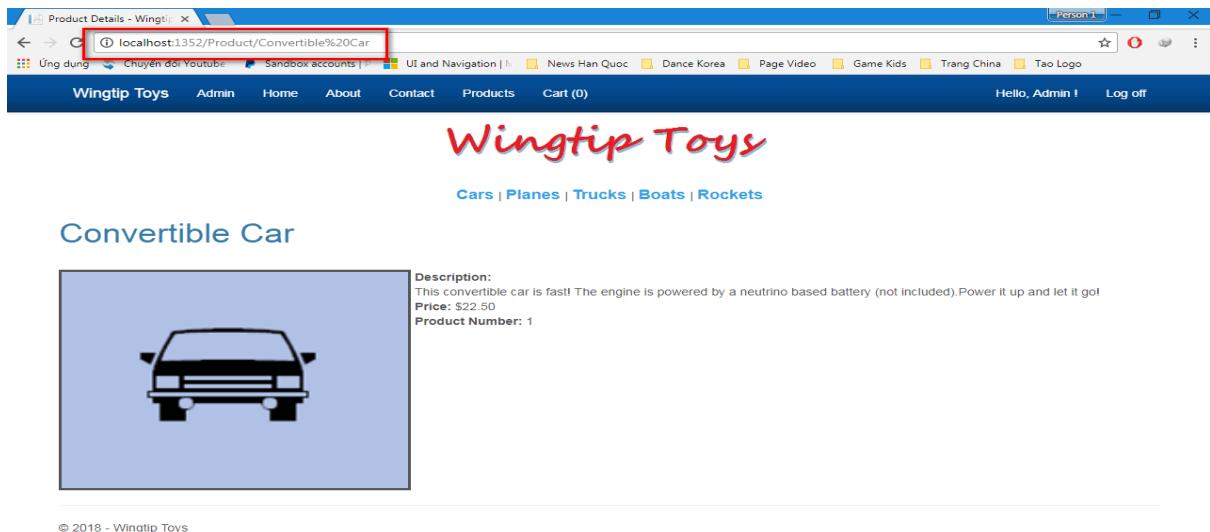
public IQueryable<Product> GetProduct(
[QueryString("ProductID")] int? productId,
[RouteData] string productName)
{
    var _db = new WingtipToys.Models.ProductContext();
    IQueryable<Product> query = _db.Products;
    if (productId.HasValue && productId > 0)
    {
        query = query.Where(p => p.ProductID == productId);
    }
    else if (!String.IsNullOrEmpty(productName))
    {
        query = query.Where(p =>
            String.Compare(p.ProductName, productName) == 0);
    }
    else
    {
        query = null;
    }
}

```

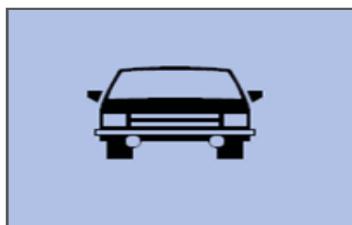
Nhấn F5 chạy ứng dụng -> Car -> Convertible Car lúc này ta sẽ thấy đường dẫn đã thay đổi từ:

Lúc đầu : <http://localhost:1352/ProductDetails?productId=1>

Bây giờ :



Convertible Car



Description:
This convertible car is fast! The engine is powered by a neutrino based battery (not included). Power it up and let it go!
Price: \$22.50
Product Number: 1

© 2018 - Wingtip Toys

Bước 22: Chuột phải thư mục Logic -> Add -> New Item -> Code -> Class -> đặt tên là :
ExceptionUtility.cs -> Add. Sau đó thay code mặc định bằng đoạn code sau :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.IO;
namespace WingtipToys.Logic
{
    // Create our own utility for exceptions
    public sealed class ExceptionUtility
    {
        // All methods are static, so this can be private
        private ExceptionUtility()
        {
        }
        // Log an Exception
        public static void LogException(Exception exc, string source)
        {
            // Include logic for logging exceptions
            // Get the absolute path to the log file
            string logFile = "App_Data/ErrorLog.txt";
            logFile = HttpContext.Current.Server.MapPath(logFile);
            // Open the log file for append and write the log
            StreamWriter sw = new StreamWriter(logFile, true);
            sw.WriteLine("***** {0} *****", DateTime.Now);
            if (exc.InnerException != null)
            {
                sw.WriteLine("Inner Exception Type: ");
                sw.WriteLine(exc.InnerException.GetType().ToString());
                sw.WriteLine("Inner Exception: ");
                sw.WriteLine(exc.InnerException.Message);
                sw.WriteLine("Inner Source: ");
                sw.WriteLine(exc.InnerException.Source);
                if (exc.InnerException.StackTrace != null)
                {
                    sw.WriteLine("Inner Stack Trace: ");
                    sw.WriteLine(exc.InnerException.StackTrace);
                }
            }
            sw.WriteLine("Exception Type: ");
            sw.WriteLine(exc.GetType().ToString());
            sw.WriteLine("Exception: " + exc.Message);
            sw.WriteLine("Source: " + source);
            sw.WriteLine("Stack Trace: ");
            if (exc.StackTrace != null)
            {
                sw.WriteLine(exc.StackTrace);
                sw.WriteLine();
            }
            sw.Close();
        }
    }
}
```

- Chuột phải Wingtip Toys -> Add -> New Item -> Web -> Web Form With Master Page -> đặt tên : ErrorPage.aspx -> Add -> Site.Master -> OK. Thay code mặc định bằng đoạn code sau:

```
<%@ Page Title="" Language="C#" AutoEventWireup="true"
MasterPageFile="~/Site.Master" CodeBehind="ErrorPage.aspx.cs"
Inherits="WingtipToys.ErrorPage" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
<h2>Error:</h2>
<p></p>
<asp:Label ID="FriendlyErrorMsg" runat="server" Text="Label" FontSize="Large"
style="color: red"></asp:Label>
<asp:Panel ID="DetailedErrorPanel" runat="server" Visible="false">
<p> </p>
<h4>Detailed Error:</h4>
<p>
<asp:Label ID="ErrorDetailedMsg" runat="server" Font-Size="Small"
/><br />
</p>
<h4>Error Handler:</h4>
<p>
<asp:Label ID="ErrorHandler" runat="server" Font-Size="Small" /><br
/>
</p>
<h4>Detailed Error Message:</h4>
<p>
<asp:Label ID="InnerMessage" runat="server" Font-Size="Small" /><br
/>
</p>
<p>
<asp:Label ID="InnerTrace" runat="server" />
</p>
</asp:Panel>
</asp:Content>
```

- Chuột phải ErrorPage.aspx -> View Code. Thay code mặc định bằng code sau :

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using WingtipToys.Logic;
namespace WingtipToys
{
    public partial class ErrorPage : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            // Create safe error messages.
            string generalErrorMsg = "A problem has occurred on this web site. Please try again. " + "If this error continues, please contact support.";
            string httpErrorMsg = "An HTTP error occurred. Page Not found. Please try again.";
            string unhandledErrorMsg = "The error was unhandled by application code.";
            // Display safe error message.
            FriendlyErrorMsg.Text = generalErrorMsg;
            // Determine where error was handled.
            string errorHandler = Request.QueryString["handler"];
            if (errorHandler == null)
            {
                errorHandler = "Error Page";
            }
            // Get the last error from the server.
            Exception ex = Server.GetLastError();
            // Get the error number passed as a querystring value.
            string errorMsg = Request.QueryString["msg"];
            if (errorMsg == "404")
            {
                ex = new HttpException(404, httpErrorMsg, ex);
                FriendlyErrorMsg.Text = ex.Message;
            }
            // If the exception no longer exists, create a generic exception.
            if (ex == null)
            {
                ex = new Exception(unhandledErrorMsg);
            }
            // Show error details to only you (developer). LOCAL ACCESS ONLY.
            if (Request.IsLocal)
            {
                // Detailed Error Message.
                ErrorDetailedMsg.Text = ex.Message;
                // Show where the error was handled.
                ErrorHandler.Text = errorHandler;
                // Show local access details.
                DetailedErrorPanel.Visible = true;
                if (ex.InnerException != null)
                {
                    InnerMessage.Text = ex.GetType().ToString() + "<br/>" +
                        ex.InnerException.Message;
                    InnerTrace.Text = ex.InnerException.StackTrace;
                }
                else
                {
                    InnerMessage.Text = ex.GetType().ToString();
                    if (ex.StackTrace != null)
                    {
                        InnerTrace.Text = ex.StackTrace.ToString().TrimStart();
                    }
                }
            }
            // Log the exception.
            ExceptionUtility.LogError(ex, errorHandler);
            // Clear the error from the server.
            Server.ClearError();
        }
    }
}

```

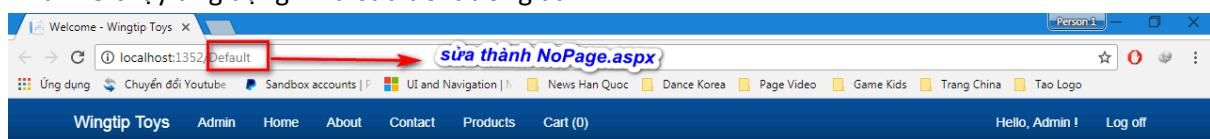
- Trong Solution Explorer, mở file Web.config. Thêm vào dưới <system.web> đoạn code sau :

```

<customErrors mode="On"
defaultRedirect="ErrorHandler.aspx?handler=customErrors%20section%20-%20Web.config">
<error statusCode="404"
redirect="ErrorHandler.aspx?msg=404&handler=customErrors%20section%20-
%20Web.config"/>

```

Nhấn F5 chạy ứng dụng. Thử sửa đổi đường dẫn



Wingtip Toys

Cars | Planes | Trucks | Boats | Rockets

Welcome.

Wingtip Toys can help you find the perfect gift.

We're all about transportation toys. You can order any of our toys today. Each toy listing has detailed information to help you choose the right toy.

© 2018 - Wingtip Toys

Kết quả :



Wingtip Toys

Cars | Planes | Trucks | Boats | Rockets

Error:

An HTTP error occurred. Page Not found. Please try again.

Detailed Error:

An HTTP error occurred. Page Not found. Please try again.

Error Handler:

customErrors section - Web.config

Detailed Error Message:

System.Web.HttpException

© 2018 - Wingtip Toys

-Chuột phải trang Default.aspx -> View Code. Thêm đoạn code được bôi vàng như sau :

```
protected void Page_Load(object sender, EventArgs e)
{
    throw new InvalidOperationException("An InvalidOperationException " +
        "occurred in the Page Load handler on the Default.aspx page.");
}
```

-Chạy trang ErrorPage.aspx kết quả :

The screenshot shows a web browser window with the title bar 'Wingtip Toys'. The address bar displays 'localhost:1352/ErrorPage'. The page content is as follows:

Wingtip Toys

Cars | Planes | Trucks | Boats | Rockets

Error:
A problem has occurred on this web site. Please try again. If this error continues, please contact support.

Detailed Error:
The error was unhandled by application code.

Error Handler:
Error Page

Detailed Error Message:
System.Exception

© 2018 - Wingtip Toys

-Chuột phải file Global.asax -> View Code. Thêm đoạn code được bôi vàng như hình :

```

    J
    1 reference
void RegisterCustomRoutes(RouteCollection routes)
{
    routes.MapPageRoute(
        "ProductsByCategoryRoute",
        "Category/{categoryName}",
        "~/ProductList.aspx"
    );
    routes.MapPageRoute(
        "ProductByNameRoute",
        "Product/{productName}",
        "~/ProductDetails.aspx"
    );
}

0 references
void Application_Error(object sender, EventArgs e)
{
    // Code that runs when an unhandled error occurs.
    // Get last error from the server
    Exception exc = Server.GetLastError();
    if (exc is HttpUnhandledException)
    {
        if (exc.InnerException != null)
        {
            exc = new Exception(exc.InnerException.Message);
            Server.Transfer("ErrorPage.aspx?handler=Application_Error%20-%20Global.asax", true);
        }
    }
}
}

```

- Chuột phải trang Default.aspx -> View Code. Thêm đoạn code được bôi vàng như hình:

```

using System.Web.UI.WebControls;

namespace WingtipToys
{
    public partial class _Default : Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            throw new InvalidOperationException("An InvalidOperationException " +
                "occurred in the Page_Load handler on the Default.aspx page.");
        }

        private void Page_Error(object sender, EventArgs e)
        {
            // Get last error from the server.
            Exception exc = Server.GetLastError();

            // Handle specific exception.
            if (exc is InvalidOperationException)
            {
                // Pass the error on to the error page.
                Server.Transfer("ErrorPage.aspx?handler=Page_Error%20-%20Default.aspx",
                    true);
            }
        }
    }
}

```

Và xóa đoạn code được bôi vàng như hình :

```

using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WingtipToys
{
    1 reference
    public partial class _Default : Page
    {
        0 references
        protected void Page_Load(object sender, EventArgs e)
        {
            throw new InvalidOperationException("An InvalidOperationException " + "occurred in the Page_Load handler on the Default.");
        }

        0 references
        private void Page_Error(object sender, EventArgs e)
        {
            // Get last error from the server.
            Exception exc = Server.GetLastError();
            // Handle specific exception.
            if (exc is InvalidOperationException)
            {
                // Pass the error on to the error page.
                Server.Transfer("ErrorPage.aspx?handler=Page_Error%20-%20Default.aspx",
                    true);
            }
        }
    }
}

```

-Tiếp theo mở class PayPalFunctions.cs trong thư mục Logic. Thêm đoạn code bôi vàng như hình;(Chú ý: Nhớ thêm “e” ngay chỗ Exception)

```

public string HttpCall(string NvpRequest)
{
    string url = pEndPointURL;

    string strPost = NvpRequest + "&" + buildCredentialsNVPString();
    strPost = strPost + "&BUTTONSOURCE=" + HttpUtility.UrlEncode(BNCode);

    HttpWebRequest objRequest = (HttpWebRequest)WebRequest.Create(url);
    objRequest.Timeout = Timeout;
    objRequest.Method = "POST";
    objRequest.ContentLength = strPost.Length;

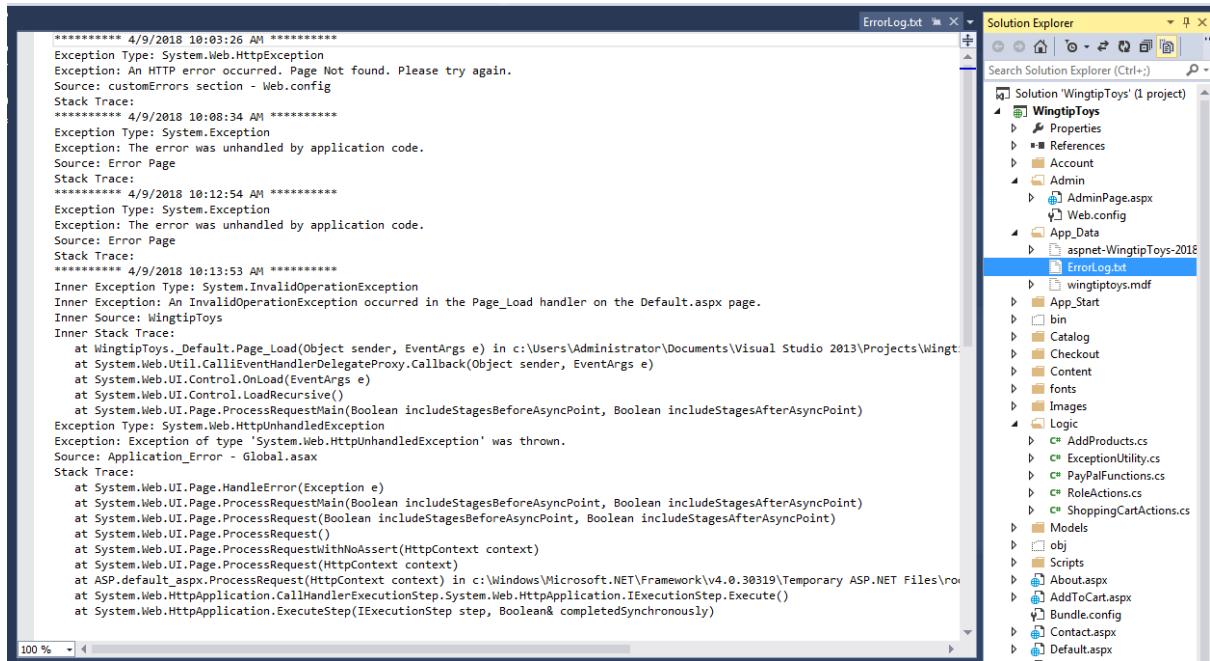
    try
    {
        using (StreamWriter myWriter = new
        StreamWriter(objRequest.GetRequestStream()))
        {
            myWriter.Write(strPost);
        }
    }
    catch (Exception e)
    {
        // Log the exception.
        WingtipToys.Logic.ExceptionUtility.LogException(e, "HttpCall in
PayPalFunction.cs");
    }

    //Retrieve the Response returned from the NVP API call to PayPal.
    HttpWebResponse objResponse = (HttpWebResponse)objRequest.GetResponse();
    string result;
    using (StreamReader sr = new StreamReader(objResponse.GetResponseStream()))
    {
        result = sr.ReadToEnd();
    }

    return result;
}

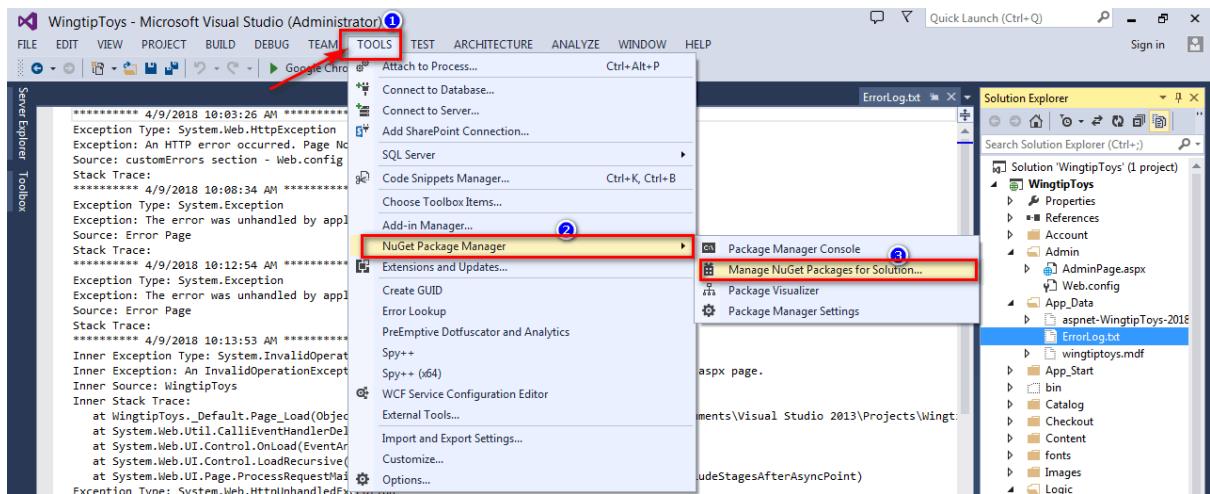
```

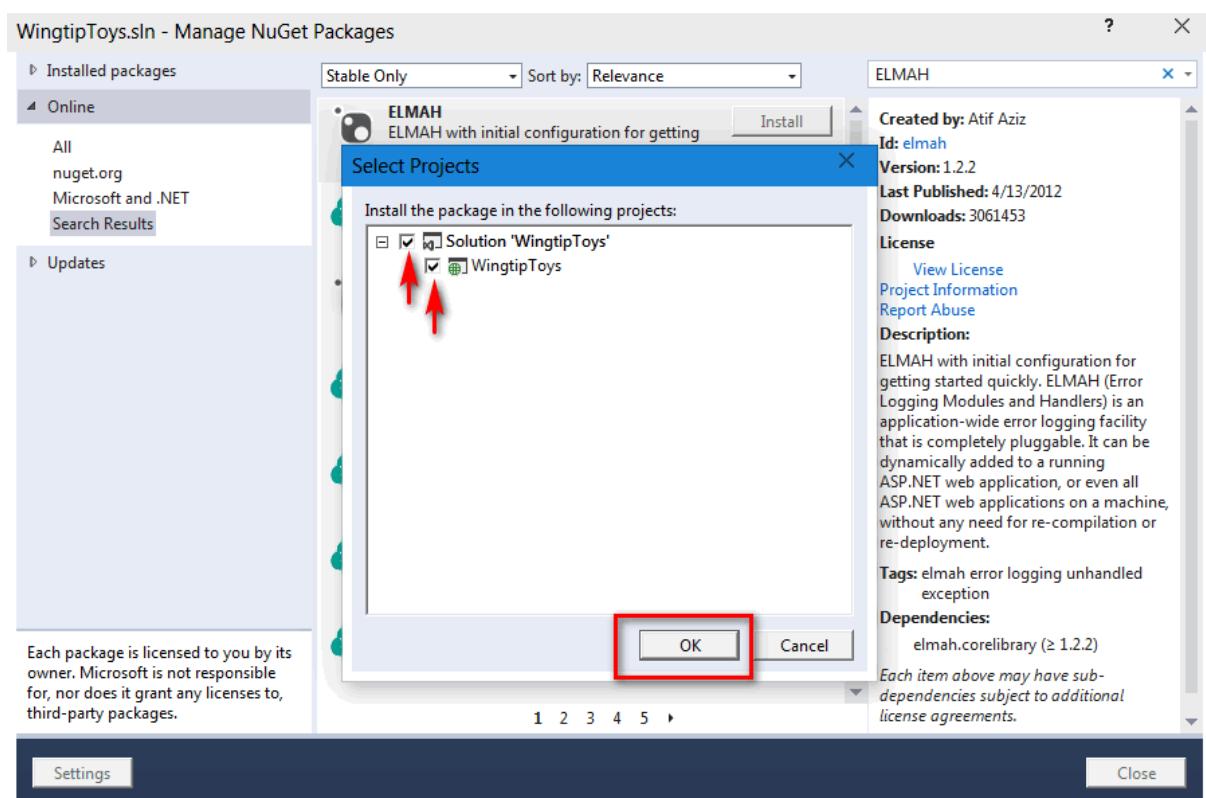
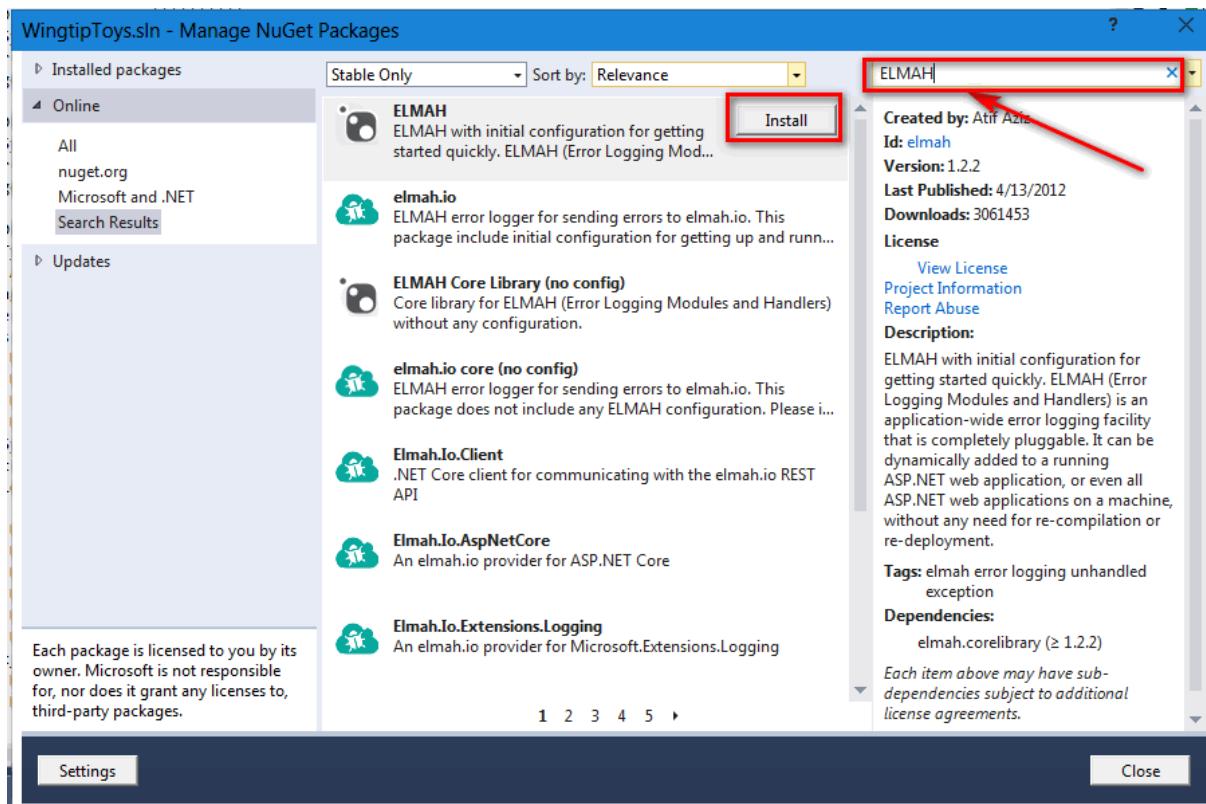
Trong Solution Explorer -> Refresh -> Show All Files. Sẽ thấy xuất hiện ErrorLog.txt trong thư mục App_Data:



```
***** 4/9/2018 10:03:26 AM *****
Exception Type: System.Web.HttpException
Exception: An HTTP error occurred. Page Not found. Please try again.
Source: customErrors section - Web.config
Stack Trace:
***** 4/9/2018 10:08:34 AM *****
Exception Type: System.Exception
Exception: The error was unhandled by application code.
Source: Error Page
Stack Trace:
***** 4/9/2018 10:12:54 AM *****
Exception Type: System.Exception
Exception: The error was unhandled by application code.
Source: Error Page
Stack Trace:
***** 4/9/2018 10:13:53 AM *****
Inner Exception Type: System.InvalidOperationException
Inner Exception: An InvalidOperationException occurred in the Page_Load handler on the Default.aspx page.
Inner Source: WingtipToys
Inner Stack Trace:
    at WingtipToys._Default.Page_Load(Object sender, EventArgs e) in c:\Users\Administrator\Documents\Visual Studio 2013\Projects\WingtipToys\WingtipToys\Default.aspx.cs:line 14
    at System.Web.Util.CallEventhandlerDelegateProxy.Callback(Object sender, EventArgs e)
    at System.Web.UI.Control.OnLoad(EventArgs e)
    at System.Web.UI.Control.LoadRecursive()
    at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
Exception Type: System.Web.HttpUnhandledException
Exception: Exception of type 'System.Web.HttpUnhandledException' was thrown.
Source: Application_Error - Global.asax
Stack Trace:
    at System.Web.UI.Page.HandleError(Exception e)
    at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
    at System.Web.UI.Page.ProcessRequest(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
    at System.Web.UI.Page.ProcessRequest()
    at System.Web.UI.Page.ProcessRequestWithNoAssert(HttpContext context)
    at System.Web.UI.Page.ProcessRequest(HttpContext context)
    at ASP.default_aspx.ProcessRequest(HttpContext context) in c:\Windows\Microsoft.NET\Framework\v4.0.30319\Temporary ASP.NET Files\root\1e0a73d\103f\ASP.default_aspx.cs:line 1
    at System.Web.HttpApplication.CallHandlerExecutionStep.System.Web.HttpApplication.IExecutionStep.Execute()
    at System.Web.HttpApplication.ExecuteStep(IExecutionStep step, Boolean& completedSynchronously)
```

Bước 23: Làm như hình:





Chạy ứng dụng và thử thay đổi đường dẫn:...../NoPage.aspx. Xem kết quả:



Error:

An HTTP error occurred. Page Not found. Please try again.

Detailed Error:

An HTTP error occurred. Page Not found. Please try again.

Error Handler:

customErrors section - Web.config

Detailed Error Message:

System.Web.HttpException

© 2018 - Wingtip Toys

Thay đoạn được đóng khung đỏ bằng elmah.axd . Xem kết quả:

