**Tạo trang login trong MVC**

**(Từ bài 3 sẽ sử dụng chung 1 Project ShopOnline)**

Bước 1. Tạo data base trong SQL server: Onlineshop

Bước 2. Tạo 1 Table: Account (Username: nchar(50), Password: nchar(50))

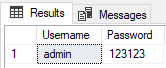
Tại sao là nchar mà ko phải là nvarchar?

N: Unicode

Char(50): nghĩa là khi nhập ít hơn thì tự động chèn thêm ký tự trắng cho đủ

INSERT INTO Account (Username, Password)

VALUES('admin','123123')



Bước 3. Tạo Store Procedure

CREATE PROCEDURE sp\_Account\_login

@Username nchar(50),

@Password nchar(50)

AS

BEGIN

DECLARE @count int

DECLARE @res bit

-- Insert statements for procedure here

SELECT @count = COUNT(1) FROM Account

WHERE Username = @Username AND Password = @Password

IF @count >0

SET @res=1

ELSE

SET @res=0

SELECT @res

END

GO

Đếm số bản ghi có @Username = trường Usename và @Password = trường Password.

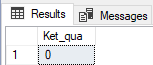
Test thử số bản ghi trong SQL:

declare @user nchar(50) = N'admin'

declare @pas nchar(50) = 123

select COUNT(1) as Ket\_qua from Account

where @user = Username and @pas = Password

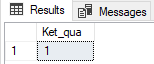


declare @user nchar(50) = N'admin'

declare @pas nchar(50) = 123123

select COUNT(1) as Ket\_qua from Account

where @user = Username and @pas = Password

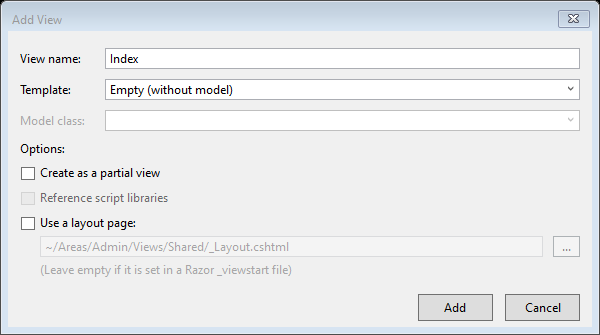


Tiếp theo xây dựng Template Login sử dụng boottrap có sẵn:

Các bước tạo, xem lại bài 04: (1)Tạo controller; (2)Từ Controller - Tạo View

1. Tạo controller:

* Ttrong thư mục Area - Admin - Add - Controller - MVC5 View Empty - Add - name = LoginController
* Add View trong Controller.cs - Addview - MVC5 View - Add - name = index - là trang Stand Alone không kế thừa hay làm trang chính cho bất kỳ Template nào - ADD



@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Index</title>

</head>

<body>

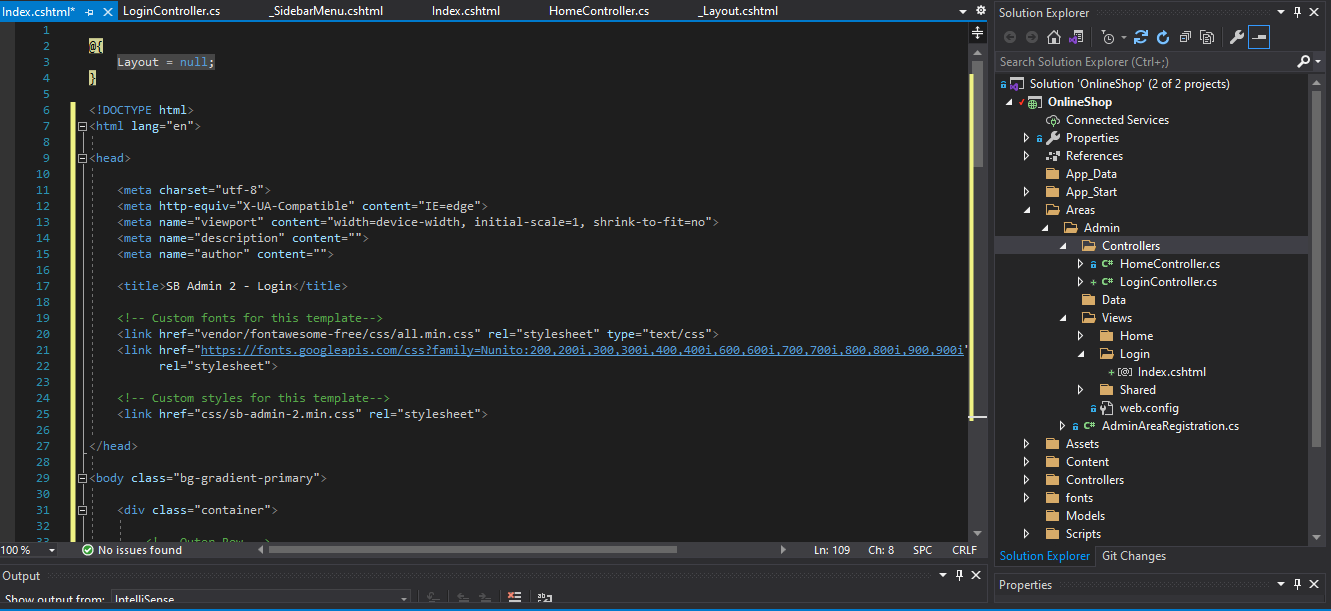
<div>

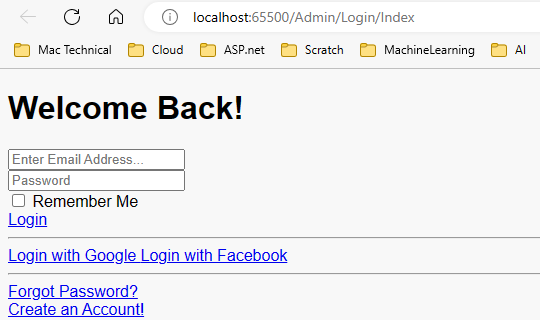
</div>

</body>

</html>

Thay thế nội dung <!DOCTYPE html> thành nội dung của trang Login mẫu





* Thiết lập đường dẫn cho các file css, js…

<!-- Custom styles for this template-->

<link href="~/Assets/admin/css/sb-admin-2.min.css" rel="stylesheet">

<!-- Bootstrap core JavaScript-->

<script src="~/Assets/admin//vendor/jquery/jquery.min.js"></script>

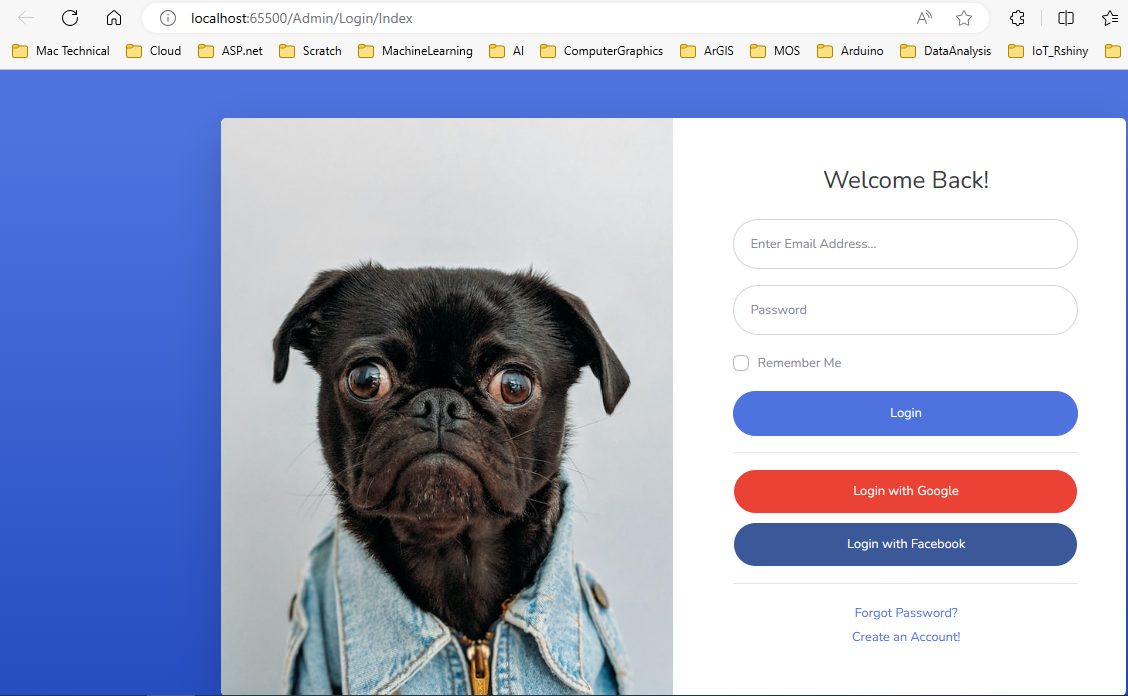
<script src="~/Assets/admin/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<!-- Core plugin JavaScript-->

<script src="~/Assets/admin/vendor/jquery-easing/jquery.easing.min.js"></script>

<!-- Custom scripts for all pages-->

<script src="~/Assets/admin/js/sb-admin-2.min.js"></script>



* Xét URL: <http://localhost:65500/Admin/Login/Index>

Admin: Vùng

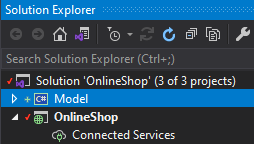
Login: Controller

Index: Method

1. Tạo VIEW

Tạo 1 tầng để truy cập dữ liệu, trong thực tế ta sẽ tạo 1 tâng riêng tách biệt với Model để phân biệt các model khác nhau giữa các chức năng.

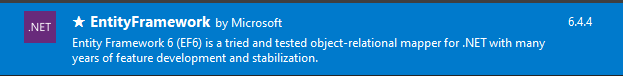
* Solution - Add New Project - Class Library (.Net Framework) - name = Model - Create



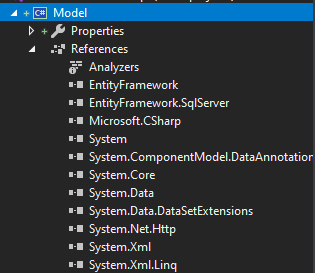
* Cài đặt Entity FrameWork: giúp tạo code từ Database

R\_click Model - Manage Nugets Package -

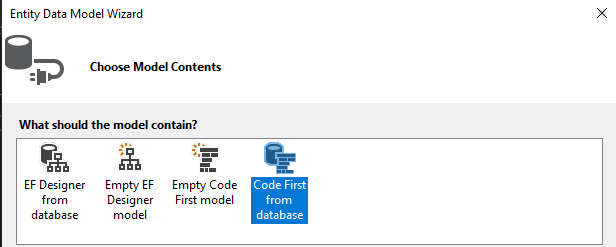
Browser: Entity Framework



Click 🡻 để tiến hành cài đặt

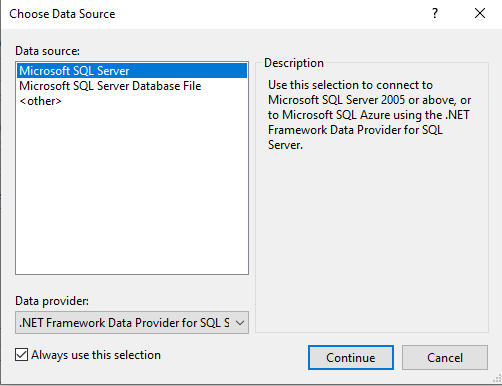


* Add Folder - Framework (trong folder Model Project vừa tạo)
* R\_click Framework - Add New Item - Data - ADO.NET Entity Data Model, name: OnlineShopDbContext (nên đặt theo cú pháp này) - mục đích là để truy cập DB chính cho toàn bộ Project - Add



Khi cài đặt Entity Framework sẽ có 2 chức năng như hình màu đỏ

* Chọn Code First from database từ 1 db có sẵn - Next
* New Connection…



* Continue

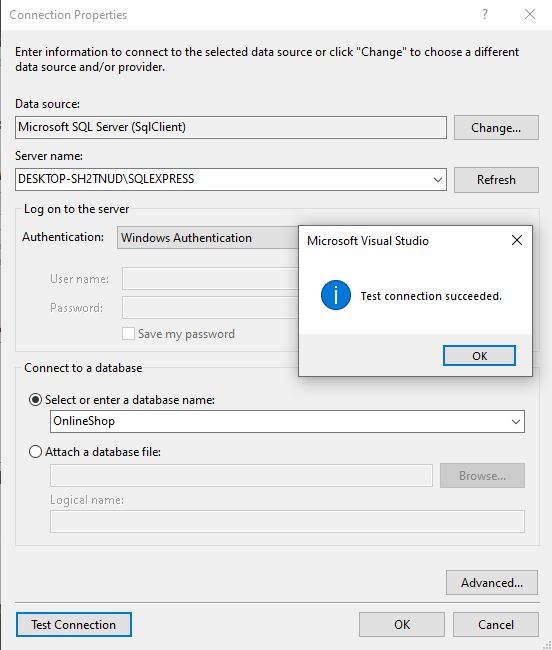
Name: DESKTOP-SH2TNUD\SQLEXPRESS

Authenticate: Windows Authentication

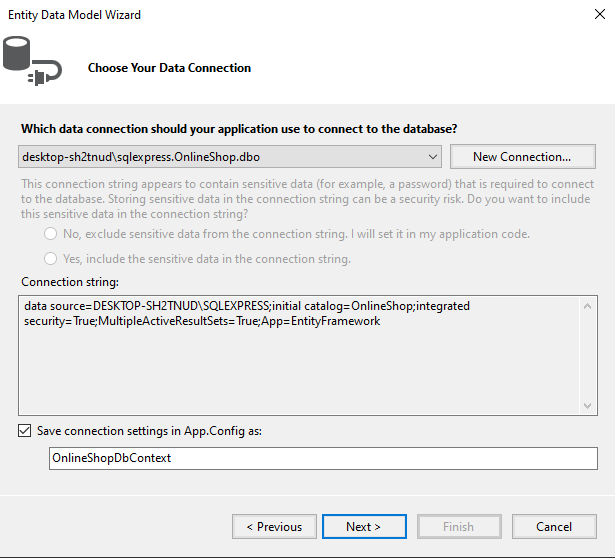
Database name: OnlineShop



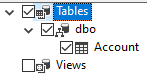
Cụ thể



* Next



* Next: chọn bảng kết nối



* Click 

1. Tạo lớp thao tác với từng bảng

R\_click Model vừa tạo - Add - New Item - Code - Class- AccountModel - Add

public class AccountModel

Tạo biến OnlineShopDbContext context

namespace Model

{

public class AccountModel

{

private OnlineShopDbContext context = null;

}

}

Tạo Contractor: AccountModel và khởi tạo biến context

namespace Model

{

public class AccountModel

{

private OnlineShopDbContext context = null;

public AccountModel()

{

context = new OnlineShopDbContext();

}

}

}

Tạo phương thức login với 2 tham số: username, pasword

public class AccountModel

{

public bool login(string username, string password)

{

}

}

Tạo phương thức login với 2 tham số: username, pasword và truyền tham số cho biến res

public bool Login(string username, string password)

{

object[] sqlParams =

{

new SqlParameter("@Username",username),

new SqlParameter("@Password",password),

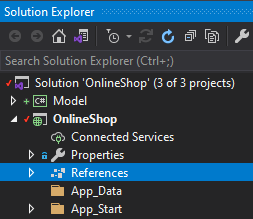
};

var res = context.Database.SqlQuery<bool>("sp\_Account\_Login @Username, @Password",sqlParams).SingleOrDefault();

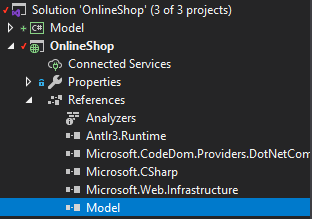
return res;

}

Gọi Model vừa tạo bằng cách Add vào Reference (của mục Online Shop)



R\_click OnlineShop\Reference\Add Reference…\Project - Chọn Model - OK



Viết các phương thức cho Controller Login trong Folder Admin

namespace OnlineShop.Areas.Admin.Controllers

{

public class LoginController : Controller

{

// GET: Admin/Login

[HttpGet]

public ActionResult Index()

{

return View();

}

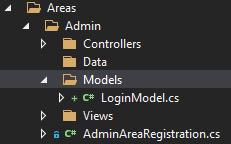
[HttpPost]

}

}

Thiết lập thuộc tính Action cho nó bằng cách tạo Login Model cho mục Admin: R\_click Areas\Admin\Models\Add New Item\C#\Code\Class\

Name= LoginModel.cs



Model này để truyền dữ liệu từ View và trả về View, cần 3 tham số

namespace OnlineShop.Areas.Admin.Models

{

public class LoginModel

{

[Required]

public string Username { set; get; }

public string Password { set; get; }

public bool RememberMe { set; get; }

}

}

Tiếp theo, viết các phương thức cho Controller Login trong Folder Admin trong trường hợp sai Username và PassWord

namespace OnlineShop.Areas.Admin.Controllers

{

public class LoginController : Controller

{

// GET: Admin/Login

[HttpGet]

public ActionResult Index()

{

return View();

}

[HttpPost]

public ActionResult Index(LoginModel model)

{

var result = new AccountModel().Login(model.Username, model.Password);

if (result && ModelState.IsValid)

{

}

else

{

ModelState.AddModelError("", "Tên đăng nhập và mật khẩu không đúng");

}

return View(model);

}

}

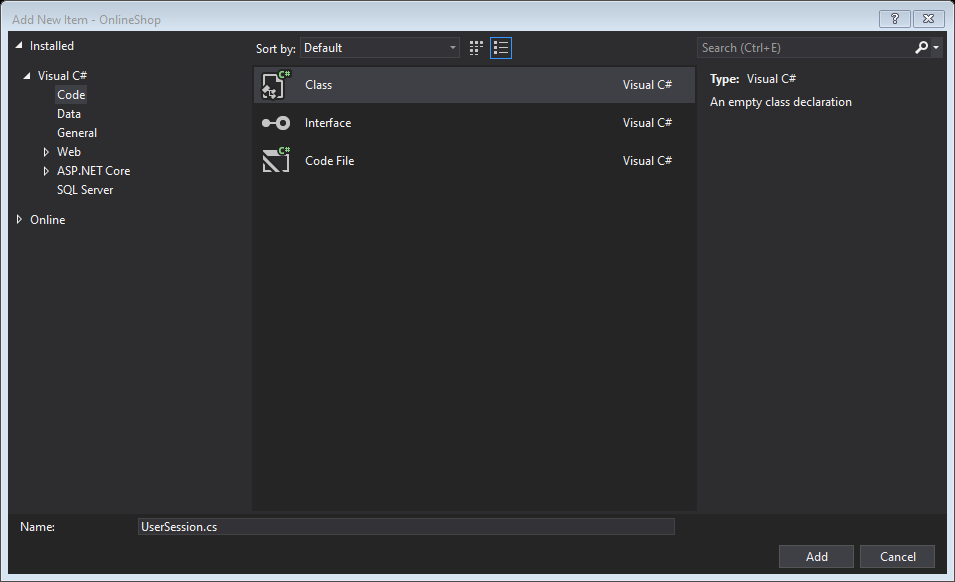
}

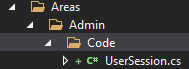
Trong trường hợp đúng mật khẩu và username, chúng ta cần xét session cho nó.

Tạo folder Code bên trong Admin



Tạo class UserSession cho thư mục Code có 1 hoặc nhiều thuộc tính để lưu các thông tin Session





Để lưu thông tin session, chúng ta cần có trường thông tin Serializable

namespace OnlineShop.Areas.Admin.Code

{

[Serializable]

public class UserSession

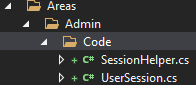
{

public string UserName { set; get; }

}

}

Tạo một lớp để set, get cho session đang tạo: R\_Click Code\Add New Item\C#\Code\Class\name = SessionHelper.cs



namespace OnlineShop.Areas.Admin.Code

{

public class SessionHelper

{

public static void SetSession(UserSession session)

{

HttpContext.Current.Session["loginSession"] = session;

}

public static UserSession GetSession()

{

var session = HttpContext.Current.Session["loginSession"];

if (session == null)

{

return null;

}

else{

return session as UserSession;

}

}

}

}

Tiếp theo hoàn thiện trường hợp login thành công cho class UserSession trả về trang Index của admin

public ActionResult Index(LoginModel model)

{

var result = new AccountModel().Login(model.Username, model.Password);

if (result && ModelState.IsValid)

{

SessionHelper.SetSession(new UserSession() { UserName = model.Username});

return RedirectToAction("Index","Home");

}

else

{

ModelState.AddModelError("", "Tên đăng nhập và mật khẩu không đúng");

}

return View(model);

}

Ngoài ra, chúng ta cần bổ sung Token cho Server và Client

public class LoginController : Controller

{

// GET: Admin/Login

[HttpGet]

public ActionResult Index()

{

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Index(LoginModel model)

{

var result = new AccountModel().Login(model.Username, model.Password);

if (result && ModelState.IsValid)

{

SessionHelper.SetSession(new UserSession() { UserName = model.Username});

return RedirectToAction("Index","Home");

}

else

{

ModelState.AddModelError("", "Tên đăng nhập và mật khẩu không đúng");

}

return View(model);

}

}

Thiết kế giao diện cho trang login