

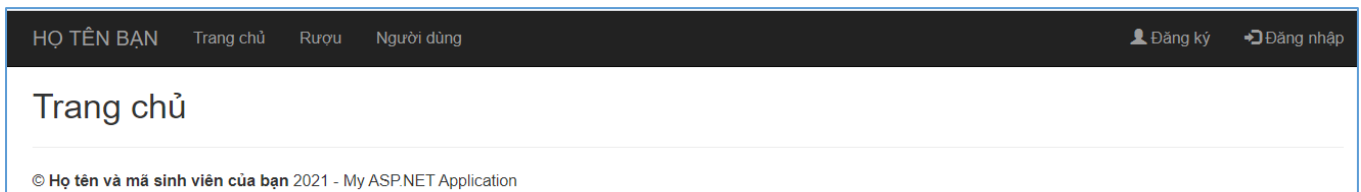
Bài tập 6

Lab 14.1 – Đăng nhập, đăng ký

- Tạo một project đặt tên là BaiTap14, chọn mẫu **MVC**.
- Sử dụng EF code first để kết nối với cơ sở dữ liệu **WineStore2** (đặt tên [ADO.NET](#) Entity Model là **WineDB2**) chỉ chọn bảng **nguoidung** và bảng **Product** trong cơ sở dữ liệu.
- Tạo folder wwwroot và copy folder images được cung cấp vào wwwroot.
- Sửa code trong lớp model **nguoidung.cs** để hiển thị tên cột là tiếng Việt, trong đó textbox mật khẩu sẽ ẩn thông tin nhập.
- Kích **ReBuild Solution** để build lại project (mỗi khi sửa model cần Rebuild lại hệ thống)
- Tạo một controller theo mẫu **MVC 5 Controller with view, using Entity Framework** gắn với model **nguoidung**.
- Sửa phần **ActionLink** trong **_Layout.cshtml** thêm các link cho đăng nhập và đăng ký

```
...  
<div class="navbar-collapse collapse">  
  <ul class="nav navbar-nav">  
    <li>@Html.ActionLink("Trang chủ", "Index", "Home")</li>  
    <li>@Html.ActionLink("Rượu", "Index", "ViewProduct")</li>  
    <li>@Html.ActionLink("Người dùng", "Index", "nguoidungs")</li>  
  </ul>  
  <ul class="nav navbar-nav navbar-right">  
    <li class="nav-item">  
      <a class="nav-link" href="@Url.Action("Register", "Home")"><span  
class="glyphicon glyphicon-user"></span> Đăng ký</a>  
    </li>  
    <li class="nav-item">  
      <a class="nav-link" href="@Url.Action("Login", "Home")"><span  
class="glyphicon glyphicon-log-in"></span> Đăng nhập</a>  
    </li>  
  </ul>  
</div>  
...
```

- Ấn F5 để được menu như hình



1. Đăng ký

- Tạo một view trong folder Home dùng template **Create** và model **nguoidung**, đặt tên view là **Register**. Sửa hiển thị nút lệnh thành tiếng Việt.

- Trong HomeController tạo các action method Register cho phần đăng ký (**Hướng dẫn: copy code của 2 action Create() GET và POST trong controllers nguoidungsController và sửa**) :

```
[HttpGet]
public ActionResult Register()
{
    return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Register([Bind(Include = "manguoidung,hoten,matkhau,email")]
nguoidung nguoidung)
{
    if (ModelState.IsValid)
    {
        db.nguoidungs.Add(nguoidung);
        db.SaveChanges();
        return RedirectToAction("Login");
    }
    return View(nguoidung);
}
```

- Ấn Ctrl+F5 chạy thử. Kích vào nút Đăng ký trên thanh menu.
- Nhập thông tin người dùng mới là: họ và tên bạn, mật khẩu và địa chỉ email của bạn kích vào nút “Đăng ký”. Sau đó kích vào Người dùng trên thanh menu để kiểm tra kết quả.

2. Đăng nhập

- Trong folder Home, tạo một view Login theo Template Empty chọn như sau:

- Trong file Login.cshtml nhập code như sau:

```
@model BaiTap15.Models.nguoidung

@{
    ViewBag.Title = "Login";
}

<h2>Đăng nhập</h2>
<div class="row justify-content-md-center">
    <div class="col-md-4">
        <div class="card">
            <div class="card-body">
                @using (Html.BeginForm("Login", "Home", FormMethod.Post))
```

```

        {
            @Html.AntiForgeryToken()
            <div class="form-group">
                @Html.LabelFor(m => m.email)
                @Html.TextBoxFor(m => m.email, "", new { @class = "form-
control", @placeholder = "Email" })

            </div>
            <div class="form-group">
                @Html.LabelFor(m => m.matkhau)
                @Html.PasswordFor(m => m.matkhau, new { @class = "form-
control", @placeholder = "Mật khẩu" })
            </div>
            <div class="form-group">
                <input type="submit" name="submit" class="btn btn-primary"
value="Đăng nhập" />
            </div>
        }
    </div>
</div>
</div>
<p>@ViewBag.error</p>

```

- Trong HomeController tạo các action method Login cho phần đăng nhập:

```

[HttpGet]
public ActionResult Login()
{
    return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Login(string email, string matkhau)
{
    if (ModelState.IsValid)
    {
        var user = db.nguoidungs.Where(u => u.email.Equals(email) &&
u.matkhau.Equals(matkhau)).ToList();
        if (user.Count() > 0)
        {
            //sử dụng Session: add Session
            Session["HoTen"] = user.FirstOrDefault().hoten;
            Session["Email"] = user.FirstOrDefault().email;
            Session["idUser"] = user.FirstOrDefault().manguoidung;
            return RedirectToAction("Index");
        }
        else
        {
            ViewBag.error = "Đăng nhập không thành công!";
        }
    }
    return View();
}

```

Ở đây nếu đăng nhập thành công thì thông tin đăng nhập của người dùng gồm Họ tên, Email và mã người dùng sẽ được lưu vào Session.

- Trong view _Layout.cshtml sửa code như sau:

```

...
<div class="navbar-collapse collapse">
    <ul class="nav navbar-nav">
        <li>@Html.ActionLink("Trang chủ", "Index", "Home")</li>
        <li>@Html.ActionLink("Rượu", "Index", "ViewProduct")</li>
        <li>@Html.ActionLink("Người dùng", "Index", "nguoidungs")</li>
    </ul>

```

```

        </ul>
        @if (Session["HoTen"] != null)
        {
            <ul class="nav navbar-nav navbar-right">
                <li><a>Xin chào, @Session["HoTen"] - @Session["Email"]</a></li>
                <li>@Html.ActionLink("Logout", "Logout", "Home",htmlAttributes: new
{ @style = "font-size:small;color:yellow" })</li>
            </ul>
        }
        else
        {
            <ul class="nav navbar-nav navbar-right">
                <li class="nav-item">
                    <a class="nav-link" href="@Url.Action("Register","Home")"><span
class="glyphicon glyphicon-user"></span> Đăng ký</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="@Url.Action("Login","Home")"><span
class="glyphicon glyphicon-log-in"></span> Đăng nhập</a>
                </li>
            </ul>
        }
    }
</div>
. . .

```

Khi người dùng đăng nhập thì sẽ hiển thị thông tin người dùng và nút Logout cùng với màn hình Trang chủ. Còn nếu chưa đăng nhập thì sẽ hiển thị nút Đăng ký và nút Đăng nhập và hiển thị màn hình Đăng nhập.

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.
 - Đăng nhập bằng tài khoản của bạn.

3. Đăng xuất

- Trong HomeController tạo action method Logout cho phần đăng xuất:

```

public ActionResult Logout()
{
    Session.Clear();//remove session
    return RedirectToAction("Login");
}

```

- Trong HomeController sửa action Index() để kiểm tra thông tin đăng nhập trong session

```

public ActionResult Index()
{
    //Kiểm tra thông tin đăng nhập trong session
    if (Session["idUser"] != null)
    {
        return View();
    }
    else
    {
        return RedirectToAction("Login");
    }
}

```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.

Lab 14.2 – Sử dụng Ajax

- Cài đặt thư viện Ajax: **Microsoft.jQuery.Unobtrusive.Ajax**

The screenshot shows the NuGet Package Manager interface. The search bar contains 'Ajax'. The results list includes 'Microsoft.jQuery.Unobtrusive.Ajax' by Microsoft, with 22.9M downloads and version v3.2.6. This package is highlighted with a red box. The right pane shows the details for this package, including its description, version, author, license, and dependencies.

Microsoft.jQuery.Unobtrusive.Ajax by Microsoft, 22.9M downloads v3.2.6
jQuery plugin that unobtrusively sets up jQuery Ajax.

Version: Latest stable 3.2.6 **Install**

Options

Description
jQuery plugin that unobtrusively sets up jQuery Ajax.

Version: 3.2.6
Author(s): Microsoft
License: View License
Date published: Saturday, October 6, 2018 (10/6/2018)
Project URL: <https://www.asp.net/>
Report Abuse: <https://www.nuget.org/packages/Microsoft.jQuery.Unobtrusive.Ajax/3.2.6/ReportAbuse>
Tags: jQuery, Unobtrusive, Ajax

Dependencies
Any, Version=v0.0
jQuery (>= 1.8.0)

Kết quả:

The screenshot shows the 'Scripts' folder in a file explorer. The files listed are: bootstrap.js, bootstrap.min.js, jquery-3.4.1.intellisense.js, jquery-3.4.1.js, jquery-3.4.1.min.js, jquery-3.4.1.min.map, jquery-3.4.1.slim.js, jquery-3.4.1.slim.min.js, jquery-3.4.1.slim.min.map, jquery.unobtrusive-ajax.js, jquery.unobtrusive-ajax.min.js, jquery.validate-vsdoc.js, jquery.validate.js, jquery.validate.min.js, jquery.validate.unobtrusive.js, jquery.validate.unobtrusive.min.js, and modernizr-2.8.3.js. The files 'jquery.unobtrusive-ajax.js' and 'jquery.unobtrusive-ajax.min.js' are highlighted with a red box.

- Trong file `_Layout.cshtml` thêm khai báo thư viện xử lý Ajax

```
...  
@Scripts.Render("~/bundles/jquery")  
@Scripts.Render("~/bundles/bootstrap")  
<script src="~/Scripts/jquery.unobtrusive-ajax.js"></script>  
<script src="~/Scripts/jquery.unobtrusive-ajax.min.js"></script>  
@RenderSection("scripts", required: false)  
</body>  
</html>
```

- Tạo một controller đặt tên là DemoAjaxController. Tạo một view Index cho action Index của DemoAjaxController. Tạo một menu Demo Ajax trên thanh menu để gọi controller DemoAjax.

HỌ TÊN BẠN

Trang chủ

Rượu

Người dùng

Demo Ajax

4. Dùng @Ajax.ActionLink

- Trong view **Index.cshtml** của folder **DemoAjax** nhập code như sau:

```
@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>
@* Sử dụng Ajax.ActionLink *@
@Ajax.ActionLink("Tiêu đề link", "TenActionAjaxLink", new AjaxOptions {
    HttpMethod = "GET", //phương thức gửi đi
    UpdateTargetId = "IdTheHienThi", //cập nhật vào thẻ div trên trang
    InsertionMode = InsertionMode.Replace, // phương thức cập nhật
    LoadingElementId = "IdAnhChoLoading" //Ảnh động chờ load
})

<div id="IdTheHienThi" >
    Hiển thị kết quả
</div>

<div id="IdAnhChoLoading" style="display:none">
    
</div>
```

- Trong controller DemoAjaxController, tạo một action để xử lý Ajax.ActionLink tên là **TenActionAjaxLink**

```
...
//Xử lý Ajax ActionLink
public ActionResult TenActionAjaxLink()
{
    System.Threading.Thread.Sleep(2000);
    return Content("Chạy Ajax Action Link");
}
```

Vì thao tác ví dụ đơn giản nên tốc độ thực hiện rất nhanh không thể nhìn thấy ảnh chờ loading vì vậy cần làm trễ hoạt động bằng đoạn code **System.Threading.Thread.Sleep(2000);** để có thể quan sát được ảnh chờ. Khi áp dụng trong ứng dụng thực tế thì cần bỏ đoạn code này.

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.

5. Dùng @Ajax.BeginForm

- Trong view **Index.cshtml** của folder **DemoAjax** sửa lại code như sau:

```
@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>
@* Sử dụng Ajax.BeginForm *@
@using (Ajax.BeginForm("TenActionAjaxForm", "DemoAjax", new AjaxOptions {
    HttpMethod = "POST", //phương thức gửi đi
    UpdateTargetId = "IdTheHienThi", //cập nhật vào thẻ div trên trang
    InsertionMode = InsertionMode.Replace, // phương thức cập nhật
    LoadingElementId = "IdAnhChoLoading" // Ảnh động chờ load
}))
{
    @Html.TextBox("txtName")
    <input type="submit" value="Submit" />
}
```

```

<div id="IdTheHienThi" >
    Hiển thị kết quả
</div>

<div id="IdAnhChoLoading" style="display:none">
    
</div>

```

- Trong controller DemoAjaxController, tạo một action để xử lý Ajax.BeginForm tên là **TenActionAjaxForm**

```

...
//Xử lý Ajax Begin Form
public ActionResult TenActionAjaxForm(FormCollection f)
{
    System.Threading.Thread.Sleep(2000);
    string KetQua = f["txtName"].ToString();
    return Content(KetQua);
}

```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.

6. Dùng @Ajax.BeginForm

- Trong view **Index.cshtml** của folder **DemoAjax** sửa lại code như sau:

```

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>
@* Sử dụng Ajax.BeginForm *@
@using (Ajax.BeginForm("TenActionAjaxForm", "DemoAjax", new AjaxOptions {
    HttpMethod = "POST", //phương thức gửi đi
    UpdateTargetId = "IdTheHienThi", //cập nhật vào thẻ div trên trang
    InsertionMode = InsertionMode.Replace, // phương thức cập nhật
    LoadingElementId = "IdAnhChoLoading" // Ảnh động chờ load
}))
{
    @Html.TextBox("txtName")
    <input type="submit" value="Submit" />
}

<div id="IdTheHienThi" >
    Hiển thị kết quả
</div>

<div id="IdAnhChoLoading" style="display:none">
    
</div>

```

- Trong controller DemoAjaxController, tạo một action để xử lý Ajax.BeginForm tên là **TenActionAjaxForm**

```

...
//Xử lý Ajax Begin Form
public ActionResult TenActionAjaxForm(FormCollection f)
{
    System.Threading.Thread.Sleep(2000);
    string KetQua = f["txtName"].ToString();
    return Content(KetQua);
}

```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.

7. Dùng Ajax JQuery

- Trong view **Index.cshtml** của folder **DemoAjax** sửa lại code như sau:

```
@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<h4>Số a: <input type="text" id="txt1" /></h4>
<h4>Số b: <input type="text" id="txt2" /></h4>
<br />
<input type="button" value="Tính" id="IdNutLenh" />

<div id="IdTheHienThi">
    Hiển thị kết quả
</div>

<div id="IdAnhChoLoading" style="display:none">
    
</div>

@* Sử dụng Ajax JQuery *@
@section scripts {
    <script>
        $(document).ready(function () {
            $("#IdNutLenh").click(function () {
                $('#IdAnhChoLoading').show();
                var n1 = $("#txt1").val();
                var n2 = $("#txt2").val();
                $.ajax({
                    type: 'GET',
                    data: { a: n1, b: n2 },
                    url: '/DemoAjax/TenActionAjaxJQuery',
                    success: function (ketqua) {
                        if (ketqua.status == "ok") {
                            $('#IdAnhChoLoading').hide();
                            $('#IdTheHienThi').html(ketqua.result);
                        }
                    }
                });
            });
        });
    </script>
}
```

- Trong controller **DemoAjaxController**, tạo một action để xử lý Ajax JQuery tên là **TenActionAjaxJQuery**

```
...
//Xử lý Ajax JQuery
public ActionResult TenActionAjaxJQuery(int a, int b)
{
    System.Threading.Thread.Sleep(2000);
    var x = a + b;
    var result = a + " + " + b + " = " + x;
    var data = new { status = "ok", result = result };
    return Json(data, JsonRequestBehavior.AllowGet);
}
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.

8. Dùng Ajax trong tìm kiếm

- Tạo một controller theo mẫu **MVC 5 Controller Empty** đặt tên là **ViewProductController**.
- Tạo View cho Action Index trong **ViewProductController** theo template sau:

Add View

View name:

Template:

Model class:

Data context class:

Options:

☐ Create as a partial view

☐ Reference script libraries

☒ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

Add Cancel

- Sửa lại view Index.cshtml trong folder **Product** như sau:

```
@{
    ViewBag.Title = "Index";
}
<h2>Index</h2>
<p>
    @using (Ajax.BeginForm("SearchAjax", "ViewProduct", new AjaxOptions
    {
        HttpMethod = "POST", //phương thức gửi đi
        UpdateTargetId = "dsRuou", //cập nhật vào thẻ div trên trang
        InsertionMode = InsertionMode.Replace, // phương thức cập nhật
        LoadingElementId = "IdAnhChoLoading" // Ảnh động chờ load
    }))
    {
        @:Tìm mặt hàng: @Html.TextBox("SearchString")
        <input type="submit" value="Tìm" />
    }
</p>
<div id="IdAnhChoLoading" style="display:none">
    
</div>
<div id="dsRuou">
    @{
        Html.RenderAction("_SanPham", "ViewProduct");
    }
</div>
```

- Tạo một PartialView đặt tên là _SanPham theo template sau:

Add View

View name:

Template:

Model class:

Data context class:

Options:

☒ Create as a partial view

☐ Reference script libraries

☒ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

Add Cancel

- Trong **_SanPham.cshtml** xóa hết phần code của các nút lệnh Create, Edit, Details và Delete đi.
- Trong **ViewProductController** thêm một action `ActionResult _SanPham()` cho PartialView

```
namespace BaiTap14.Controllers
{
    public class ViewProductController : Controller
    {
        private WineDB2 db = new WineDB2();
        // GET: ViewProduct
        public ActionResult Index()
        {
            return View();
        }

        [ChildActionOnly]
        public ActionResult _SanPham()
        {
            var hangs = db.Products.Select(p=>p);
            return PartialView(hangs);
        }
    }
}
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.
- Trong **ViewProductController**, tạo Action **SearchAjax** xử lý việc tìm kiếm:

```
...
public ActionResult SearchAjax(string searchString)
{
    var hangs = db.Products.Select(p=>p);
    //Lọc theo tên hàng
    if (!String.IsNullOrEmpty(searchString))
    {
        hangs = hangs.Where(p => p.ProductName.Contains(searchString));
    }
    return PartialView("_SanPham", hangs);
}
```

- Sửa lại view **Index.cshtml** phần hiển thị sản phẩm như sau:

```
...
<div id="dsRuou">
    @if (ViewBag.Begin == 1)
    {
        Html.RenderAction("_SanPham", "ViewProduct");
        ViewBag.Begin = 0;
    }
</div>
```

- Sửa lại action **Index** trong **ViewProductController**

```
...
public ActionResult Index()
{
    ViewBag.Begin = 1;
    return View();
}
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.