



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

VNUHCM - UIT

LẬP TRÌNH WEB

CHƯƠNG 7 SỬ DỤNG IDENTITY

NỘI DUNG

1

- Tạo project sử dụng ASP.NET Identity

2

- Cấu trúc tổ chức thư mục của ASP.NET Identity

3

- Chức năng đăng ký thành viên (Register)

4

- Chức năng đăng nhập (Login)

5

- Chức năng đăng xuất (Log off)

6

- Sự hiển thị của menu

7

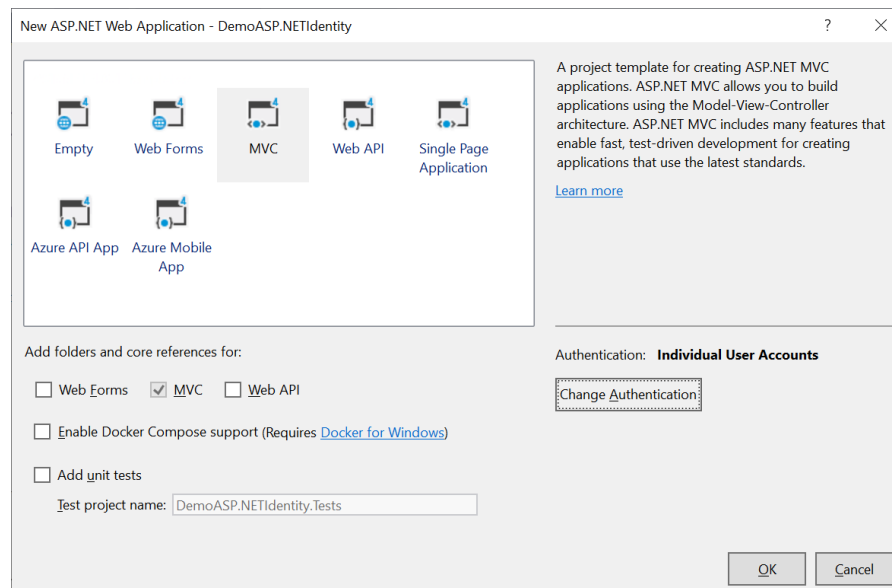
- Xác thực email khi đăng ký thành viên

Giới thiệu

- ❖ **ASP.NET Identity** là 1 công nghệ ra đời sau **ASP.NET Membership**.
- ❖ **ASP.NET Identity** khắc phục một số yếu điểm của **ASP.NET Membership** và bổ sung thêm nhiều tính năng hấp dẫn để bắt kịp xu thế như hỗ trợ đăng nhập qua mạng xã hội, đăng nhập thông qua 2 bước, gửi mail xác nhận khi đăng ký,...

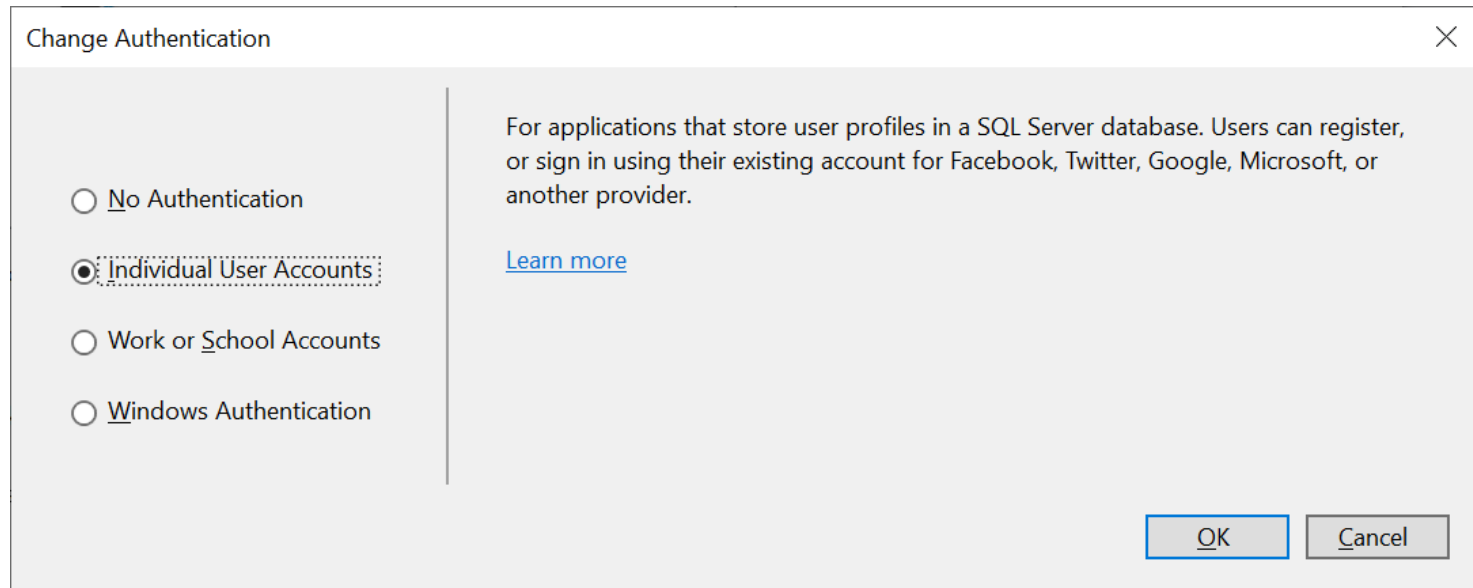
1. Tạo project sử dụng ASP.NET Identity

- ❖ ASP.NET Identity hỗ trợ từ ASP.NET MVC5 trở lên.
- ❖ **Bước 1:** Tạo mới project ASP.NET Web Application.
- ❖ **Bước 2:** Trong khung New ASP.NET Project hiện ra, chọn MVC và check vào checkbox MVC.



1. Tạo project sử dụng ASP.NET Identity

- ❖ **Bước 3:** Ấn nút **Change Authentication** và chọn **Individual User Account** để Visual Studio tích hợp sẵn code của **ASP.NET Identity** trong project **ASP.NET MVC** được tạo.



2. Cấu trúc tổ chức thư mục của ASP.NET Identity

- ❖ **App_Start/IdentityConfig.cs:** chứa các lệnh để cấu hình ASP.NET Identity
- ❖ **Controller/AccountController:** controller chứa các action method có tác dụng xác thực người dùng như Login, Register, ForgotPassword,...
- ❖ **Controller/ManageController:** controller chứa các action method có tác dụng quản lý user (khi user đã login vào web) như ChangePassword, SetPassword,...
- ❖ **Model/AccountViewModels:** chứa các View Model hiển thị trong các view của AccountController.

2. Cấu trúc tổ chức thư mục của ASP.NET Identity

- ❖ **Model/ManageViewModels**: chứa các View Model hiển thị trong các view của ManageController
- ❖ **Model/IdentityModels**: chứa class **ApplicationUser** để quản lý thông tin user và class **ApplicationDbContext** để quản lý kết nối với database ở dạng Entity Framework Code First để có thể bổ sung thêm các field cho user hoặc loại bỏ bớt các field mà bạn không cần thiết một cách dễ dàng và ít lỗi.

3. Chức năng đăng ký thành viên (Register)

- ❖ Lưu ý cấu hình chuỗi kết nối (connectionString) trong file **Web.config** ở thư mục gốc của project.
- ❖ Khi đăng ký thành viên thành công thì Visual Studio sẽ tự tạo mới một database (nếu database chưa tồn tại) chứa các table của ASP.NET Identity hoặc add thêm các table của ASP.NET Identity vào database (nếu database đó đã tồn tại) trong SQL Server.

3. Chức năng đăng ký thành viên (Register)

```
<connectionStrings>
  <add name="DefaultConnection"
        connectionString="Data Source=DESKTOP-M0S38PN;
        Initial Catalog=QLSINHVIEN;Integrated Security=True"
        providerName="System.Data.SqlClient" />
</connectionStrings>
```

❖ Bây giờ chúng ta tiến hành dùng chức năng đăng ký thành viên (register) trong ASP.NET Identity.

3. Chức năng đăng ký thành viên (Register)

❖ **Bước 1:** Chạy website và click vào link Register ở góc phải menu.

Application name Home About Contact

Register.

Create a new account.

Email

Password

Confirm password

© 2023 - My ASP.NET Application

3. Chức năng đăng ký thành viên (Register)

❖ Bước 2: Nhập thông tin để đăng ký tài khoản.

Application name Home About Contact

Register.

Create a new account.

- The Email field is not a valid e-mail address.

Email

dungta.uit

Password

.....

Confirm password

.....

Register

© 2023 - My ASP.NET Application

3. Chức năng đăng ký thành viên (Register)

Register - My ASP.NET Application x +

localhost:62029/Account/Register

Anh Dung Diep Nhu Thẻ mới Download Ebook Bấ... Bí Quyết Giàu Có V... Download and Inst...

Application name Home About Contact Register Log in

Register.

Create a new account.

- Passwords must have at least one non letter or digit character. Passwords must have at least one lowercase ('a'-'z'). Passwords must have at least one uppercase ('A'-'Z').

Email

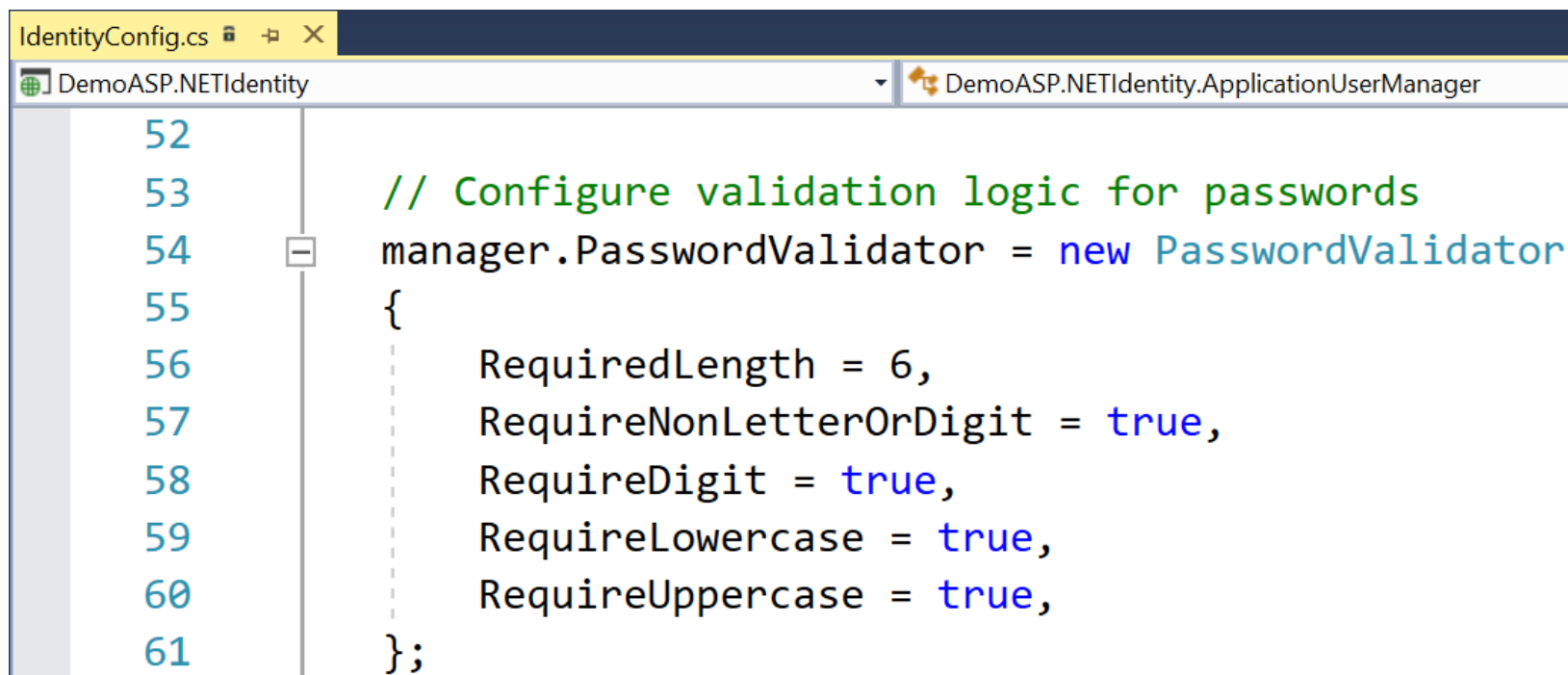
Password

Confirm password

© 2023 - My ASP.NET Application

3. Chức năng đăng ký thành viên (Register)

- ❖ ASP.NET Identity mặc định thiết lập rằng buộc password rất "chặt chẽ".

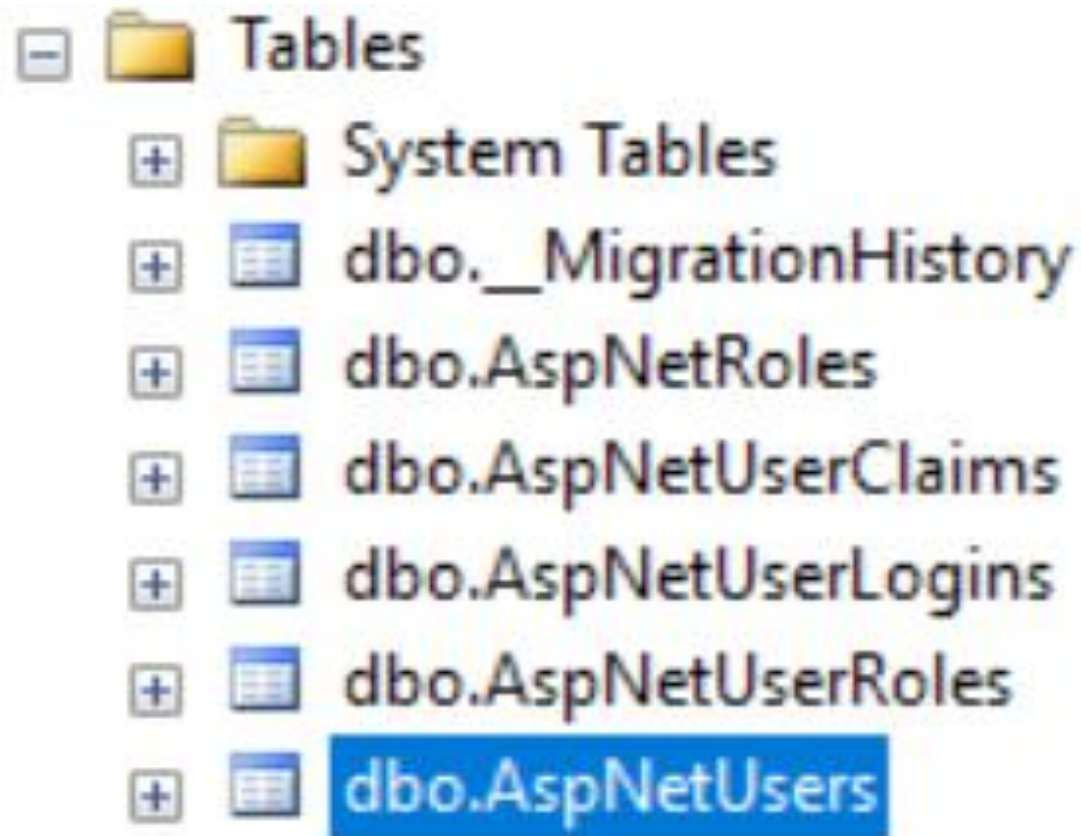


The screenshot shows a code editor window titled 'IdentityConfig.cs' with a solution explorer on the left showing 'DemoASP.NETIdentity' and 'DemoASP.NETIdentity.ApplicationUserManager'. The code in the editor is as follows:

```
52
53 // Configure validation logic for passwords
54 manager.PasswordValidator = new PasswordValidator
55 {
56     RequiredLength = 6,
57     RequireNonLetterOrDigit = true,
58     RequireDigit = true,
59     RequireLowercase = true,
60     RequireUppercase = true,
61 };
```

3. Chức năng đăng ký thành viên (Register)

- ❖ Sau khi đăng ký thành công, trong database xuất hiện thêm 6 table:



3. Chức năng đăng ký thành viên (Register)

❖ Cơ chế hoạt động của chức năng đăng ký trong ASP.NET Identity

- Khi người dùng click vào link **Register** thì ASP.NET sẽ gọi tới action **Register** (dòng 140) trong **AccountController**.

```
138      |  
139      |  
140  [ - ] | // GET: /Account/Register  
141      | | [AllowAnonymous]  
142      | | public ActionResult Register()  
143      | | {  
      | |     return View();  
      | | }  
      | |
```

3. Chức năng đăng ký thành viên (Register)

❖ Cơ chế hoạt động của chức năng đăng ký trong ASP.NET Identity

- Nội dung của view Register được tìm thấy file **Views/Account/Register.cshtml**
- View định nghĩa model là 1 đối tượng của class **RegisterViewModel** được định nghĩa trong file **Models/AccountViewModel**.

A screenshot of a web browser tab. The tab is yellow with a dark blue border. It contains the text 'Register.cshtml*' in a dark blue font. To the left of the text is a small icon of a document with a plus sign, and to the right is a small 'X' icon for closing the tab.

A screenshot of a code editor. The text '@model DemoASP.NETIdentity.Models.RegisterViewModel' is displayed. The '@model' part is highlighted in yellow, and 'RegisterViewModel' is highlighted in blue.

3. Chức năng đăng ký thành viên (Register)

❖ Cơ chế hoạt động của chức năng đăng ký trong ASP.NET Identity

```
public class RegisterViewModel
{
    [Required]
    [EmailAddress]
    [Display(Name = "Email")]
    public string Email { get; set; }
    [Required]
    [StringLength(100,
        ErrorMessage="The {0} must be at least {2} characters long.",MinimumLength=6)]
    [DataType(DataType.Password)]
    [Display(Name = "Password")]
    public string Password { get; set; }
    [DataType(DataType.Password)]
    [Display(Name = "Confirm password")]
    [Compare("Password",
        ErrorMessage="The password and confirmation password do not match.")]
    public string ConfirmPassword { get; set; }
}
```

3. Chức năng đăng ký thành viên (Register)

❖ Các ràng buộc trong **RegisterViewModel** như sau:

Thuộc tính	Quy tắc
Email	<ul style="list-style-type: none">• Không được bỏ trống• Đúng định dạng Email• Tên hiển thị là Email
Password	<ul style="list-style-type: none">• Không được bỏ trống• Tối thiểu 6 ký tự, tối đa 100 ký tự• Định dạng Password• Tên hiển thị là Password
ConfirmPassword	<ul style="list-style-type: none">• Định dạng Password• Tên hiển thị Confirm Password• Giá trị phải khớp với thuộc tính Password

3. Chức năng đăng ký thành viên (Register)

- ❖ Mặc định các **tên hiển thị** và các **thông báo lỗi** ở dạng **tiếng Anh**. Chúng ta có thể tùy chỉnh sang tiếng Việt.

```
public class RegisterViewModel
{
    [Required(ErrorMessage = "Bạn chưa nhập email")]
    [EmailAddress(ErrorMessage = "Email không hợp lệ, xin vui lòng kiểm tra lại")]
    [Display(Name = "Email")]
    public string Email { get; set; }
    [Required(ErrorMessage = "Bạn chưa nhập mật khẩu")]
    [StringLength(100,
        ErrorMessage = "Mật khẩu tối thiểu {2} và tối đa {1} ký tự", MinimumLength=6)]
    [DataType(DataType.Password)]
    [Display(Name = "Mật khẩu")]
    public string Password { get; set; }
    [DataType(DataType.Password)]
    [Display(Name = "Xác nhận mật khẩu")]
    [Compare("Password", ErrorMessage = "Mật khẩu và xác nhận mật khẩu không khớp.")]
    public string ConfirmPassword { get; set; }
}
```

3. Chức năng đăng ký thành viên (Register)

Application name

Home

About

Contact

Register.

Create a new account.

- Mật khẩu tối thiểu 6 và tối đa 100 ký tự
- Mật khẩu và xác nhận mật khẩu không khớp.

Email

dungta.uit@gmail.com

Mật khẩu

...

Xác nhận mật khẩu

•|

Register

3. Chức năng đăng ký thành viên (Register)

- ❖ Khi người dùng ấn **Register** thì form sẽ được đệ trình về phía server sau đó server sẽ gọi đến **action Register** có **attribute**

[HttpPost] trong **AccountController**.

```
146 // POST: /Account/Register
147 [HttpPost]
148 [AllowAnonymous]
149 [ValidateAntiForgeryToken]
150 public async Task<ActionResult> Register(RegisterViewModel model)
151 {
152     if (ModelState.IsValid){
153         var user = new ApplicationUser { UserName = model.Email, Email = model.Email };
154         var result = await UserManager.CreateAsync(user, model.Password);
155         if (result.Succeeded){
156             await SignInManager.SignInAsync(user, isPersistent:false, rememberBrowser:false);
157             // For more information on how to enable account confirmation and password reset
158             // Send an email with this link
159             // string code = await UserManager.GenerateEmailConfirmationTokenAsync(user.Id);
160             // var callbackUrl = Url.Action("ConfirmEmail", "Account", new { userId = user.Id
161             // await UserManager.SendEmailAsync(user.Id, "Confirm your account", "Please conf
162             return RedirectToAction("Index", "Home");
163         }
164         AddErrors(result);
165     }
166     return View(model);
167 13/03/2023 }
```

3. Chức năng đăng ký thành viên (Register)

❖ Cơ chế hoạt động của chức năng đăng ký trong ASP.NET Identity

- Action `Register` sử dụng thêm attribute `[ValidateAntiForgeryToken]` để bảo mật website tránh hacker mạo danh website và gửi 1 request không hợp lệ đến server.
- Chi tiết về tác dụng của attribute này các bạn tham khảo thêm ở link sau: <http://www.oasearch.net/Tai-sao-su-dung-ValidateAntiForgeryToken-trong-ASPNET-MVC-83>

4. Chức năng đăng nhập (Login)

The screenshot displays the ASP.NET website interface. At the top, there is a browser tab labeled 'Home Page - My ASP.NET Applic' and a address bar showing 'localhost:62029'. Below the browser, a navigation bar contains links for 'Application name', 'Home', 'About', and 'Contact', along with 'Register' and 'Log in' on the right. The main content area features the 'ASP.NET' logo and a description: 'ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.' A blue button labeled 'Learn more »' is positioned below the text. Further down, there are three columns: 'Getting started' with a description of ASP.NET MVC and a 'Learn more »' button; 'Get more libraries' with a description of NuGet and a 'Learn more »' button; and 'Web Hosting' with a description of finding a hosting company and a 'Learn more »' button.

Home Page - My ASP.NET Applic x +

localhost:62029

Anh Dung Diep Nhu Thẻ mới Download Ebook Bí Quyết Giàu Có V... Download and Inst... Download and Inst... Đưa chức năng Rep... Trần Diệp Như

Application name Home About Contact Register Log in

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.

[Learn more »](#)

Getting started

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.

[Learn more »](#)

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)

4. Chức năng đăng nhập (Login)

Application name Home About Contact Register Log in

Log in.

Use a local account to log in.

Email

dungta.uit@gmail.com

Password

.....

☒ Remember me?

Log in

[Register as a new user](#)

Use another service to log in.

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.

© 2023 - My ASP.NET Application

4. Chức năng đăng nhập (Login)

- ❖ Có thể chỉnh sang tiếng Việt bằng cách sửa định nghĩa của `class LoginViewModel` trong file `Models/AccountViewModel.cs`

```
49 public class LoginViewModel
50 {
51     [Required]
52     [Display(Name = "Email")]
53     [EmailAddress]
54     public string Email { get; set; }
55
56     [Required]
57     [DataType(DataType.Password)]
58     [Display(Name = "Password")]
59     public string Password { get; set; }
60
61     [Display(Name = "Remember me?")]
62     public bool RememberMe { get; set; }
63 }
```

4. Chức năng đăng nhập (Login)

```
public class LoginViewModel
{
    [Required(ErrorMessage = "Bạn chưa nhập email")]
    [Display(Name = "Email")]
    [EmailAddress(ErrorMessage = "Email không hợp lệ, xin kiểm tra lại")]
    public string Email { get; set; }

    [Required(ErrorMessage = "Bạn chưa nhập mật khẩu")]
    [DataType(DataType.Password)]
    [Display(Name = "Mật khẩu")]
    public string Password { get; set; }

    [Display(Name = "Ghi nhớ?")]
    public bool RememberMe { get; set; }
}
```

4. Chức năng đăng nhập (Login)

❖ Action **Login** trong **AccountController**

```
56      // GET: /Account/Login
57      [AllowAnonymous]
58      public ActionResult Login(string returnUrl)
59      {
60          ViewBag.ReturnUrl = returnUrl;
61          return View();
62      }
63
```

❖ Dòng **61** trả ra 1 view **Login**

4. Chức năng đăng nhập (Login)

- ❖ Nội dung của view Login này được tìm thấy tại file **Views/Account/Login.cshtml**

```
Login.cshtml  AccountController.cs  Register.cshtml  AccountViewModels.cs
1  @using DemoASP.NETIdentity.Models
2  @model LoginViewModel
3  @{
4      ViewBag.Title = "Log in";
5  }
6
7  <h2>@ViewBag.Title.</h2>
8  <div class="row">
9      <div class="col-md-8">
10         <section id="loginForm">
11             @using (Html.BeginForm("Login", "Account",
12                 {
13                     @Html.AntiForgeryToken()
14                     <h4>Use a local account to log in.</h4>
```

4. Chức năng đăng nhập (Login)

- ❖ Khi user ấn nút **Log in** thì form sẽ đệ trình về server sau đó server gọi đến **action Login** có attribute **[HttpPost]** tại dòng 66 trong **AccountController**.

```
65 // POST: /Account/Login
66 [HttpPost]
67 [AllowAnonymous]
68 [ValidateAntiForgeryToken]
69 public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
70 {
71     if (!ModelState.IsValid)
72     {
73         return View(model);
74     }
75
76     // This doesn't count login failures towards account lockout
77     // To enable password failures to trigger account lockout, change to shouldLockout
78     var result = await SignInManager.PasswordSignInAsync(model.Email, model.Password,
79     switch (result)
80     {
```

4. Chức năng đăng nhập (Login)

❖ Đoạn lệnh `switch - case` để kiểm tra giá trị `result`

```
switch (result)
{
    case SignInStatus.Success:
        return RedirectToLocal(returnUrl);
    case SignInStatus.LockedOut:
        return View("Lockout");
    case SignInStatus.RequiresVerification:
        return RedirectToAction("SendCode",
            new { returnUrl = returnUrl, RememberMe = model.RememberMe });
    case SignInStatus.Failure:
    default:
        ModelState.AddModelError("", "Invalid login attempt.");
        return View(model);
}
```

5. Chức năng đăng xuất (Log off)

- ❖ Cơ chế xử lý của chức năng này thì cực kỳ đơn giản.
- ❖ Khi các bạn click vào link **Log off** thì server sẽ gọi tới **action LogOff** (dòng 391) trong **AccountController**. Action này gọi tới phương thức **SignOut** trong class **AuthenticationManager** kèm theo cookie của ứng dụng để ASP.NET Identity remove những cookie đã lưu khi user đăng nhập, cuối cùng thì redirect về trang chủ.

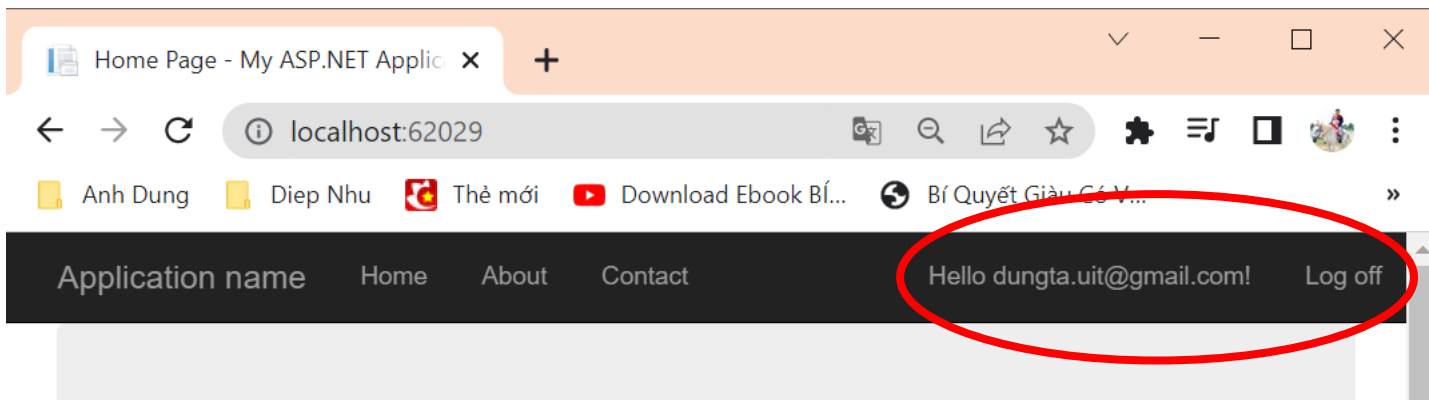
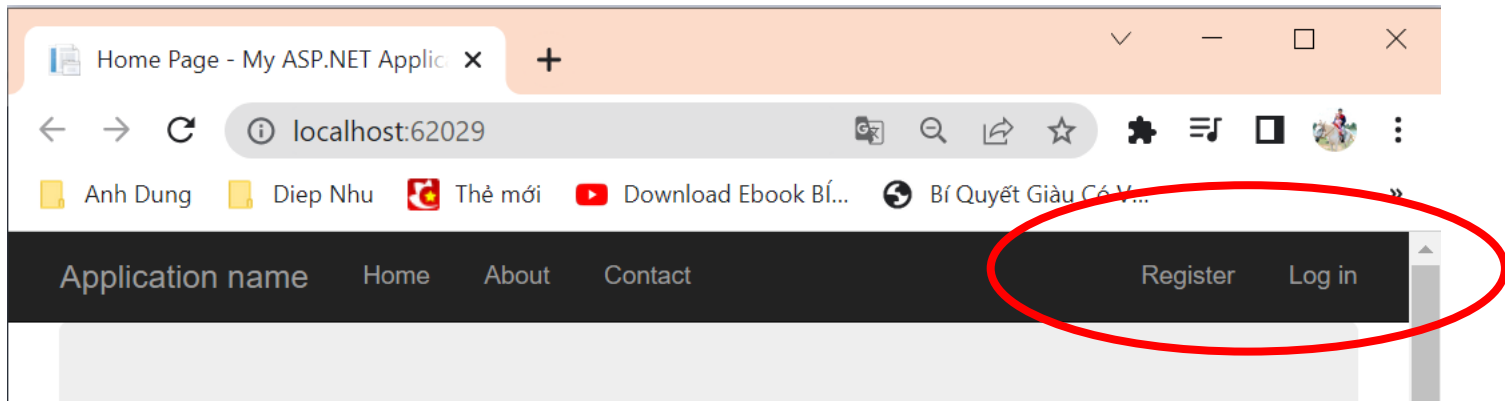
5. Chức năng đăng xuất (Log off)

❖ Action **LogOff** của controller **AccountController.cs**

```
//  
// POST: /Account/LogOff  
[HttpPost]  
[ValidateAntiForgeryToken]  
public ActionResult LogOff()  
{  
    AuthenticationManager.SignOut(  
        DefaultAuthenticationTypes.ApplicationCookie);  
    return RedirectToAction("Index", "Home");  
}
```


6. Sự hiển thị của menu

❖ Khi chưa đăng nhập và đã đăng nhập:



6. Sự hiển thị của menu

❖ View `_Layout.cshtml` ở thư mục `Views/Shared`

```
_Layout.cshtml  X
20      @Html.ActionLink("Application name", "Index", "Home", new {
21      }
22      </div>
23      <div class="navbar-collapse collapse">
24      <ul class="nav navbar-nav">
25          <li>@Html.ActionLink("Home", "Index", "Home")</li>
26          <li>@Html.ActionLink("About", "About", "Home")</li>
27          <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
28      </ul>
29      @Html.Partial("_LoginPartial")
30      </div>
31  </div>
```

6. Sự hiển thị của menu

❖ View `_LoginPartial` ở thư mục `Views/Shared`

```
_LoginPartial.cshtml  _Layout.cshtml
1  @using Microsoft.AspNet.Identity
2  @if (Request.IsAuthenticated)
3  {
4      using (Html.BeginForm("LogOff", "Account", FormMethod.Post, new { id = "logoutForm", @class
5      {
6          @Html.AntiForgeryToken()
7
8          <ul class="nav navbar-nav navbar-right">
9              <li>
10                 @Html.ActionLink("Hello " + User.Identity.GetUserName() + "!", "Index", "Manage", ro
11                 </li>
12                 <li><a href="javascript:document.getElementById('logoutForm').submit()">Log off</a></li>
13             </ul>
14         }
15     }
16     else
17     {
18         <ul class="nav navbar-nav navbar-right">
19             <li>@Html.ActionLink("Register", "Register", "Account", routeValues: null, htmlAttribute
20             <li>@Html.ActionLink("Log in", "Login", "Account", routeValues: null, htmlAttributes: ne
21         </ul>
22     }
23
```

7. Xác thực email khi đăng ký thành viên

- ❖ Các bước cấu hình để gửi mail thông qua tài khoản gmail.
- ❖ **Bước 1:** Mở file **IdentityConfig.cs** trong folder **App_Start**.
 - Khai báo sử dụng 2 namespace: **System.Net.Mail** và **System.Net**
 - Sau đó cài đặt lại phương thức **SendAsync** trong class **EmailService** với nội dung như slide phía sau.

7. Xác thực email khi đăng ký thành viên

```
22 public Task SendAsync(IdentityMessage message)
23 {
24     var client = new SmtpClient
25     {
26         Host = "smtp.gmail.com",
27         Port = 587,
28         UseDefaultCredentials = false,
29         DeliveryMethod = SmtpDeliveryMethod.Network,
30         Credentials = new NetworkCredential("your email", "pass"),
31         EnableSsl = true
32     };
33     var from = new MailAddress("your email", "pass");
34     var to = new MailAddress(message.Destination);
35     var mail = new MailMessage(from, to)
36     {
37         Subject = message.Subject,
38         Body = message.Body,
39         IsBodyHtml = true
40     };
41     client.Send(mail);
42     return Task.FromResult(0);
43 }
```

7. Xác thực email khi đăng ký thành viên

❖ **Bước 2:** Truy cập vào link sau:

➤ <https://www.google.com/settings/security/lesssecureapps>

➤ Bật quyền truy cập cho các ứng dụng kém an toàn.

❖ **Bước 3:** Tìm đến action **Register** trong **AccountController** và uncomment đoạn chú thích và thay đổi code để hiển thị thông báo cho user biết rằng họ cần vào email để xác thực tài khoản.

7. Xác thực email khi đăng ký thành viên

❖ Action **Register** trong **AccountController** đã sinh tự động như sau:

```
146 // POST: /Account/Register
147 [HttpPost]
148 [AllowAnonymous]
149 [ValidateAntiForgeryToken]
150 public async Task<ActionResult> Register(RegisterViewModel model)
151 {
152     if (ModelState.IsValid){
153         var user = new ApplicationUser { UserName = model.Email, Email = model.Email };
154         var result = await UserManager.CreateAsync(user, model.Password);
155         if (result.Succeeded){
156             await SignInManager.SignInAsync(user, isPersistent:false, rememberBrowser:false);
157             // For more information on how to enable account confirmation and password reset
158             // Send an email with this link
159             // string code = await UserManager.GenerateEmailConfirmationTokenAsync(user.Id);
160             // var callbackUrl = Url.Action("ConfirmEmail", "Account", new { userId = user.Id
161             // await UserManager.SendEmailAsync(user.Id, "Confirm your account", "Please conf
162             return RedirectToAction("Index", "Home");
163         }
164         AddErrors(result);
165     }
166     return View(model);
167 }
```

7. Xác thực email khi đăng ký thành viên

```
if (ModelState.IsValid){
    var user = new ApplicationUser { UserName = model.Email, Email = model.Email };
    var result = await UserManager.CreateAsync(user, model.Password);
    if (result.Succeeded){
        await SignInManager.SignInAsync(user, isPersistent:false, rememberBrowser:false);
        // For more information on how to enable account confirmation and password reset
        // Send an email with this link
        string code = await UserManager.GenerateEmailConfirmationTokenAsync(user.Id);
        var callbackUrl = Url.Action("ConfirmEmail", "Account",
            new { userId = user.Id, code = code }, protocol: Request.Url.Scheme);
        await UserManager.SendEmailAsync(user.Id, "Xác thực tài khoản của bạn",
            "Vui lòng click vào <a href=\"" + callbackUrl
            + "\">đây</a> để thực hiện việc xác thực");
        ViewBag.Message = "Chúng tôi đã gửi email cho việc xác thực tài"
            + "khoản đến email của bạn. Vui lòng kiểm tra email để thực hiện xác thực";
        return View(model);
    }
    AddErrors(result);
}
return View(model);
```


7. Xác thực email khi đăng ký thành viên

- ❖ **Bước 4:** Mở view Register và bổ sung đoạn lệnh sau vào sau thẻ `<h2>@ViewBag.Title</h2>`:

```
8      <h2>@ViewBag.Title.</h2>
9
10     <div class="text-success">
11         <p>
12             @ViewBag.Message
13         </p>
14     </div>
```

7. Xác thực email khi đăng ký thành viên

❖ **Bước 5:** view **ConfirmEmail** ở file **Views/Account/ConfirmEmail.cshtml**, đây là view hiển thị thông báo khi user xác thực email thành công.



```
1 @{
2     ViewBag.Title = "Confirm Email";
3 }
4
5 <h2>@ViewBag.Title.</h2>
6 <div>
7     <p>
8         Thank you for confirming your email.
9         Please @Html.ActionLink("Click here to Log in", "Login", "Account",
10             routeValues: null, htmlAttributes: new { id = "loginLink" })
11     </p>
12 </div>
```

7. Xác thực email khi đăng ký thành viên

❖ Bước 6: Build và chạy để kiểm tra

Register - My ASP.NET Applicatio x +

localhost:62029/Account/Register

Anh Dung Diep Nhu Thẻ mới Download Ebook Bí... Bí Quyết Giàu Có V...

Application name Home About Contact Register Log in

Register.

Create a new account.

Email

Mật khẩu

Xác nhận mật khẩu

Register

7. Xác thực email khi đăng ký thành viên

The screenshot shows a web browser window with the Google Account Security page. The browser's address bar shows the URL `myaccount.google.com/security?hl=en`. The page title is "Google Account" and the main heading is "Security". Below the heading, it says "Settings and recommendations to help you keep your account secure".

On the left, there is a navigation menu with the following items: Home, Personal info, Data & privacy, Security (highlighted), People & sharing, Payments & subscriptions, and About. At the bottom of the menu are links for Privacy, Terms, Help, and About.

The main content area is divided into two sections. The top section is titled "You have security tips" and includes a green shield icon with a checkmark. It says "Security tips found in the Security Checkup" and has a link to "Review security tips". The bottom section is titled "Recent security activity" and lists two entries: "New sign-in on Windows" on Mar 9 from Ho Chi Minh City, Vietnam, and "New sign-in on Windows" on Mar 6 from Vietnam. Each entry has a right-pointing arrow.

On the right side of the page, there is a "Help" section. It contains a notification box stating: "Your account, dungta.uit@gmail.com, has a recovery email set up. Make sure your recovery info is updated in case you get locked out of your account." Below this is a section titled "Less secure apps & your Google Account". It contains a warning: "To help keep your account secure, from May 30, 2022, Google no longer supports the use of third-party apps or devices which ask you to sign in to your Google Account using only your username and password." It also includes an "Important" note: "This deadline does not apply to Google Workspace or Google Cloud Identity customers. The enforcement date for these customers will be announced on the Workspace blog at a later date." and a link to "For more information, continue to read."

At the bottom of the page, there is a footer with the text "If an app or site doesn't meet our security standards," followed by a link to "security standards".

Một số chức năng mở rộng ASP.NET Identity

1

- Khóa (Lock) user khi đăng nhập sai nhiều lần

2

- Đổi mật khẩu

3

- Lấy lại mật khẩu

4

- Bổ sung thông tin của User

1. Khóa (Lock) user

- ❖ Ở một số website khi các bạn truy cập vào trang Login của họ và nhập sai mật khẩu quá nhiều lần thì sẽ hiển thị thông báo rằng tài khoản của bạn tạm thời bị khóa.
- ❖ Đầu tiên để bật chức năng này thì các bạn vào action **Login** có attribute **[HttpPost]** trong **AccountController**, sau đó thay giá trị của đối số **shouldLockout** thành **true**.

1. Khóa (Lock) user

❖ Thay giá trị của đối số `shouldLockout` thành `true`

```
// POST: /Account/Login
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }

    // This doesn't count login failures towards account lockout
    // To enable password failures to trigger account lockout, change to shou
    var result = await SignInManager.PasswordSignInAsync(model.Email,
        model.Password, model.RememberMe, shouldLockout: true);
    switch (result)
    {

```

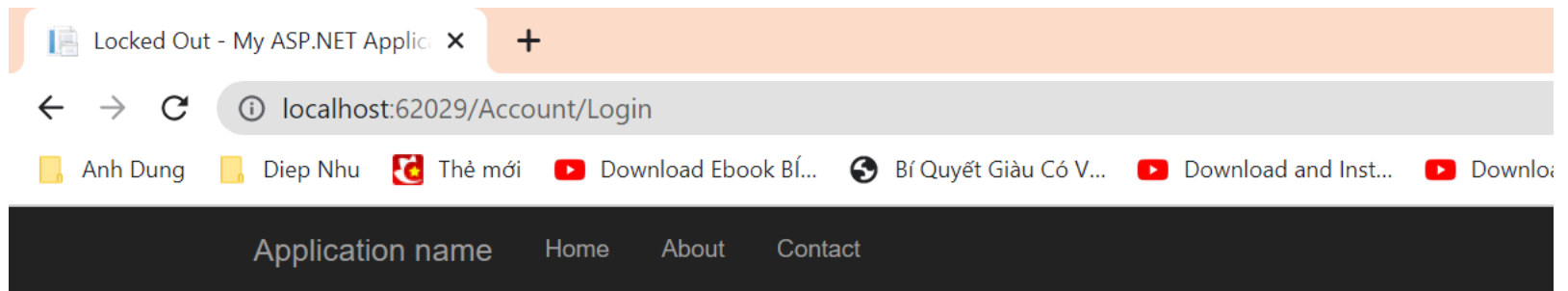
1. Khóa (Lock) user

- ❖ Đoạn code cấu hình lock user được tìm thấy ở dòng 86 trong file `App_Start/IdentityConfig.cs`. Hình bên dưới thể hiện đoạn code này:

```
// Configure user lockout defaults
manager.UserLockoutEnabledByDefault = true;
manager.DefaultAccountLockoutTimeSpan = TimeSpan.FromMinutes(5);
manager.MaxFailedAccessAttemptsBeforeLockout = 5;
```


1. Khóa (Lock) user

- ❖ Nếu user nhập sai tài khoản hoặc mật khẩu từ 5 lần trở lên thì user đó sẽ bị khóa 5 phút và sau 5 phút phút bị khóa user đó mới có thể nhập tiếp.



Locked out.

This account has been locked out, please try again later.

© 2023 - My ASP.NET Application

1. Khóa (Lock) user

- ❖ View thể hiện thông báo lỗi được tìm thấy tại file `Lockout.cshtml` trong folder `Views/Shared`. Các bạn có thể thiết kế lại view này sao cho nó thân thiện với khách hàng.



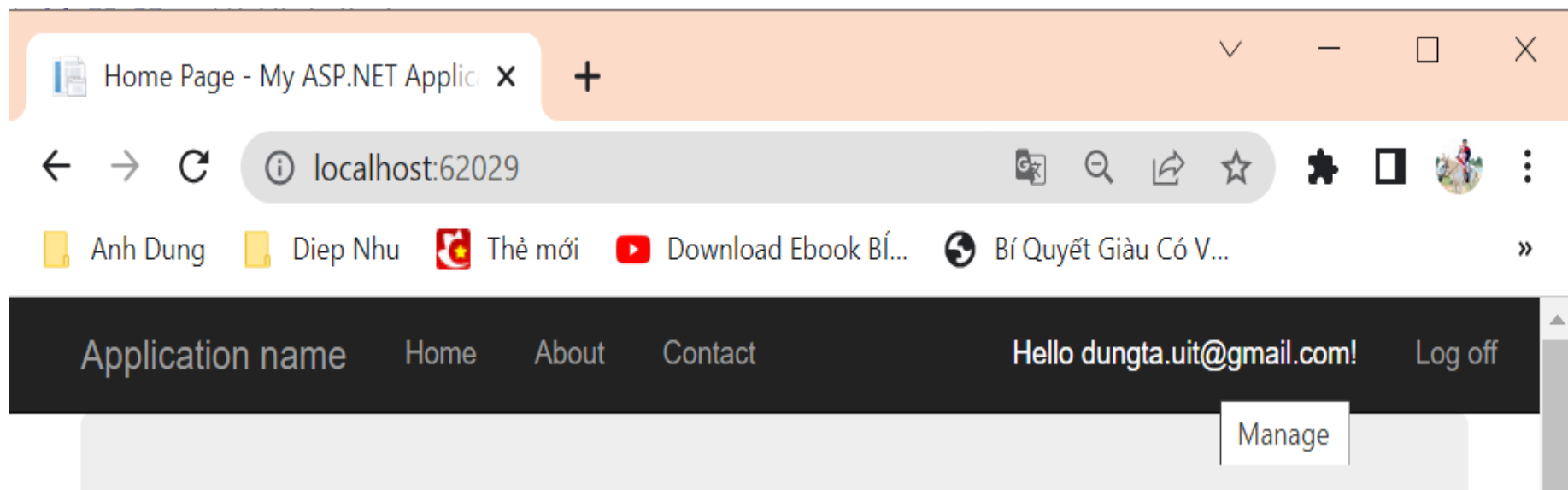
Tạm khóa

Tài khoản này đã bị khóa, vui lòng thử lại sau.

2. Đổi mật khẩu

❖ Các bước để đổi mật khẩu:

- **Bước 1:** Login vào website và click vào link Hello emailcuaban. Trang Manage sẽ hiện ra, đây là trang cá nhân của user đó các bạn.



2. Đổi mật khẩu

❖ Các bước để đổi mật khẩu:

➤ **Bước 2:** Click vào link Change your password để vào trang Change Password.

Application name Home About Contact Hello dungta.uit@gmail.com! Log off

Manage.

Change your account settings

Password:	[Change your password]
External Logins:	0 [Manage]
Two-Factor Authentication:	There are no two-factor authentication providers configured. See this article for details on setting up this ASP.NET application to support two-factor authentication.

© 2023 - My ASP.NET Application

2. Đổi mật khẩu

❖ Các bước để đổi mật khẩu:

➤ **Bước 3:**

[Application name](#) [Home](#) [About](#) [Contact](#)

Change Password.

Change Password Form

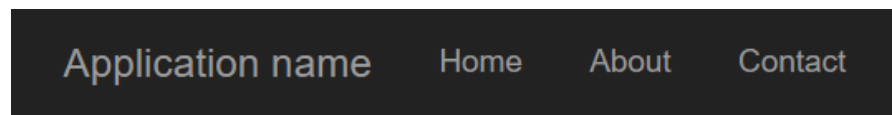
Current password	<input type="password"/>
New password	<input type="password"/>
Confirm new password	<input type="password"/>
<input type="button" value="Change password"/>	

© 2023 - My ASP.NET Application

2. Đổi mật khẩu

❖ Các bước để đổi mật khẩu:

- **Bước 4:** Ấn nút Change Password để chấp nhận đổi mật khẩu.



Manage.

Your password has been changed.

Change your account settings



Cập nhật mật khẩu?

Tên người dùng

dungta.uit@gmail.com

Mật khẩu

.....

Cập nhật mật khẩu

Không, cảm ơn

Bạn có thể dùng các mật khẩu đã lưu trên mọi thiết bị. Các mật khẩu đó được lưu vào [Trình quản lý mật khẩu](#) cho dungta@uit.edu.vn.

2. Đổi mật khẩu

- ❖ Khi click vào link **Change your password** thì server sẽ gọi đến **action ChangePassword** trong **ManageController** (dòng thứ 218). Nội dung của view này được tìm thấy tại file **ChangePassword.cshtml** ở folder **Views/Manage**.

```
217 // GET: /Manage/ChangePassword
218 public ActionResult ChangePassword()
219 {
220     return View();
221 }
```

2. Đổi mật khẩu

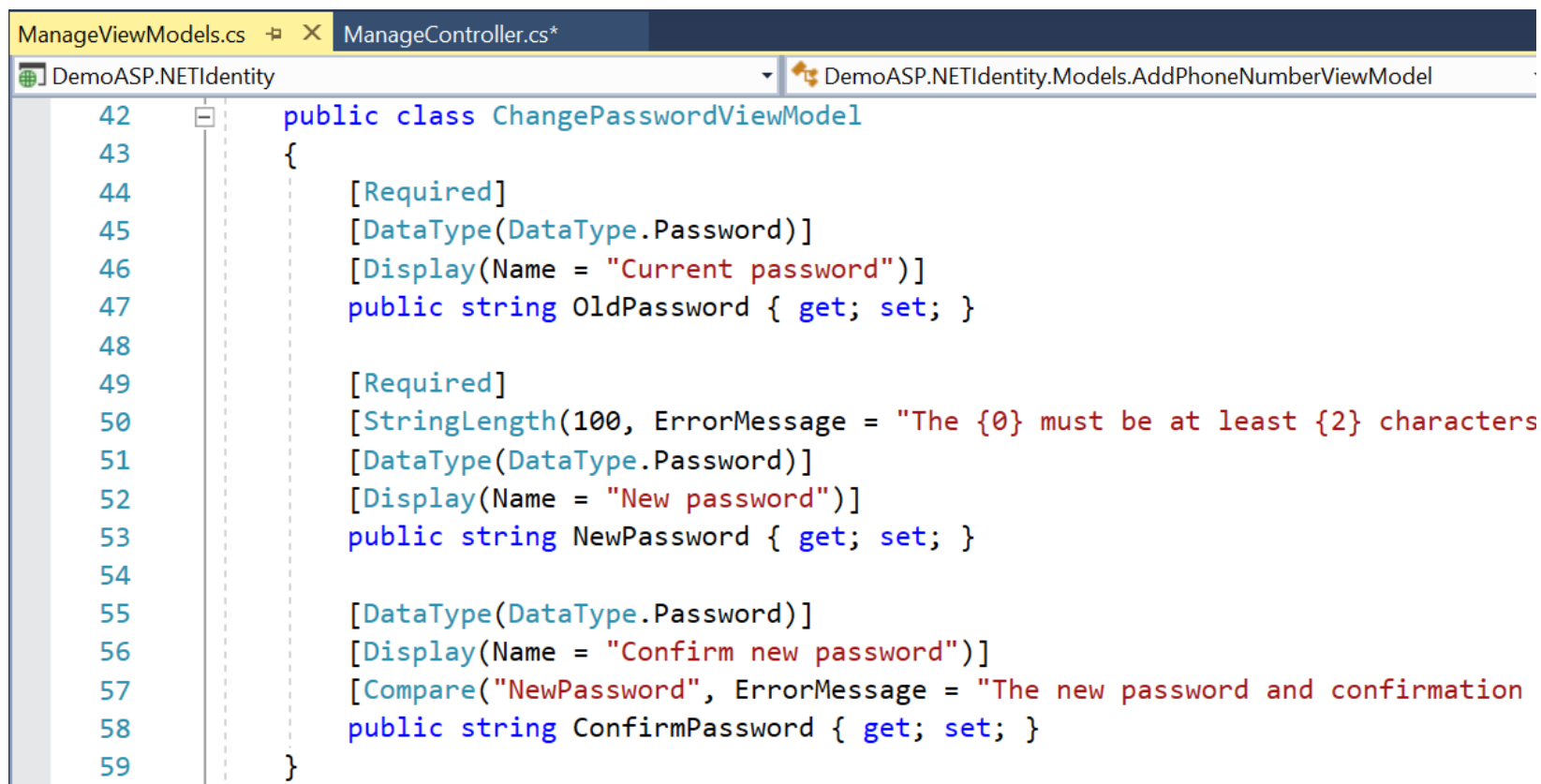
- ❖ Form sẽ được đệ trình và gửi thông tin đến **action** **ChangePassword** có attribute **[HttpPost]** (dòng 227).

```
// POST: /Manage/ChangePassword
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult> ChangePassword(ChangePasswordViewModel model)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }
    var result = await UserManager.ChangePasswordAsync(User.Identity.GetUserId(),
                                                       model.OldPassword, model.NewPassword);

    if (result.Succeeded)
    {
        var user = await UserManager.FindByIdAsync(User.Identity.GetUserId());
        if (user != null)
        {
            await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);
        }
        return RedirectToAction("Index", new { Message = ManageMessageId.ChangePasswordSuccess });
    }
    AddErrors(result);
    return View(model);
}
```


2. Đổi mật khẩu

- ❖ Chúng ta cấu hình để hiển thị lỗi tiếng Việt thay vì tiếng Anh ở trang Change Password.



```
42 public class ChangePasswordViewModel
43 {
44     [Required]
45     [DataType(DataType.Password)]
46     [Display(Name = "Current password")]
47     public string OldPassword { get; set; }
48
49     [Required]
50     [StringLength(100, ErrorMessage = "The {0} must be at least {2} characters")]
51     [DataType(DataType.Password)]
52     [Display(Name = "New password")]
53     public string NewPassword { get; set; }
54
55     [DataType(DataType.Password)]
56     [Display(Name = "Confirm new password")]
57     [Compare("NewPassword", ErrorMessage = "The new password and confirmation")]
58     public string ConfirmPassword { get; set; }
59 }
```

2. Đổi mật khẩu

```
public class ChangePasswordViewModel
{
    [Required(ErrorMessage = "Bạn chưa nhập mật khẩu hiện tại")]
    [DataType(DataType.Password)]
    [Display(Name = "Mật khẩu hiện tại")]
    public string OldPassword { get; set; }

    [Required(ErrorMessage = "Bạn chưa nhập mật khẩu mới")]
    [StringLength(100,
        ErrorMessage="Mật khẩu mới tối thiểu {2} và tối đa {1} ký tự.",MinimumLength=6)]
    [DataType(DataType.Password)]
    [Display(Name = "Mật khẩu mới")]
    public string NewPassword { get; set; }

    [DataType(DataType.Password)]
    [Display(Name = "Xác thực mật khẩu mới")]
    [Compare("NewPassword",
        ErrorMessage = "Mật khẩu mới và xác thực mật khẩu mới không khớp.")]
    public string ConfirmPassword { get; set; }
}
```

3. Lấy lại mật khẩu

- ❖ Chức năng này chỉ có thể được sử dụng khi bạn đã cấu hình gửi mail xác nhận khi user đăng ký thành viên.
- ❖ Vì khi user yêu cầu lấy lại mật khẩu thì ASP.NET Identity sẽ gửi 1 link đến mail của user, khi user đó ấn vào link này thì sẽ trình duyệt sẽ hiện lên trang **Reset Password** cho user đó thực hiện đặt mật khẩu.

3. Lấy lại mật khẩu

- ❖ Để sử dụng chức năng này, mở file `Views/Login.cshtml` lên và uncomment thẻ `<p></p>`:

```
45      @Html.ActionLink("Register as a new user", "Register")
46  </p>
47  @* Enable this once you have account confirmation enabled for password
48      <p>
49          @Html.ActionLink("Forgot your password?", "ForgotPassword")
50      </p>*@
```

- ❖ Sau khi uncomment:

```
47  //Enable this once you have account confirmation enabled for password
48  //reset functionality
49  <p>
50      @Html.ActionLink("Forgot your password?", "ForgotPassword")
51  </p>
```

3. Lấy lại mật khẩu

Application name Home About Contact

Log in.

Use a local account to log in.

Email

dungta.uit@gmail.com

Mật khẩu

.....

☐ Ghi nhớ?

Log in

[Register as a new user](#)

[Forgot your password?](#)

© 2023 - My ASP.NET Application

Application name

Home

About

Contact

Forgot your password?.

Enter your email.

Email

dungta.uit@gmail.com

Email Link

© 2023 - My ASP.NET Application

3. Lấy lại mật khẩu

- ❖ Khi user nhập Email và ấn nút **Email Link** thì form sẽ được đề trình và gửi thông tin đến action **ForgotPassword** (dòng 202) có attribute **[HttpPost]** trong **AccountController**

```
198 // POST: /Account/ForgotPassword
199 [HttpPost]
200 [AllowAnonymous]
201 [ValidateAntiForgeryToken]
202 public async Task<ActionResult> ForgotPassword(ForgotPasswordViewModel model)
203 {
204     if (ModelState.IsValid)
205     {
206         var user = await UserManager.FindByNameAsync(model.Email);
207         if (user == null || !(await UserManager.IsEmailConfirmedAsync(user.Id)))
208         {
209             // Don't reveal that the user does not exist or is not confirmed
210             return View("ForgotPasswordConfirmation");
211         }
212
213         // For more information on how to enable account confirmation and password r
214         // Send an email with this link
215         // string code = await UserManager.GeneratePasswordResetTokenAsync(user.Id);
216         // var callbackUrl = Url.Action("ResetPassword", "Account", new { userId = u
217         // await UserManager.SendEmailAsync(user.Id, "Reset Password", "Please reset
218         // return RedirectToAction("ForgotPasswordConfirmation", "Account");
219     }
```

4. Bổ sung thông tin của user

- ❖ Mặc định các bạn thấy table **dbo.AspNetUsers** (table chứa thông tin các user trên web) trong CSDL có sẵn các cột do **ASP.NET Identity** tự sinh ra.
- ❖ Tuy nhiên bạn muốn bổ sung thêm 1 số thông tin cho user như Họ tên, địa chỉ, giới tính, ... thì làm sao?
- ❖ Để làm được điều này thì các bạn cần đến kiến thức căn bản về **Entity Framework Code First**.

4. Bổ sung thông tin của user

❖ Giả sử bây giờ muốn bổ sung thêm thông tin họ tên, địa chỉ, giới tính cho các user thì thực hiện như sau:

➤ **Bước 1:** Tìm đến class `ApplicationUser` trong file `Models/IdentityModels.cs`

```
// You can add profile data for the user by adding more properties to your ApplicationUser class, please visit http://
public class ApplicationUser : IdentityUser
{
    public async Task<ClaimsIdentity> GenerateUserIdentityAsync(UserManager<ApplicationUser> manager)
    {
        // Note the authenticationType must match the one defined in CookieAuthenticationOptions.AuthenticationType
        var userIdentity = await manager.CreateIdentityAsync(this, DefaultAuthenticationTypes.ApplicationCookie);
        // Add custom user claims here
        return userIdentity;
    }
}
```


4. Bổ sung thông tin của user

❖ Bước 2: using namespace System và bổ sung thêm 3

// You can add profile data for the user by adding more properties to your ApplicationUser class, please visit [http](http://)

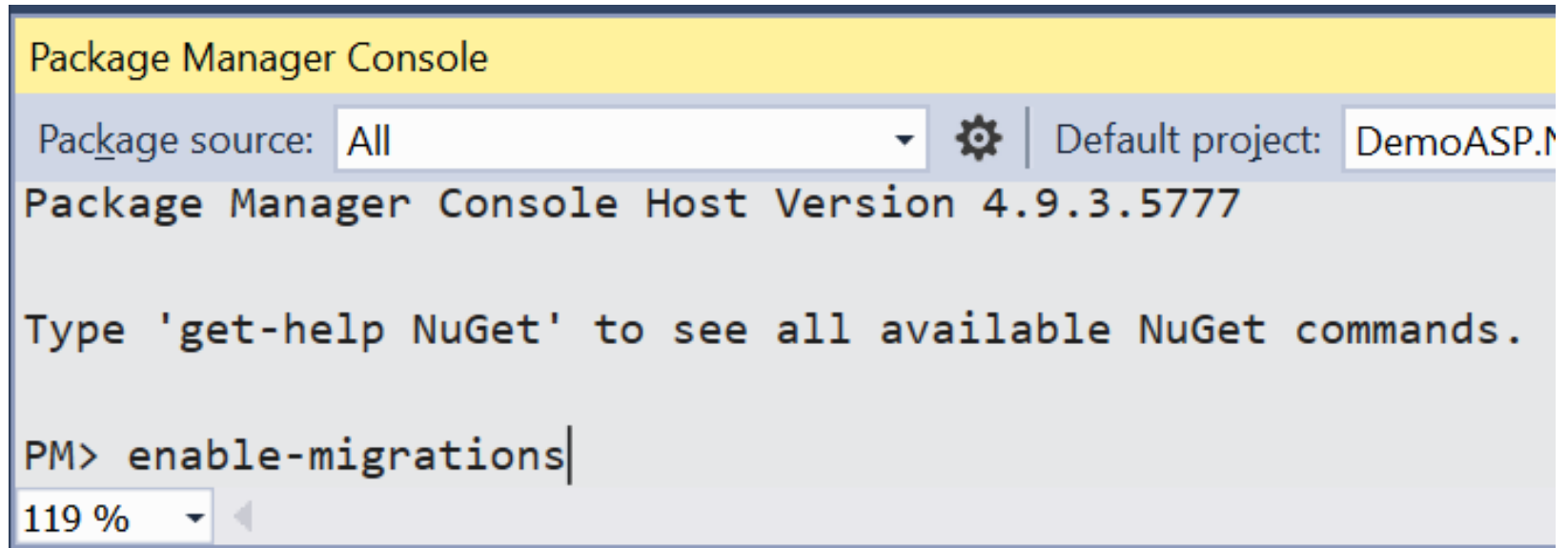
```
public class ApplicationUser : IdentityUser
{
    public string FullName { get; set; }
    public string Address { get; set; }
    public Nullable<bool> Gender { get; set; }
    public async Task<ClaimsIdentity> GenerateUserIdentityAsync(UserManager<ApplicationUser> manager)
    {
        // Note the authenticationType must match the one defined in CookieAuthenticationOptions.AuthenticationType
        var userIdentity = await manager.CreateIdentityAsync(this, DefaultAuthenticationTypes.ApplicationCookie);
        // Add custom user claims here
        return userIdentity;
    }
}
```

4. Bổ sung thông tin của user

- ❖ **Bước 3:** Mở màn hình **Package Manage Console** lên bằng cách click vào menu Tools -> NuGet Package Manage -> Package Manage Console ở Visual Studio.

4. Bổ sung thông tin của user

- ❖ **Bước 4:** Ở màn hình Package Manage Console nhập `enable-migrations` và **nhấn Enter**



The screenshot shows the Package Manager Console interface. At the top, there's a yellow header bar with the text "Package Manager Console". Below this, there's a light blue bar containing "Package source: All" with a dropdown arrow and a gear icon, and "Default project: DemoASP.N". Below the blue bar, the text "Package Manager Console Host Version 4.9.3.5777" is displayed. The main area is white and contains the instruction "Type 'get-help NuGet' to see all available NuGet commands." At the bottom, the command prompt shows "PM> enable-migrations|" with the cursor at the end of the line. Below the command prompt, there's a status bar with "119 %" and a left arrow.

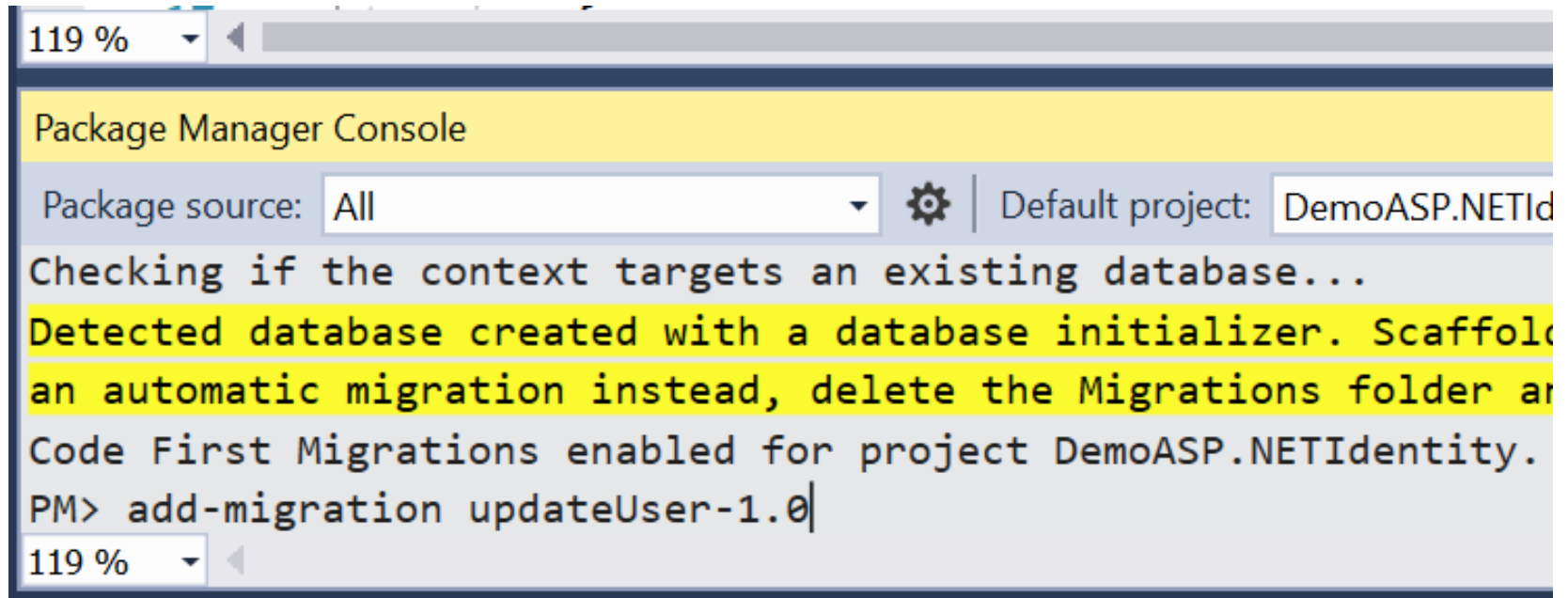
```
Package Manager Console
Package source: All | Default project: DemoASP.N
Package Manager Console Host Version 4.9.3.5777

Type 'get-help NuGet' to see all available NuGet commands.

PM> enable-migrations|
119 %
```

4. Bổ sung thông tin của user

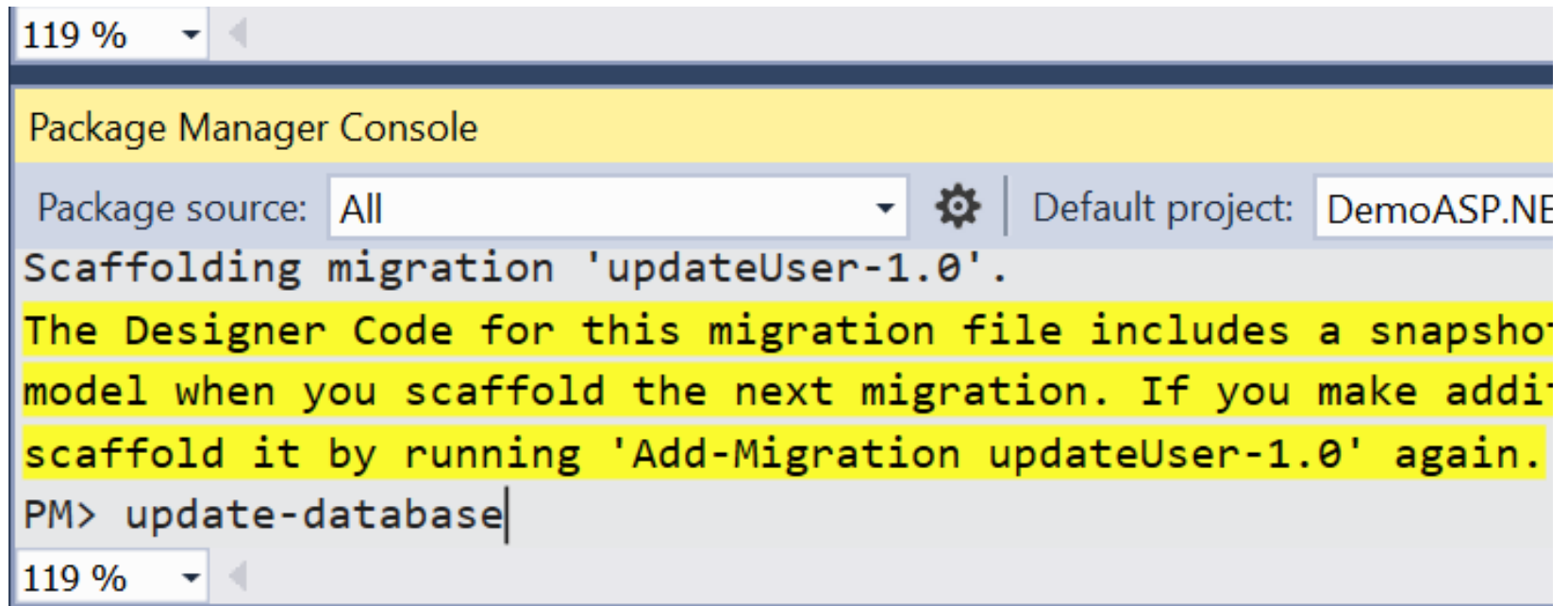
- ❖ **Bước 5:** Nhập `add-migration updateUser-1.0` và **nhấn Enter**.



```
119 %  
Package Manager Console  
Package source: All | Default project: DemoASP.NETId  
Checking if the context targets an existing database...  
Detected database created with a database initializer. Scaffold  
an automatic migration instead, delete the Migrations folder and  
Code First Migrations enabled for project DemoASP.NETIdentity.  
PM> add-migration updateUser-1.0  
119 %
```

4. Bổ sung thông tin của user

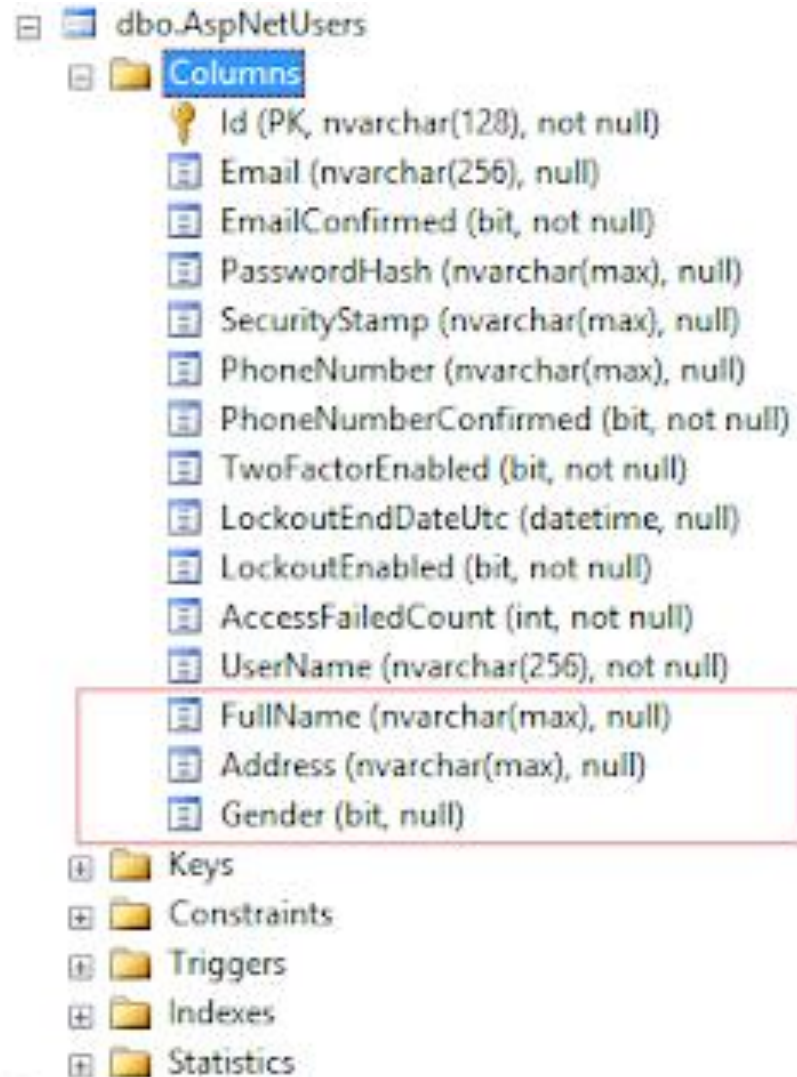
❖ **Bước 6:** Gõ `update-database` và nhấn Enter



```
119 %  
Package Manager Console  
Package source: All | Default project: DemoASP.NET  
Scaffolding migration 'updateUser-1.0'.  
The Designer Code for this migration file includes a snapshot  
model when you scaffold the next migration. If you make additional  
scaffold it by running 'Add-Migration updateUser-1.0' again.  
PM> update-database  
119 %
```

4. Bổ sung thông tin của user

- ❖ Mở CSDL và đảm bảo rằng 3 cột **FullName**, **Address**, **Gender** đã được thêm vào table **dbo.AspNetUsers** như hình dưới là OK



4. Bổ sung thông tin của user

- ❖ Bây giờ chúng ta tiến hành cho người dùng nhập những thông tin này khi đăng ký thành viên.
- ❖ **Bước 1:** Tìm đến class `RegisterViewModel` trong file `Models/AccountViewModels.cs` và bổ sung các lệnh được tô đen như ở slide sau.

4. Bổ sung thông tin của user

```
public class RegisterViewModel
{
    [Required(ErrorMessage = "Bạn chưa nhập email")]
    [EmailAddress(ErrorMessage = "Email không hợp lệ, xin vui lòng kiểm tra lại")]
    [Display(Name = "Email")]
    public string Email { get; set; }
    [Required(ErrorMessage = "Bạn chưa nhập mật khẩu")]
    [StringLength(100,
        ErrorMessage= "Mật khẩu tối thiểu {2} và tối đa {1} ký tự", MinimumLength=6)]
    [DataType(DataType.Password)]
    [Display(Name = "Mật khẩu")]
    public string Password { get; set; }
    [DataType(DataType.Password)]
    [Display(Name = "Xác nhận mật khẩu")]
    [Compare("Password", ErrorMessage="Mật khẩu và xác nhận mật khẩu không khớp.")]
    public string ConfirmPassword { get; set; }
    [Display(Name = "Họ tên")]
    [Required(ErrorMessage = "Bạn chưa nhập họ tên")]
    public string FullName { get; set; }
    [Display(Name = "Địa chỉ")]
    [Required(ErrorMessage = "Bạn chưa nhập địa chỉ")]
    public string Address { get; set; }
    [Display(Name = "Giới tính")]
    public bool? Gender { get; set; }
}
```


4. Bổ sung thông tin của user

- ❖ **Bước 2:** Mở file **Views/Account/Register.cshtml** và thêm code xử lý vào vị trí chú thích như sau:

```
Register.cshtml* + X
1 @model DemoASP.NETIdentity.Models.RegisterViewModel
2 @{
3     ViewBag.Title = "Register";
4 }
5 <h2>@ViewBag.Title.</h2>
6 <div class="text-success">
7     <p>
8         @ViewBag.Message
9     </p>
10 </div>
11 @using (Html.BeginForm("Register", "Account", FormMethod.Post, new { @class = "form-horizontal", role = "form" }))
12 {
13     @Html.AntiForgeryToken()
14     <h4>Create a new account.</h4>
15     <hr />
16     @Html.ValidationSummary("", new { @class = "text-danger" })
17
18     //Chèn code để tạo 3 field mới vào vị trí này
19
20     <div class="form-group">
21         @Html.LabelFor(m => m.Email, new { @class = "col-md-2 control-label" })
22         <div class="col-md-10">
23             @Html.TextBoxFor(m => m.Email, new { @class = "form-control" })
24         </div>
25     </div>
```

4. Bổ sung thông tin của user

//Chèn code để tạo 3 field mới vào vị trí này

```
<div class="form-group">
  @Html.LabelFor(m => m.FullName, new { @class = "col-md-2 control-label" })
  <div class="col-md-10">
    @Html.TextBoxFor(m => m.FullName, new { @class = "form-control" })
  </div>
</div>

<div class="form-group">
  @Html.LabelFor(m => m.Address, new { @class = "col-md-2 control-label" })
  <div class="col-md-10">
    @Html.TextBoxFor(m => m.Address, new { @class = "form-control" })
  </div>
</div>

<div class="form-group">
  @Html.LabelFor(m => m.Gender, new { @class = "col-md-2 control-label" })
  <div class="col-md-10">
    Nam @Html.RadioButtonFor(m => m.Gender.Value, true, new { @checked = "checked" })
    Nữ @Html.RadioButtonFor(m => m.Gender.Value, false)
  </div>
</div>
```

4. Bổ sung thông tin của user

- ❖ **Bước 3:** Mở action method **Register** có attribute **[HttpPost]** trong **AccountController** và sửa câu lệnh:

```
var user = new ApplicationUser { UserName =  
model.Email, Email = model.Email };
```

- ❖ **Thành câu lệnh sau:**

```
var user = new ApplicationUser { UserName =  
model.Email, Email = model.Email, FullName =  
model.FullName, Address = model.Address, Gender  
= model.Gender };
```

4. Bổ sung thông tin của user

Application name

Home

About

Contact

Register.

Create a new account.

Họ tên

Địa chỉ

Giới tính

Nam ☒ Nữ ☐

Email

Mật khẩu

Xác nhận mật khẩu

Register

