

## **Przygotowanie teoretyczne**

### **Przykładowy problem**

Tworząc aplikacje internetowe często napotykamy problem związany z dodaniem możliwości uwierzytelniania użytkowników naszego serwisu. Dodatkowo musimy zapewnić mechanizm ograniczenia dostępu do zasobów dla określonych użytkowników lub grup użytkowników. Uwierzytelnianie użytkowników to nie tylko formularz, w którym wpisuje się nazwę i hasło. Za tym zagadnieniem kryje się szereg innych kwestii: jak i gdzie przechowywać informacje o użytkownikach, jak umożliwić rejestrację, jak ułatwić proces odzyskiwania hasła, jak w końcu ograniczyć dostęp do opcji menu i katalogów aplikacji.

W tym celu wykorzystujemy mechanizmy uwierzytelnienia i autoryzacji. Tworzenie takich mechanizmów nie jest proste i musi być poprzedzone dogłębną analizą problemu. W aplikacjach internetowych zależy nam na szybkim tworzeniu rozwiązań. W celu zapewnienia odpowiedniego poziomu zabezpieczeń możemy skorzystać z mechanizmów dostarczanych przez ASP.NET.

Tworzenie, rejestracja, logowanie, zmiana hasła, przypomnienie hasła to najczęstsze przypadki użycia zarządzania użytkownikami. Tworząc takie formularze od podstaw musimy nie tylko zapewnić odpowiedni interfejs użytkownika, ale również zapewnić odpowiednie zapisanie tych danych do repozytorium. ASP.NET dostarcza nam gotowy zbiór podstawowych kontrolek ułatwiających te zadania.

## Podstawy teoretyczne

### *Uwierzytelnianie użytkowników*

*Uwierzytelnianie* (ang. *authentication*) jest to proces, w którym sprawdza się, czy dany użytkownik jest tym, za kogo się podaje. Proces ten najczęściej wiąże się z podaniem przez użytkownika nazwy i hasła. W ASP.NET wspierane są różne mechanizmy uwierzytelniania:

- uwierzytelnianie Windows
- uwierzytelnianie za pomocą formularza
- uwierzytelnianie przy użyciu Windows Live ID

#### Uwierzytelnianie Windows

Uwierzytelnianie Windows można wykorzystywać, jeśli baza użytkowników znajduje się na kontrolerze domeny Windows. Najważniejszą zaletą takiego uwierzytelnienia jest możliwość korzystania z istniejących kont systemowych oraz to, że nie trzeba pisać dodatkowego kodu zarządzającego uwierzytelnieniem. Serwer IIS identyfikuje użytkownika za pomocą żetonów przyznanych użytkownikowi przy zalogowaniu do serwera.

W celu konfiguracji uwierzytelniania Windows należy w pliku **Web.config** w sekcji `<system.web>` dopisać:

```
<authentication mode="Windows"/>
```

Uwierzytelnianie Windows może być wykorzystywane tylko w przypadku posiadania kont przez użytkowników na serwerze. Taka sytuacja ma miejsce najczęściej w przypadku aplikacji intranetowych. Uwierzytelnianie Windows jest domyślną metodą uwierzytelniania w aplikacjach ASP.NET.

#### Uwierzytelnianie za pomocą formularza

W przypadku gdy projektowana aplikacja ma działać w sieci Internet, posiadanie konta na serwerze przez wszystkich potencjalnych użytkowników aplikacji jest niemożliwe. Dodatkowo istnieje potrzeba przechowywania informacji o użytkownikach w innym miejscu niż system użytkowników



Windows. Najczęściej dane takie przechowywane są w bazie danych. Uwierzytelnianie formularzy umożliwia łatwe i bezpieczne potwierdzanie tożsamości.

Użytkownik w przypadku tej metody uwierzytelniania musi skorzystać ze specjalnej strony do wprowadzenia nazwy i hasła, które następnie jest sprawdzane z danymi zapisanymi w bazie danych, specjalnych plikach lub innych źródłach. W przypadku małej ilości użytkowników możliwe jest przechowywanie tych danych w pliku **Web.config**, jednak dla większych rozwiązań lepiej jest wykorzystać bazę danych.

Standardowo uwierzytelnianie to wykorzystuje plik cookie do przechowywania informacji o uwierzytelnieniu między stronami. Plik taki jest wysyłany do serwera wraz z każdym żądaniem. Istnieje możliwość wykorzystania adresu URL do przechowywania tej informacji w przypadku, kiedy przeglądarka użytkownika ma wyłączoną obsługę ciasteczek.

Poniżej została przedstawiona podstawowa konfiguracja tego typu uwierzytelniania z dodatkowym znacznikiem `<forms>` określającym stronę logowania **Zaloguj.aspx**.

```
<authentication mode="Forms">
  <forms loginUrl="Zaloguj.aspx" />
</authentication>
```

#### Uwierzytelnianie przy użyciu Windows Live ID

*Windows Live ID* to ogólnodostępny, jednolity system, umożliwiający dowolnym użytkownikom – nie tylko klientom Microsoft – korzystanie ze wszystkich stron Microsoft wymagających logowania. Każdy zarejestrowany użytkownik otrzymuje swój indywidualny identyfikator, który pomaga mu w poruszaniu się po wszystkich zasobach Microsoft, na całym świecie. Windows Live ID jest np. konieczny do wzięcia udziału w spotkaniu (konferencji lub seminarium) – dzięki niemu rejestracja uczestników odbywa się łatwiej i szybciej.

Windows Live ID został udostępniony programistom, aby łatwo mogli dołączyć globalne uwierzytelnianie do swoich aplikacji. Dzięki temu użytkownik nie musi posiadać na naszej stronie dodatkowego hasła. Wystarczy, że zaloguje się na dowolnej stronie wspierającej tę technologię i ma dostęp do wszystkich zasobów.

### ***Przechowywanie informacji o użytkownikach w bazie danych***

W przypadku uwierzytelnienia za pomocą formularza ASP.NET wykorzystuje standardowych dostawców `SqlMembershipProvider` i `SqlRoleProvider`. Wykorzystują one bazę danych SQL Server do przechowywania informacji o użytkownikach i rolach. Informacje te są przechowywane w szeregu tabel i dostęp do nich następuje przez procedury składowane. Standardowo informacje te są przechowywane w bazie **ASPNETDB.MDF** w katalogu `App_Data`.

Umieszczając aplikację internetową na serwerze dostawcy mamy dość często ograniczenie ilości baz. Z tego względu, jeśli w aplikacji wykorzystywane są również inne dane, dość często są one dodawane do tej właśnie bazy. Nie jest to rozwiązanie bezpieczne, ponieważ znajomość nazwy bazy danych może ułatwić atak. Dodatkowo jeśli posiadamy już gotowe rozwiązanie bazodanowe i chcemy rozszerzyć naszą aplikację o możliwość uwierzytelniania, możemy te informacje umieścić w naszej bazie danych.

Aby umieścić niezbędne tabele i procedury w naszej bazie, musimy wykorzystać *ASP.NET SQL Server Registration Tool* (**aspnet\_regsql.exe**). Narzędzie to może być używane z linii poleceń lub z poziomu graficznego kreatora. W przypadku użycia linii poleceń mamy większe możliwości konfiguracji narzędzia.

W tym celu musimy uruchomić linię poleceń VS 2008. W systemie Windows XP/2003 musimy wybrać **Start -> Programy -> Microsoft Visual Studio 2008 -> Visual Studio Tools -> Visual Studio 2008 Command Prompt**, a następnie wydać polecenie:



```
aspnet_regsql -A all -C "Data Source=.\SQLEXPRESS;Integrated
Security=True;User Instance=True" -d "X:\Projekt\APP_DATA\moja_baza.mdf"
```

Gdzie X:\Projekt to ścieżka do naszego projektu strony, a moja\_baza.mdf, to plik zawierający bazę danych aplikacji.

Po utworzeniu niezbędnych tabel i procedur należy poinformować dostawców o konieczności korzystania z naszej bazy danych. W tym celu do pliku **Web.config** musimy dodać do sekcji <system.web> następujące informacje:

```
<roleManager enabled="true" defaultProvider="CustomizedRoleProvider">
  <providers>
    <add name="CustomizedRoleProvider"
        type="System.Web.Security.SqlRoleProvider"
        connectionStringName="ConnectionString" />
  </providers>
</roleManager>

<membership defaultProvider="CustomizedMembershipProvider">
  <providers>
    <add name="CustomizedMembershipProvider"
        type="System.Web.Security.SqlMembershipProvider"
        connectionStringName="ConnectionString" />
  </providers>
</membership>
```

Gdzie ConnectionString to nazwa naszego połączenia do bazy skonfigurowana w sekcji connectionStrings.

## **Autoryzacja użytkowników**

**Autoryzacja** (ang. *authorization*) to proces, w którym sprawdzane jest, czy użytkownik o ustalonej wcześniej tożsamości ma prawo dostępu do zasobów, o które prosi. Uprawnienia mogą dotyczyć np. dostępu do katalogu lub pliku.

### **Dostęp do plików**

Autoryzacja dostępu do pliku jest związana z prawami, jakie użytkownik ma w systemie, zatem aby używać tej metody należy stosować uwierzytelnianie Windows. W celu konfiguracji uprawnień należy wykorzystać narzędzia systemu i skonfigurować odpowiedni poziom dostępu dla użytkowników.

### **Dostęp do pliku w obrębie aplikacji**

W przypadku plików w obrębie aplikacji możliwe jest określenie dostępu do pojedynczej strony lub wirtualnego katalogu w aplikacji sieciowej. Możliwe jest użycie tego sposobu autoryzacji ze wszystkimi sposobami uwierzytelnienia.

Dostęp do katalogu można ograniczyć w pliku **Web.config** umieszczonym w głównym katalogu aplikacji lub katalogu, którego dotyczą dane prawa.

W celu ustalenia praw w wybranym katalogu należy utworzyć w nim plik **Web.config**, a następnie w sekcji <authorization> zezwolić (element <allow>) lub zabronić (element <deny>) dostępu do niego określonym użytkownikom lub grupom użytkowników.

Do określania roli dopuszcza się również używanie następujących symboli zastępczych:

- \* – określa dowolnego użytkownika
- ? – określa użytkownika anonimowego

Przykład – zezwolenie dostępu dla użytkowników Adam, Karol oraz Michał i zabronienie dostępu dla użytkowników anonimowych.

```

<authorization>
  <allow users="Adam, Karol, Michal" />
  <deny users="?" />
</authorization>

```

W celu ustalenia praw dla całej aplikacji z określeniem katalogów aplikacji należy w głównym pliku **Web.config** dodać sekcję `<location path="ściezka_do_katalogu">`, a następnie sekcję `<system.web>`, a w niej sekcję `<authorization>`. Przykład:

```

<location path="Admin">
  <system.web>
    <authorization>
      <allow roles="Administrator" />
      <deny users="*" />
    </authorization>
  </system.web>
</location>

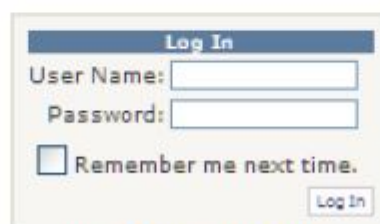
```

### ***Kontrolki logowania***

Kontrolki logowania stanowią zbiór kontrolek serwerowych implementujących wszystkie najważniejsze elementy niezbędne do zarządzania użytkownikami i ich uwierzytelniania.

#### **Kontrolka Login**

Kontrolka Login (Rys. 2) umożliwia uwierzytelnienie użytkownika. Uwierzytelnienie odbywa się z pomocą formularza przy użyciu obiektu `MembershipProvider` zdefiniowanego w pliku **Web.config**.



*Rys. 2 Kontrolka Login*



Mechanizmy uwierzytelniania umożliwiają zapamiętanie użytkownika w pliku cookie, dzięki temu nie musi on za każdym razem wpisywać danych logowania.

#### Kontrolka ChangePassword

Kontrolka ChangePassword (Rys. 3) umożliwia zalogowanemu użytkownikowi zmianę hasła. Udana zmiana hasła powoduje wysłanie do użytkownika wiadomości e-mail.

The image shows a web form titled "Change Your Password" in a blue header bar. Below the header, there are three text input fields. The first is labeled "Password:", the second "New Password:", and the third "Confirm New Password:". At the bottom of the form, there are two buttons: "Change Password" and "Cancel".

Rys. 3 Kontrolka ChangePassword

Wysłanie poczty elektronicznej jest możliwe po wcześniejszym skonfigurowaniu parametrów serwera SMTP w pliku **Web.config**. Można to również zrobić przy pomocy ASP.NET Web Site Administration Tool.



Przykład konfiguracji serwera poczty w pliku **Web.config**:

```
<system.net>  
  <mailSettings>  
    <smtp from="od">  
      <network host="adresSerwera" password="haslo" userName="uzytkownik" />  
    </smtp>  
  </mailSettings>  
</system.net>
```

#### Kontrolka CreateUserWizard

Kontrolka CreateUserWizard (Rys. 4) umożliwia rejestrację nowego użytkownika. Nowy użytkownik musi podać nazwę, hasło, adres e-mail oraz sekretne pytanie i odpowiedź. Sekretne pytanie i odpowiedź służą do odzyskiwania hasła w przypadku, gdy użytkownik je zapomni.

Udane utworzenie użytkownika powoduje wysłanie wiadomości elektronicznej na podany adres.



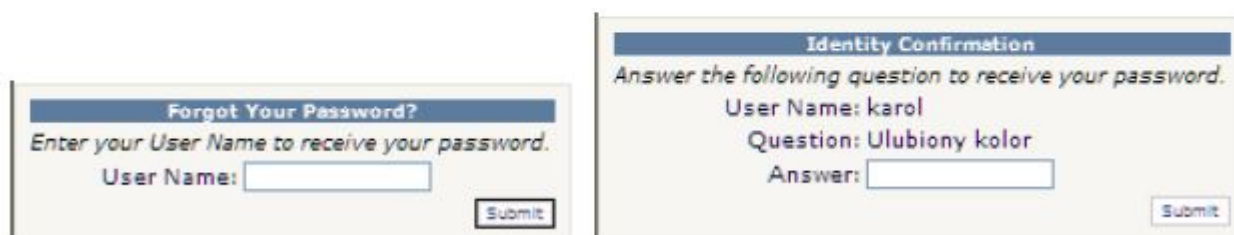
The image shows a web form titled "Sign Up for Your New Account". It contains the following fields and controls:

- User Name:
- Password:
- Confirm Password:
- E-mail:
- Security Question:
- Security Answer:
- Create User:

*Rys. 4 Kontrolka CreateUserWizard*

### Kontrolka PasswordRecovery

Kontrolka PasswordRecovery umożliwia użytkownikowi przypomnienie hasła. Kontrolka po wpisaniu nazwy użytkownika wymaga podania odpowiedzi na sekretne pytanie. Jeśli odpowiedź jest prawidłowa, to wysyłana jest wiadomość zawierająca nowe hasło.



The image displays two sequential windows from the PasswordRecovery control. The first window, titled "Forgot Your Password?", prompts the user to "Enter your User Name to receive your password." It features a text input field for the "User Name" and a "Submit" button. The second window, titled "Identity Confirmation", instructs the user to "Answer the following question to receive your password." It shows the "User Name: karol" and the "Question: Ulubiony kolor". Below this, there is an "Answer:" text input field and a "Submit" button.

Rys. 5 Kontrolka PasswordRecovery

### Kontrolka LoginView

Kontrolka LoginView jest kontenerem umożliwiającym wyświetlenie zawartości w zależności od tego, czy i kto jest zalogowany.

Standardowo kontrolka rozpoznaje użytkownika anonimowego (<AnonymousTemplate>) i zalogowanego (<LoggedInTemplate>). Dla każdego ze stanów możliwe jest zdefiniowanie osobnej zawartości. Przykład:

```
<asp:LoginView ID="LoginView1" runat="server">
  <LoggedInTemplate>
    Zawartość dla użytkownika zalogowanego
  </LoggedInTemplate>
  <AnonymousTemplate>
    Zawartość dla użytkownika anonimowego
  </AnonymousTemplate>
</asp:LoginView>
```

### Kontrolka LoginStatus

Kontrolka LoginStatus umożliwia wyświetlenie konfigurowalnych odnośników w zależności od tego, czy użytkownik jest zalogowany (Logout) czy anonimowy (Login). Jeśli użytkownik nie jest zalogowany, to po kliknięciu na odnośnik **Login** zostaje przekierowany na stronę logowania zdefiniowaną w pliku **Web.config** w atrybucie loginUrl sekcji <authentication>. Jeśli użytkownik jest zalogowany, to kliknięcie przycisku Logout zostaje wylogowany i przekierowany na stronę podaną w atrybucie LogoutPageUrl kontrolki. Przykład:

```
<asp:LoginStatus ID="LoginStatus1" runat="server"
  LogoutAction="Redirect" LogoutPageUrl="Default.aspx" />
```

### Kontrolka LoginName

Kontrolka LoginName wyświetla nazwę użytkownika. Jest wykorzystywana najczęściej do wyświetlenia komunikatu powitalnego dla użytkownika. Przykład:

```
Witaj <asp:LoginName ID="LN1" runat="server" />!
```



### *Ukrywanie opcji menu*

W aplikacjach internetowych zachodzi często potrzeba ukrywania pewnych elementów nawigacyjnych przed użytkownikami, którzy nie mają odpowiednich uprawnień, np. wybranych opcji menu.

Aby tego dokonać, należy dodać atrybut `roles` do pliku **Web.sitemap** oraz w pliku **Web.config** ustawić właściwość `securityTrimmingEnabled` obiektu `XmlSiteMapProvider` na wartość `true`.

Należy pamiętać o tym, że ukrycie przed użytkownikiem odnośnika do strony nie zabrania mu do niej dostępu. Aby ograniczyć dostęp do folderu lub pliku, należy skonfigurować sekcję `<authorization>` w pliku **Web.config**.

Koniecznym zatem jest jawne określenie roli jednocześnie na poziomie uprawnień do folderów (**Web.config**), jak i węzłów, z którymi nie jest skojarzony adres URL (**Web.sitemap**).

#### Konfiguracja atrybutu `roles` w pliku **Web.sitemap**

Konfiguracja atrybutu `roles` odbywa się poprzez dodanie go do sekcji `<siteMapNode>` i przypisanie do niego odpowiednich ról. Przykład:

```
<siteMapNode
  title="Administracja"
  description="Zarządzaj witryną"
  roles="Administrator">
  ...
</siteMapNode>
```

#### Konfiguracja pliku **Web.config**

W pliku **Web.config** w sekcji `<system.web>` należy dodać domyślnego dostawcę mapy serwisu `XmlSiteMapProvider`. Przykład:

```
<siteMap defaultProvider="XmlSiteMapProvider" enabled="true" >
  <providers>
```

```
<add
  name="XmlSiteMapProvider"
  description="Domyślny dostawca mapy serwisu."
  type="System.Web.XmlSiteMapProvider"
  siteMapFile="Web.sitemap"
  securityTrimmingEnabled="true" />
</providers>
</siteMap>
```

## Dodatkowe źródła informacji

1. Scott Mitchell, *Examining ASP.NET 2.0's Membership, Roles, and Profile*, <http://aspnet.4guysfromrolla.com/articles/120705-1.aspx>

12-częściowy artykuł poruszający praktyczne zagadnienia związane z zarządzaniem użytkownikami, rolami i profilami. Nie tylko dobrze przedstawiona teoria, ale przede wszystkim porady praktyczne i rozwiązania najczęściej spotykanych problemów.

2. Jakub Zagórski, *Portal Internetowy w ASP.NET 2.0 – z czym to się je?*, <http://www.codeguru.pl/article-560.aspx>

Artykuł dotyczący wielu aspektów związanych z tworzeniem portalu internetowego, z ciekawym opisem kwestii związanych z uwierzytelnieniem.

3. *How To: Use Membership in ASP.NET 2.0*, <http://msdn.microsoft.com/en-us/library/ms998347.aspx>

Obszerny artykuł zespołu patterns & practices przedstawiający kwestie wykorzystania uwierzytelniania w aplikacjach ASP.NET.

4. Stefan Schackow, *Professional ASP.NET 2.0 Security, Membership, and Role Management*, Wrox, 2006

Autor w książce poruszył wiele zagadnień związanych z bezpieczeństwem, włącznie z zabezpieczaniem konfiguracji. Książka zawiera szczegółowe informacje dotyczące wszystkich najważniejszych obszarów bezpieczeństwa aplikacji ASP.NET.



## Laboratorium podstawowe

### Problem 1 (czas realizacji 20 min)

Przygotowujesz aplikację internetową dla firmy Adventure Works, która planuje ekspansję na rynku internetowym w Polsce. Aktualnie aplikacja umożliwia już przeglądanie produktów oraz zawiera kilka ciekawych elementów ułatwiających sprzedaż produktów. Kolejnym krokiem w rozwoju aplikacji jest implementacja uwierzytelniania jej użytkowników. W bazie dostarczonej do rozbudowy istnieją już informacje o użytkownikach, jednak ze względu na krótki czas implementacji aplikacji Twój zespół zaproponował wykorzystanie wbudowanych mechanizmów uwierzytelniania. Na szczęście klient się zgodził. Twój zespół sprawdził, że mechanizm ten potrzebuje bazy danych ASPNETDB. Niestety na serwerze wykupionym przez klienta nie ma możliwości dodania drugiej bazy danych. Musisz zatem zintegrować bazę uwierzytelniania z istniejącą bazą danych oraz utworzyć dwóch użytkowników: **admin** i **karol**.

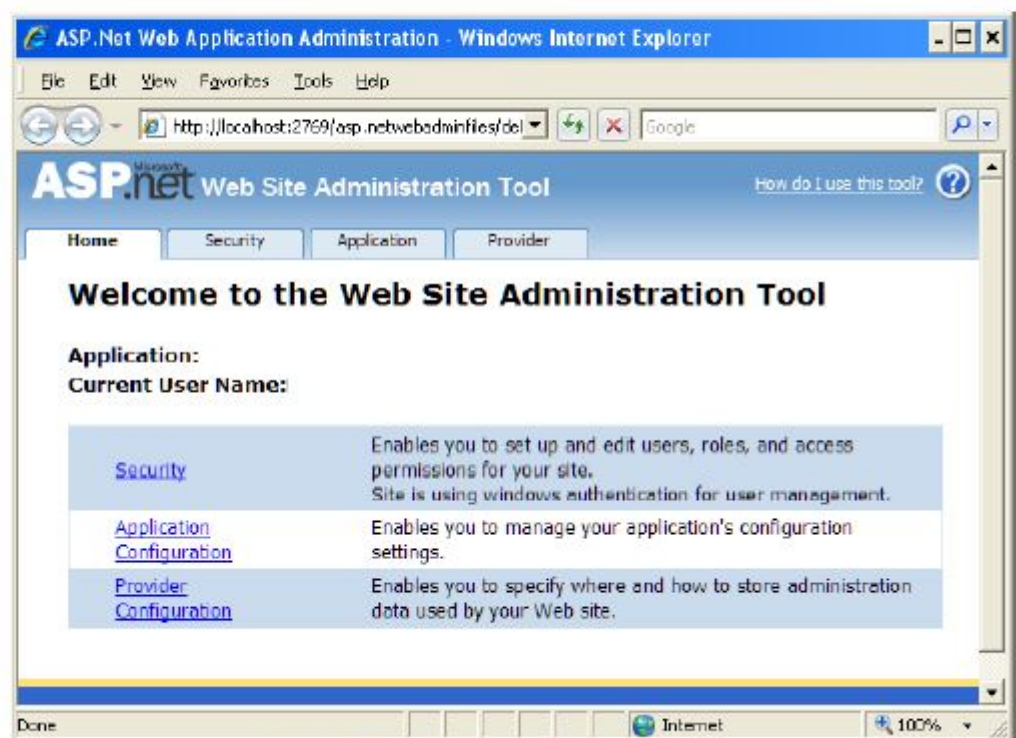
Zadanie	Tok postępowania
1. Dodaj tabele i procedury z uwierzytelniające do istniejącej bazy danych	<ul style="list-style-type: none"><li>• Otwórz aplikację internetową przygotowaną w poprzednim module.</li><li>• W oknie <b>Solution Explorer</b> wybierz bazę danych <b>AdventureWorksLT_Data.mdf</b> znajdującą się w katalogu <b>App_Data</b>. Skopiuj ścieżkę dostępu znajdującą się w oknie <b>Properties</b> w polu <b>FullPath</b>.</li><li>• W systemie Windows XP/2003 wybierz <b>Start -&gt; Programy -&gt; Microsoft Visual Studio 2008 -&gt; Visual Studio Tools -&gt; Visual Studio 2008 Command Prompt</b>.</li><li>• W linii poleceń wpisz nie kopiuj: <code>aspnet_regsql -A all -C "Data Source=.\SQLEXPRESS;Integrated Security=True;User Instance=True" -d</code></li><li>• Dodaj na końcu wpisanego polecenia odstęp, a następnie wklej skopiowaną ścieżkę (kliknij prawym przyciskiem myszy, a następnie wybierz <b>Wklej</b>).</li></ul>

<p>2. Dodaj informację o korzystaniu z innej bazy do pliku Web.config</p>	<ul style="list-style-type: none"> <li>• Otwórz plik <b>Web.config</b>.</li> <li>• Do sekcji <b>&lt;system.web&gt;</b> dodaj następujące informacje: <pre> &lt;roleManager enabled="true" defaultProvider="CustomizedRoleProvider"&gt;   &lt;providers&gt;     &lt;add name="CustomizedRoleProvider"       type="System.Web.Security.SqlRoleProvider"       applicationName="AdventureWorks"       connectionStringName="ConnectionString" /&gt;   &lt;/providers&gt; &lt;/roleManager&gt;  &lt;membership defaultProvider="CustomizedMembershipProvider"&gt;   &lt;providers&gt;     &lt;add name="CustomizedMembershipProvider"       type="System.Web.Security.SqlMembershipProvider"       applicationName="AdventureWorks"       connectionStringName="ConnectionString" /&gt;   &lt;/providers&gt; &lt;/membership&gt; </pre> </li> <li>• Odnajdź sekcję <b>&lt;connectionStrings&gt;</b>, a następnie znacznik <b>&lt;add&gt;</b>. Skopiuj zawartość właściwości <b>name</b> i wklej dwukrotnie w miejsce <b>ConnectionString</b> (wyróżnione miejsca w kodzie powyżej) w dodanej</li> </ul>
---	---

uprzednio fragmencie konfiguracji.

3. Uruchom narzędzie administracyjne

- Uruchom **ASP.NET Web Site Administration Tool**, wybierając z menu głównego **Websites -> ASP.NET Configuration**.
- Zostanie uruchomiona przeglądarka i wyświetli się strona przedstawiona na Rys. 6.



Rys. 6 ASP.NET Web Site Administration Tool



<p>4. Dodaj użytkowników do aplikacji</p>	<ul style="list-style-type: none"> <li>• Wybierz z menu zakładkę <b>Security</b>, a następnie kliknij odnośnik <b>Use the security Setup Wizard to configure security step by step</b>. Naciśnij przycisk <b>Next</b>.</li> <li>• W kroku drugim (<b>Select Access Method</b>) określ metodę dostępu, zaznaczając opcję <b>From the Internet</b>. Naciśnij przycisk <b>Next</b>.</li> <li>• W kroku trzecim (<b>Data Store</b>) naciśnij przycisk <b>Next</b>.</li> <li>• W kroku czwartym (<b>Define Roles</b>) zaznacz opcję <b>Enable roles for this Web site</b>. Naciśnij przycisk <b>Next</b>. W polu tekstowym <b>New Role Name</b> wpisz: <ul style="list-style-type: none"> <li>– <b>Administrator</b> i wciśnij <b>Add Role</b></li> <li>– <b>User</b> i wciśnij <b>Add Role</b></li> </ul> </li> <li>• Naciśnij przycisk <b>Next</b>.</li> <li>• W kroku piątym (<b>Add New Users</b>) dodaj nowego użytkownika, podaj następujące dane: <ul style="list-style-type: none"> <li>– <b>User Name:</b> admin</li> <li>– <b>Password:</b> Pa\$\$word</li> <li>– <b>Confirm Password:</b> Pa\$\$word</li> <li>– <b>E-mail:</b> {adres e-mail}</li> <li>– <b>Security Question:</b> Ulubiony kolor</li> <li>– <b>Security Answer:</b> zielony</li> </ul> </li> <li>• Naciśnij przycisk <b>Create User</b>. Dodaj kolejnego użytkownika: <ul style="list-style-type: none"> <li>– <b>User Name:</b> karol</li> <li>– <b>Password:</b> Pa\$\$word</li> </ul> </li> </ul>
---	---

	<ul style="list-style-type: none"> <li>– <b>Confirm Password:</b> Pa\$\$word</li> <li>– <b>E-mail:</b> {adres e-mail}</li> <li>– <b>Security Question:</b> Ulubiony kolor</li> <li>– <b>Security Answer:</b> zielony</li> </ul> <ul style="list-style-type: none"> <li>• Naciśnij przycisk <b>Create User</b>, a następnie przycisk <b>Next</b>.</li> <li>• W kroku szóstym (<b>Add New Access Rules</b>), pozwalającym określić prawa do katalogów dla poszczególnych użytkowników lub grup, naciśnij przycisk <b>Next</b>.</li> <li>• Krok siódmy informuje o zakończeniu procesu konfiguracji zabezpieczeń. Naciśnij przycisk <b>Finish</b>.</li> <li>• Na zakładce <b>Security</b> sprawdź, czy ilość użytkowników (<b>Users</b>) wynosi 2 oraz ilość ról (<b>Roles</b>) – 2.</li> </ul>
5. Przypisz konta użytkowników do ról	<ul style="list-style-type: none"> <li>• W zakładce <b>Security</b> kliknij łącze <b>Manage users</b>.</li> <li>• Przy użytkowniku <b>admin</b> kliknij łącze <b>Edit Roles</b> i przypisz obie role do użytkownika zaznaczając pola wyboru.</li> <li>• Przy użytkowniku <b>karol</b> kliknij łącze <b>Edit Roles</b> i przypisz rolę <b>User</b> do użytkownika zaznaczając odpowiednie pole wyboru.</li> </ul>

6. Skonfiguruj konto pocztowe	<ul style="list-style-type: none"> <li>Wybierz zakładkę <b>Application</b>, a następnie kliknij łącze <b>Configure SMTP e-mail settings</b>. Skonfiguruj ustawienia serwera poczty wysyłającej, wpisując adres dowolnego serwera pocztowego, z którego na co dzień korzystasz: <ul style="list-style-type: none"> <li><b>Server Name:</b> adres serwera poczty wysyłającej</li> <li><b>Server Port:</b> port serwera poczty wysyłającej</li> <li><b>From:</b> Informacja, która pojawi się w liście jako <b>Od:</b> (niektóre serwery blokują podawanie innych nazw, niż nazwa konta)</li> <li><b>Authentication:</b> zaznacz <b>Basic</b></li> <li><b>Sender's user name:</b> nazwa użytkownika poczty</li> <li><b>Sender's password:</b> hasło użytkownika</li> </ul> </li> </ul> <p> Pamiętaj, że wszystkie dane konta, w tym hasło, zostaną umieszczone w postaci niezakodowanej w pliku <b>Web.config</b>.</p>
-------------------------------	---

## Problem 2 (czas realizacji 20 min)

Po dodaniu i skonfigurowaniu użytkowników czas na dodanie do aplikacji możliwości logowania. Na stronie **Zaloguj.aspx** użytkownik ma mieć możliwość podania danych do uwierzytelnienia lub przekierowania na stronę **Odzyskiwanie.aspx**, na której będzie mógł odzyskać zapomniane hasło. Użytkownik ma mieć również możliwość utworzenia konta w systemie na stronie **Zarejestruj.aspx**. Na stronie **ZmianaHasla.aspx** użytkownik zalogowany ma mieć możliwość zmiany swojego dotychczasowego hasła.

Zadanie	Tok postępowania
1. Dodaj wymagane strony do aplikacji	<ul style="list-style-type: none"> <li>Wybierz <b>Website</b> -&gt; <b>Add New Item</b>.</li> <li>Z listy <b>Visual Studio installed templates</b> wybierz <b>Web Form</b>.</li> <li>W polu <b>Name</b> wpisz <b>Zarejestruj.aspx</b>.</li> <li>Z listy <b>Language</b> wybierz <b>Visual C#</b>.</li> <li>Upewnij się, że opcja <b>Place code in separate file</b> jest zaznaczona.</li> <li>Upewnij się, że opcja <b>Select master page</b> jest zaznaczona.</li> <li>Kliknij <b>OK</b>.</li> <li>Podobnie dodaj stronę <b>Zaloguj.aspx</b></li> </ul>



2. Dodaj kontrolkę LoginView	<ul style="list-style-type: none"> <li>• Na stronie wzorcowej <b>SzablonStrony.master</b> w obszarze <b>div</b> o <b>ID="ObszarLogowania"</b> umieść kontrolkę <b>LoginView</b>.</li> <li>• Kliknij Smart Tag i wybierz <b>AnonymousTemplate</b>. W kontrolce wpisz <b>Zarejestruj</b>, zaznacz wpisany tekst i naciśnij <b>Ctrl+L</b> lub wybierz <b>Format-&gt;Convert to Hyperlink</b>. Kliknij przycisk <b>Browse</b> i wybierz plik <b>Zarejestruj.aspx</b>. Naciśnij <b>OK</b>. Ponownie naciśnij <b>OK</b>.</li> <li>• Kliknij Smart Tag i wybierz <b>LoggedInTemplate</b>. W kontrolce wpisz <b>"Witaj "</b>, a następnie dodaj kontrolkę <b>LoginName</b>.</li> </ul>
3. Dodaj kontrolkę LoginStatus	<ul style="list-style-type: none"> <li>• Na stronie wzorcowej <b>SzablonStrony.master</b> w obszarze <b>div</b> o <b>ID="ObszarLogowania"</b> za kontrolką <b>LoginView</b> wpisz symbol <b> </b>, a następnie umieść kontrolkę <b>LoginStatus</b>. W oknie <b>Properties</b>: <ul style="list-style-type: none"> <li>– w polu <b>CssClass</b> wpisz <b>LinkObszaruLogowania</b></li> <li>– w polu <b>LoginText</b> wpisz <b>Zaloguj</b></li> <li>– w polu <b>LogoutAction</b> wpisz <b>Redirect</b></li> <li>– w polu <b>LogoutPageUrl</b> wpisz <b>~/Default.aspx</b></li> <li>– w polu <b>LogoutText</b> wpisz <b>Wyloguj</b></li> </ul> </li> <li>• Do pliku <b>Style.css</b> dodaj definicję klasy <b>LinkObszaruLogowania</b>: <pre>.LinkObszaruLogowania { color: #000033; }</pre> </li> </ul>
4. Określ stronę logowania	<ul style="list-style-type: none"> <li>• W pliku <b>Web.config</b> zamień znacznik <b>&lt;authentication mode="Forms" /&gt;</b> na: <pre>&lt;authentication mode="Forms"&gt;   &lt;forms loginUrl="~/Zaloguj.aspx" /&gt; &lt;/authentication&gt;</pre> </li> </ul>

<p>5. Dodaj kontrolkę logowania</p>	<ul style="list-style-type: none"> <li>• Na stronie <b>Zaloguj.aspx</b> w widoku <b>Design</b> napisz <b>Zaloguj się w serwisie</b>, a następnie umieść kontrolkę <b>Login</b>. W oknie <b>Properties</b>: <ul style="list-style-type: none"> <li>– w polu <b>FailureText</b> wpisz <b>Logowanie nie powiodło się. Upewnij się, że poprawnie wpisałeś nazwę użytkownika i hasło.</b></li> <li>– w polu <b>LoginButtonText</b> wpisz <b>Zaloguj</b></li> <li>– w polu <b>PasswordLabelText</b> wpisz <b>Hasło:</b></li> <li>– w polu <b>PasswordRequiredErrorMessage</b> wpisz <b>Wprowadź hasło.</b></li> <li>– w polu <b>RememberMeText</b> wpisz <b>Zapamiętaj mnie</b></li> <li>– w polu <b>UserNameLabelText</b> wpisz <b>Login:</b></li> <li>– w polu <b>UserNameRequiredErrorMessage</b> wpisz <b>Wprowadź nazwę.</b></li> </ul> </li> <li>• Do strony w widoku <b>Source</b> dodaj: <pre>&lt;a href="Odzyskiwanie.aspx"&gt;Zapomniałem hasła&lt;/a&gt;</pre> </li> <li>• Zapisz zmiany na stronie.</li> </ul>
<p>6. Dodaj kontrolkę tworzenia użytkownika</p>	<ul style="list-style-type: none"> <li>• Na stronie <b>Zarejestruj.aspx</b> w widoku <b>Design</b> umieść kontrolkę <b>CreateUserWizard</b>. W oknie <b>Properties</b> zmień właściwości tak, aby w kontrolce były wyświetlane komunikaty w języku polskim.</li> <li>• Zapisz zmiany na stronie.</li> </ul>
<p>7. Dodaj kontrolkę odzyskiwania hasła</p>	<ul style="list-style-type: none"> <li>• Do aplikacji dodaj stronę <b>Odzyskiwanie.aspx</b> opartą o szablon strony <b>SzablonStrony.master</b>. W widoku <b>Design</b> umieść kontrolkę <b>PasswordRecovery</b>. W oknie <b>Properties</b> zmień właściwości tak, aby w kontrolce były wyświetlane komunikaty w języku polskim.</li> <li>• Zapisz zmiany na stronie.</li> </ul>



8. Dodaj kontrolkę zmiany hasła	<ul style="list-style-type: none"> <li>• Do projektu dodaj katalog <b>Zarzadzanie</b>.</li> <li>• Do katalogu dodaj stronę <b>ZmianaHasla.aspx</b> opartą na szablonie strony <b>SzablonStrony.master</b>.</li> <li>• Na stronie napisz <b>Zmień hasło</b> i dodaj kontrolkę <b>ChangePassword</b>. W oknie <b>Properties</b> zmień właściwości tak, aby w kontrolce były wyświetlane komunikaty w języku polskim.</li> </ul>
9. Sprawdź działanie aplikacji	<ul style="list-style-type: none"> <li>• Sprawdź poprawność działania aplikacji.</li> <li>• Zaloguj się korzystając z użytkownika <b>karol</b>.</li> <li>• Dodaj nowego użytkownika i sprawdź możliwość zalogowania na podane dane.</li> </ul>



**Problem 3 (czas realizacji 5 min)**

Ostatnim elementem związanym z zabezpieczeniami jest ukrycie opcji menu oraz zabezpieczenie katalogu przed dostępem niepowołanych użytkowników. Menu **Zarządzanie** powinno być widoczne tylko dla użytkowników znajdujących się w roli **User** i **Administrator**. Pozostali użytkownicy nie powinni mieć dostępu do katalogu.

Zadanie	Tok postępowania
1. Uzupełnij brakujące strony w pliku Web.sitemap	<ul style="list-style-type: none"><li>Otwórz plik <b>Web.sitemap</b>. Po sekcji wyświetlającej menu dla produktów dodaj: <pre>&lt;siteMapNode title="Zarządzanie" description="Zarządzaj"&gt;   &lt;siteMapNode url="Zarządzanie/ZmianaHasła.aspx"     title="Zmiana hasła" description="Zmień hasło" /&gt; &lt;/siteMapNode&gt;</pre></li></ul>
2. Dodaj atrybut roles do pliku Web.sitemap	<ul style="list-style-type: none"><li>Dodaj do znacznika <b>siteMapNode</b> atrybut <b>roles="*"</b>, gdy menu ma być prezentowane dla wszystkich użytkowników strony (nawet anonimowych).</li><li>Dodaj do znacznika <b>siteMapNode</b> atrybut <b>roles="X"</b>, gdzie <b>X</b> jest nazwą roli (możliwe podanie wielu ról oddzielonych przecinkiem np. <b>Administrator, User</b>), gdy menu ma być prezentowane dla konkretnych użytkowników przypisanych do roli <b>X</b>.</li><li>Ustal odpowiednie prawa do menu. Dla węzła <b>siteMapNode</b> o właściwości:<ul style="list-style-type: none"><li><b>title="Produkty"</b> ustal <b>roles="*"</b></li><li><b>title="Zarządzanie"</b> ustal <b>roles="User,Administrator"</b></li><li><b>title="Ankiety"</b> ustal <b>roles="*"</b></li><li><b>title="Informacje"</b> ustal <b>roles="*"</b></li></ul></li></ul>
3. Określ dostawcę informacji o ścieżce nawigacyjnej	<ul style="list-style-type: none"><li>Do pliku <b>Web.config</b> do znacznika <b>&lt;system.web&gt;</b> dodaj następujący fragment: <pre>&lt;siteMap defaultProvider="XmlSiteMapProvider" enabled="true" &gt;   &lt;providers&gt;     &lt;add name="XmlSiteMapProvider"       description="Domyślny dostawca mapy serwisu."       type="System.Web.XmlSiteMapProvider"       siteMapFile="Web.sitemap"       securityTrimmingEnabled="true" /&gt;   &lt;/providers&gt; &lt;/siteMap&gt;</pre></li></ul>

4. Sprawdź działanie aplikacji	<ul style="list-style-type: none"> <li>• Sprawdź poprawność działania aplikacji.</li> <li>• Sprawdź, które opcje menu są prezentowane.</li> <li>• Zaloguj się korzystając z użytkownika <b>admin</b>.</li> <li>• Sprawdź, które opcje menu są teraz prezentowane</li> <li>• Jeśli jesteś zalogowany, wyloguj się z aplikacji. Sprawdź, czy możesz wywołać stronę <b>Zarządzanie/ZmianaHasla.aspx</b>.</li> </ul> <p> Dlaczego strona została wyświetlona?</p>
5. Zabezpiecz katalog Zarządzanie przed niepowołanym dostępem	<ul style="list-style-type: none"> <li>• Otwórz plik <b>Web.config</b>.</li> <li>• Do sekcji <b>configuration</b> dodaj: <pre> &lt;location path="Zarządzanie"&gt;   &lt;system.web&gt;     &lt;authorization&gt;       &lt;allow roles="User,Administrator" /&gt;       &lt;deny users="*" /&gt;     &lt;/authorization&gt;   &lt;/system.web&gt; &lt;/location&gt; </pre> </li> </ul>
6. Sprawdź działanie aplikacji	<ul style="list-style-type: none"> <li>• Sprawdź poprawność działania aplikacji.</li> <li>• Jeśli jesteś zalogowany, wyloguj się z aplikacji. Sprawdź, czy możesz wywołać stronę <b>Zarządzanie/ZmianaHasla.aspx</b>.</li> </ul> <p> Co się teraz stało? Dlaczego?</p>