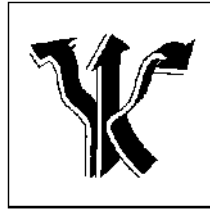


**ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО «ВИШИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«МІЖРЕГІОНАЛЬНА АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ» »**



МАУП

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
ЩОДО ОРГАНІЗАЦІЇ
САМОСТІЙНОЇ РОБОТИ
СТУДЕНТІВ З ДИСЦИПЛІНИ**

**“РОЗРОБКА ІНТЕРНЕТ-ЗАСТОСУВАНЬ ЗА ТЕХНОЛОГІЄЮ
ASP.NET”**

(для магістрів)

Київ 2018

Підготовлено доцентом кафедри комп'ютерних інформаційних систем та технологій М.П.Дяченко.

Затверджено на засіданні кафедри комп'ютерних інформаційних систем та технологій (Протокол № 1 від 23 серпня 2018 р.)

Схвалено Вченою радою Міжрегіональної Академії управління персоналом

Дяченко М.П. Методичні матеріали щодо забезпечення самостійної роботи студентів з дисципліни “Розробка інтернет-застосувань за технологією ASP.NET” (для освітньо-кваліфікаційного рівня «магістр»). — К.: МАУП, 2018— 33 с.

Методична розробка містить пояснювальну записку, тематичний зміст, питання щодо самостійного вивчення студентами і самоконтролю та список літератури.

ПОЯСНЮВАЛЬНА ЗАПИСКА

Технологія ASP.NET є універсальною платформою для розробки web-застосунків корпоративного рівня. Вона підтримує зручну модель та інфраструктуру програмування, що дозволяє реалізовувати масштабовні і захищені клієнт-серверні рішення для мережі інтернет. Вона пропонує web-розробнику також інструментальну підтримку при розв'язанні ряду типових задач web-програмування, як-то:

- конфігурування застосунку системним адміністратором;
- аутентифікація і авторизація користувачів;
- розгортання готового проекту шляхом простого копіювання вмісту розробки в спеціальну папку на сервері;
- забезпечення високої продуктивності внаслідок компіляції коду;
- гнучке кешування сторінок у відповідності до потреб застосунку;
- підтримка національних мов і мобільних пристроїв;
- наявність інструментів відлагодження і тестування;
- підтримка інтеграції web-застосунку з середовищем net framework.

Крім того, завдячуючи компіляції вихідного коду, ASPNET має переваги в швидкості, порівнянно із скриптовими технологіями.

Метою курсу є формування у студента базових принципів розробки інтернет-застосунків з використанням технології ASPNET.

Кращою формою засвоєння теоретичних положень курсу і набуття необхідних практичних навичок зазвичай є власноручне створення web-застосунку. В даній методичній розробці студентам пропонується самостійно пройти через всі фази розробки web-сайту за технологією Asp.NET на прикладі створення сайту працевлаштування. В якості інструменту розробки можуть бути використані комерційна система Microsoft Visual Studio [1], або вільне середовище програмування SharpDevelop [2].

I. ПОСТАНОВКА ЗАДАЧІ

Об'єктом вивчення є технологія розробки сайту працевлаштування за технологією AspNET. Користувачами сайту можуть бути роботодавці та потенційні кандидати на працевлаштування, тобто, сайт має надавати послуги обом категоріям користувачів. Розпочнемо з формулювання змісту функціональних вимог кожної категорії.

1. Функціональність для кандидатів, що шукають роботу:

Аутентифікація и безпека.

Це значить, якщо ви зареєстрований користувач, то ви можете переглянути сторінку, доступну за умовчанням (default.aspx), інакше має бути видане повідомлення про помилку.

В тому випадку, коли користувач звертається до іншої сторінки, що потребує аутентифікації, і він ще не ввів власні дані, має відкритися сторінка login.aspx. Ця сторінка забезпечує механізм аутентифікації.

Пошук пропозицій роботи.

Деталі стосовно можливої роботи мають надаватися іншою категорією користувачів – роботодавцями, при здійсненні пошуку потенційних працівників. Деталі мають зберігатися в окремій базі даних. Пропозиції щодо змісту роботи підбираються в залежності від області професійних інтересів (інформатика, фінанси, тощо.) та рівня кваліфікації кандидата (з використанням технології XML).

Надання резюме

Якщо кандидат є новим користувачем, то він має надати резюме про себе. Після подачі резюме йому має бути надіслане підтвердження на адресу електронної пошти, введеної кандидатом з поштового сервісу (за умовчанням є Outlook).

Для конвертації оцінок кандидата в бали (проценти (%)) може бути використаний відповідний веб-сервіс.

Редагування резюме

Якщо кандидат хоче оновити дані свого резюме, то у відповідній формі повинні відображатись всі попередні дані.

Всі внесені до форми зміни мають бути відіслані кандидату електронною поштою.

Системою має бути забезпечена щомісячна розсилка кандидатам повідомлень про підходящі пропозиції через електронну пошту.

Система має підтримувати лічильник переглядів.

2. Функціональність для роботодавців, яким потрібні кадри.

Перевірка обробки даних та безпека

Зареєстрований роботодавець має доступ до сторінки default page (умовчанням) інакше має бути видане повідомлення про помилку. В тому випадку, коли користувач звертається до іншої сторінки, що потребує аутентифікації, і він ще не ввів власні дані, має відкритися сторінка login.aspx, тобто, також має бути забезпечена дія механізму аутентифікації.

Пошук резюме.

Результати пошуку надаються із бази даних. База даних може переглядатися двома способами:

1. Всі упорядковані дані підряд.
2. Відфільтровані дані – в залежності від поточної потреби.

Заповнення форми клієнта.

Новий клієнт має заповнити вхідну форму (Entry form). Після заповнення йому має бути надіслане підтвердження на адресу електронної пошти, введеної кандидатом з поштового сервісу (за умовчанням є Outlook).

Заповнення даних щодо пропонованої роботи.

Якщо компанія хоче надати вакансію, то деталі про неї потрібно завантажити в базу даних. Компанії, після введення даних має бути

надіслане повідомлення – підтвердження про введену інформацію. Для скорочення списку кандидатів підраховуються зважені оцінки, отримані кандидатами зза результатами конкурсу з використанням веб-сервісу. Також має бути реалізований лічильник переглядів.

3. Веб-сервіси на сайті.

Компаніям слід надати можливість, при виборі кандидатів, користуватись сервісом їх оцінки за результатами конкурсу.

4. Робота сайту.

На головній сторінці (Home Page) всім користувачам надається дві опції:

1. Пошук роботи - Find Work (вхід для тих, хто шукає роботу)
2. Пошук талантів - Find Talent (вхід для компаній)

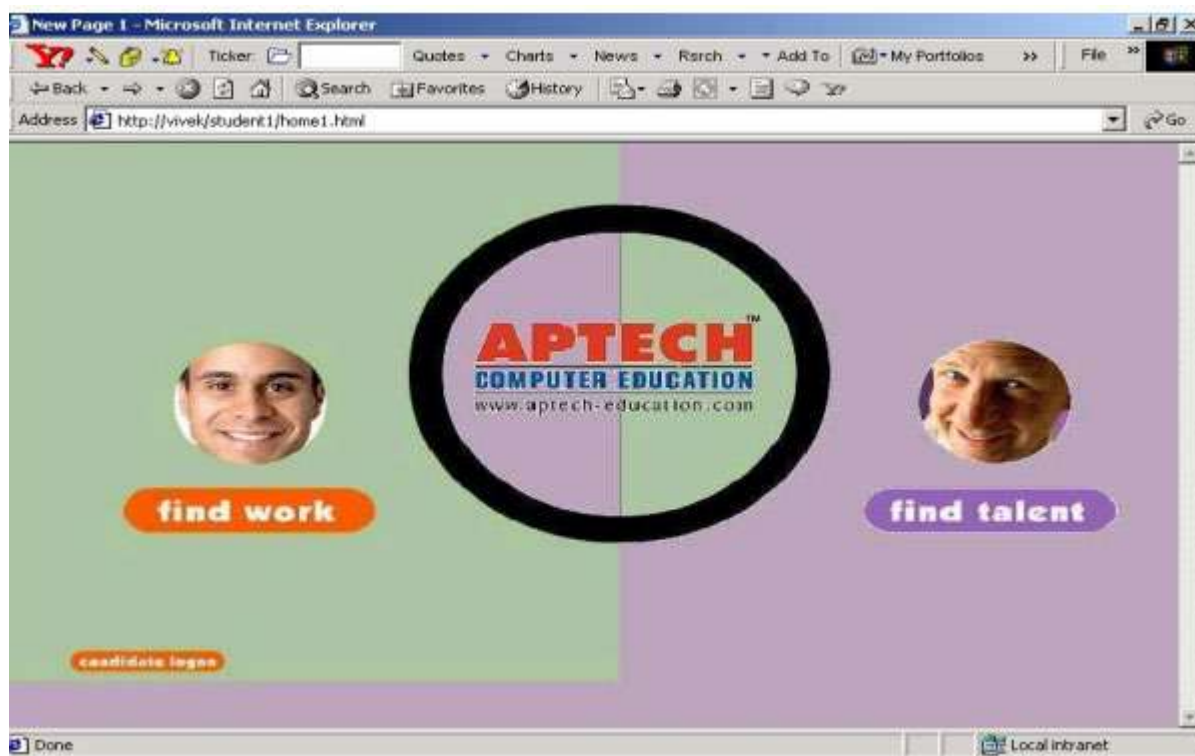


Рис. 1. Головна сторінка (Home page) сайту

Після того, як користувач вибере одну із опцій, він повинен ввести логін і пароль для доступу до детальної інформації. Якщо відвідувач новий, то він має заповнити відповідну форму реєстрації.

В застосунку мають бути передбачені

- перевірка всіх лічильників;
- можливість інформування користувачів про порядок дій, якщо введений пароль неправильний, або користувач його забув;
- захист інформації від незареєстрованих користувачів;
- захист від доступу зареєстрованих користувачів до інших сторінок.

5. Функціональність пропонованої системи

Створювана система повинна забезпечити:

- перевод оцінок CGPA в проценти %;
- використання аутентифікації, що ґрунтується на закладках (cookie);
- надання детальної інформації щодо кандидатів;
- надання детальної інформації щодо компаній;
- доступ до даних, які зберігаються в базі даних SQL server;
- надання резюме;
- редагування резюме;
- пошук роботи он-лайн;
- пошук резюме (інформації про кандидатів) он-лайн;
- доступ до даних про компанії-роботодавці;
- роботу лічильників перегляду.

6. Переваги пропонованої системи:

надання веб-сервісів за стандартами Інтернет;

- розподілене обчислювальне середовище;
глобальний доступ;
- накопичення бази даних про випускників; накопичення бази даних про роботодавців;

надання засобів для ефективного пошуку;

- хороший навчальний зразок.

I. РОЗРОБКА КАРКАСУ САЙТА

У цьому розділі створимо домашню сторінку сайту та заготовки для подальшої роботи. Домашня сторінка показує два посилання:

- “find work” (знайти роботу) та
- “find talent” (знайти таланти).

Ті, хто шукає роботу, натискають ‘Find Work’, а ті, хто пропонує роботу, натискають ‘Find Talent’. Таким чином, домашня сторінка містить два посилання на два розділи веб-сайту (порталу):

- пошук роботи;
- пошук працівників.

Практична частина

1. Створіть HTML-сторінки, як показано на рис. 2, 3, 4. Підходящі рисунки можна знайти в інтернеті, або створити самому за допомогою Photoshop. Сторінку, подібну до рис. 2, збережіть як home.html.



Рис. 2. home.html

Створіть сторінку, подібну до рис. 3, і збережіть її як left_frame.htm.

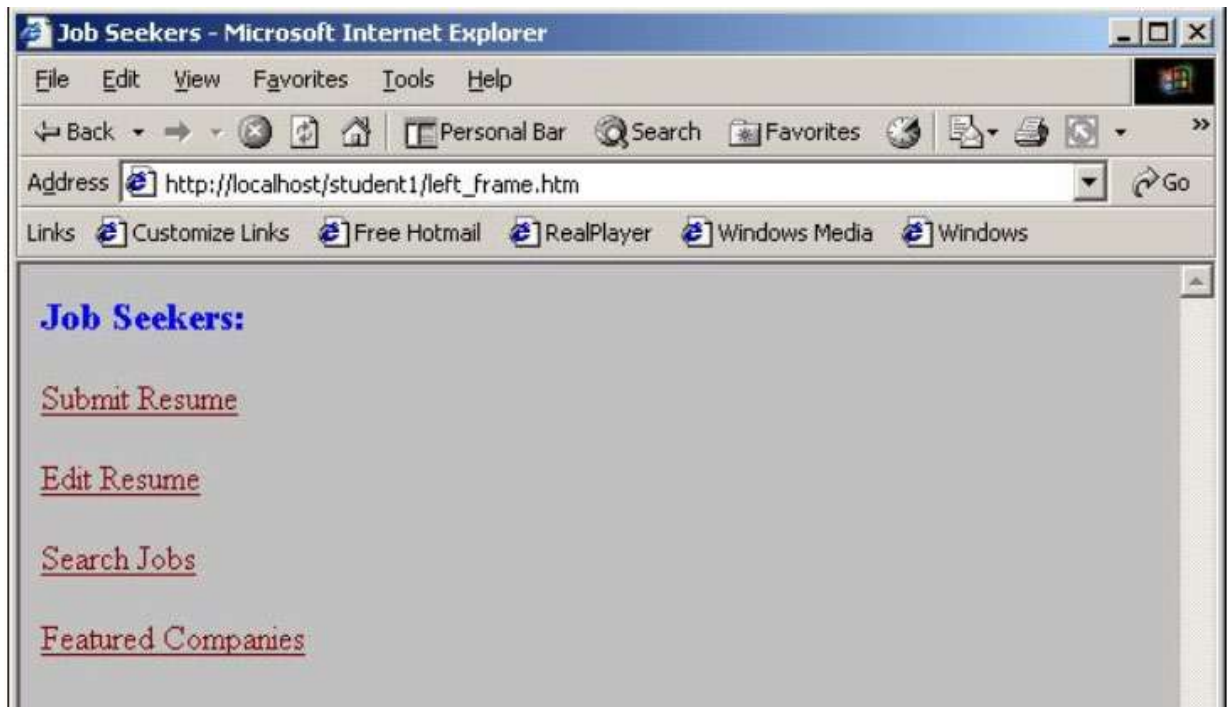


Рис. 3. left_frame.html

Сторінку, за наступним зразком, збережіть як main_frame.html.



Рис. 4. main_frame.html

Тепер оформіть сторінку (construction.html), що виводить текст “This page is under construction” (сторінка в процесі розробки) та зробіть гіперпосилання на неї із усіх заготовок незавершених сторінок.

Контрольні питання

1. Яку сторінку називають динамічною?
2. Які методи для створення динамічних інтерактивних веб-сторінок вам відомі?
3. Які програмні засоби підтримують створення динамічних сторінок?
4. Чим відрізняється динамічна сторінка від статичної?
5. Якими засобами підтримується динамічна сторінка?
6. Чи можна вбудувати код ASP.NET в HTML документ?
7. Чи може користувач проглянути вихідний код сторінки ASP.NET?
8. Що виконується на боці сервера?
9. Що виконується на боці клієнта?
10. Чи вірно, що скрипти серверної частини застосунку виконуються у вікні браузера клієнта?

II.СТВОРЕННЯ СТОРІНКИ ЗА УМОВЧУВАННЯМ

Створимо веб-сторінку під іменем default.aspx, що відображується за умовчужанням, як це показано на рис. 5. Ця aspx сторінка (default.aspx) має містити фрейми, які ділять її на два вікна. Ліве вікно зв'язить із сторінкою left_frame.html, а праве – із сторінкою main_frame.html,

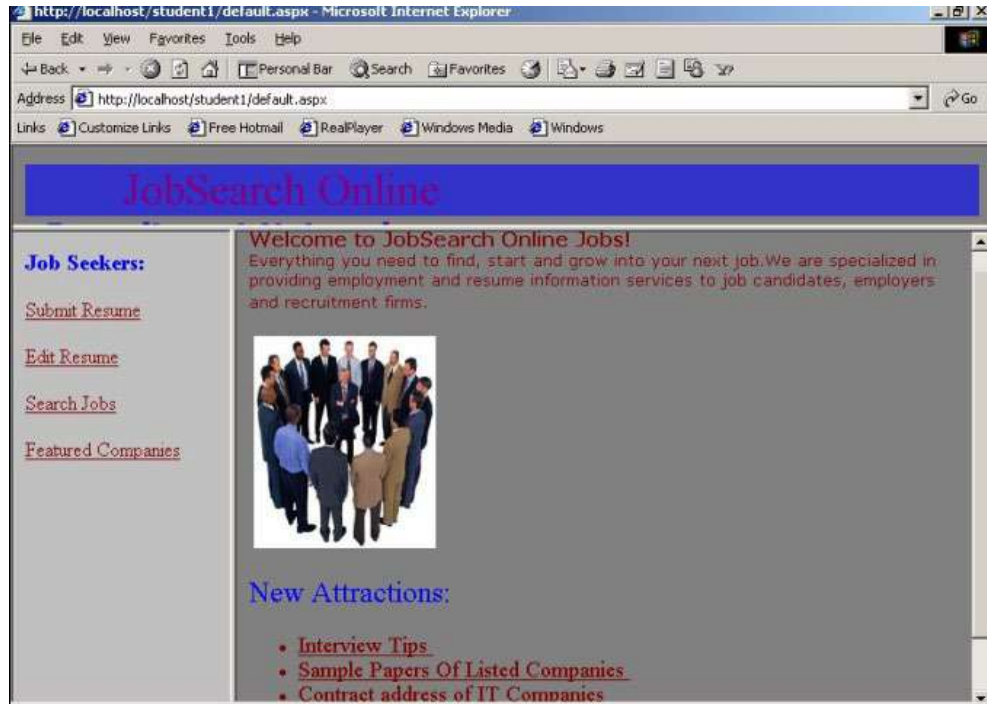


Рис. 5. default.aspx

Доповніть код сторінки “login.aspx” текстом

```
<html>
<script language="C#" runat="server" Debug="true"> </script>
<body bgcolor="#C0C0C0" text="#800080">
  <form method="post" runat="server">Welcome to
  <h3>"JobsSearch OnLine"</h3>
  We are Continuously Working for you.
  <h2>
    Please Enter Login Details :
  </h2>
  <p>
    User Id
    ..... Code for creating a Text box control for User id ....
    Password
    ..... Code for creating a Text box control that accepts
    password .....
  </p>
  <p>
  </p>
  <p>
  <asp:_____ /> // write
```

```

Code for creating a Button control that submits the
</p>
</form>
</body>
</html>

```

щоб вона мала наступний вигляд

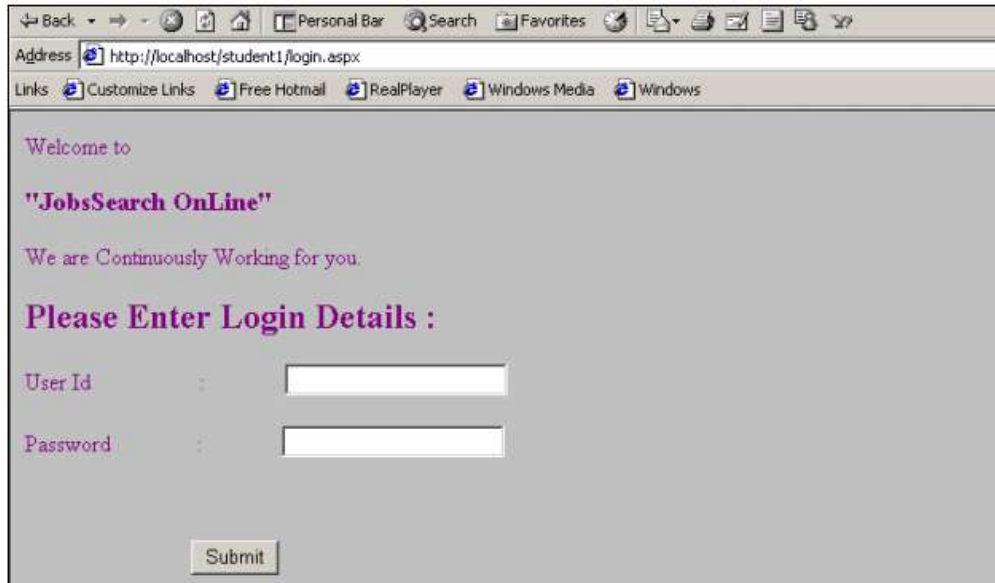


Рис. 6. login.aspx

Зробіть сторінку так, щоб вона показувала повідомлення про помилку при неправильній реєстрації на тій же сторінці, як це показано на рис.



Рис. 7. login.aspx (з повідомленням про помилку)

Контрольні питання

1. Для чого використовується директива @ Page ?
2. Для чого використовується елемент керування _VIEWSTATE ?
3. Перерахуйте основні об'єкти ASP.NET.

III. СТОРІНКА ВВЕДЕННЯ ДАНИХ

Створимо сторінку, вигляд якої показано на рис. 8 і назвемо її ‘entry.aspx’. Ця сторінка буде використовуватись для введення даних і збереження їх в базі даних. Пока не будемо додавати до неї контроль правильності введення та її взаємодію з СКБД (системою керування базою даних).

Зовнішній вигляд цієї сторінки досить великий, щоб вміститись в одному кадрі, тому розділимо її на 2 частини меншого розміру (а і б).

The image shows a web form with a grey background. At the top, there is a heading 'Please Enter Your's Login Details:' in purple. Below this heading, there are three input fields: 'User ID', 'Password', and 'Confirm Password', each with a corresponding label and a colon. Below these fields, there is a section heading 'Personal Information' in purple. This section contains seven fields: 'Title' (a dropdown menu with '--Select One'), 'First Name', 'LastName', 'Gender' (a dropdown menu with '--Select-'), 'Date of Birth' (a text input field with a placeholder '(dd/mm/yyyy)'), 'Nationality', and 'Marital Status' (a dropdown menu with '--Select-').

Рис. 8 (а). entry.aspx

Contact Information

Address :

State :

Pin code :

ContactNo. :

Email Id :

Qualification

Highest Professional qualification :

Have any previous working exp. :

Skills :

Academic Background:

Degree	College/University	from(date)	To(date)	%MarksObtained
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

No. of Jobs Taken in the Past :

No. of years of working experience :

Latest Job Function :

Interested overseas work opportunities in :

Рис. 8 (б). entry.aspx

Контрольні питання

1. Назвіть елемент керування для введення тексту та наведіть приклад його використання.
11. Перерахуйте елементи керування, що забезпечують перемикання між сторінками.
12. Перерахуйте елементи керування для вибору опцій та наведіть приклади їх використання.
13. Наведіть приклади використання елемента керування, який може бути контейнером для інших елементів.
14. Що означає термін «база даних»?
15. Що означає аббревіатура СКБД?
16. Якою мовою можна спілкуватись з базою даних?
17. Назвіть відомі вам типи СКБД, які можна використовувати в ASP.NET.
18. Яке призначення у елемента розмітки `<form>`?
19. Чим відрізняються поняття аутентифікації та авторизації?
20. Яке призначення у cookies файлів?
21. Де зберігаються cookies?

IV. РОЗРОБКА ФОРМИ РЕДАГУВАННЯ

Створіть форму, що перевіряє особисті дані користувача і дозволяє йому маніпулювати інформацією сайту. В цій формі здійснюється контроль імені користувача і пароль, після чого йому дозволено доступ до інформації. Переконайтесь в тому, що виконуються наступні перевірки:

- Поле введення імені користувача не повинно залишатись порожнім;
- Поле введення пароля також не повинно залишатись порожнім.

Для перевірки пароля скористайтесь елементом керування Compare Validator.

Зовнішній вигляд форми редагування зображено на рис. 9.

Відповідну сторінку назвемо “edit_form.aspx”.



The image shows a web form with a grey background. At the top, there is a title in purple text: "Please enter Your's login details to Edit Your Resume". Below the title, there are two rows of labels and input fields. The first row has the label "UserId" in purple, followed by a colon, and a white text input box. The second row has the label "PassWord" in purple, followed by a colon, and another white text input box. At the bottom center of the form, there is a button with the text "Submit to Edit Resume".

Рис. 9. edit_form.aspx

Застосуйте відповідні техніки перевірки елементів керування, які ви вже використовували при створенні сторінки “entry.aspx” вище .

Поради:

- Поля UserId і Password не повинні бути порожніми.

- Дата народження має вводитись у форматі dd/mm/yyyy.
- Номер телефону повинен відповідати зразку: xxx-xx-xxxxxx.
- В стовпчику % Marks Obtained слід обмежити користувача введенням тільки числових значень із інтервалу 0 – 100.
- В адресі електронної пошти повинен бути присутнім значок @.
- Поле Pincode повинно бути числом із 6-ма цифрами.

Дайте відповідь на наступні питання.

Контрольні питання

1. Перерахуйте основні елементи для контролю введення даних в технології ASP.NET.
2. Які значення може сприймати властивість Display елементів перевірки даних та який результат породжують ці значення?
3. Наведіть приклад використання елемента CompareValidator.
4. Наведіть приклад використання елемента RangeValidator.
5. Наведіть приклад використання елемента RegularExpression Validator.

V. СТОРІНКА ІНФОРМУВАННЯ КОРИСТУВАЧІВ

1. Прийшов час створити сторінку, яка відповідає наступним вимогам:
 - показує привітання відвідувачам сайта та інформує про кількість відвідувачів, що заходили на сайт (назвемо її Global.aspx);
 - показує вітальне повідомлення на сторінці реєстрації (login.aspx).
2. Тепер створіть веб-сторінку, що показує ім'я сервера, на якому еревірте виконується застосунок.
3. Перевірте введення користувача і переправте перевірені дані іншій сторінці для обробки (з використанням Server.UrlEncode).

Поради:

Із сторінки входу (login.aspx) перешліть ім'я користувачата пароль на сторінку default.aspx.

4. Створіть веб-сторінку, яка показує шлях до файла, який виконується в даний момент.

Контрольні питання.

1. Перерахуйте основні події об'єкту Application.
2. Наведіть приклад використання змінних рівня застосунку.
3. В якій послідовності викликаються події застосунку, сесії та сторінки при зверненні до сторінки ASP.NET?
4. Де слід прописувати обробників подій застосунку та сесії?
5. Як знаходиться шлях до активного файлу?
6. Як пересилати інформацію від одної сторінки до іншої?
7. Як визначається ім'я сервера?
8. Що таке сесія?
9. Що міститься в cookies файлах?

VI. КЕРУВАННЯ ПЕРЕМІЩЕННЯМ ДАНИХ

На сторінці реєстрації “login.aspx” організуйте прийом гіпотетичного логіну і пароля та збережіть ці дані в якості закладки cookie з парою значень і’мя/пароль.

2. На сторінці “default.aspx” прочитайте значення із закладки и відобразіть їх.

3. Виконайте завдання попередніх пунктів, використовуючи змінні об’єкту session.

4. Створіть веб-сторінку, яка виконує наступні дії:

- Приймає значення валюти від користувача;
- Показує список (listbox) з локальними ідентифікаторами різних країн;
- Показує значення валюти в форматі вибраної країни.

Контрольні питання

1. Як зберігаються дані сесії?
2. Як організувати список?
3. Опишіть використання закладок (cookies)?
4. Перерахуйте параметри cookies.
5. Наведіть приклад використання закладки с єдиним значенням та закладки-словника.
6. Де зберігаються cookies?
7. Охарактеризуйте поняття Session.
8. Наведіть приклад використання змінних сесії.
9. Перерахуйте основні властивості об’єкту Session.
10. Як організувати двосторонню взаємодію між клієнтом і сервером?

VII. СТВОРЕННЯ ПОШУКОВОЇ СТОРІНКИ

1. Створіть пошукову сторінку з іменем “search_job.aspx”, вигляд якої наведено нижче на рис. 10

The image shows a web form with a light gray background. At the top, the text "Please Fill Search Details:" is displayed in a purple font. Below this, there are two rows of labels and dropdown menus. The first row has the label "Job in Area of" in purple, followed by a colon and a dropdown menu containing the text "IT". The second row has the label "Min. Qualification" in purple, followed by a colon and a dropdown menu containing the text "B.Tech". At the bottom center of the form is a rectangular button with the text "Submit to see posts".

Рис. 10. Search_job.aspx

Два списки list box повинні показувати Job Areas (сферу роботи) та Min Qualification (мінімальна кваліфікація). Значення для цих списків мають бути закодовані в програмі.

При натисканні на кнопку Submit ця сторінка повинна викликати сторінку “search_job1.aspx”, яку ми створимо нижче.

2. Наберіть наступний код і збережіть його як сторінку aspx с іменем “search_job1.aspx”. Ми повинні встановити зв'язок так, щоб ця сторінка могла бути викликана із сторінки “Search_job.aspx”.

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SqlClient"%>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System.Xml" %>
<html>
<script language="C#" runat="server">
SqlConnection myConnection;
protected void Page_Load(Object Src, EventArgs E)
{
myConnection = new SqlConnection("server=<Server
name>; uid=sa;pwd=;database=pubs");
```

```

string sCommand = "SELECT company_name, jobarea,
job_title, min_qualification, expertise, experience,
salary, location, mailid, lastdate FROM recruitment
WHERE jobarea = '" + Session["jobarea"].ToString() +
"' AND min_qualification = '" +
Session["min_qualification"].ToString()+ "'";
DataSet ds = new DataSet();
SqlDataAdapter myCommand1 = new SqlDataAdapter();
SqlCommand SqlCom1 = new
SqlCommand(sCommand,myConnection);
myCommand1.SelectCommand = SqlCom1;
myCommand1.Fill(ds,"recruitment");
DataGrid1.DataSource=ds.Tables[0].DefaultView;
DataGrid1.DataBind();
}
</script>
<body bgcolor="#C0C0C0" text="#800080">
<form method="post" runat="server">
<asp:DataGrid id="DataGrid1" runat="server"
/>
</form>
</body>
</html>

```

Модифікуйте запит в наведеному вище коді відповідно до вимог:

- знайти детальну інформацію із таблиці recruitment для роботи;
- 2. коефіцієнт кваліфікації мінімальний “МВА”.
- 3. результат має містити наступні поля:

Company name, experience, salary, location.

Контрольні питання.

1. Опишіть структуру об’єкта DataSet.
2. Охарактеризуйте об’єкт Connection.
3. Наведіть приклади використання елемента DataGrid.

VIII. ЗВ'ЯЗУВАННЯ СТОРІНКИ ВХОДУ З БАЗОЮ ДАНИХ

1. Модифікуйте сторінку входу (login.aspx) так, щоб вона могла діставати ім'я і пароль користувача із бази даних SQL server.

Для цього додайте нову функціональність, включивши в текст код з'єднання з базою даних:

```
protected void Page_Load(Object Src, EventArgs E)
{
//.....Connection string code
}
```

Щоб відкрити зєднання і прочитати дані із бази даних додайте наступний код на сторінку login.page:

```
public void login_click(object sender,
System.EventArgs E)
{
if (Page.IsValid)
{
HttpCookie cookie1 = new HttpCookie("username");
.....
Response.Cookies.Add(cookie1);
.....
.....
.....
string insertCmd = "SELECT password1 FROM
authentication
WHERE userid='" + userid.Text + "'";
SqlCommand myCommand = new
SqlCommand(insertCmd,
myConnection);
.....
.....
.....
.....
}
```

Додайте також відповідний код в метод `login_click()`, щоб порівнювати пароль, введений користувачем с паролем, отриманим із бази даних. Якщо паролі співпадають, здійснити виклик сторінки `default.aspx`.

На сторінку `Edit_form.aspx` додайте наступну функціональність: перевіряти ім'я користувача і пароля, що були отримані із форми і при їх коректних значеннях викликати сторінку `construction.html`

Порада: використовуйте аутентифікацію таблиці.

Контрольні питання.

1. Для чого потрібен об'єкт `DataGridView`?
2. Наведіть приклад використання об'єкта `DataGridView`.
3. Для чого потрібен об'єкт `DataReader`?
4. В яких випадках слід використовувати об'єкт `DataReader`? Наведіть приклад його використання.
5. Для чого потрібен об'єкт `DataAdapter`?
6. В яких випадках слід використовувати об'єкт `DataAdapter`? Наведіть приклад його використання.
7. Що таке ADO.NET?
8. Розкажіть про основні переваги технології ADO.NET.

ІХ. ЗВ'ЯЗУВАННЯ СТОРІНКИ ВХОДУ З SQL SERVER.

1. Для зберігання данх будемо використовувати наступні таблиці:

Personalinfo для особистої іформації

Ім'я стовпчика	Тип даних	Розмір	Опис
userid	Varchar	20	
title	varchar	10	
fname	varchar	20	
lname	Varchar	20	
gender	Varchar	15	
dob	Varchar	15	
nationality	Varchar	15	
m_status	Varchar	10	
Address	Vachar	50	
state	Varchar	15	
Pincode	Varchar	15	
contactno	Varchar	20	
emailid	Varchar	20	

Academics для кваліфікації

Ім'я стовпчика	Тип даних	Розмір	Опис
Naqualification	Varchar	20	
Ftexperience	Varchar	20	
skills	Varchar	100	
No_of_job	Varchar	2	
experience	Varchar	10	
latest_job_fun	Varchar	20	
overseas_opportunities	Varchar	20	
Userid	Varchar	20	

Асабаск для академичного досвіду роботи

Ім'я стовпчика	Тип даних	Розмір	Опис
Userid	Varchar	20	
do1	Varchar	10	
cu1	Varchar	50	
fd1	Varchar	15	
td1	Varchar	15	
mo1	Integer	9	
do2	Varchar	10	
cu2	Varchar	50	
fd2	Varchar	15	
td2	Varchar	15	
mo2	Integer	9	
do3	varchar	10	
cu3	varchar	50	
fd3	varchar	15	
td3	varchar	15	
mo3	Integer	9	

2. Модифікуйте сторінку “search_job1.aspx” так, щоб вона могла читати дані із таблиці recruitment так же як і із файлів XML, та показувати ці дані. Структура файла xml наведена нижче:

```

<NewDataSet>
  <recruitment>
    <company_name>IBM</company_name>
    <jobarea>IT</jobarea>
    <job_title>Consultant</job_title>
    <min_qualification>M.Tech</min_qualification>
    <expertise>Consultancy</expertise>
    <experience>2-5 years</experience>
    <salary>40000</salary>
  </recruitment>
</NewDataSet>

```

```

<location>Bangalore</location>
<mailid>ibmjjobs@aptech.com</mailid>
<lastdate>1/12/2001</lastdate>
</recruitment>
<recruitment>
<company_name>InfoSys</company_name>
<jobarea>IT</jobarea>
<job_title>Educator</job_title>
<min_qualification>Ph. D</min_qualification>
<expertise>Education</expertise>
<experience>3-5 years</experience>
<salary>60000</salary>
<location>Mumbai</location>
<mailid>ibmjjobs@ibm.com</mailid>
<lastdate>3/08/2001</lastdate>
</recruitment>
</NewDataSet>

```

Результат, який має бути отриманий для тих, хто шукає роботу, наведено нижче.

company_name	jobarea	job_title	min_qualification	expertise	experience	salary	location	mailid	lastdate
Aptech	IT	developer	B.Tech	Programming	1-3 years	10000	Mumbai	apjobs@aptech.com	30/09/2001
IBM	IT	Tech Writer	B.Tech	documenting	1-3 years	10000	Bangalore	ibmjjobs@ibm.com	30/08/2001
IBM	IT	Consultant	M.Tech	Consultancy	2-5 years	40000	Bangalore	ibmjjobs@aptech.com	1/12/2001
InfoSys	IT	Educator	Ph. D	Education	3-5 years	60000	Mumbai	ibmjjobs@ibm.com	3/08/2001

Рис 11.

Таблиця Recruitment для опису видів роботи:

Ім'я стовпчика	Тип даних	Розмір	Опис
company_name	Varchar	20	
Jobarea	Varchar	20	
job_title	Varchar	20	
min_qualification	Varchar	20	
expertise	Varchar	20	
Exp	Varchar	10	
Salary	Integer		
Location	Varchar	20	
Mailed	Varchar	30	
Lastdate	Varchar	20	

4. Створіть сторінку з іменем `update_form.aspx`, яка дозволить користувачам модифікувати власні резюме.
пользователям модифицировать их резюме.

Поради:

- Скористайтесь логікою організації сторінки, що і `entry.aspx`.
- Прочитайте дані із таблиць `authentication`, `acaback`, `personalinfo`, `academic` та заповніть форму даними.
- Обновіть змінені дані у відповідній таблиці.
- Замініть сторінку `construction.html`, на котру є посилання із сторінки `edit_form.aspx` на сторінку `update_form.aspx`. Після цього користувач, при введенні коректних імені та паролю, отримає доступ із сторінки `edit_form.aspx` до сторінки `update_form.aspx`.

5. Контрольні питання.

1. Розкажіть про використання колекції `Parameters` об'єкту `Command`.
- 2 Розкажіть про використання `EditCommandColumn` елементу керування `DataGrid`.
3. Яка властивість елементу керування `DataGrid` містить номер редагованого рядка?
4. Розкажіть про використання `ButtonColumn` елементу керування `DataGrid`.

6.Завдання

Для категорії тих, хто шукає працівників слід створити відповідну сторінку самостійно за викладеним вище зразком. Вона повинна відповідати двом наступним вимогам:

- надавати детальні дані про компанії, яким потрібні кадри, за вище викладеною схемою та
- містити лічильник відвідувань.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Поняття web-застосунку.
2. Класифікація web-застосунків.
3. Патерни проєтування.
4. Архітектурні шаблони web-застосунку.
5. Організація зв'язку між компонентами застосунку.
6. Слабкий зв'язок між компонентами застосунку.
7. Механізм впровадження залежностей.
8. Контейнер впровадження залежностей.
9. Що таке Ninject?
10. Розпізнавач залежностей.
11. Шаблон MVC.
12. Порівняння MVC та MVVM.
13. Модель предметної області.
14. Слабкий зв'язок між компонентами.
15. Що таке JSON?
16. Де використовується JSON?
17. Що таке Asp.Net?
18. Реалізація MVC в Asp.Net.
19. Інтелектуальний користувацький інтерфейс.
20. Загальна характеристика Asp.Net.
21. Відношення Asp.Net та MVC.
22. Переваги і недоліки Asp.Net.
23. Структура застосунку Asp.Net.
24. Механізм посилань на браузері.
25. Поняття тестування.
26. Що таке TDD?
27. Модульне тестування.

28. Поняття моккінга.
29. Що таке Razor?
30. Синтаксис виразів Razor.
31. Умовні оператори в Razor.
32. Що таке представлення?
33. Що таке компоновка?
34. Що таке контроллер?
35. Що таке модель?
36. Що таке предметна область?
37. Що таке система маршрутизації?
38. Що таке шаблон URL?
39. Що таке регулярний вираз?
40. Як використовуються регулярні вирази в маршрутизації?
41. Охарактеризуйте маршрутизацію на основі угод.
42. Охарактеризуйте маршрутизацію на основі атрибутів.
43. Зв'язок моделі з базою даних..
44. Що таке прив'язка моделі?.
45. Що таке – активатор дій? .
46. Що таке – зв'язувач моделей?
47. Види зв'язувачів.
48. Механізм вибіркового виключення властивостей із моделі.
49. Прив'язка масивів.
50. Прив'язка колекцій.
51. Автоматична і ручна прив'язки.
52. Постачальник даних.
53. Суть і мета перевірки достовірності моделей.
54. Способи перевірки достовірності моделей.
55. Помилки рівня властивостей.
56. Помилки рівня моделі.

- 57.Що таке фільтрація в Asp.Net&
- 58.Способи організації фільтрації.
- 59.Як організована видача повідомлень про порушення достовірності?
- 60.Чи можна виконати перевірку достовірності у зв'язувачі моделей?.
- 61.Що таке метадані?.
- 62.Атрибути перевірки достовірності моделі.
- 63.Що таке самоперевіряюча модель?.
- 64.Як організована перевірка достовірності на боці клієнта?
- 65.Призначення пакета NuGet/
- 66.Призначення WebAPI.
- 67.Односторінкові застосунки.
- 68.Застосунки повного обміну.
- 69.Призначення бібліотеки KnockOut/
- 70.Визначення контроллера API.
- 71.Методи дії контроллера API.
- 72.Характеристика запитів AJAX..
- 73.Події і обробники подій.
- 74.З чого починати проектування web-застосунку?
- 75.Як здійснюється стилізація контенту?
- 76.Охарактеризуйте інструментарій Bootstrap.
- 77.Що таке панель адміністрування?
- 78.Яка різниця між аутентифікацією і авторизацією?
- 79.Як встановлюється зв'язок між моделлю та базою даних?
- 80.Чи використовується в Asp.Net мова запитів SQL?
- 81.Як взаємодіють між собою компоненти застосунку?
- 82.Що таке розгортання web-застосунку?
- 83.Які дії виконуються при розгортанні web-застосунку?
- 84.Як здійснюється експорт даних?

СПИСОК ЛІТЕРАТУРИ

Основна література

1. ASP.NET. <https://uk.wikipedia.org/wiki/ASP.NET>, 2017.
2. Введение в Asp.Net Core (Фирменная документация), 2018. <https://docs.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-2.1>
3. Начало работы с MVC ASP.NET Core и Visual Studio (tutorial), 2017. <https://docs.microsoft.com/ru-ru/aspnet/core/tutorials/>
4. Начало работы с Razor Pages в ASP.NET Core (Фирменная документация). <https://docs.microsoft.com/ru-ru/aspnet/core/tutorials/razor-pages/razor-pages-start?view=aspnetcore-2.1>
5. Эспозито Д. Программирование с использованием Microsoft ASP.Net 4. / Д. Эспозито, Е. Матвеев. – СПб.: Питер, 2013.
6. . Фримен А. ASP.Net MVC 3.0 для профессионалов. / А. Фримен, С. Сандерсон. – М.: Вильямс, 2011.
7. Макки А. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов. / А. Макки. М: Вильямс, 2010

Додаткова література

1. Шилдт Г. С# 4.0. Полное руководство./ Г. Шилдт. – М.: Вильямс, 2011
2. Павловская Т. С#. Программирование на языке высокого уровня. Учебник для вузов. / Т. Павловская. – СПб.: Питер, 2008. – 461 с.
3. С# 4.0 и платформа .Net для профессионалов. / Б. Ивѐн, Дж. Глин, К. Уотсон, К. Нейгел, М. Скиннел. – М.:Диалектика, 2011. – 1440 с.
5. Фримен А. ASP.Net MVC 3.0 для профессионалов. / А. Фримен, С. Сандерсон. – М.: Вильямс, 2011. – 672 с.

6. Мак-Доналд М. WPF 4: Windows Presentation Foundation в .NET 4.0 с примерами на C# 2010 для профессионалов / М. Мак-Доналд. – М.: Вильямс, 2011.
7. Сеппа Д. Программирование на Microsoft ADO.Net. / Д. Сеппа. – М.: Русская редакция, 2007. – 784 с.
8. Чедвик Дж. Разработка реальных веб-приложений с помощью ASP.Net MVC. / Чедвик Дж., Хришикеш П., Снайдер Т. – М.: Вильямс, 2013. – 432 с
9. Культин Н. C# в задачах и примерах. / Н. Культин. – СПб.: BHV, 2007. – 288с.
10. Натан А. WPF 4. Подробное руководство. / А. Натан. – М.: Символ, 2011. – 880 с.
7. Павлова Е. Технологии разработки современных информационных систем на платформе Microsoft .Net. / Е. Павлова. – М.:

ЗМІСТ

ПОЯСНЮВАЛЬНА ЗАПИСКА.....3