**Лабораторная работа 1. Создание основы веб-сайта приглашения на семинар**

**Постановка задачи**

Требуется создать веб-сайт, который дал бы возможность приглашать на семинар, а приглашенным отправлять ответы на приглашение (repondez s'il vous plait - RSVP).

Должны быть следующие основные средства:

− страница, которая отображает информацию о семинаре и форму RSVP;

− проверка достоверности для формы RSVP, которая будет отображать страницу подтверждения;

− страница, на которой выводится список ответов от приглашенных.

***Упражнение 1. Создание основы веб-сайта ASP.NET***

**Создание веб-сайта**

В этом упражнении вы создадите простой веб-сайт ASP.NET.

1. Запустите Visual Studio.

2. Создайте веб-сайт, для этого выберите пункт **File** (Файл) – **New** (Создать) – **Web Site** (Веб-сайт).

3. Откроется диалоговое окно New Web Site (Новый веб-сайт), создайте **Empty ASP.NET Web Site (Пустой веб-сайт)** с именем RSVP.

Шаблон *ASP.NET Empty Web Site* – простейший из всех шаблонов сайтов. Он создает сайт, содержащий лишь файл Web.config, в котором находится конфигурационная информация для приложения ASP.NET.

**Создание модели данных**

В веб-сайте будет использоваться модель данных.

1. Добавьте в проект класс. Для этого в контекстном меню проекта сайта выберите в контекстном меню пункт **Add** **Class** (Добавить Класс).

2. Укажите в качестве имени класса GuestResponse.cs и щелкните на кнопке Add (Добавить). Согласитесь на предупреждение о помещении файла в папку **App\_Code**.

Среда Visual Studio создаст новый файл класса C# и откроет его для редактирования.

3. Объявите в класс **GuestResponse** и в нем автоматически реализуемые свойства:

public class GuestResponse

{

public int GuestResponseIdId { get; set; }

public string Name { get; set; }

public string Email { get; set; }

public string Phone { get; set; }

public bool? WillAttend { get; set; }

public DateTime Rdata {get; set;}

}

Первое свойство определяет уникальное значение, в дальнейшем оно будет использовано в качестве первичного ключа. Суффикс Id в имени требуется для генерации в базе данных первичного ключа. По умолчанию при генерации базы данных в качестве первичных ключей будут рассматриваться свойства с именами Id или [Имя\_класса]Id .

Свойство WillAttend определено с типом bool, допускающим null. Это означает, что свойство может принимать значения true, false или null. Для представления ответов от приглашенных будут применяться экземпляры класса GuestResponse.

4. Добавьте в класс два конструктора, первый по умолчанию, второй с параметрами, принимающий имя, электронный адрес, телефон и информацию о согласии, дата должна вычисляться в теле конструктора как текущее значение:

public GuestResponse() { }

public GuestResponse(string name, string email, string phone, bool? willattend)

{

Name = name;

Email = email;

Phone = phone;

WillAttend = willattend;

Rdata = DateTime.Now;

}

**Создание хранилища данных**

Созданные объекты GuestResponse должны записываться в хранилище.

Позднее в качестве хранилища будет выступать база данных. Пока будет реализовано хранение объектов в памяти. Преимущество такого подхода связано с простотой реализации, однако следует учитывать, что данные будут теряться при каждом останове или перезапуске приложения, поэтому лучше сохранять коллекцию в файл с помощью механизма сериализации.

1. Чтобы определить хранилище, добавьте в папку **App\_Code** новый файл класса по имени ResponseRepository.cs и поместите в него код, показанный ниже:

// Ставим курсор на название проекта о обозревателе решений (там, где Solution Explorer – прямо под “Solution”), нажимаем ПКМ и в выпадающем меню выбираем ADD. Потом выбираем Class и определяем его название. Класс можно перетащить мышкой в указанную папку.

using System.Collections.Generic;

public class ResponseRepository

{

private static ResponseRepository repository = new ResponseRepository();

private List<GuestResponse> responses = new List<GuestResponse>();

public static ResponseRepository GetRepository()

{

return repository;

}

public IEnumerable<GuestResponse> GetAllResponses()

{

return responses;

}

public void AddResponse(GuestResponse response)

{

responses.Add(response);

}

}

Хранилище обычно располагает методами для создания, чтения, обновления и удаления объектов данных (вместе называемых методами CRUD (creating, reading, updating, deleting – создание, чтение, обновление, удаление), но в этом приложении реализуется только возможность чтения всех объектов данных и добавления новых объектов данных.

***Упражнение 2. Создание форм веб-сайта***

**Создание основной формы**

Следующий шаг заключается в создании стартовой страницы, которая содержит информацию о семинаре.

1. Добавьте в проект сайта новую форму с именем Start.aspx.

// Ставим курсор на название проекта о обозревателе решений (там, где Solution Explorer – прямо под “Solution”), нажимаем ПКМ и в выпадающем меню выбираем ADD. Потом выбираем Web Form и определяем название.

2. В раздел <div> внесите необходимые изменения для отображения информации на форме:

<div>

<div class="rek">

<h1>Приглашение на семинар</h1>

<p>Вы приглашены на наш семинар</p>

<p>Подтвердите свое согласие, пройдя регистрацию</p>

</div>

<div class="info">

Семинар состоится 1 января 2020 года в 7.30

</div>

</div>

3. Постройте решение.

4. Задайте форму Start.aspx в качестве начальной страницы, выбрав соответствующую команду контекстного меню файла формы в обозревателе решений.

// Ставим курсор на указанную форму в обозревателе решений, нажимаем ПКМ. Выбираем “Set As Start Page”.

5. Запустите проект, выбрав пункт **Start Debugging** в меню **Debug** и проверьте отображение стартовой страницы.

**Создание формы регистрации**

Следующий шаг заключается в создании страницы регистрации, которая содержит информацию о клиенте, позволяющую приглашенным подготовить ответ.

1. Добавьте в проект сайта новую форму с именем Reg.aspx.

// Ставим курсор на название проекта о обозревателе решений (там, где Solution Explorer – прямо под “Solution”), нажимаем ПКМ и в выпадающем меню выбираем ADD. Потом выбираем Web Form и определяем название

2. Внесите необходимые изменения для отображения информации на форме

<div>

<h1>Приглашаем на семинар</h1>

<p></p>

</div>

3. Добавьте серверные элементы, предназначенные для сбора информации от пользователей и кнопку для ее отправки:

<div>

<label>Ваше имя:</label><asp:TextBox ID="name" runat="server"></asp:TextBox>

</div>

<div>

<label>Ваш email:</label><asp:TextBox ID="email" runat="server"></asp:TextBox>

</div>

<div>

<label>Ваш телефон:</label><asp:TextBox ID="phone" runat="server"></asp:TextBox>

</div>

<div>

<label>Вы будете делать доклад?</label>

<asp:CheckBox ID="CheckBoxYN" runat="server" />

</div>

<div>

<button type="submit">Отправить ответ на приглашение RSVP</button>

</div>

4. Откройте страницу в браузере и посмотрите, как выглядят указанные элементы.

**Стилизация созданных форм**

Элементы веб-формы стилизуются с применением каскадных таблиц стилей (Cascading Style Sheets - CSS).

1. Добавьте к приложению таблицу стилей с помощью команды контекстного меню **Add** **StyleSheet** (Добавить Таблица стилей). В появившемся диалоговом окне в качестве имени укажите Styles и щелкните на кнопке OK.

// Ставим курсор на название проекта о обозревателе решений (там, где Solution Explorer – прямо под “Solution”), нажимаем ПКМ и в выпадающем меню выбираем ADD. Потом выбираем StyleSheet и указываем имя.

Среда Visual Studio добавит к проекту новый файл Styles.css.

2. Приведите содержимое этого файла в соответствие с указанным ниже:

h1 {

margin: 0px;

padding: 2px 0;

font-size: 30px;

font-weight: bold;

}

#form1 label {

width: 120px;

display: inline-block;

}

#form1 input {

margin: 2px;

margin-left: 4px;

width: 150px;

}

#form1 select {

margin: 2px 0;

width: 154px;

}

button[type=submit] {

margin-top: 15px;

padding: 5px;

}

3. Добавьте элемент link (таблица стилей CSS ассоциируется с веб-формой с помощью этого элемента) в раздел head файлов Reg.aspx и Start.aspx:

...

<head runat="server">

<title></title>

<link rel="stylesheet" href="Styles.css" />

</head>

...

Обратите внимание, что для ссылки на файл, содержащий базовые стили CSS, используется стандартный HTML-элемент.

4. Постройте приложение. Просмотрите влияние стилей на отображение формы в браузере.