

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

ПО ДИСЦИПЛИНЕ

«ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ» НА ТЕМУ «РАЗРАБОТКА ФОРМ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ASP.NET»

Вариант – 2

Факультет: АВТ Преподаватель: Дубков И.С.

Группа: АВТ-918

Студент: Ванин К.Е.

Цель работы

Получить практические навыки по созданию форм в приложениях ASP.NET для реализации обработки информации на стороне сервера.

Задание

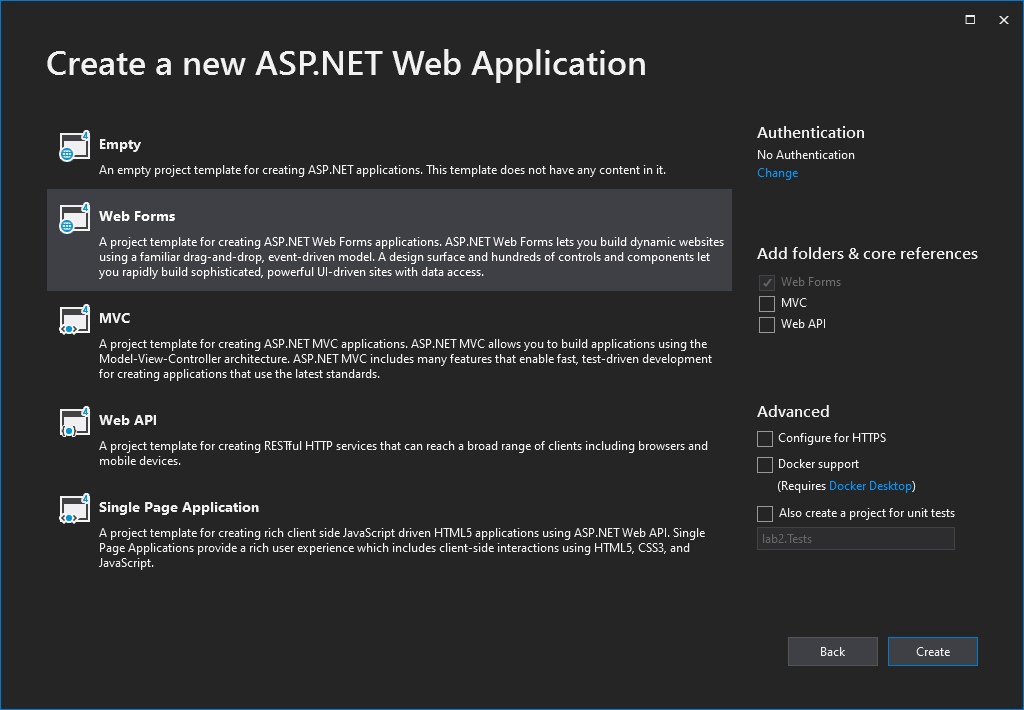
1. Создать пустой проект.
2. Добавить в проект веб-страницу.
3. Создать форму, содержащую серверные элементы управления и элементы управления HTML.
4. Создать код для обработки информации на стороне сервера.
5. Запустить приложение, проверить его работоспособность.
6. В отчете привести код страницы и скриншот формы.
7. Пояснить назначение отдельных фрагментов кода.
8. Пояснить отличие серверных элементов и элементов HTML.
9. Провести анализ выполненной работы и сформулировать выводы.

**Вариант** 2:

Обмен текстовыми сообщениями между сервером и клиентом

Ход работы

Для выполнения лабораторной работы был выбран ASP.NET Web Forms для системы Windows.



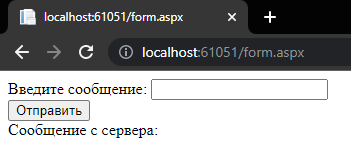
*Рис. 1 – Создание проекта*

Согласно заданию, был создан пустой проект в Visual Studio 2019, в который была добавлена веб-форма (form.aspx). В форму был помещен элемент asp:TextBox, в который пользователю предлагается ввести сообщение, и кнопка asp:Button, а также элемент label. На этой же странице будет выводиться сообщение от сервера.

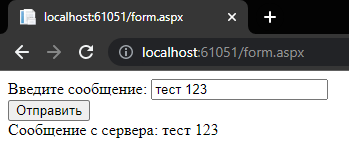
Атрибут runat="server" говорит о том, что события для элементов управления данной формы должны обрабатываться сервером. Если этот элемент управления указан не будет, то в этом случае форма будет вести себя как обычная форма HTML.

Для страницы form.aspx был написан обработчик. При нажатии на кнопку Button1 происходит обработка переданного из формы значения. Значение из TextBox1 выводится в Label1.

Скриншоты



*Рис. 2 – Первоначальный вид страницы index.aspx.*



*Рис. 3 – Вывод содержимого сообщения от сервера.*

Отличие серверных элементов от элементов HTML

Отличие встроенных элементов ASP.NET от стандартных элементов HTML5 заключается в следующем:

* Элементы управления ASP.NET – это специализированные объекты .NET, которые имеют встроенные механизмы создания и обработки событий. Эти механизмы работают схожим образом с теми, что представлены в Windows-приложениях.
* Для серверных элементов управления ASP.NET сохраняется информация о состояниях.
* ASP.NET генерирует HTML-разметку для соответствующих элементов управления в зависимости от типа браузера и устройства клиента путем сопоставления user-agent со своим каталогом. Хоть и не исключена вероятность, что user-agent в каталоге будет не найден, но наличие этой возможности позволяет обеспечить дополнительную совместимость.
* Серверные элементы управления являются наследниками общих предков, благодаря чему обладают общим набором свойств, методов и событий. При этом у каждого элемента могут быть определены свои свойства, методы или события.
* В коде HTML для серверных элементов управления используется префикс пространства имен XML – «asp».

Выводы

В ходе выполнения лабораторной работы были получены практические навыки по созданию форм в приложениях ASP.NET для реализации обработки информации на стороне сервера.

Приложение А

Листинг программы

Файл form.aspx

1. <%@ Page Language="C#" AutoEventWireup="true" CodeBehind="form.aspx.cs" Inherits="lr1.form" %>
3. <!DOCTYPE html>
5. <html xmlns="http://www.w3.org/1999/xhtml">
6. <head runat="server">
7. <title></title>
8. </head>
9. <body>
10. <form id="form1" runat="server">
11. <div>
12. Введите сообщение: <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
13. </div>
15. <asp:Button ID="Button1" runat="server" OnClick="Button1\_Click" Text="Отправить" />
17. </form>
18. <div>
19. Сообщение с сервера:
20. <asp:Label ID="Label1" runat="server"></asp:Label>
21. </div>
22. </body>
23. </html>

Файл form.aspx.cs

1. **using** System;
3. **namespace** lr1
4. {
5. **public** **partial** **class** form : System.Web.UI.Page
6. {
7. **protected** **void** Page\_Load(**object** sender, EventArgs e)
8. {
10. }
12. **protected** **void** Button1\_Click(**object** sender, EventArgs e)
13. {
14. Label1.Text = TextBox1.Text;
15. }
16. }
17. }

Приложение Б

Ответы на вопросы

1. **В чем различия серверного элемента управления от элемента HTML**

Отличие встроенных элементов ASP.NET от стандартных элементов HTML5 заключается в следующем:

* Элементы управления ASP.NET – это специализированные объекты .NET, которые имеют встроенные механизмы создания и обработки событий. Эти механизмы работают схожим образом с теми, что представлены в Windows-приложениях.
* Для серверных элементов управления ASP.NET сохраняется информация о состояниях.
* ASP.NET генерирует HTML-разметку для соответствующих элементов управления в зависимости от типа браузера и устройства клиента путем сопоставления user-agent со своим каталогом. Хоть и не исключена вероятность, что user-agent в каталоге будет не найден, но наличие этой возможности позволяет обеспечить дополнительную совместимость.
* Серверные элементы управления являются наследниками общих предков, благодаря чему обладают общим набором свойств, методов и событий. При этом у каждого элемента могут быть определены свои свойства, методы или события.
* В коде HTML для серверных элементов управления используется префикс пространства имен XML – «asp».

1. **Особенности построения веб-форм по технологии ASP.NET**

Web-форма состоит из двух частей:

* отображаемого содержимого, или презентации формы;
* программного кода, который описывает логику взаимодействия с визуальными элементами.

Физически Web-форма представлена файлом с расширением .aspx. Особенность Web-форм состоит в способе, при помощи которого программный код отделяется от самой формы. Исходный код может храниться в отдельном не скомпилированном файле. Или же программный код может быть вложен в .aspx-файл. И, наконец, код может храниться в предварительно скомпилированной сборке, содержащей динамически подключаемую библиотеку (DLL). Когда страницы загружаются Web-сервером, выполняется код, описывающий пользовательский интерфейс. Этот код динамически формирует отображаемую клиентом страницу.

1. **Синтаксис и назначение директивы Page.**

Директива Page позволяет установить свойства страницы, которые будут использованы во время компиляции. Эта директива используется чаще остальных.

Основной синтаксис директивы:

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" Trace="true" %>

Наиболее важные атрибуты директивы перечислены в табл. 1:

Таблица 1. Атрибуты директивы Page.

|  |  |
| --- | --- |
| **Атрибуты** | **Описание** |
| AutoEventWireup | Автоматическая обработка событий страницы |
| Buffer | Управляет буферизацией страницы. По умолчанию буферизуется |
| ClassName | Позволяет назначать имя класса, сгенерированного данной страницей |
| CodeFile | Название файла с отделенным кодом для данной страницы |
| Culture | Устанавливает набор региональных параметров, т.е. язык, формат валюты, даты, чисел |
| Debug | Если true, на страницу выводится отладочная информация |
| Trace | Вывод трассировочной информации |
| EnableViewState | Сохранение состояния страницы. По умолчанию она сохраняется |
| EnableTheming | Позволяет включить или выключить поддержку тем оформления. По умолчанию включено |
| Inherits | Класс, от которого наследуется класс данной страницы в технологии отделенного кода |
| IsAsync | Показывает, обрабатывается ли страница асинхронно. |
| Language | Язык, используемый во внедренном коде |
| WarningLevel | Наибольший допустимый уровень предупреждений компилятора |
| CompilerOptions | Опции компилятора |

1. **Синтаксис и назначение директивы DOCTYPE.**

DOCTYPE указывает тип документа. Директива DOCTYPE необходима, чтобы браузер понимал, как обрабатывать текущую веб-страницу, так как существует несколько версий языка HTML, а еще имеется XHTML (EXtensible HyperText Markup Language, расширенный язык разметки гипертекста), который похож на HTML, но различается с ним по синтаксису.

В табл. 2 приведены основные типы документов.

Таблица 2. Основные типы документов.

|  |  |
| --- | --- |
| **DOCTYPE** | **Описание** |
| **HTML 4.01** | |
| <!DOCTYPE HTML PUBLIC «-//W3C//DTD HTML 4.01//EN» «http://www.w3.org/TR/html4/strict.dtd»> | Строгий синтаксис HTML |
| <!DOCTYPE HTML PUBLIC «-//W3C//DTD HTML 4.01 Transitional//EN» «http://www.w3.org/TR/html4/loose.dtd»> | Переходный синтаксис HTML |
| <!DOCTYPE HTML PUBLIC «-//W3C//DTD HTML 4.01 Frameset//EN» «http://www.w3.org/TR/html4/frameset.dtd»> | В HTML-документе применяются фреймы |
| **HTML 5** | |
| <!DOCTYPE html> |  |
| **XHTML 1.0** | |
| <!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Strict//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd»> | Строгий синтаксис XHTML |
| <!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»> | Переходный синтаксис XHTML |
| <!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Frameset//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd»> | XHTML-документе применяются фреймы |
| **XHTML 1.1** | |
| <!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.1//EN» «http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd»> |  |

Пример использования DOCTYPE в HTML5:

<!DOCTYPE html>  
<html>  
  <head>  
    <title>HTML5 Page</title>  
  </head>  
  <body>  
    <p>Content text</p>  
  </body>  
</html>

1. **Способы позиционирования элементов на форме**

Для изменения дизайна веб-страницы можно использовать CSS (каскадные таблицы стилей). Для использования CSS на необходимых веб-страницах необходимо подключить соответствующий файл с описанием стилей с помощью тега <link> внутри <head>.

Пример:

<link rel="stylesheet" href="./assets/css/styles.css">

Для того, чтобы позиционировать элемент, необходимо задать элементу класс, в котором описать его позиционирование свойством position – оно может быть относительное (relative), абсолютное (absolute), фиксированное (fixed), или статическое (static).

Пример:

.custom-label

{

position: absolute;

left: 15px;

top: 30px;

}

Классы для ASP.NET-элементов задаются через атрибут CssClass.

Пример:

<asp:Label ID="Label1" runat="server" Text="Label" CssClass="custom-label" />