

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
АНО ПОО «КОЛЛЕДЖ ЦИФРОВОЙ ЭКОНОМИКИ И ТЕХНОЛОГИЙ»

Образовательная программа среднего профессионального образования

09.02.07 Информационные системы и программирование

ОТЧЕТ ПО ПРЕДДИПЛОМНОЙ ПРАКТИКЕ

Выполнил студент
Никитина Т.О.

(подпись)

Руководитель практики от
образовательного учреждения
Андреев В.В.

(подпись)

Саратов 2021 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
АНО ПОО «КОЛЛЕДЖ ЦИФРОВОЙ ЭКОНОМИКИ И ТЕХНОЛОГИЙ»

Образовательная программа среднего профессионального образования

09.02.07 Информационные системы и программирование

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПЕРИОД ПРАКТИКИ

1. Выявление целей и задач преддипломной практики
2. Описание выполненных заданий
3. Создание примера ASP.Net web-формы

Руководитель практики:

Андреев В.В. _____

СОДЕРЖАНИЕ

Введение.....	5
Теория и методы разработки веб-приложений.....	7
Описание решаемых задач за время практики.....	10
Заключение.....	13
Список используемой литературы.....	14
Отзыв руководителя практики.....	15

ВВЕДЕНИЕ

В последние годы интернет-технологии стремительно развиваются, неуклонно растет число различных web-приложений: интернет-магазины, сайты компаний, приложения для мобильных устройств. В результате растет и число различных способов, методов и технологий для создания web-приложений. Все чаще компании прибегают к услугам разработки web-приложений, чтобы эффективно решать широкий спектр бизнес-задач. Каждый день в интернете появляются огромное количество разных сайтов. С помощью таких сайтов можно узнать множество интересной и полезной информации, не выходя из дома. На сегодняшний день мало организаций, которые не имели бы свой собственный сайт. Целью создания сайта для организаций является в первую очередь, реклама, а также информация о нем и о его услугах, которые она предоставляет людям или описание продукции, которую изготавливает. На таких сайтах пользователи могут задавать вопросы, которые их интересуют о продукции или оставлять свои отзывы о работе данной организации, комментарии. Благодаря этому, учитывая все пожелания, фирма может улучшить свою продукцию и дать людям нужный ассортимент услуг на рынке. Web-сайты – это как электронные визитки организаций, это залог того, что о нем узнает большое количество потенциальных клиентов не выходя из дома.

Начинающим разработчикам web-приложений бывает сложно сориентироваться в многообразии методов и технологий, которые имеются на просторах сети. Поэтому необходимо структурирование и упорядочивание имеющихся теоретических и методологических подходов к разработке web-приложений. стремительно развиваются, постепенно вытесняя настольные решения и становясь важнейшим компонентом бизнеса в современном мире. Раскрыв и обосновав основные из имеющихся современных теорий, методов, подходов и технологий разработки веб-приложений, возможно проще будет подобрать подходящий для конкретного проекта вариант.

В ходе прохождения преддипломной практики, на основании индивидуального задания «Теория и методы разработки веб-приложений», был создан пример веб-формы обратной связи с использованием технологии ASP.Net Web Forms.

Теория и методы разработки веб-приложений

Web-приложение - это клиент-серверное приложение, в котором клиентом выступает браузер, а сервером - web-сервер. Логика web-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому web-приложения являются межплатформенными сервисами.

Под клиентом в среде клиент-сервер принято понимать программу, которую человек использует для запуска приложения. Клиент-серверная среда – это среда, в которой несколько компьютеров обмениваются информацией, например, вводят информацию в базу данных. «Клиент» – это приложение, используемое для ввода информации, а «сервер» – это приложение, используемое для хранения информации.

Само web-приложение может выступать в качестве клиента других служб, например, базы данных или другого web-приложения, расположенного на другом сервере.

Под современными технологиями web-программирования, сегодня понимают в основном, Интернет-технологии, включающие в себя концептуальные знания www, HTML, Java, клиентских и серверных скриптов и языков запросов к базам данных, основы web-дизайна. Развитие не останавливается, с каждым годом появляются все новые, более совершенные технологии, дорабатываются ранее созданные.

При разработке сайтов, web-приложений и прочих программных продуктов используется достаточно много отличных методологий, выбор зависит непосредственно от задачи, которая стоит перед разработчиком, особенностей проекта, бюджета и других нюансов. Современные методы

проектирования и разработки web-приложений базируются на предыдущем опыте, анализе аудитории пользователей и характеризуются четкой структурированностью этапов, на каждом из которых текущая версия web-приложения проверяется и оценивается, а затем расширяется или модифицируется для того, чтобы удовлетворять как уже собранным требованиям, так и вновь появившимся. В процессе разработки любого web-приложения существует определенная последовательность действий, она включает в себя этапы. На каждом этапе осуществляется определенная деятельность, закладываются существенные компоненты, каждый из этих этапов важен и необходим для создания web-приложения.

Все подходы к разработке web-приложений могут быть разделены на 3 большие категории:

- Подходы, основанные на программировании или скриптах: внешние программы или скрипты; расширения web-сервера.

- Подходы, основанные на использовании шаблонов web-страниц, включающих вставки кода скриптов и специальных серверных тэгов.

- Объектные среды (каркасы, фреймворки, frameworks).

Хотя между этими категориями и имеются пересечения (а также различные мнения о том, к какой категории относится конкретная технология разработки), большинство широко известных подходов связаны с одной конкретной категорией.

Объектные среды (фреймворки, frameworks) представляют собой следующий уровень совершенствования разработки web-приложений. Вместо объединения разметки и логики в единый модуль, объектные среды (frameworks) поддерживают принцип отделения содержания от представления. Модули, ответственные за создание контента, отделяются от модулей, которые показывают это содержание в конкретном формате.

Подход на основе web-форм является дальнейшим развитием скриптовых серверных технологий. В данном подходе в HTML-документы добавляются специальные тэги, обрабатываемые на стороне сервера. Кроме

этого, можно описывать и использовать собственные тэги в виде классов на универсальных языках программирования (Java, C#, Visual- Basic и подобные), создавать на стороне сервера объектной модели web-приложения, аналогичные объектной модели локального приложения. Одним из основных объектных подходов на основе форм является технология Microsoft ASP.Net Web Forms.

Описание решаемых задач за время практики

В результате прохождения практики были изучены теоретические основы и возможности web-приложений, наиболее популярные языки, методы, технологии, средства и инструменты разработки web-приложений, проанализированы существующие подходы к разработке web-приложений.

Технология ASP.Net Web Forms является объектным подходом на основе форм. Она представляет собой развитие шаблонного подхода с использованием средств полностью объектно-ориентированного программирования. Web-приложение является набором web-форм, которые состоят из двух частей: шаблона, включающего HTML-код и специальные теги, и программного кода с обработчиками событий для объектов, используемых при обработке запроса к шаблону.

Для создания нового пустого web-приложения в форме сайта для начала выполняется команда "File=>New Web Site=>ASP.Net Empty Web Site". Далее, после указания места расположения и имени папки Visual Studio создаст новое web-приложение. Для добавления web-формы необходимо выполнить команду "Web Site=>Add New Item", выбрать шаблон нового файла "Web Form" и назначить ему имя. После создания файла с шаблоном web-формы его можно просматривать и редактировать в двух основных режимах – design (графический режим) и source (режим исходного кода). Каждая web-форма является страницей с HTML-формой.

Для создания форм в html используются теги группы form. К ним относятся:

- <form> - контейнер формы;
- <input> - элементы формы (поля) различных типов;
- <button> - кликабельная кнопка;
- <select> - выпадающий список с множественным выбором;
- <option> - пункт списка;

- <optgroup> - группа пунктов;
- <datalist> - список predefined вариантов;
- <textarea> - текстовая область;
- <label> - подпись, метка поля;
- <fieldset> - группа полей;
- <legend> - название группы полей;
- <output> - результат вычислений.

Для создания простой web-формы, необходимо поместить в код страницы тег <form> и разместить внутри него элементы формы.

Тег <form> является корневым элементом web-формы. Все элементы формы должны находиться внутри тега <form>, либо быть связанными с формой при помощи атрибута form.

Для примера рассмотрим создание формы обратной связи. Стандартная форма обратной связи содержит следующие элементы: список выбора тематики, текстовые поля для ввода имени, контакта (например, email) и текстовую область для ввода сообщения. Визуальный интерфейс и результаты работы данной формы показаны на рисунке 1.

Рис. 1. Пример web-формы

Описание шаблона web-формы, показанной на рис. 1, приведено на рисунке 2.

Основным элементом всех шаблонов форм является элемент Form. Код данной формы включает в себя атрибут class который определяет название CSS класса для стилизации формы, name - имя формы, method="POST"

используется для того, что бы данные введенные пользователем были переданы в теле HTTP запроса. Атрибут action содержит URL обработчик формы, он определяет куда будет отправлена форма после наступления события submit (submit - событие отправки формы). Внутри тега <form>, помимо текстовых элементов, находятся элементы формы: <select> - список, <input> text и email - поля ввода, <textarea> - текстовая область и <input type="submit"> - кнопка отправки формы.

Каждый элемент содержит атрибут name. При отправке формы, значения введенные пользователем будут переданы в переменных с именами указанными в этом атрибуте. Атрибут type определяет тип элемента <input>. Атрибут required делает элемент обязательными для заполнения. Форма не будет отправлена, пока пользователь не заполнит все обязательные элементы.

```
<form class="guruweba_example_form" name="feedback" method="POST" action="/basa.php">
  <div class="guruweba_example_caption">Введите данные</div>
  <div class="guruweba_example_infofield">Тема обращения</div>
  <select name="theme" required="required">
    <option value="">Выберите вариант</option>
    <option>Группа 1</option>
    <option>Группа 2</option>
    <option>Группа 3</option>
  </select>
  <div>Ваше имя</div>
  <input type="text" name="name" required="required">
  <div>Ваш email</div>
  <input type="email" name="email" required="required">
  <div>Сообщение</div>
  <textarea name="message"></textarea>
  <input type="submit" name="submit_btn" value="Отправить">
</form>
```

Рис. 2. Описание примера шаблона web-формы

Такой программный код web-формы исполняется средой выполнения ASP.Net.

ЗАКЛЮЧЕНИЕ

Для достижения успехов в освоении выбранной профессии необходимо было отмечено получение теоретических и практических знаний в написании, обновлении, исправлении и совершенствовании алгоритмов для приложений, сайтов и отдельных элементов, с использованием разных языков программирования. За период прохождения практики мне удалось освоить необходимые для создания ASP.NET веб-формы навыки. В результате выполнения задачи преддипломной практики, удалось создать пустое приложение в Visual Studio, сформировать страницу с HTML-формой, освоить процесс создания веб-формы обратной связи с использованием технологии ASP.NET, а так же были получены навыки использования тегов группы form, для написания HTML-формы.

При разработке web-приложений с использованием ASP.NET можно отметить определенные достоинства. Данная технология предоставляет отличные возможности для быстрой разработки web-приложений, относительно проста и понятна для создания web-приложений, работающих с большими объемами данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Джон Дакетт, HTML и CSS. Разработка и создание веб-сайтов. Переводчик: Михаил Райтман, Эксмо 2013 г.
2. Эрик Фримен, Элизабет Фримен. Изучаем HTML, XHTML и CSS. Переводчик: В. Черник. Питер, 2019 г.
3. Венц К. Программирование в ASP.NET Ajax. Перевод А.Киселева. СПб: Символ Плюс 2008 г.
4. Чамберс Д., Пэккетт Д., Тиммс С. ASP.NET Core. Разработка приложений. Издательство: Питер. 2018 г.
5. Робин Никсон. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 (pdf+epub). 2018 г.
6. Эд Титтел, Джефф Ноубл. HTML, XHTML и CSS для чайников, 7-е издание. «Диалектика». 2010 г.
7. Стивен Шафер. HTML, XHTML и CSS. Библия пользователя, 5-е издание. «Диалектика». 2011 г.
8. Гончаров, А.Н. Самоучитель HTML [Текст] / А.Н. Гончаров.- СПб.: Питер, 2002 г.

