

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Компьютерные системы и сети (КСиС)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему

Веб-сайт доска объявлений

БГУИР КП 1-40 01 0 26 ПЗ

Студент: гр. 351002 Гучок О.А.

Руководитель: асс. Третьяков Ф.И.

Минск 2015

Учреждение образования
«Белорусский государственный университет информатики и
радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ
Заведующий кафедрой ПОИТ

(подпись)

2015 г.

ЗАДАНИЕ
по курсовому проектированию

Студенту Гучку Олегу Анатольевичу

1. Тема работы Веб-сайт доска объявлений
2. Срок сдачи студентом законченной работы DD.MM.YYYY
3. Исходные данные к работе Среда разработки Visual Studio 2013.
4. Содержание расчётно-пояснительной записки (перечень вопросов, которые подлежат разработке)
Введение. 1. Анализ литературных источников. 2. Постановка задачи
3.Разработка программного средства. 4. Руководство по
использованию веб-сайта. Заключение. Приложения.
5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)
1. Схема алгоритма
6. Консультант по курсовой работе
Третьяков Ф.И.
7. Дата выдачи задания DD.MM.YYYY
8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и процентом от общего объёма работы):
раздел 1 к DD.MM.YYYY – 15 % готовности работы;
разделы 2, 3 к DD.MM.YYYY – 30 % готовности работы;
раздел 4 к DD.MM.YYYY – 60 % готовности работы;
раздел 5, 6 к DD.MM.YYYY – 90 % готовности работы;

оформление пояснительной записки и графического материала к
DD.ММ.YYYY – 100 % готовности работы.
Защита курсового проекта с DD по DD декабря YYYY г.

РУКОВОДИТЕЛЬ _____ Третьяков Ф.И.
(подпись)

Задание принял к исполнению _____ Гучок О.А. DD.ММ.YYYYг.
(дата и подпись студента)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 АНАЛИЗ ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ	7
1.1 Технологии создания web-страниц	7
1.2 Появление ASP.NET MVC	8
1.3 Паттерн MVC	9
1.4 Преимущества ASP.NET MVC	10
2 ПОСТАНОВКА ЗАДАЧИ	11
2.1 Сравнение приложения с существующим аналогом	11
2.2 Постановка задачи	11
3 РАЗРАБОТКА ПРИЛОЖЕНИЯ	13
4 РУКОВОДСТВО ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ ВЕБ-САЙТА	14
ЗАКЛЮЧЕНИЕ	15
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	16
ПРИЛОЖЕНИЕ А	17

ВВЕДЕНИЕ

Интернет впервые был создан в 60-х гг. прошлого века как проект Министерства обороны США. Он получил название ARPANET (Advanced Research Projects Agency Network — сеть Агентства по перспективным исследовательским проектам). Его основной целью было объединение различных компьютеров, разбросанных по Западному побережью США с тем, чтобы они могли взаимодействовать друг с другом даже в случае начала атомной войны. В то время компьютеры были очень редкими и дорогостоящими устройствами, которые могли позволить себе только университеты и правительственные организации. Связав воедино несколько компьютеров, можно было организовать общий доступ к различной информации и данным, что еще больше увеличивало ценность каждого отдельного компьютера. Протокол, предназначенный для работы в данной сети, был разработан таким образом, чтобы быть устойчивым к нарушениям целостности сети. Это было сделано для того, чтобы даже при выходе из строя в результате ядерной атаки одного или нескольких компьютеров, сеть сохраняла бы свою работоспособность.

Результатом разработки ARPANET явилось создание TCP/IP (Transmission Control Protocol/Internet Protocol — протокол управления передачей/межсетевой протокол). Это протокол, позволяющий вести обмен сигналами, в которых содержатся компьютерные данные и информация, по телефонным линиям, по волоконно-оптическим кабелям и через спутники. Протокол TCP/IP позволяет различным компьютерам обмениваться между собой информацией по сети, и его принятие расчистило дорогу для появления Интернета в том виде, в котором он знаком нам сегодня.

Бесчисленное множество новых технологий, вызванных бурным ростом информатизации общества, делает нашу жизнь невозможной без быстрого доступа к информации. В наше время очень легко получить информацию, одним из способов быстрого доступа к ней является сайт.

Создание сайтов на сегодняшний день, становится одной из наиболее актуальных и востребованных услуг. Именно поэтому, большинство компаний уже оценили все преимущества такого предложения как создание сайтов и позаботились о разработке подходящего ресурса.

Пользователю приятно посещать те Web-страницы, которые имеют стильное оформление, не отягощены чрезмерно графикой и анимацией, быстро загружаются и правильно отображаются в окне Web-браузера. Но может возникнуть и другая проблема - сайт может оказаться не интересным пользователю и та информация, которую он несет, окажется не

востребованной. Именно поэтому важно, чтобы сайт отвечал всем требованиям пользователя.

Актуальность данной работы заключается в том, что с учетом скорости развития сети Интернет и направления e-commerce, рынок WEB-разработки огромен и очень перспективен.

Целью работы является формирование теоретических знаний по проектированию web-сайта и практических навыков по его разработке. Выбранная технология - ASP.NET MVC.

Инфраструктура ASP.NET MVC Framework реализует шаблон MVC и при этом обеспечивает существенно улучшенное разделение ответственности. На самом деле в ASP.NET MVC внедрен современный вариант MVC, который особенно хорошо подходит для веб-приложений.

За счет принятия и адаптации шаблона MVC инфраструктура ASP.NET MVC Framework составляет сильную конкуренцию Ruby on Rails и аналогичным платформам, выводя модель MVC в авангард развития мира .NET. Обобщая опыт и наиболее рекомендуемые приемы, обнаруженные разработчиками, которые используют другие платформы, ASP.NET MVC во многих отношениях превзошла даже то, что может предложить Rails.

1 АНАЛИЗ ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ

1.1 Технологии создания web-страниц

Итак, Интернет — это цельная связанная сеть компьютеров, которая охватывает весь земной шар. На миллионах серверов располагаются миллиарды web-страниц, групп электронной почты, дискуссионных страниц и FTP-сайтов. Интернет целиком работает в режиме онлайн, включая электронную почту, FTP, новости, Gopher-сайты, чат-программы и т. п. World Wide Web появилась позже Интернета. Первый web-сервер заработал в 1991 г., но на сегодняшний день Всемирная паутина — это огромная часть Интернета, состоящая из взаимосвязанных web-страниц, в которых содержится текст, графика и мультимедиа. Ключевым компонентом World Wide Web являются гиперссылки (текстовые или графические), которые позволяют пользователю переходить на другие web-страницы. Это оказывается возможным благодаря использованию HyperText Markup Language (HTML, язык гипертекстовой разметки).

Интернет представляет собой клиент-серверную систему. Т.е. вся информация хранится на серверах, которые по мере надобности посылают информацию клиентам. Сервер — это приложение, выполняющееся на некотором компьютере и обеспечивающее доступ к информации, файлам или данным, которые запрашиваются каким-либо другим лицом, приложением или компьютером. Местоположение серверов определяется с помощью Uniform Resource Locator (URL — универсальный указатель ресурса). Это особый адрес web-страницы, по которому она располагается в World Wide Web.

Клиентом называется особая часть программного обеспечения, выполняющаяся на компьютере. Наиболее распространенным клиентом в Интернете является браузер — специальная программа, позволяющая пользователю вводить адреса или использовать гиперссылки на web-страницах для поиска новых web-страниц. Она управляет отправкой запросов web-серверу и выводом на экран полученной от web-сервера web-страницы.

Вся работа по отправке и получению web-страниц ведется через протокол HTTP (Hypertext Transfer Protocol). Это прикладной протокол, позволяющий браузерам и серверам взаимодействовать друг с другом и обмениваться между собой данными.

Использование одного лишь HTML невозможно в наше время. Во-первых это связано с созданием web-страниц: очень сложно обновлять страницы вручную. Во-вторых, в чистом HTML отсутствует какая-либо интерактивность и возможность взаимодействовать на серверах на пользовательском уровне. Например, не было бы почтовых служб, чатов и форумов. Проще говоря, это означало бы полную статичность.

Чтобы «оживить» Web, т.е. перейти от структурного предъявления

гипертекстовой информации к событийному был разработан DHTML (Dynamic HyperText Markup Language). Основной отличительной особенностью DHTML от HTML является возможность взаимодействия DHTML-документов с пользователем на клиентском компьютере, что в значительной степени обогащает возможности создаваемых с их помощью Web-страниц и Web-приложений и в то же время сводит часть взаимодействия пользователя с сервером к взаимодействию пользователя с DHTML-документом. Динамичность осуществляется при помощи языков сценариев, таких как JavaScript и VBScript. Но эта динамичность присуща только для клиентской части, взаимодействие с сервером все так же остается на примитивном уровне и позволяет отправлять всем пользователям только одну и ту же статическую информацию.

Поэтому был разработан Common Gateway Interface (CGI — интерфейс общего шлюза), который позволял web-страницам вызывать приложения, расположенные на web-серверах. Такие приложения могли создаваться на произвольном языке программирования, но чаще всего они писались на C/C++ или на Perl. Но CGI требовало огромных ресурсов, — для каждого подключившегося к серверу пользователя запускался свой экземпляр CGI, ни один популярный сервер не смог бы работать в нормальном режиме.

В итоге было создано большое количество различных инструментов и языков программирования, которые позволяют создавать активные web-страницы. Вот некоторые из этих технологий:

- Active Server Pages
- PHP
- ColdFusion
- WebSphere
- Java Server Pages

Несмотря на то, что PHP является лидером этой сферы деятельности, ASP даёт программистам большую свободу и простоту. Главным недостатком является цена и производительность Windows серверов.

1.2 Появление ASP.NET MVC

ASP.NET MVC - это инфраструктура для разработки веб-приложений от Microsoft, которая сочетает в себе эффективность и аккуратность архитектуры "модель-представление-контроллер" (model-view-controller - MVC), новейшие идеи и приемы гибкой разработки, а также все лучшее из существующей платформы ASP.NET. Она представляет собой полномасштабную альтернативу традиционной технологии ASP.NET Web Forms, предоставляя преимущества для всех проектов веб-разработки, кроме самых тривиальных.

Важно различать архитектурный шаблон MVC и инфраструктуру ASP.NET MVC Framework. Шаблон MVC далеко не нов (его появление датируется 1978 г. и связано с проектом Smalltalk в Xerox PARC), но в наши

дни он завоевал огромную популярность в качестве шаблона для веб-приложений по перечисленным ниже причинам:

- Взаимодействие пользователя с приложением MVC осуществляется в соответствии с естественным циклом: пользователь предпринимает действие, в ответ на которое приложение изменяет свою модель данных и доставляет обновленное представление пользователю. Затем цикл повторяется. Это хорошо укладывается в схему веб-приложений, предоставляемых в виде последовательностей запросов и ответов HTTP.

- Веб-приложения, нуждающиеся в комбинировании нескольких технологий (например, баз данных, HTML-разметки и исполняемого кода), обычно разделяются на ряд слоев или уровней. Полученные в результате шаблоны естественным образом вписываются в концепции MVC.

1.3 Паттерн MVC

Шаблон MVC, лежащий в основе новой платформы, подразумевает взаимодействие трех компонентов: контроллера (controller), модели (model) и представления (view)(Рис. 1).

Контроллер (controller) представляет класс, с которого собственно и начинается работа приложения. Этот класс обеспечивает связь между моделью и представлением. Получая вводимые пользователем данные, контроллер исходя из внутренней логики при необходимости обращается к модели и генерирует соответствующее представление.

Представление (view) - это собственно визуальная часть или пользовательский интерфейс приложения - например, html-страница, через которую пользователь, зашедший на сайт, взаимодействует с веб-приложением.

Модель (model) представляет набор классов, описывающих логику используемых данных.

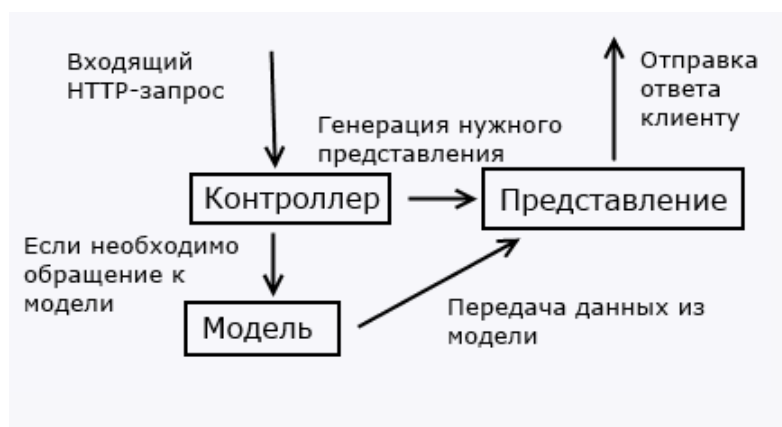


Рис. 1: Схема паттерна MVC

1.4 Преимущества ASP.NET MVC

ASP.NET MVC имеет следующие преимущества:

- **Разделение ответственности.** В MVC приложение состоит из трех частей: контроллера, представления и модели, каждая из которых выполняет свои специфичные функции. В итоге приложение будет легче поддерживать и модифицировать в будущем.
- **В силу разделения ответственности** приложения MVC обладают лучшей тестируемостью. И мы можем тестировать отдельные компоненты независимо друг от друга.
- **Соответствие протоколу HTTP.** Приложения MVC в отличие от веб-форм не поддерживают объекты состояния (ViewState). Ясность и простота платформы позволяют добиться большего контроля над работой приложения.
- **Гибкость.** Вы можете настраивать различные компоненты платформы по своему усмотрению. Изменять какие-либо части конвейера работы MVC или адаптировать его к своим нуждам и потребностям.
- **Инфраструктура ASP.NET MVC** имеет открытый код. В отличие от предшествующих платформ веб-разработки производства Microsoft, первоначальный исходный код ASP.NET MVC доступен для свободной загрузки и даже для модификации и компиляции с целью получения собственной версии этой инфраструктуры. Это буквально неоценимо при отладке кода, обращающегося к системному компоненту, когда требуется пошагово выполнить его код (и даже ознакомиться с комментариями программистов, написавших этот код). Это также полезно, если вы создаете усовершенствованный компонент и хотите видеть, какие существуют возможности разработки, или узнать, как действительно работают встроенные компоненты.

2 ПОСТАНОВКА ЗАДАЧИ

2.1 Сравнение приложения с существующим аналогом

В интернете множество сайтов объявлений, например: kufar.by, ixr.by, doska.by. Данные сайты предусматривают просмотр и поиск объявлений, а также добавление, удаление, изменение объявления. Любой пользователь может зарегистрироваться на сайте и добавить свое объявление. На некоторых сайтах таких как kufar.by можно свободно добавлять свои объявления и откликаться на другие. На таких сайтах как auto.by необходимо подтверждение администратором объявления. Таким образом администратор осуществляет проверку на цензуру, на сам контент объявления.

Также на многих сайтах объявлений обычно предусмотрена функция редактирования профиля. Пользователи могут указывать свои данные, к примеру: фамилию, имя, телефон, адрес, и тд, если это необходимо. На продвинутых сайтах, таких как aliexpress.com можно регистрироваться через сторонние сервисы. Эта область на данный момент развивается. На многие сайты можно зайти через facebook, vk, google+, yahoo, linkedIn, youtube. Пользователи находят это удобным и предпочитают аутентифицироваться через сторонние сервисы.

На данном этапе развития информационных технологий большинство пользователей очарованы тэгами. Их можно оставлять везде и всему. Любой сущности можно оставить свой комментарий в виде тэга и не один. Данное приспособление привлекает пользователей, поскольку удобно осуществлять фильтрацию по тэгам. И любой объект можно пометить чем-то своим. Удобно осуществлять поиск по тэгам и таким образом сразу находить интересующий нас предмет/область/статью.

Большинство сайтов сейчас работают быстро, однако наиболее удобным является технология обновления какой-либо части веб-страницы без ее перезагрузки. К примеру, чтобы "лайк" ставился сразу, чтобы объявление помечалось как "избранное" по щелчку. Поэтому большинство разработчиков стараются предусмотреть такой поворот событий и сделать свое веб-приложение наиболее удобным и комфортным для пользователей.

2.2 Постановка задачи

Разработать программное средство “Сайт Доска Объявлений” с использованием языка программирования C sharp.

Реализовать функциональный набор:

1. Регистрация/вход пользователя в систему;
2. Просмотр объявлений, размещенных на сайте;
3. Добавление, изменение, удаление объявлений для зарегистрированных пользователей;

4. Создание 2 видов пользователей: администратор и пользователь с соответствующими функциями;

5. Поиск объявлений;

6. Добавление изображений для объявлений;

Дополнение: Предусмотреть аутентификацию через сторонние сервисы(vk, google+, facebook), реализовать возможность редактирования профиля (изменение аутентификационных данных, загрузка аватара), реализовать админку (главный администратор сайта проверяет каждое объявление и разрешает или запрещает его публикацию)

Приложение разрабатывать на платформе ASP.NET MVC

3 РАЗРАБОТКА ПРИЛОЖЕНИЯ

4 РУКОВОДСТВО ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ ВЕБ-САЙТА

ИСПОЛЬЗОВАНИЕ

ЗАКЛЮЧЕНИЕ

Что получилось в результате выполнения курсовой работы, что планируется добавить в будущем.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

ПРИЛОЖЕНИЕ А
(обязательное)
Исходный код программы

`<html>HTML-код</html>`