МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УО «Витебский государственный университет имени П.М. Машерова»

*Кафедра прикладной математики*

*и механики*

Курсовая работа

**Анализ возможностей технологии ASP.NET Core 1.0**

Студентка 53 гр. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ О. А. Урбан

подпись, дата

Руководитель,

старший преподаватель

кафедры ПМ и М \_\_\_\_\_\_\_\_\_\_\_\_\_\_ О. П. Оганджанян

подпись, дата

Работа защищена «\_\_»\_\_\_\_\_\_\_\_\_ 2016г. с оценкой «\_\_\_\_\_\_\_\_\_\_\_»

Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись Фамилия И.О.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись Фамилия И.О.

Витебск, 2016 г.

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc468352595)

[Задание на курсовую работу 4](#_Toc468352596)

[Раздел 1. Технология ASP.NET Core 1.0 5](#_Toc468352597)

[1.1 Концепция технологии 5](#_Toc468352598)

[1.2 История развития технологии ASP компании Microsoft 7](#_Toc468352599)

[1.3 Особенности разработки приложений 9](#_Toc468352600)

[1.4 Начало работы с ASP.NET Core 1.0 14](#_Toc468352601)

[1.4.1 Создание проекта 14](#_Toc468352602)

[1.4.2 Структура проекта ASP.NET Core 16](#_Toc468352603)

[Раздел 2. Web-приложение «ESharp» 18](#_Toc468352604)

[2.1 Общая характеристика 18](#_Toc468352605)

[2.2 Проектирование 19](#_Toc468352606)

[2.2.1 Интерфейс 19](#_Toc468352607)

[2.2.2 Хранилище данных, обработка данных 23](#_Toc468352608)

[ЗАКЛЮЧЕНИЕ 24](#_Toc468352609)

[СПИСОК ЛИТЕРАТУРЫ 25](#_Toc468352610)

# ВВЕДЕНИЕ

Развитие информационных технологий не стоит на месте. Каждый год выпускаются десятки новых фреймворков, дополнений, обновлений для облегчения разработки и объединения разных компонентов сложного программного проекта. Современный высококвалифицированный разработчик программного обеспечения должен быть в курсе последних разработок в сфере IT, а также уметь быстро изучать их и применять на практике. Однако выбор подходящей технологии должен основываться не только на личных предпочтениях разработчика, но и на анализе авторитетных источников и опыта использования различных подходов.

Одной из самых популярных технологий для разработки web-приложения является ASP.NET от компании Microsoft. Она является составной частью платформы Microsoft. NET и развитием более старой технологии Microsoft ASP.

В 2015 году была представлена последняя революционная версия ASP.NET Core. «ASP.NET Core – это работа над ошибками классического ASP.NET MVC, возможность начать с чистого листа. » - отмечает независимы эксперт Морис де Бейер. [1] В своей последней разработке компания Microsoft акцентировала свое внимание на кроссплатформенность, модульность. Благодаря ASP.NET Core уже в 2016 году у пользователей операционных систем Mac OS и Linux появилась возможность разрабатывать приложения на платформе .NET.

Впервые в странах СНГ технология ASP.NET Core обсуждалась на конференции для разработчиков DotNext 2015 в Москве[2]. Специалисты в области разработки приложений на платформе .NET М. Филиппов и А. Акиньшин в своих докладах активно обсуждали будущее ASP.NET Core и перспективы развития технологии.

Изучением технологий на платформе .NET занимается опытный специалист в области информационных технологий Adam Freeman. Его самые известные книги « Pro ASP.NET Core MVC» [3] и «Pro ASP.NET MVC 4»[3] пользуются большим спросом у разработчиков во всём мире.

К сожалению, в настоящее время технология считается сырой, также не весь функционал с ASP.NET 4.6 был перенесен. Поэтому пока не рекомендуют спешно переходить на новую технологию. Однако ситуация может поменяться в течении нескольких лет. По мнению специалистов через 2-3 года технология прочно займет своё место и сменит устаревшую ASP.NET 4.6.

# Задание на курсовую работу

1. Тема курсовой работы.

Анализ возможностей технологии ASP.NET Core 1.0.

2. Исходные данные к курсовой работе.

3. Перечень подлежащих разработке вопросов или краткое содержание

расчетно-пояснительной записки:

* Изучение основ технологий ASP.Net Core 1.0 и ASP.NET Core MVC;
* Проектирование пользовательского интерфейса, структуры и хранилища информации для web-приложения «ESharp»
* Разработка главной страницы web-приложения «ESharp» на основе данных технологий.

4. Перечень функциональных требований к web-приложению.

5. Примерный календарный график выполнения курсовой работы.

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование этапов выполнения работы | Сроки выполнения |
| 1 | Изучение технологии ASP.NET Core 1.0 | сентябрь - октябрь 2016г. |
| 2 | Разработка дизайна приложения. | октябрь 2016 г. |
| 3 | Проектирование структуры и хранилища информации | октябрь-ноябрь  2016г. |
| 4 | Реализация главной страницы с помощью выбранной технологии | ноябрь – декабрь 2016 г. |
| 5 | Подготовка пояснительной записки, презентации, защита курсовой работы. | декабрь 2016 г. |

5. Дата выдачи задания 12 сентября 2016 г.

6. Срок сдачи законченной курсовой работы 02 декабря 2016г.

# Раздел 1. Технология ASP.NET Core 1.0

## Концепция технологии

ASP.NET Core - это кроссплатформенный фреймворк с открытым исходным кодом для создания современных облачных web-приложений. Приложения ASP.NET Core могут быть запущены под NET Core или под полной версией .NET Framework. Фреймворк состоит из модульных компонентов, что дает вам гибкость при создании решений. Вы можете разрабатывать и запускать ASP.NET Core приложения под Windows, Mac OS X и Linux. ASP.NET Core имеет открытый исходный код на  GitHub.[5]

Компания Microsoft объясняет, почему они так кардинально поменяли название своей технологии: «Почему 1.0? Потому что эти инструменты новые. Вся концепция .NET Core – новая. Инструменты .NET Core 1.0 CLI– совершенно новые. И это не единственная причина, .NET Core не содержит всего функционала .NET Framework 4.6. Мы все еще работаем над серверными графическими библиотеками. Мы все еще работаем над устранением отставания ASP.NET Core 1.0 от ASP.NET 4.6.» [6]

На рисунке 1 представлена структура ASP.NET Core приложения. Приложения можно разрабатывать на одной из трёх операционных систем:

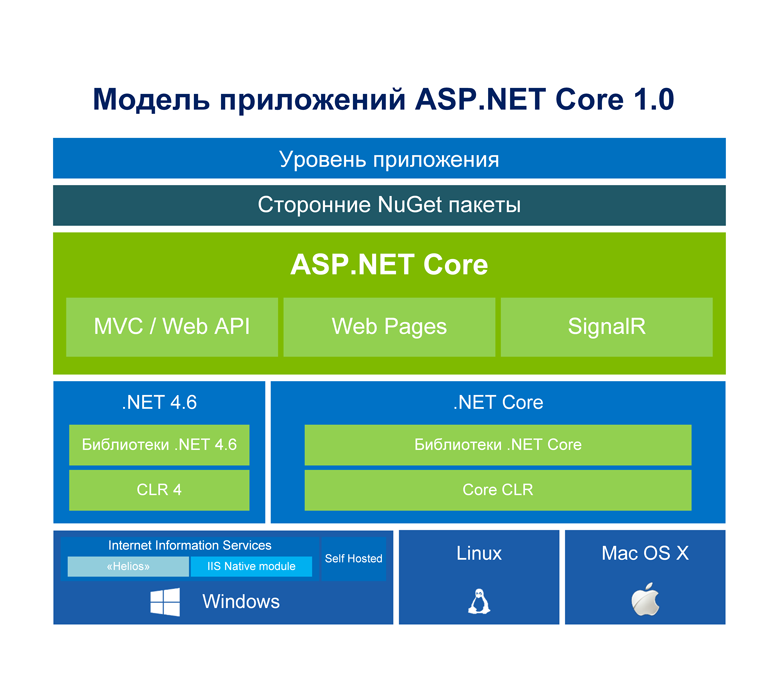
Windows, Linux, Mac Os X. На нее устанавливается набор специальных библиотек для взаимодействия операционной системы с приложениями и технологиями. – платформа. В настоящее время используется два вида платформ для ОС Windows: .NET Framework 4.6 и более современная .NET Core. Для ОС Linux и Mac OS X используется только платформа .NET Core.

Рисунок 1. Структура приложения ASP.NET Core

Как можно видеть из рисунка, ключевыми свойствами новой технологии являются полная модульность и кроссплатформенность. Почти вся платформа написана с нуля и на управляемом коде, старого кода тут нет. Весь исходный код новейшей версии web-платформы доступен на GitHub. Технологии ASP.NET MVC и Web API соединены в одну, а Web Forms больше не используется в версии ASP.NET Core.

## История развития технологии ASP компании Microsoft

В конце ХХ века компания Microsoft активно интересовалась созданием технологии для разработки web-приложений. В 1996 г. она предложила концепцию Active Server Pages, известная теперь как ASP. ASP приложения имели следующий принцип работы: на сервере размещалась web-страница, которая содержала серверные скрипты на VBScript или Jscript, при запросе страницы осуществлялся запуск и выполнение кода скриптов с последующей генерацией HTML. По сравнению с предыдущей технологией CGI был совершен большой шаг вперед. С выходом платформы Microsoft .NET Framework в 2002 году вышла совершенно новая платформа для разработки веб-приложений ASP.NET, которая была обновлена в 2003 году до версии ASP.NET 1.1. Именно с этого момента началась история ASP.NET. На то время это была совершенной новая технология, которая имела массу преимуществ перед ASP. Это был уже полноценный фреймворк для разработки web-приложений.

Через 2 года было выпущено крупное обновление для ASP.NET. Компания Microsoft в свою разработку добавила такие элементы как темы, мастер-страницы и т. д. Web-разработка развивалась быстрыми темпами, страницы становились всё более интерактивными, насыщенными. Разработчики начали применять популярную в настоящее время технологию AJAX.

В 2007 году было выпущено еще одно крупное обновление – ASP.NET 3.5. Именно тогда появилась поддержка таких мощных средств как: ASP.NET AJAX, LINQ, Dynamic Data. С выпуском ASP.NET 3.5 SP1 в 2009 году был добавлен новый подход для обработки запросов веб-страниц на основе шаблона MVC. Теперь их стало целых два: Web Forms и MVC. В настоящее время эти два подхода сливаются в один универсальный. Заметим, что MVC практически полностью вытеснила технологию Web Forms.

Одновременно с запуском .NET Framework 4 была обновлена и ASP.NET до четвёртой версии. Ещё одно крупное обновление платформы было выпущено с выходом ASP.NET 4.5. Появилось новое мощное средство для разработки RESTful приложений – Web API, а также SingnalR.

С выходом среды разработки Visual Studio 2013 были обновлены все компоненты платформы, появилась новая спецификция OWIN (Open Web Server Interface for .NET). ASP.NET 4.5.1 была обновлена и получила новую версию – ASP.NET 4.6. Следует отметить, что под ASP.NET 4.6 подразумевается весь следующий спектр технологий: Web Forms, MVC 5.x, Web API 2.x и SignalR 2.x.

Версия 4.6 получила большую популярность и множество положительных отзывов. Она используется для разработки приложений и в настоящее время. Однако разработчики не остановились и продолжили совершенствование технологии. Дальнейшее развитие спецификации OWIN легло в основу ASP.NET 5. В 2015 году компания Microsoft неожиданно переименовала платформу ASP.NET 5 в ASP.NET Core v1.0. По какой причине? Насколько серьёзные изменения ждут технологию ASP.NET в будущем? Такие эксперты в мире IT, как Scott Hanselman, Rick Strahl, Simone Chiaretta и другие, в своих статьях объясняли причину смену названия технологии. Они говорили о том, что цифра в версии может вводить в заблуждение, т.к. основной особенностью разработки является кроссплатформенность. Microsoft переименовала платформу, чтобы внести больше ясности в планы по дальнейшему развитию. Как говорит один из экспертов, далее следует ждать внедрение в ASP.NET Core новых технологий и языков: SignalR, WebPages, VB, F# и других полезный вещей.

Хронология релизов ASP.NET наглядно представлена на рисунке 2

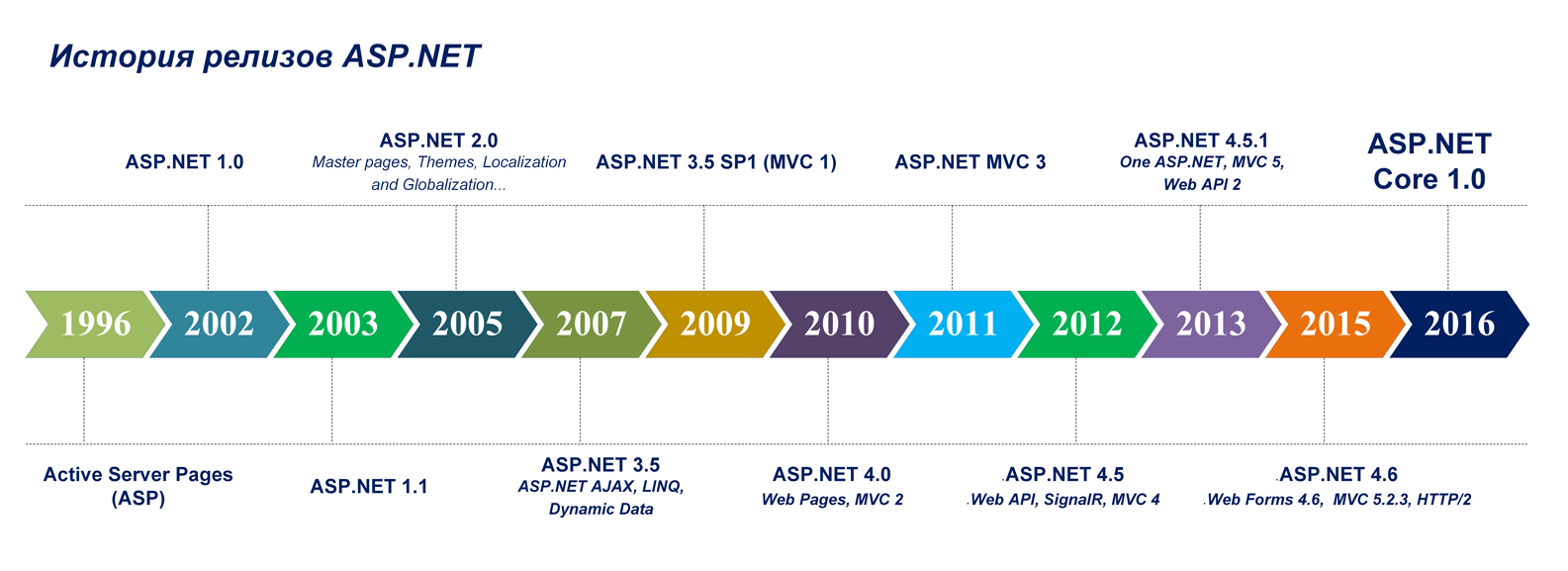


Рисунок 2. История релизов ASP.NET

## Особенности разработки приложений

В ASP.NET Core произошло большое число архитектурных изменений, в результате чего фреймворк стал компактным и модульным. Рассмотрим некоторые из них, наиболее популярные и масштабные.

**Открытый код**

Компания Microsoft и его сообщество разработчиков сделали акцент на прозрачности и открытости кода. Все ресурсы технологии ASP.NET Core 1.0 теперь доступны в системе GitHub. На сайте можно найти большое количество активных репозиториев, которые включают большую часть функционала ASP.NET Core 1.0, а также промежуточные программы, демоверсии и другие полезные инструменты. Все проекты активно обновляются участниками Microsoft и сообществом в целом. Такой подход способствует более быстрому развитию и совершенствованию кода. [7]

**Модульность**

Ключевым свойством новой технологии является его полная модульность. Все необходимые компоненты для приложений загружаются через диспетчер пакетов NuGet. Этот диспетчер использовался еще со времен Visual Studio 2010, однако теперь он был расширен для того, чтобы включать в себя основные функции ASP.NET. У разработчиков появилась возможность выбрать, какой функционал включать в свои решения.

**Объединение MVC и Web API**

Web API – это фреймворк для разработки сервис ориентированных web-приложений, в то время как MVC является более универсальным средством. В предыдущих версиях технологии фреймфорки основывались на различных базовых классах: System.Web.MVC и System.Web.Http / System.Net.Http, соответственно. Различия в понятиях, пространствах имен, классах привели к путанице. Эта проблема была решена с помощью ASP.NET MVC 1.0 Core. В настоящее время существует один набор объектов в пределах одного пространства имен (Microsoft.AspNet.Mvc). Этот стандартизированный подход упрощает разработку как MVC и Web API приложений.

**Отказ от Web Forms**

Web Forms – это средство позволяет графически создавать страницы ASP.NET – точно так же, как создаются окна в Visual Basic 6 или в C++ Builder. Другими словами, потребуется перетащить необходимые элементы управления из панели инструментов на поверхность формы, затем слегка подкорректировать код формы и написать обработчики событий для элементов управления.

В ASP.NET Core компания Microsoft отказалась от технологии Web Forms. Разработчики фреймворка объединили некоторый функционал Web Form с MVC и Web Pages в единую технологию. Известный эксперт в сообществе разработчиков ASP Stephen Walther в своём блоге отметил одну из причин отказа: «Web Forms приложения не имеют преимуществ перед новыми подходами, которые реализованы в ASP.NET Core».[8]

**Кроссплатформенность и сервера Kestrel, WebListener**

Главным отличием технологии ASP.NET Core от её предыдущих версий является кроссплатформенность. Это значит, что платформа .NET Core, на которой построена технология, может быть развернута на основных популярных операционных системах: Windows, Mac OS X, Linux. Хотя Windows в качестве среды для разработки и развертывания приложения до сих пор превалирует, но теперь разработчики не ограничены только этой операционной системой. Появилась возможность запускать web-приложения не только на ОС Windows, но и на Linux и Mac OS X.

На данный момент ASP.NET Core поддерживает развертывание приложения на стандартных веб-серверах IIS и IIS Express, а также предоставляет возможность запускать приложение без IIS в рамках собственного процесса с помощью двух дополнительных серверов, которые поставляются вместе с ASP.NET Core:

• Microsoft.AspNetCore.Server.WebListener (WebListener)

• Microsoft.AspNetCore.Server.Kestrel (Kestrel)

WebListener – это HTTP сервер, который работает только на Windows. Для работы с сервером необходимо добавить в приложение зависимость «Microsoft.AspNet.Server.WebListener». На данный момент существует лишь предварительная версия сервера.

Kestrel представляет кроссплатформенный веб-сервер, основанный на кросплатформенной библиотеке асинхронного ввода/вывода libuv. Kestrel использует сокеты и полностью состоит из управляемого кода. Kestrel развивается как opensource-проект, и при необходимости на гитхабе можно посмотреть его исходный код. К слову, на операционных системах Mac OS X и Linux приложения разворачиваются только на сервере Kestrel. [9]

**Пакетный менеджер Bower**

Каждый разработчик .NET приложений хоть раз сталкивался с менеджером пакетов NuGet для загрузки дополнительного функционала. Начиная с версии Core компания Microsoft добавила еще один менеджер пакетов – Bower.

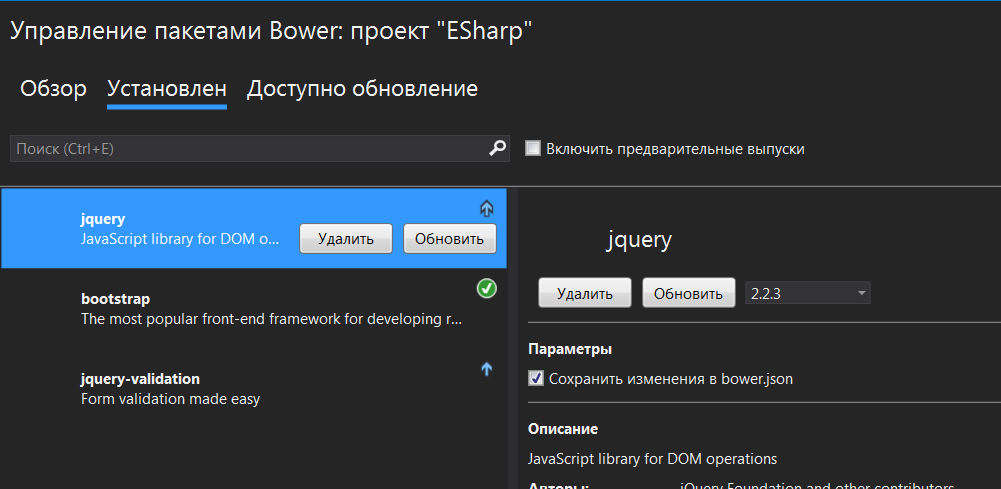
Bower представляет собой менеджер, предназначенный для управления пакетами и библиотеками пользовательского интерфейса, таких как JavaScript, CSS и др. Стандартный проект ASP.NET Core по типу Web Application уже имеет встроенную поддержку Bower. На сегодняшний день менеджер поддерживает более 30 тысяч различных пакетов, которые можно подключить в свой проект. Добавляют библиотеки так же, как и в NuGet: выбираем вкладку «Dependencies/Bower», в контекстном меню находим «Manage Bower Packages». После этого на экране появится окно менеджера (рисунок 3), в строку «Search» вводим название необходимой библиотеки и скачиваем. Установка происходит автоматически после скачивания.

Рисунок 3. Менеджер пакетов Bower

Список всех клиентских пакетов можно найти в файле bower.json, который по умолчанию скрыт. В листинге 1 представлено содержимое файла по умолчанию:

Листинг 1 - Файл bower.json

{

  "name": "asp.net",

  "private": true,

  "dependencies": {

    "bootstrap": "3.3.6",

    "jquery": "2.2.0",

    "jquery-validation": "1.14.0",

    "jquery-validation-unobtrusive": "3.2.6"

  }

}

Данное описание означает, что в проект подключены библиотеки Bootstrap и JQuery. Также название пакета можно ввести вручную или поменять его версию.

**Тэг-хэлперы.**

«Тэг-хелперы – это одна из особенностей, которая может оказать наибольшее влияние на путь создания своих представлений в приложении ASP.NET MVC. Tag Помощники являются лучшей альтернативой использованию традиционных хелперы MVC »[6] – отмечает в своей работе известный специалист в области разработки .NET приложений Stephen Walther. Тэг-хэлперы используют в представлениях и выглядят они как обычные HTML тэги, однако при работе приложения они обрабатываются на стороне сервера. Их можно понимать как более удобный аналог HTML helpers. На рисунке 4 представлен пример кода с использованием тэг-хэлперов и без него.

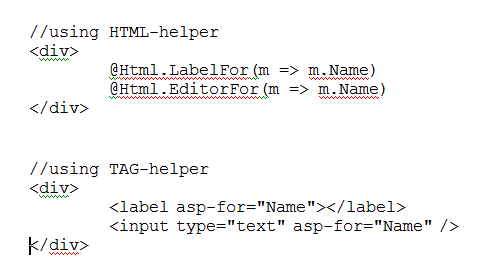


Рисунок 4 – Пример кода с использованием Tag helpers

**View Component**

View Component (или компонент представлений) представляет собой новую идею, которая появилась в ASP.NET MVC Core. Она во многом схожа с частичными представлениями. Однако при этом View Component предназначен для таких задач, которые, возможно, сложно решить с помощью одних чаcтичных представлений, например, создание динамических меню, облако тегов, панель логина, корзина покупок и так далее. View Component можно понимать как функцию или модуль, который и обрабатывает данные, и в результате формирует часть предcтавления. Главное отличие от чаcтичных предcтавлений – это формирование модуля на cтороне контроллера.

View Component соcтоит из двух частей: класса на языке C# и частичного представления, которое вызывает методы этого класса. При этом View Component не может обрабатывать HTTP-запросы, а генерируемый им cодержимое включается в код родительского предcтавления, в котором вызывается компонент.

Также подобно контроллерам компонент должен представлять класс с модификатором public. Также, данный клаcсе не должен быть вложенным или абстрактным.

Есть три способа определения компонента. Рассмотрим один из них. Для использования View Component необходимо:

1. Создать класс с окончанием ViewComponent (например, Phones ViewComponent)
2. Определить метод, который будет выполнять формирование модуля (например, create)
3. Вызвать модуль в представлении (например, @await Component.createAsync(“Phones”))

В работе рассмотрена только небольшая часть нововведений в технологии ASP.NET Core, которая необходима для начальной разработки приложений. Компания Microsoft добавила в свой проект большое количество специального функционала, такие как инверсия зависимости, технология middleware и другие.

## Начало работы с ASP.NET Core 1.0

### 1.4.1 Создание проекта

Для работы с технологией ASP.NET Core необходима среда разработки Visual Studio 2015 с пакетом обновлений не ниже Update 3. Если среда была уже установлена ранее, то программа сама попросит обновление. Достаточно обратить внимание на символ флага в верхней части экрана. Обновление программы может занять довольно продолжительное время. Если среда разработки не установлена, то все необходимые компоненты для загрузки можно найти на официальной странице среды разработки [www.visualstudio.com](http://www.visualstudio.com). В данном случае не важно, какой выпуск Visual Studio использовать – платный Enterprise или бесплатный Community. Оба этих выпуска имеют встроенные средства для создания приложений на ASP.NET Core. Также для работы с ASP.NET Core необходимо установить инструмент .NET Core 1.0.1 for Visual Studio.

Компания Microsoft, специально к выходу обновления Update 3, выпустила еще одну среду разработки – Visual Code, которая является кроссплатформенной. Программу можно использовать на ОС Windows, Linux, Mac OS. В этом случае нам также надо установить .NET Core SDK для целевой операционной системы, который можно найти на официальном сайте компании Microsoft.

16 ноября 2016 года на конференции разработчиков «Connect() //2016» была представлена превью-версия среды Visual Studio для ОС Mac OS X (рисунок 5). Среда уже доступна для скачивания. Как отмечают специалисты: «Visual Studio для Mac — это полноценная среда разработки, предназначенная для создания мобильных приложений, облачных сервисов и приложений для Mac. Первый превью-релиз включает поддержку создания нативных iOS и Android приложений на C# и F# с помощью Xamarin, а также создание бэкэнда с помощью Azure сервисов, и, естественно, поддержку ASP.NET Core» [10] «Как и в случае с Visual Studio для Windows, в дополнении будет идти Visual Studio Code для случаев, когда вам не нужна полноценная интегрированная среда разработки, а нужен лишь легкий автономный редактор исходного кода», - отмечает представитель компании Microsoft. [11]

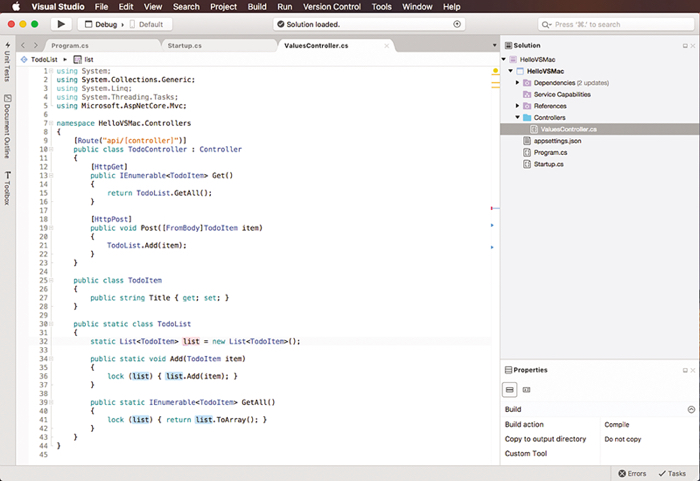


Рисунок 5 – Visual Studio, запущенная в Mac OS

В курсовом проекте использовалась операционная система Windows 7, среда Visual Studio 2015. Рассмотрим создание проекта ASR.NET Core MVC .

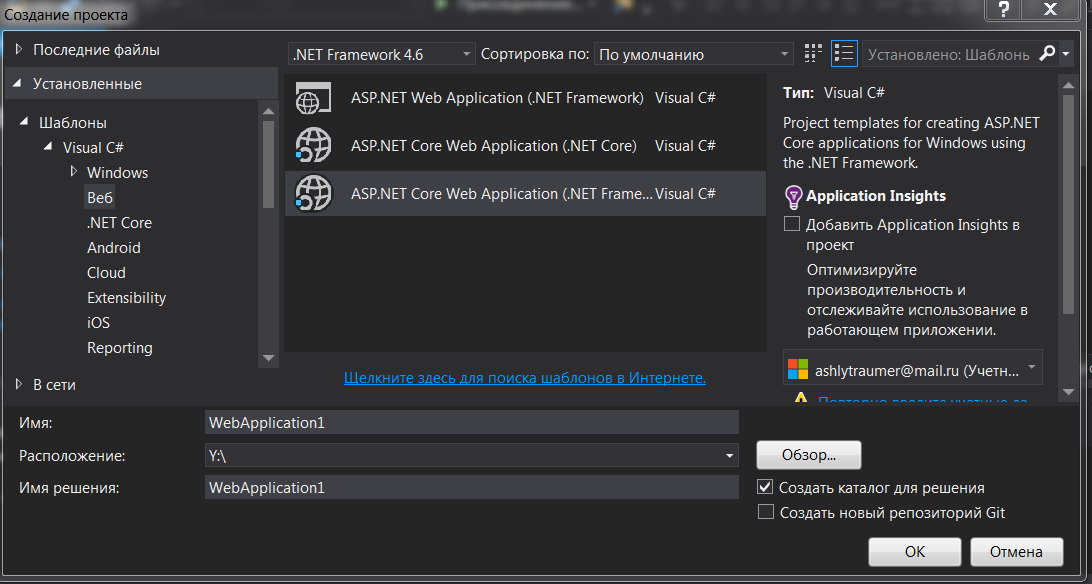
В меню File (Файл) выберем пункт New и далее пункт Project. Откроется диалоговое окно создания проекта. (рисунок 6)

Рисунок 6 – Создание проекта ASP.NET Core

Вкладка «Веб» предлагает следующие три типа проектов:

* ASP.NET Web Application (.NET Framework) - создание стандартных проектов – Web Forms, MVC 5, Web API.
* ASP.NET Core Web Application (.NET Core) - создание web-приложений ASP.NET Core, которое будет использовать среду .NET Core.
* ASP.NET Web Application (.NET Framework) – создание web-приложения ASP.NET Core, которое будет использовать полный .NET Framework.

Для разработки приложений ASP.NET Core применяются два последних типа проектов. Первый использует кроссплатформенную среду .NET Core, а второй – полноценный, но устаревший .NET Framework.

В курсовом проекте использовался тип приложения - ASP.NET Web Application (.NET Framework).

### 1.4.2 Структура проекта ASP.NET Core

Рассмотрим базовую структуру стандартного проекта ASP.NET Core. На рисунке 7 представлены структуры проектов ASP.NET 4.6 MVC и ASP.NET Core 1.0 MVC.

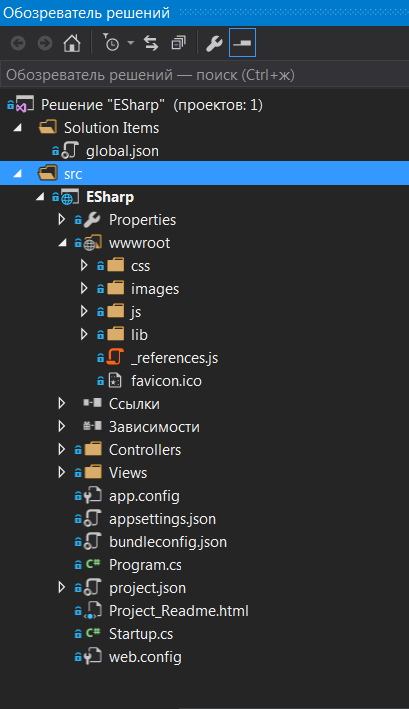
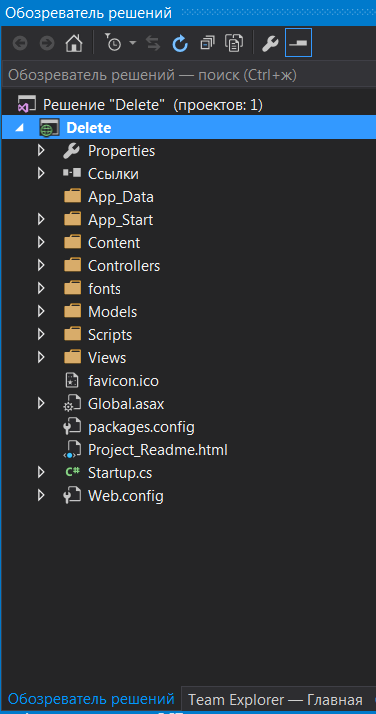


Рисунок 7 – Структуры проектов ASP.NET 4.6 MVC (слева) и ASP.NET Core MVC (справа)

Первое, на что следует обратить внимание - это глобальный узел Solution Items. Он содержит файл global.json. Файл используется для конфигурации всего решения. Он содержит всего две секции projects и sdk (Листинг 2):

Листинг 2 – Файл global.json

{

  "projects": [ "src", "test" ],

  "sdk": {

    "version": "1.0.0-preview2-003121"

  }

}

Элемент sdk определяет версию .NET Core, которая будет использоваться приложением.

Элемент projects показывает, в какой папке на жестком диске будет располагаться исходный код. По умолчанию это папка src, содержащая проект. Также здесь может быть указана папка для проекта тестов – tests. Создается и указывается она вручную. Каталог src содержит следующие новые папки и файлы:

* References: все добавленные в проект пакеты и библиотеки.
* wwwroot: узел предназначен для хранения различных статических файлов - картинок, скриптов javascript, файлов css и т.д., которые используются в приложении. Основной целью добавления узла wwwroot по сравнению с предыдущими версиями ASP.NET, состоит в том, чтобы разграничить доступ к статическим файлам, к которым разрешен доступ со стороны клиента (например, style.css) и к которым доступ запрещен (например project.json).
* Dependencies: узел служит для хранения зависимостей, которые используются при разработке клиентской части приложения.
* Program.cs: главный файл приложения, с которого и начинается его выполнение. Код файла настраивает и запускает сервер (обычно IIS), в рамках которого разворачивается приложение.
* project.json: определяет все зависимости, которые используются в проекте, а также различные конфигурационные настройки.
* Project\_Readme.html: файл приветствия, открывающий в Visual Studio по умолчанию при создании проекта.
* Startup.cs: файл, который определяет класс Startup, содержит логику обработки входящих запросов.
* web.config: файл, который настраивает модуль AspNetCoreModule. Файл web.config указывает на команду запуска и аргументы запуска, которые будет использоваться для запуска приложения .NET Core.

Данная структура не представляет проект полнофункционального приложения, однако достаточна для разработки простых приложений.

# Раздел 2. Web-приложение «ESharp»

## 2.1 Общая характеристика

Web-приложение «ESharp» (Electronic Sharp) - это наглядно-справочное электронное пособие, которое позволяет студентам, молодым специалистам освежить в памяти знания об языке программирования C#, восполнить пробелы, обобщить недавно полученные знания в области программирования. Пособие позволяет повторить и систематизировать свои знания перед выполнением лабораторных работ по языку C# и технологии ASP.NET, зачетом или экзаменом. Благодаря пособию можно подготовиться к собеседованию в компании, либо быстро освежить свои знания перед и во время выполнения рабочих проектов.

Главный принцип web-приложения – «Наглядность». Практически вся информация представлена в виде картинок, схем, таблиц, примеров кода, а также видео файлов. Наглядность предоставляет возможность усвоить большое количество информации намного быстрее и качественнее.

Приложение разработано на базе ASP.NET Core 1.0. Технология позволяет быстро и качество «собирать» web-приложения, модернизировать его на любой операционной системе.

Web-приложение состоит из двух основных представлений: страница пользователя, система администратора. На странице пользователя расположен весь контент пособия. Благодаря технологии AJAX обработка запросов будет происходить в асинхронном режиме, незаметно для пользователя. Система администратора позволяет добавлять новые разделы, статьи, редактировать и удалять их. Встроенная система шаблонов позволит организовать этот процесс максимально быстро и с минимальными ошибками.

## 2.2 Проектирование

### 2.2.1 Интерфейс

Основной задачей практической части курсового проекта является разработка макета пользовательского интерфейса, проектирование структуры web-приложения и хранилища информации. На рисунке 8 представлен макет главной страницы web-приложения. Наполнение сайта информацией планируется провести весной в рамках дипломного проекта:

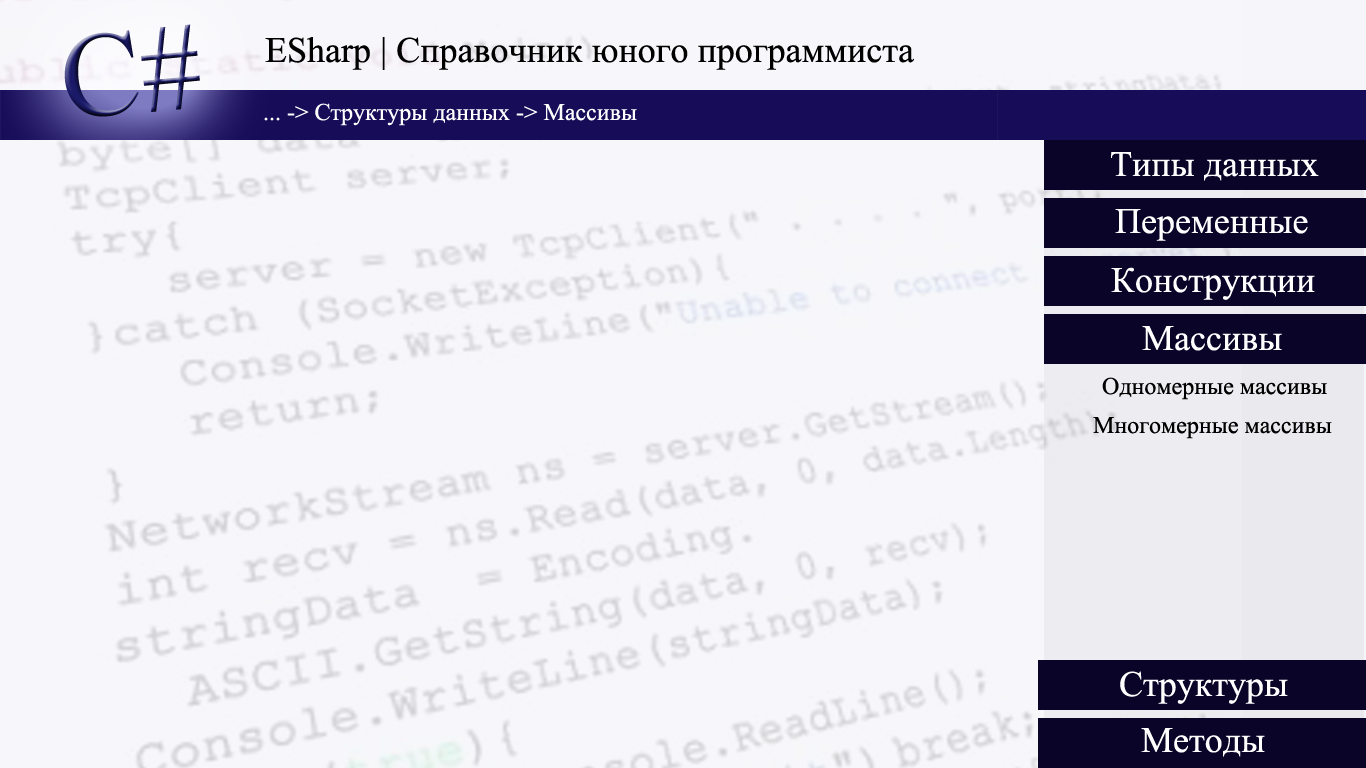


Рисунок 8 – Главная страница web-приложения

По мнению психологов, синий или голубой цвет являются одними из наиболее благоприятных для человека. Студентам, ученикам, да и всем людям умственного труда следует использовать немного синего цвета в своём окружении: синие стены или шторы, синие шрифты или синие обои на рабочий стол. Считается, что этот цвет усиливает интеллект, а также обладает успокаивающими свойствами.

Навигация по статьям будет осуществляться через меню в правой части экрана. Меню будет разрабатывалось с помощью библиотеки JQuery, объект «Аккордеон» (листинг 3 и 4). Запросы будут обрабатываться с помощью технологии AJAX.

Листинг 3 – Файл «script.js». Вызов объекта «Аккордеон»

$(function () {

$("#accordion").accordion({ autoHeight: true });

}

Листинг 4 – Файл «index.html». Формирование главного меню

<div class="right\_content" id="accordion">

<h4>Типы данных</h4>

<div style="padding:0px">

<ul>По значению</ul>

<ul>Ссылочные</ul>

</div>

<h4>Переменные</h4>

<div></div>

<h4>Конструкции</h4>

<div style="padding:0px">

<ul>Условный оператор</ul>

<ul>Цикл с предусловием</ul>

<ul>Цикл с постусловием</ul>

</div>

<h4>Массивы</h4>

<div style="padding:0px">

<ul>Одномерный</ul>

<ul>Многомерный</ul>

</div>

<h4>Структуры</h4>

<div></div>

<h4>Методы</h4>

<div></div>

</div>

Редактировать статьи имеет право только администратор. Для того, чтобы войти в режим редактирования необходимо ввести уникальный ключевой код. Форма для ввода кода будет появляться во всплывающем окошке браузера. На рисунке 9 представлен макет начальной страницы в режиме редактирования.

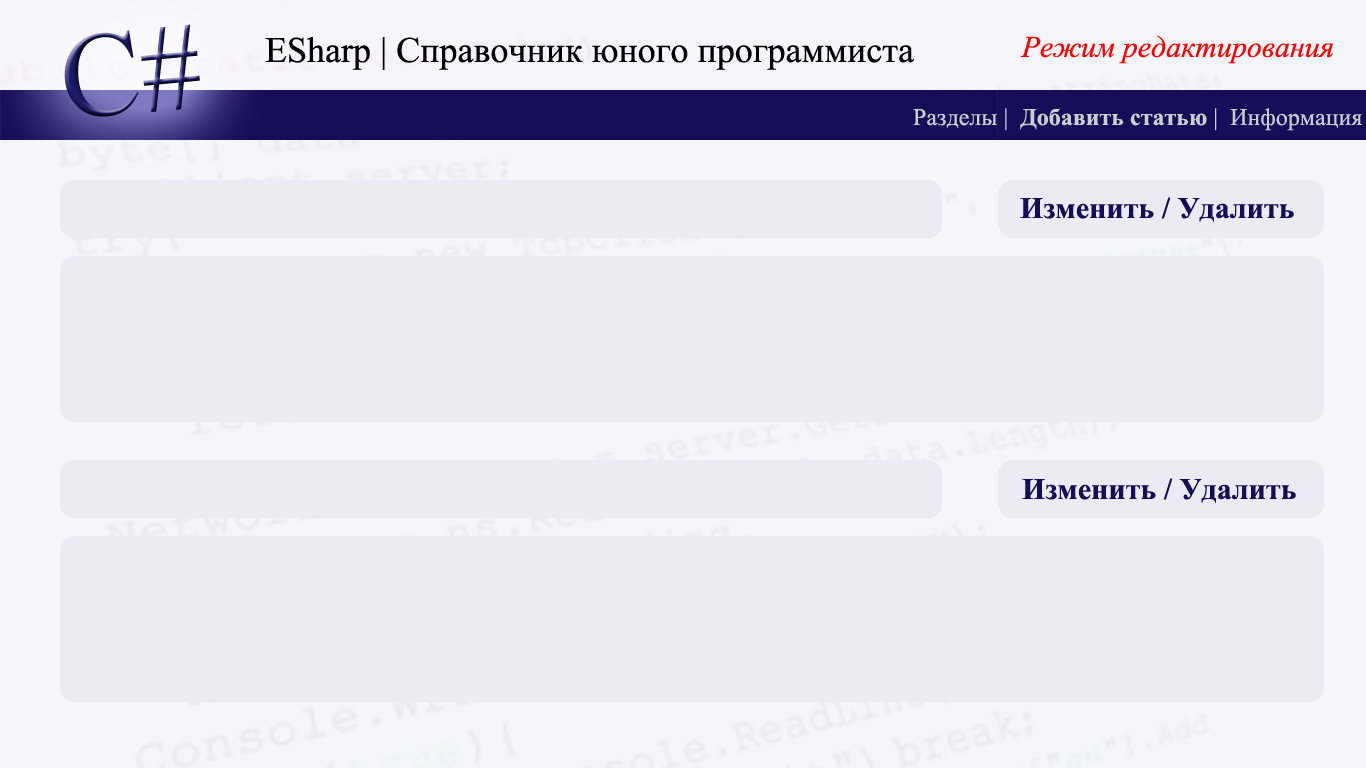
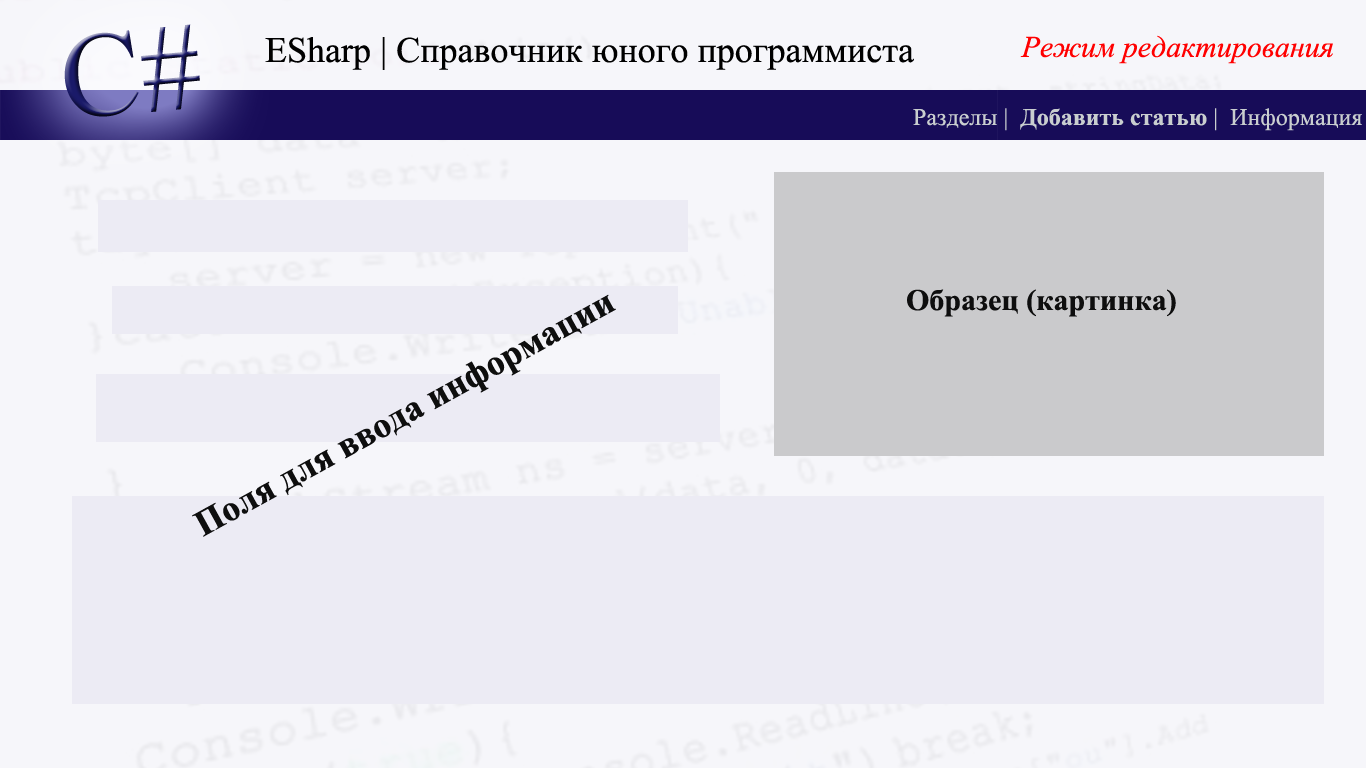
Чтобы получить возможность редактировать боковое меню, перейдите во вкладку «Разделы». При нажатии на ссылку «Добавить статью» откроется выплывающее окошко, на котором следует выбрать шаблон, по которому будет строиться статья. После этого следует заполнить форму (рисунок 10).

Рисунок 9 – Главная страница в режиме редактирования

Рисунок 10 – Редактирование статьи

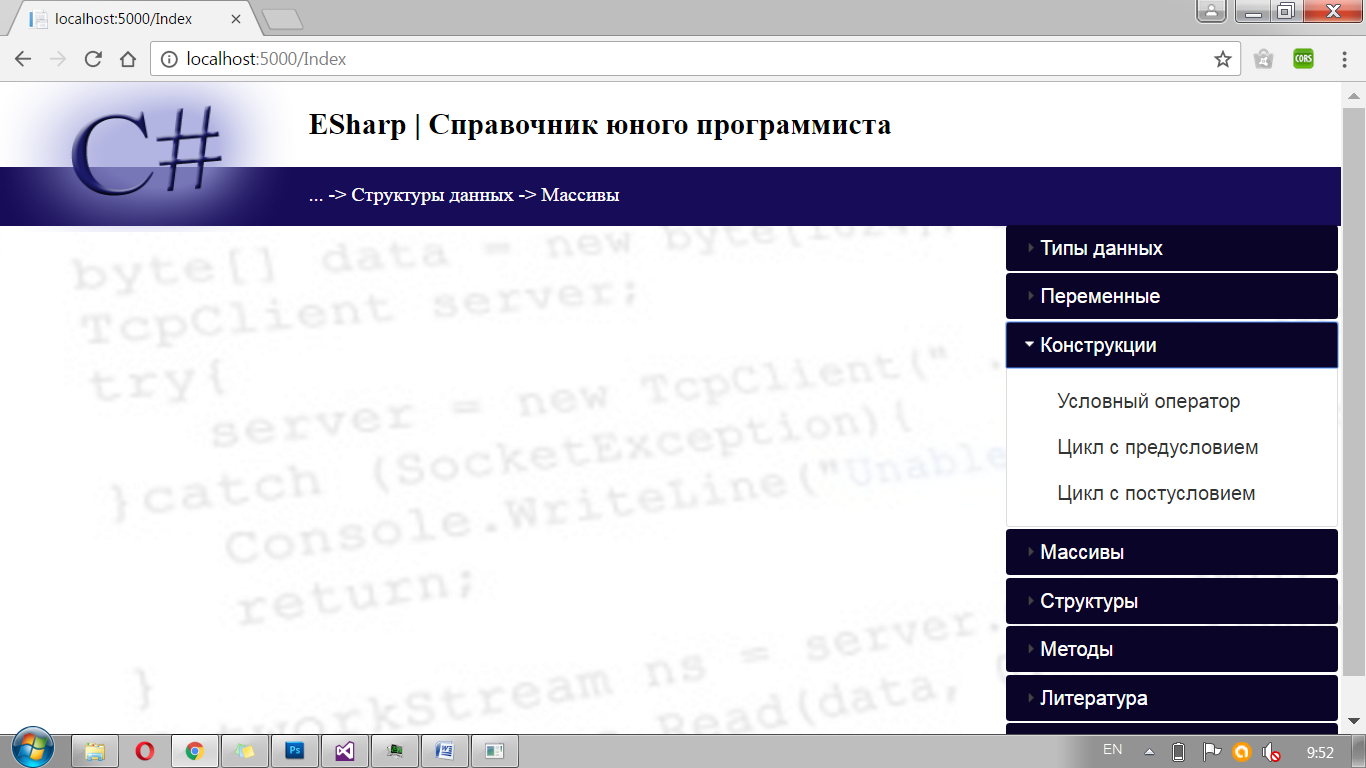
В браузере Google Chrome главная страница выглядит следующим образом (рисунок 11):

Рисунок 11 – Главная страница сайта в браузере

### 2.2.2 Хранилище данных, обработка данных

Особенностью web-приложения «ESharp» является файловое хранение данных и использование шаблонов для формирования страниц. Поскольку в проекте используется большое количество разнородной информации (картинки, текст), то было бы неуместно использовать базу данных для поиска файлов, достаточно строго определить структуру хранилища. Рассмотрим её подробнее.

«Storage» - это главная папка для хранения данных. Она располагается в каталоге проекта. Включает в себя 2 подкаталога:

* «Pages» - каталог хранит содержимое страниц;
* «Templates» - хранений файлов с шаблонами.

Для каждой страницы сайта создается своя папка в каталоге «Pages», имеющая уникальный номер. Например, мы хотим добавить новую статью об условных операторах в раздел «Конструкции» (рисунок 11). В данном случае создается папка с названием «3.1».

Каталог страницы может содержать следующие файлы:

* Info.txt – обязательный файл, содержащий служебную информацию о странице (номер шаблона, количество картинок, ключевые слова и т д);
* pictureN.png– изображение для страницы, где N – номер картинки. Рекомендованный формат данный – «.png»;
* example.txt – пример кода, где N – порядковый номер;
* textN.txt – текст страницы, где N – порядковый номер;
* links.txt – внешние ссылки;
* и т д.

Количество и название папок может быть изменено, одна смысл останется прежним. Ниже представлен пример дерева файлов (рисунок 12):

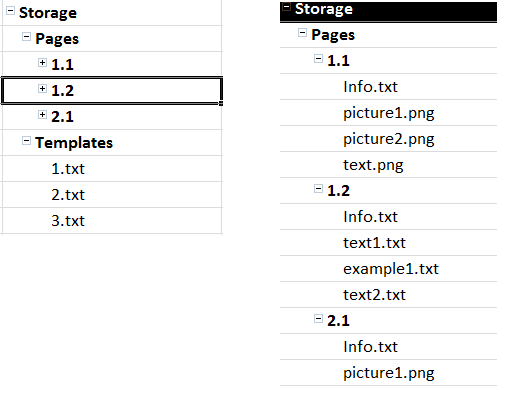


Рисунок 12 – Структура хранилища данных

Каталог «Templates» содержит шаблоны, по которым будет формироваться сама страница. Представляют собой txt-файлы с соответствующими именами: «1.txt», «2.txt» и т д. Внутри файлов располагается html-код, мета, в которые после будет вставляться картинка или файл, выделены служебным именем. В процессе формирования будет происходить поиск служебных слов и замена их на содержимое. Механизм шаблонов напоминает загрузку файлов по протоколу HTTP. Пример шаблона представлен в листинге 5:

Листинг 5 – Пример шаблона

<div>

<h2>**SetText**</h2>

<img src=”**SetImg**”></img>

<h4>**SetText**</h4>

</div>

Структура данных позволяется в случае необходимости быстро добавлять новые форматы данных, например файлы PDF или DOC, создавать новые шаблоны, редактировать информацию.

# ЗАКЛЮЧЕНИЕ

В результате работы были изучены возможности ASP.NET Core 1.0, а также особенности разработки web-приложений на этой платформе, разработан макет сайта «ESharp», продумана структура проекта, хранения данных, разработана главная страница сайта для демонстрации изученных возможностей технологии ASP.NET Core. В будущем планируется полная реализация приложения «ESharp» в рамках дипломного проекта.

# СПИСОК ЛИТЕРАТУРЫ

1. ASP.NET Core сегодня: за или против [Электронный ресурс]. – 08.09.2016. – Режим доступа: http://security-corp.org/programming/33426-aspnet-core-segodnya-za-i-protiv.html
2. Конференция DotNext [Электронный ресурс]. – 08.09.2016. – Режим доступа: <http://dotnext-moscow.ru/>
3. Freeman, Adam. Pro ASP. NET Core MVC. Apress, 2016. – 1018 p
4. Freeman A., Sanderson S. Pro Asp. net Mvc 4. – Apress, 2012. – 698 p
5. Введение в ASP.NET Core [Электронный ресурс] / Daniel Roth, Rick Anderson, Shaun Luttin. – 08.09.2016. – Режим доступа: http://dotnet.today/ru/aspnet5-vnext/intro.html
6. ASP.NET 5 мертв — представляем ASP.NET Core 1.0 и .NET Core 1.0 [Электронный ресурс] / А.Еремеева. – 21.01.2016. – Режим доступа: https://blogs.msdn.microsoft.com/rudevnews/2016/01/21/asp-net-5-asp-net-core-1-0-net-core-1-0/.
7. An Overview of What’s New in ASP.NET Core 1.0 / [Электронный ресурс]. – 07.04.2016. – Режим доступа: https://www.intertech.com/Blog/an-overview-of-whats-new-in-asp-net-core-1-0/
8. ASP.NET Core сегодня: за и против [Электронный ресурс].– 25.05.2016 Режим доступа: http://security-corp.org/programming/33426-aspnet-core-segodnya-za-i-protiv.html
9. Top 10 Changes in ASP.NET 5 and MVC 6 / [Электронный ресурс]. – Режим доступа: http://stephenwalther.com/archive/2015/02/24/top-10-changes-in-asp-net-5-and-mvc-6. – Дата доступа: 27.10.2016.
10. Серверы / [Электронный ресурс]. – 2016 – Режим доступа: http://dotnet.today/ru/aspnet5-vnext/fundamentals/servers.html
11. Visual Studio для Mac и другие новости конференции Connect(); //2016 [Электронный ресурс]. – 2016 – Режим доступа: <https://habrahabr.ru/company/microsoft/blog/315444/>
12. Microsoft выпускает Visual Studio для MacOS [Электронный ресурс]. – 2016 – Режим доступа: http://www.cnews.ru/news/top/2016-11-14\_microsoft\_vypuskaet\_visual\_studio\_dlya\_kompyuterov