**АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ**

**Факультет: Информационные технологии и управление**

**Кафедра: Компьютерная инженерия**

**Предмет: «Веб-системы и технологии»**

**Курсовая работа**

**Тема** «Создание онлайновой информационной системы для городских телефонных служб»

**Группа:** 680.22

**Курc:** 3

**Специальность:** Информационные технологии

**Код специальности:** 050616

**Студент**: Усейнова Лейла

**Руководитель:** **асс. Халилов М. Э.**

**Зав.кафедрой: доц.Рагимова Н.А.**

**АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ**

**АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ**

**ЗАДАНИЕ ПО КУРСОВОЙ РАБОТЕ**

**ФАКУЛЬТЕТ: Информационные технологии и управление**

**КАФЕДРА:** **Компьютерная инженерия**

**Группа 680.22 курс III**

**Специальность Информационные технологии**

**Студент Усейнова Лейла**

**Зачетная книжка**

**Руководитель курсовой работы асс. Халилов М. Э.**

**Срок выдачи**

**Дата сдачи**

**Тема курсовой работы «** Создание онлайновой информационной

системы для городских телефонных служб **»**

**Отзыв руководителя курсовой:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Подпись студента\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Подпись рук.курсовой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Подпись зав.кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дата защиты курсовой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Оценка\_\_\_\_\_\_\_\_\_\_**

**Пред.комиссии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

**Члены комиссии: 1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

**2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

**3.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

**СОДЕРЖАНИЕ**

**ВВЕДЕНИЕ………………………………………………………………………4**

1. **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ………...………………………………………6**
   1. **Web-программирование…………………………………………….……...6**
   2. **HTML и XHTML………………………………………………...…………..7**
   3. **Dynamic HTML……………………………………………………………..13**

**1.4.1 Каскадные таблицы стилей……………………………………...13**

**1.4.2 JavaScript…………………………………………………………...16**

**1.5 Bootstrap……………………………………………………………………..17**

1. **ПРАКТИЧЕСКАЯ ЧАСТЬ…………………………………………...……20**
   1. **Описание предметной области…………………………………………...20**
   2. **Создание окружения для проекта………………………………………..21**
   3. **Верстка HTML страницы…………………………………………………21**
   4. **Использование стилей CSS………………………………………………..29**

**3.РЕЗУЛЬТАТ…………………………………………………………………..44 4.ЛИТЕРАТУРА………………………………………………………………..45**

**5.ПРИЛОЖЕНИЯ……………………………………………………………...46**

**ВВЕДЕНИЕ**

Веб-системы и технологии относятся к различным инструментам, языкам и платформам, которые используются для разработки, развертывания и обслуживания веб-приложений. Эти технологии развивались на протяжении многих лет, чтобы не отставать от растущего спроса на веб-сервисы, что привело к появлению разнообразных языков программирования, фреймворков, библиотек и баз данных. Когда появились Web-технологии, пользоваться компьютерной техникой начали абсолютно новые категории граждан. Список социальных категорий населения, подсоединяющихся к сети интернет с целью поиска информации во всемирной паутине, постоянно увеличивается за счёт людей, которые не считают себя специалистами в сфере информационных технологий. Web-технология в корне изменила понятия об информационной обработке, как и работе с компьютерной техникой тоже.

Оказывается стандартные характеристики, описывающие прогресс в сфере вычислительного оборудования, а именно, производительность, пропускная способность, объём памяти, практически не включали в свой состав главное системное «узкое место», именуемое пользовательским интерфейсом. Устаревшая техника взаимоотношений пользователя с информационной системой тормозила продвижение передовых технологий и нивелировала преимущества от их внедрения. И лишь когда интерфейс пользователя, обеспечивающий взаимодействие человека и компьютера, разработчики довели до простого и естественного понимания обычными людьми, произошёл невообразимый взрывной рост интереса к достоинствам и возможностям компьютерного оборудования.

Web-технологиями является весь набор средств, позволяющих организовать WWW (World Wide Web), то есть всемирную паутину. Так как каждый сеанс является взаимодействием двух сторон, а именно, сервера и клиента, то и Web-технологии делятся на следующие группы:

1. Технологии серверной стороны (server-side).
2. Технологии клиентской стороны (client-side).

Технологии клиентской стороны включают в свой состав весь набор технологий по созданию веб-страниц (HTML, JavaScript, DHTML и другие), а технологии серверной стороны состоят из технологий доступа к информационным базам данных в сети интернет (CGI, PHP).

Серверные программы обеспечивают предоставление тех или иных ресурсов клиентским программам. Клиенты, когда им требуется какой-либо файл или просто какая-то информация от сервера, вырабатывают специальный запрос клиента и отправляют его серверу. Серверная программа выполняет обработку запроса и отправляет ответ сервера, который содержит запрошенную информацию или же извещение об ошибке, в случае недоступности требуемых данных. Данная компьютерная организация, или по другому, принципы формирования вычислительных систем или сетей, именуется архитектурой «клиент-сервер» или двухзвенной организацией. Как раз на базе двухзвенной архитектуры работают практически все интернет - сервисы, включая и WWW.

Данная курсовая работа рассматривает пример создания информационной системы для продуктовых магазинов.  
Разработка онлайн-информационной системы будет включать в себя следующие этапы:

1. Сбор требований: Первым шагом будет сбор требований от клиента, таких как продукты, которые он хочет продавать, целевая аудитория, функции, которые он хочет видеть на веб-сайте.

2. Дизайн: Следующим шагом будет разработка дизайна веб-сайта, включая создание каркасов и макетов, выбор цветовых схем и шрифтов, а также определение общего макета и пользовательского интерфейса.

3. Разработка: Это предполагает создание веб-сайта с использованием HTML, CSS, JavaScript и других веб-технологий.

4. Тестирование: После завершения разработки веб-сайт необходимо будет протестировать, чтобы убедиться, что он должным образом функционирует в различных браузерах и устройствах.

WEB-сайт Интернет-магазина – это система, ориентированная большей частью на покупателя. Он должен предоставлять сервисы просмотра товара, удобного поиска товара, легкость и удобство при заказе товара.   
 И так, целью моей курсовой работы является создание сайта Интернет-магазина, обеспечивающего удобство, безопасность и интуитивную понятность интерфейса пользователю.

1. **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**1.1** **Web-программирование**

Web-программирование (Web-разработка) - это бурно развивающийся раздел программирования, ориентированный на разработку динамических Интернет-приложений. Языки Web-программирования делятся на две группы: клиентские и серверные. Клиентские языки обрабатываются на стороне пользователя (в основном в браузере). Соответственно, обработка скрипта зависит от браузера пользователя, и пользователь имеет полномочия настроить свой браузер так, чтобы тот вообще игнорировал скрипты. При этом если браузер старый, он может не поддерживать тот или иной язык или версию языка, на которую опирался разработчик. С современными браузерами таких проблем возникать не должно, к тому же языки программирования не так уж часто кардинально обновляются (раз в несколько лет) и лучшие из них давно известны. Код клиентского скрипта может посмотреть каждый, выбрав в меню своего браузера «Исходный код страницы». Преимущество клиентского языка заключается в том, что обработка скриптов на таком языке может выполняться без отправки документа на сервер. Программа сразу проверит правильное заполнение формы перед отправкой, и, если необходимо, выведет ошибку. Отсюда же вытекает и то ограничение, что с помощью клиентского языка программирования ничто не может быть записано на сервер. Самым распространенным из клиентских языков признан JavaScript, разработчиками которого является компания Netscape совместно с компанией Sun Microsystems. Еще один популярный язык - это VBScript. Помимо этого в последнее время набрали популярность такие технологии, как AJAX, Adobe Flash, Microsoft Silverlight и др. Серверные языки программирования открывают перед программистом большие просторы для деятельности. Когда пользователь делает запрос на какую-либо страницу (переходит на нее по ссылке, или вводит адрес в адресной строке своего браузера), то вызванная страница сначала обрабатывается на сервере (то есть выполняются все программы, связанные со страницей) и только потом возвращается к посетителю по сети в виде файла. Этот файл может иметь расширения: HTML, PHP, ASP, Perl, SSI, XML, DHTML, XHTML.

**1.2****HTML и XHTML**

HyperTextMarkupLanguage (HTML) - это язык разметки документов во Всемирной паутине, принятый за стандартный. Большая доля всех Web-страниц в Интернете создана при помощи языка HTML (или XHTML), поэтому мы рассмотрим его подробно.

Язык HTML позволяет форматировать текст и другие элемента Web-страницы:

*Цвет, жирность, стиль, название шрифта* для текста.

*Позволяет выделять фрагменты и символы* например: ударение в слове, заголовок страницы или абзаца, сам абзац, пункт списка.

*Гипертекстовые ссылки*, позволяют переходить между документами и между фрагментами одного документа.

*Формы* для введения данных, как правило, данные из форм обрабатываются с помощью скриптов на языках программирования, ориентированных на Web, например PHP.

*Отображение мультимедийных файлов*, их может отображать сам браузер - изображения, аудиофайлы, или внешние приложения, взаимодействующие с браузером, например Flash-ролики, Java-апплеты.

HTML - язык разметки документов основанный на тэгах. Документ на языке HTML представляет собой набор элементов, при этом начало и конец каждого элемента обозначается служебными символами - тегами. Все тэги HTML начинаются с «<« (левой угловой скобки) и заканчиваются символом «>« (правой угловой скобки). Завершающий тег выглядит также, как начальный, и отличается от него прямым слэшем перед текстом внутри угловых скобок.

**<HTML></HTML>**

HTML регистронезависимый язык, теги могут быть написаны как строчными, так и заглавными буквами (в отличие от XHTML). Теги могут быть вложенными друг в друга.

<**HTML>**

**<HEAD>**

**<TITLE>**

Заголовок страницы

**</TITLE>**

**</HEAD>**

**</HTML>**

Тэги могут быть пустыми, то есть не содержать текста или других вложенных конструкций (например, <**br>**который переводит строку). Закрывающий тег в таком случае не указывается.

Также, элементы разметки могут иметь атрибуты, задающие их свойства (например, размер шрифта, цвет, расположение). Атрибуты задаются в начале тега.

**<ahref=«**http://www.yandex.ru**»>**Пример элемента с атрибутом **href.</a>**

Теги можно разделить на следующие группы.

*Гиперссылки*

**<AHREF=«**filename**»target=«\_self»>**текст ссылки**</A>**

где filename - имя файла (может быть и локальным) или адрес страницы вInternet, на который нужно совершить переход.

текст ссылки - текст гипертекстовой ссылки, который будет отображаться в браузере, как правило, выделяется подчеркиванием.

target - задает окно или фрейм, в котором будет открыт документ, при переходе по ссылке. Он может принимать значения:

\_top - документ откроется в текущем окне

\_blank - документ откроется в новом окне

\_self - документ откроется в текущем фрейме

\_parent - документ откроется в родительском фрейме

По умолчанию принимает значение \_self.

*Текстовые ссылки.*

**<H1></H1>**,**<H2></H2>**, … ,**<H6></H6>** - заголовки 1-6 уровней. Применяются для выделения частей выводимого текста (заголовок 1 - будет выведен очень большим, 6 - будет размером сопоставимым с обычным текстом).

**<P>** - обозначает начало нового абзаца. Закрывающий тег**</P>**, не является необязательным.

**<BR>** - переход на новую строку. Закрывающий тег**</br>**, отсутствует.

**<HR>** - горизонтальная линия.

**<BLOCKQUOTE></BLOCKQUOTE>** - цитата. Выделение заданного текста как цитаты.

**<PRE></PRE>** - режим предпросмотра. При этом текст заключается в рамку и выводиться не форматированным (то есть все теги, кроме </PRE>, игнорируются, но при этом переводы строки ставятся там, где они присутствуют в исходном документе).

**<DIV></DIV>** - блок текста (как правило, применяется для использования каскадных стилей CSS).

**<SPAN></SPAN>** - строка (как правило, применяется для использования каскадных стилей CSS).

*Теги форматирования текста*

**<EM></EM>** - выделение символа, на который падает ударение (обычно отображается курсивом).

**<STRONG></STRONG>** - выделение символа, на который падает усиленное ударение (обычно отображается жирным текстом).

**<I></I>** - выделение текста курсивом.

**<B></B>** - выделение текста жирным шрифтом.

**<U></U>** - подчёркивание текста

**<S></S>** - зачёркивание текста.

**<STRIKE></STRIKE>** - то же самое, что <S> … </S>

**<BIG></BIG>** - увеличение шрифта.

**<SMALL></SMALL>** - уменьшение шрифта.

**<BLINK></BLINK>** - мигающий текст.

**<MARQUEE></MARQUEE>** - сдвигающийся по экрану текст.

**<SUB></SUB>** - вывод текста под строкой. Например, **H<SUB>2</SUB> О**тобразиться в виде текста H2O.

**<SUP></SUP>** - вывод текста над строкой. Например, **E=mc<SUP>2</SUP>**отобразиться в виде текста E=mc2.

**<FONT атрибуты></FONT>** - задание атрибутов у используемого шрифта. Атрибуты могут быть следующими:

**COLOR=color** - указание цвета. Цвет может быть указан шестнадцатеричным числом в формате #rrggbb (первые 2 шестнадцатеричные цифры указывают интенсивность красного, следующие 2 - зелёного, последние 2 - синего) или названием самого цвета.

FACE= указываем имя шрифта.

SIZE= позволяет изменить размер шрифта. Размеры могут быть от 1 до 7, по умолчанию размер 3.

SIZE=+ размер или SIZE=-размер - размер больше или меньше стандартного. Например,SIZE=+2 указывает размер на 2 больше стандарта, то есть размер 5.

*Списки.*

Данная конструкция

**<UL>**

<LI> первый элемент списка </LI>

<LI> второй элемент списка </LI>

<LI> третий элемент списка </LI>

</UL>

создаёт список вида:

- первый элемент;

- второй элемент;

- третий элемент.

Также стоит отметить, что тегов есть параметры, позволяющие менять вид списка.

*Объекты.*

EMBED - вставка объектов различных типов

APPLET - вставка Java-апплетов

SCRIPT - вставка различных скриптов, например JavaScript

*Изображения.*

IMG - тег для вставки изображения. Это не закрывающийся тег.

SRC - имя локального файла или путь к нему в виде URL

ALT - текст картинки (отобразится, в виде текста, если не удалось отобразить картинку)

TITLE - подсказка (показывается при попадании курсора в область картинки)

WIDTH, HEIGHT - размеры изображения (выводимое изображение будет масштабировано до указанных размеров)

ALIGN - обтекание текста

*Таблицы.*

TABLE - тег создание таблицы. Тег имеет следующие параметры:

BORDER - задает толщину границу таблицы

CELLSPACING - задает расстояние от ячейки до ячейки

CAPTION - задает заголовок таблицы (необязательный тег)

TR - добавление строки в таблицы

TH - задает заголовок столбца (необязательный тег)

TD - добавление ячейки таблицы

WIDTH, HEIGHT - размеры таблицы

*Формы.*Формы ввода данных могут быть самыми разнообразными. Поэтому рассмотрим только основные теги:

FORM - тег для создания формы

INPUT - добавление элемента ввода

TEXTAREA - добавление текстового поля

SELECT - добавление списка (как правило, это выпадающее меню)

OPTION - пункт списка

*Символы.*Некоторые символы не могут быть выведены напрямую. Для их вывода требуется использовать их определения, например, символ апмерсанд& в коде HTML будет иметь вид **&amp**, символ меньше < будет иметь вид **&lt**, символ больше > будет **&gt**. Это ограничение введено, так как эти символы уже используются в языке HTML как служебные.

Любая HTML-страница должна иметь обозначение начала и конца документа обрамленные тегами**<html>** и **</html>** соответственно. Внутри них должны находиться теги заголовка **<head>** и **</head>**, итеги, обозначающие тело документа **<body>** и **</body>**. А внутри них могут быть произвольные комбинации из групп тегов описанных ранее.

Также рассмотрим ExtensibleHypertextMarkupLanguage (XHTML) это расширяемый язык разметки гипертекста. Стоит отметить, что язык XHTML это ни описание самого языка, а список отличий XHTML от HTML. Рассмотрим основные отличия HTML и XHTML. В XHTML все используемые теги должны иметь закрывающий тег. Теги, не имеющие закрывающего тега должны оканчиваться символом /. Например тег **<br>**, должен иметь закрывающий его тег **<br />**. В XHTML допускается писать теги и их атрибуты только строчными буквами. В XHTML очень строгая проверка синтаксиса не допускается использовать< и &, даже в URL, вместо них должны быть **&lt** и **&amp**. Браузеры, обнаружив ошибку синтаксиса в XHTML, должны прекратить его обработку и вывести ошибку на экран. В стандарте HTML браузер должен попытаться отобразить запрашиваемый документ. Стоит отметить, что XHTML расширяемый язык - за счет указания типа документа и возможности использовать свои теги.

Как мы видим, язык разметки HTML предоставляет широкие возможности для отображения информации, для этого в нем содержится большое количество тегов для различного форматирования выводимой информации. Язык XHTML очень похож на HTML, но более строгий, грамматические правила в XHTML менее сложные, и как следствие при созданий Web-страниц будет меньше ошибок.

**1.3 Dynamic HTML**

Dynamic HTML или DHTML - так принято называть связку языка HTML, каскадных таблиц стилей, скриптового языка и объектной модели документов. Скриптовым языком может выступать JavaScript или VisualBasic, но именно первый язык получил большую популярность и сегодня используется повсеместно. При помощи DHTML можно создавать интерактивные Web страницы, он позволяет легко и гибко обрабатывать данные запроса и формировать динамический ответ. DHTML может быть использован для реализации интерфейса Drag'n'Drop. На его основе создаются игры и другие интерактивные сервисы. Стоит отметить, что для реализации принципов DHTML достаточно лишь браузера, который будет обрабатывать содержимое страницы. То есть, нет необходимости, например, в обращениях к базе данных.

***1.3.1******Каскадные таблицы стилей***

Рассмотрим подробнее каскадные таблицы стилей - CSS (CascadingStyleSheets). Это стандарт позволяющий задавать описание внешнего вида некоторых элементов страницы на HTML. CSS используется при создании Web-страниц для определения шрифта, цвета, расположения и прочих атрибутов, используемых в документе, элементов.

Основная цель использования CSS разграничить само содержимое страницы, созданное на языке HTML и описание оформления, которое написано на CSS. Такое разделение упрощает создание и изменение документа, документом более легко управлять, а также снижается избыточность в исходном коде. Также, при помощи CSS можно легко представить один и тот же документ, но с разным форматированием или использовать разные методы вывода, такие как отображение на экране, печать документа, чтение голосом и тому подобных.

Описание стиля на CSS при открытии страницы может быть взято из разных мест: оно может быть встроенным стилем - это блоки CSS внутри страницы на языке HTML.

**<style type=«**text/css**»>**

**body {**

**color:**green;

**}**

**</style>**

Из отдельного файла .css, в котором описывается таблица стиля, на этот файл делается ссылка внутри страницы:

**<link rel=«**stylesheet**» type=«**text/css**» href=«**style.css**» />**

Это может быть Inline-стиль, при этом в HTML документе информация о стиле элемента указывается как атрибут style.

**<p style=«**font-size: 21px; color: green; **»>**Выводимый текст**</p>**

Также в любом браузере есть свой стандартный стиль, используемый по умолчанию.

Таблица стилей — это набор описаний форматов. Каждое правило стиля, в свою очередь, имеет один или несколько селекторов, которые разделяются запятыми. В любом стиле также должен присутствовать блок определений, который обозначается фигурными скобками { }, и включает в себя набор свойств и их значений.

селектор, селектор {

свойство1: значение;

свойство2: значение;

свойство3: значение;

}

В стандарте CSS присутствуют приоритеты, согласно которым применяются правила стилей, например, если для элемента подходят свойства сразу нескольких правил. Это и называется каскадом. Стоит обратить внимание и на порядок расположения применяемых свойств - у свойства, указанного позже, будет более высокий приоритет.

До создания CSS оформление стиля Web-страницы производилось непосредственно внутри самого содержимого документа. Однако с использованием CSS появилась возможность принципиально разделить содержания и описание стиля документа. Благодаря этому стало возможным лёгкое применение единого стиля при создании похожих документов, а также это оформления можно изменить легко и быстро. Появилась возможность создать несколько вариантов дизайна страницы для разных устройств, на котором ее можно просмотреть. Например, на мониторе дизайн страницы будет иметь большую ширину, при отправке документа на печать буду отбрасываться такие элементы как меню, а на КПК или сотовом будет масштабирование уменьшающее искажение текста страницы. За счет того, что таблицы стилей хранятся в отдельном файле CSS, уменьшается время загрузки страницы. В этом случае браузер получает только описание структуры документа и данные, находящиеся на Web-странице, а стиль оформления этих данных загружается браузером при первом обращении и потом берется из КЭШа. Последующее изменения дизайна сайта так же значительно упрощается. Нет необходимости изменять каждую страницу, лишь нужно внести изменения в CSS-файл. Но у CSS есть также и недостатки. Различное отображение вёрстки в различных браузерах, которые по-разному отображают одни и те же данные CSS.

Для работы с CSS есть большое количество различных программных продуктов, но они не получили большого распространения. Как правило, объем CSS, кода по соотношению с HTML, не такой большой. И почти все HTML-редакторы имеют поддержку синтаксиса CSS. Из отдельных решений можно выделить TopStyle, данный программный продукт предназначен для создания стилей CSS. Очень полезным в нем является модуль Stylechecker, с помощью него можно не только проверить правильность синтаксиса таблицы стилей, но и проверить формат представления данных в зависимости от браузера. Таким образом, при его помощи можно создавать стили, которые будут одинаково отображаться в разных браузерах. Также существуют визуальные CSS-редакторы, но из них сложно выделить какой-то наиболее достойный.

***1.3.2 JavaScript***

Несмотря на то, что JavaScript не является стандартом W3C, мы рассмотрим его в данном контексте, и как часть DHTML.

JavaScript — это объектно-ориентированный скриптовый язык программирования. JavaScript встраиваемый язык, используемый для доступа к объектам в приложениях. Он нашел очень широкое применение при создании Web-страниц. JavaScript очень похож на язык Си, но все же имеет кардинальные отличия: Структуру объектов и тип объектов (например переменных, функций) можно определить в процессе выполнения кода. В JavaScript действует автоматическое приведение типов данных. Также автоматической является контроль памяти и защита от утечек. Функции в языке программирования JavaScript могут быть без имени. Все описания переменных, функции и других элементов регистрозависимы, названия переменных могут содержать буквы, символ подчёркивания, символ доллара и арабские цифры. При этом названия переменных не должны начинаться с цифры. Стоит отметить, что JavaScript не поддерживает области видимости, не умеет работать с файловой системой и потоками ввода/вывода, не имеет сетевых интерфейсов. Это сделано для обеспечения безопасности, так как скрипт на языке JavaScript легко запустить на любом компьютере, для этого будет достаточным открыть на нем Web-страницу, содержащую скрипт. Также для повышения безопасности скрипт не может получить доступ к свойствам другой страницы.

Использовать JavaScript на странице, можно при помощи тегов**<script></script>.** Например:

<**script type=«**text/javascript**»>**

**alert ('**Hello, World!**');**

**</script>**

Также скрипт может быть сохранен в отдельном файле.

<scripttype=«text/javascript»src=http://Путь\_к\_файлу\_со\_скриптом**></script>**

Язык JavaScriptдает создателю страницы очень широкие возможности для выполнения скриптов и как следствие качественно новые горизонты для создания динамических элементов на Web-страницах. JavaScript позволяет автоматически заполнять данные в формах, изменять форматирование и стиль страницы. Скрывать ненужное содержимое и отображать нужное содержимое, менять поведение клиентской части Web-приложений, изменять элементы управления на странице и так далее. При этом язык довольно простой для понимания, и большая часть функции выполняется автоматически.

Выделить какой-то редактор для JavaScript сложно, но отметим что в среде Eclipse, есть поддержка JavaScript при помощи подключаемого модуля, при этом становиться доступен весь функционал среды Eclipse. А также практически все современные браузеры имеют встроенные средства для отладки JavaScript-скриптов.

* 1. **Bootstrap**

Bootstrap — это открытый и бесплатный [HTML-](https://blog.skillfactory.ru/glossary/html/" \t "_blank), [CSS-](https://blog.skillfactory.ru/glossary/css/) и [JS-фреймворк](https://blog.skillfactory.ru/glossary/javascript/), который используют веб-разработчики для быстрой верстки адаптивных дизайнов сайтов и веб-приложений. Включает в себя CSS- и HTML-шаблоны оформления для веб-форм, меток, типографики, кнопок, блоков навигации и других компонентов веб-интерфейса.

Bootstrap используется, когда:

* у сайта много страниц;
* страницы собраны из простых базовых элементов — кнопок или таблиц;
* не будет глобального редизайна;
* шаблонность страниц окупается скоростью внедрения.

По сути, Bootstrap — это набор файлов. После их подключения к странице для верстки станет доступно большое количество готовых компонентов и классов. Они позволяют быстро и качественно создавать адаптивный дизайн сайта.

Классы в Bootstrap делятся на 3 большие группы:

* для создания сетки — адаптивного макета страницы;
* для стилизации контента — текста, изображений, кода, таблиц и прочей информации;
* служебные — для решения популярных вспомогательных задач, таких как отображение и скрытие элементов, выравнивание текста на странице, настройка цвета фона, отступов и пр.

Кроме классов, в Bootstrap есть компоненты (готовые объекты интерфейса). Это хлебные крошки (путь от начального элемента до уровня иерархии, который сейчас просматривает пользователь), кнопки, выпадающие списки и подсказки, модальные окна, формы, навигационные меню, всплывающие панели и пр.

Чтобы начать работать с Bootstrap, нужен текстовый редактор для работы с кодом (Visual Studio Code, Atom, Sublime Text и т.п.) или IDE — интегрированная среда разработки, а также браузер, в котором можно отслеживать изменения. Он должен быть обновлен до версии, которую поддерживает последний вариант Bootstrap.

Существует несколько способов установки Bootstrap. Получить актуальную версию фреймворка можно на [официальном сайте.](https://getbootstrap.com/)

1. Подключение скомпилированных файлов через [BootstrapCDN](https://www.bootstrapcdn.com/" \t "_blank) — общедоступную сеть доставки контента. Достаточно создать шаблон HTML и разместить внутри него ссылку на фреймворк.
2. Скачивание скомпилированных файлов CSS и JS с их подключением к проекту через ссылки.
3. Скачивание исходных файлов. Для работы потребуются [компилятор CSS](https://bootstrap-5.ru/docs/5.1/getting-started/contribute/#sass) и [автопрефиксатор](https://github.com/postcss/autoprefixer" \t "_blank).
4. Установка исходных файлов с помощью менеджеров пакетов yarn, npm, RubyGems, NuGet, Composer.

**2) Практическая часть**

**2.1 Описание предметной области**

Сайт CityTel представляет собой онлайн-платформу для предоставления услуг в сфере связи. Основной задачей сайта является обеспечение пользователей необходимой информацией о компании, услугах и контактных данных, а также возможность связи с компанией через форму обратной связи.

Основными разделами сайта являются:

* **Главная страница:** представление общей информации о компании, а также кнопки для навигации по основным разделам.
* **О нас:** подробная информация о компании, её истории и миссии.
* **Контакты:** форма обратной связи для связи с компанией, а также предоставление контактной информации.
* **FAQ:** раздел с часто задаваемыми вопросами, что позволяет пользователю быстро найти ответы на стандартные запросы.

Сайт должен быть удобным и функциональным, обеспечивая комфортный доступ как с мобильных устройств, так и с настольных ПК. Главными требованиями к веб-страницам являются:

1. **Юзабилити:** Структура и оформление сайта должны обеспечивать удобство навигации и использования.
2. **Адаптивность:** Сайт должен корректно отображаться на различных устройствах (мобильных телефонах, планшетах, ПК).
3. **Доступность:** Информация должна быть доступна пользователям с различными возможностями, включая поддержку экранных читалок и другие технологии.

В данной курсовой работе рассматривается создание и верстка сайта CityTel с учётом всех этих аспектов, а также использование современных стандартов HTML5, CSS и JavaScript для реализации функционала и дизайна.

**2.2 Создание окружения для проекта**

Перед началом разработки сайта CityTel было необходимо создать и настроить рабочее окружение. Этот процесс включает установку всех необходимых инструментов, программ и создание структуры проекта. Ниже описаны основные шаги и инструменты, использованные при подготовке среды разработки.

**Выбор редактора кода:**

Для верстки и написания скриптов использовался редактор Visual Studio Code (VS Code) благодаря его широким возможностям, поддержке расширений и удобному интерфейсу.

**Пояснение:** VS Code позволяет быстро создавать файлы, подключать плагины (например, Live Server для предпросмотра HTML-страниц), подсвечивает синтаксис и облегчает работу с HTML, CSS и JavaScript.

**Структура проекта:**

CityTel/

├── index.html

├── about.html

├── contact.html

├── style.css

└── script.js

**Пояснение:** Структура проекта включает три HTML-страницы: главную, страницу "О нас" и "Контакты". Внешний файл стилей style.css содержит все CSS-правила для сайта. В script.js размещён JavaScript-код для динамических элементов.

**Подключение CSS и JavaScript:**

Во всех HTML-документах CSS-файл подключается через тег link, а JavaScript — через script:

<link rel="stylesheet" href="style.css">

...

<script src="script.js"></script>

**Пояснение:** Эти строки подключают внешний файл стилей и скриптов, обеспечивая единый стиль и поведение на всех страницах сайта.

**Используемые технологии:**

* **HTML5** — для структурирования контента
* **CSS3** — для оформления внешнего вида страниц
* **JavaScript** — для реализации динамических функций (мобильное меню, аккордеон, валидация формы)

**Инструменты и расширения:**

* Расширение Live Server — для локального предпросмотра HTML-страниц
* Расширение Prettier — для автоматического форматирования кода
* Инструмент DevTools в браузере — для отладки и анализа элементов страницы

**Заключение:**

Создание и настройка среды разработки является важным этапом в работе над любым веб-проектом. В случае сайта CityTel была организована простая и эффективная структура, использованы современные инструменты и обеспечена возможность дальнейшего масштабирования и поддержки проекта.

**2.3 Верстка HTML страницы**

HTML-код сайта CityTel отвечает за структуру и содержимое всех страниц: главной, страницы "О нас", "Контакты" и других. Он реализован с использованием стандартной семантической разметки HTML5, что делает страницы понятными как для пользователей, так и для поисковых систем. Далее приведён подробный построчный разбор ключевых элементов с примерами кода и пояснениями.

**Пример структуры страницы:**

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>CityTel — Контакты</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<!-- Навигационное меню -->

</header>

<main>

<!-- Основной контент -->

</main>

<footer>

<!-- Подвал сайта -->

</footer>

</body>

</html>

**Пояснение:** Основная структура страницы, включающая заголовок head, тело body, и ключевые блоки: header, main, footer. Указан язык документа и подключен внешний файл стилей.

**Навигационное меню:**

<header>

<nav>

<ul>

<li><a href="index.html">Главная</a></li>

<li><a href="about.html">О нас</a></li>

<li><a href="contact.html">Контакты</a></li>

</ul>

<button class="menu-toggle">Меню</button>

<div class="mobile-menu">

<a href="index.html">Главная</a>

<a href="about.html">О нас</a>

<a href="contact.html">Контакты</a>

</div>

</nav>

</header>

**Пояснение:** Меню состоит из списка ссылок, кнопки открытия мобильного меню и блока с мобильными ссылками. Это обеспечивает адаптивность интерфейса.

**Основной контент страницы "Контакты":**

<main>

<section>

<h1>Свяжитесь с нами</h1>

<form id="contact-form">

<input type="text" id="name" name="name" placeholder="Ваше имя" required>

<input type="email" id="email" name="email" placeholder="Ваш email" required>

<textarea id="message" name="message" placeholder="Ваше сообщение" required></textarea>

<button type="submit">Отправить</button>

</form>

</section>

<section>

<h2>Часто задаваемые вопросы</h2>

<button class="accordion">Как с вами связаться?</button>

<div class="panel">

<p>Вы можете использовать форму выше или написать нам напрямую на email.</p>

</div>

</section>

</main>

**Пояснение:** Контент содержит два раздела: форма обратной связи и блок FAQ с аккордеоном. Используются семантические теги section, h1, form, input, textarea, button.

**Подвал сайта:**

<footer>

<p>&copy; 2025 CityTel. Все права защищены.</p>

</footer>

**Пояснение:** Простая структура подвала — один абзац с копирайтом. Используется тег footer, что соответствует семантике HTML5.

**Заключение:**

HTML-верстка сайта CityTel построена по современным стандартам. Использование семантической разметки, разбивка на логические блоки и адаптивные элементы обеспечивают удобство восприятия, поддержку мобильных устройств и простоту масштабирования проекта. Такая структура делает сайт понятным для пользователя и легко поддерживаемым в будущем.

**2.4 Использование стилей CSS**

CSS-файл сайта CityTel определяет внешний вид всех страниц: главной, страницы "О нас", "Контакты" и других. Он содержит как общие стили, так и стили, специфичные для различных компонентов пользовательского интерфейса. Далее приводится подробный построчный анализ с пояснениями и вставками соответствующих фрагментов кода.

**Общие стили:**

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

**Пояснение:** Универсальный селектор \* применяется ко всем элементам на странице. Он сбрасывает стандартные отступы и задаёт модель расчета размеров через border-box, при которой в размеры элемента входят отступы и рамка.

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

background-color: #f4f4f4;

color: #333;

}

**Пояснение:** Оформление основного тела сайта. Устанавливается шрифт, светлый фон и тёмный цвет текста для читабельности.

a {

color: inherit;

text-decoration: none;

}

**Пояснение:** Ссылки наследуют цвет текста родителя и лишены подчеркивания для стилистической согласованности.

**Стили для шапки сайта (header):**

header {

background-color: #333;

color: #fff;

padding: 20px 0;

text-align: center;

}

**Пояснение:** Шапка имеет тёмный фон и белый текст, выравнивание текста по центру и вертикальные отступы.

nav ul {

list-style-type: none;

display: flex;

justify-content: center;

gap: 20px;

}

**Пояснение:** Меню навигации — это flex-контейнер, элементы которого располагаются по центру с промежутками между ними. Убираются маркеры у списка.

nav ul li {

display: inline;

}

**Пояснение:** Элементы списка располагаются в одну строку.

nav ul li a {

color: white;

font-weight: bold;

transition: color 0.3s;

}

**Пояснение:** Ссылки в меню — белые, жирные, с плавным переходом цвета при наведении.

nav ul li a:hover {

color: #ffd700;

}

**Пояснение:** При наведении на ссылку цвет меняется на золотистый (#ffd700).

**Стили для главного контента:**

main {

padding: 20px;

}

**Пояснение:** Основному блоку сайта задается внутренний отступ.

h1, h2, h3 {

margin-bottom: 15px;

}

**Пояснение:** Заголовки отделяются друг от друга и от основного текста с помощью нижнего отступа.

section {

margin-bottom: 40px;

background-color: white;

padding: 20px;

border-radius: 10px;

box-shadow: 0 2px 5px rgba(0,0,0,0.1);

}

**Пояснение:** Каждый раздел (section) имеет отступ снизу, белый фон, внутренний отступ, скругленные углы и лёгкую тень для приподнятого эффекта.

**Стили для формы на странице "Контакты":**

form {

display: flex;

flex-direction: column;

gap: 15px;

max-width: 500px;

margin: auto;

}

**Пояснение:** Форма организована вертикально, с промежутками между элементами. Центрируется и ограничивается по ширине.

input, textarea {

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

font-size: 16px;

}

**Пояснение:** Поля ввода и текстовые области оформлены с отступами, границами и скругленными углами.

button {

background-color: #333;

color: white;

padding: 10px;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s;

}

**Пояснение:** Кнопка отправки формы имеет тёмный фон, белый текст и плавное изменение фона при наведении.

button:hover {

background-color: #555;

}

**Пояснение:** Цвет фона кнопки становится светлее при наведении для визуальной обратной связи.

**Стили для аккордеона (FAQ):**

.accordion {

cursor: pointer;

padding: 15px;

width: 100%;

text-align: left;

border: none;

outline: none;

transition: background-color 0.3s;

background-color: #eee;

font-weight: bold;

}

**Пояснение:** Элемент аккордеона оформлен как кнопка. Устанавливается фон, убирается обводка, задаются отступы и курсор указателя.

.accordion:hover {

background-color: #ddd;

}

**Пояснение:** При наведении фон становится темнее.

.panel {

padding: 0 15px;

display: none;

background-color: white;

overflow: hidden;

}

**Пояснение:** Контент панели изначально скрыт, открывается при клике. Внутренние отступы и белый фон обеспечивают читаемость.

.panel.show {

display: block;

}

**Пояснение:** Класс show делает панель видимой при раскрытии.

**Стили для футера:**

footer {

background-color: #333;

color: white;

text-align: center;

padding: 20px;

margin-top: 40px;

}

**Пояснение:** Нижняя часть сайта имеет тёмный фон, белый текст, выравнивание по центру и вертикальные отступы.

**Стили для мобильного меню:**

.mobile-menu {

display: none;

flex-direction: column;

background-color: #333;

padding: 10px;

}

**Пояснение:** Мобильное меню скрыто по умолчанию. При отображении оно становится вертикальным блоком с тёмным фоном.

.mobile-menu a {

color: white;

padding: 10px;

border-bottom: 1px solid #555;

}

**Пояснение:** Ссылки в мобильном меню — белые, отделены друг от друга нижней границей.

.menu-toggle {

display: none;

background-color: #333;

color: white;

padding: 10px;

cursor: pointer;

border: none;

font-size: 18px;

}

**Пояснение:** Кнопка открытия мобильного меню оформлена в стилистике сайта, но по умолчанию скрыта.

@media (max-width: 768px) {

nav ul {

display: none;

}

.menu-toggle {

display: block;

}

.mobile-menu {

display: flex;

}

}

**Пояснение:** Медиазапрос изменяет поведение меню на экранах шириной до 768px: обычное меню скрывается, появляется кнопка и мобильное меню.

**Заключение:**

CSS-оформление сайта CityTel демонстрирует аккуратный, современный и отзывчивый интерфейс, построенный на основе flex-разметки, плавных анимаций и адаптивной верстки. Использование единых цветовых решений и стилистической гармонии делает интерфейс удобным и визуально приятным.

Таким образом, стили CSS обеспечивают не только эстетическую привлекательность, но и функциональность интерфейса сайта, делая его доступным на различных устройствах и понятным для пользователей.

**3.РЕЗУЛЬТАТ**

За время выполнения курсовой работы были изучены принципы разработки web-сайта:

1. Мы создали веб-страницу используя HTML.
2. Стили выполнены с использованием CSS
3. Создана адаптивная веб-страница с использованием медиа-запросов.

В результате была создана современная, функциональная и эстетичная веб-страница, которая удовлетворяет требованиям заказчиков, удобна для пользователей и соответствует современным трендам в веб-дизайне.

Ссылка на веб-страницу: <https://kursovaya-for-leyla.vercel.app/>

**4****. ЛИТЕРАТУРА**

Книги:

1. «*HTML и CSS. Разработка и дизайн веб-сайтов*» — Джон Дакетт  
   *(Лучшая книга для начинающих, много визуальных примеров)*
2. «*JavaScript и jQuery. Интерактивная веб-разработка*» — Джон Дакетт  
   *(Продолжение первой книги, посвящено JS и jQuery)*
3. «*Вы не знаете JS*» (серия книг) — Кайл Симпсон  
   *(Глубокое погружение в JavaScript, бесплатно на GitHub)*
4. «*CSS. Карманный справочник*» — Эрик Мейер  
   *(Удобный справочник по свойствам CSS)*
5. «*JavaScript. Подробное руководство*» — Дэвид Флэнаган  
   *(Полное руководство по JS, подходит для продвинутых)*
6. **ПРИЛОЖЕНИЯ**

Репозиторий:

<https://github.com/shalbuzz/Kursovaya_for_Leyla.git>

Страница сайта на хостинге GitHub Pages:

<https://kursovaya-for-leyla.vercel.app/>