ГБПОУ Республики Марий Эл «МРМТ»

Допущен к защите

Зам. директора по УМР

\_\_\_\_\_\_\_\_\_И.Ю. Бурханова

«\_\_\_»\_\_\_\_\_\_\_\_\_ 2021 г.

КУРСОВАЯ РАБОТА

РАЗРАБОТКА САЙТА ДЕТЕЙЛИНГ-ЦЕНТРА

Работу выполнил

Студент группы П-32

\_\_\_\_\_\_\_\_\_Софронов М. И.

Руководитель

\_\_\_\_\_\_\_\_\_Эйбин С.А.

Йошкар-Ола 2021

СОДЕРЖАНИЕ

￼

￼

￼

￼

￼

￼

￼

￼

￼

￼

￼

￼

￼

￼

￼

￼

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящей выпускной квалификационной работе применяют следующие сокращения и обозначения:

CSS - Cascading Style Sheets

CMS - Content management system

HTML - HyperText Markup Language

JS - JavaScript

VS Code - Visual Studio Code

WAAPI - Web Animations API

# ВВЕДЕНИЕ

Интернет стал, наиболее эффективным средством рекламы и продвижения и является одним из важных элементов современной цивилизации. Интернет может удовлетворить все потребности современного человека: это покупки, заключение деловых отношений, поиск клиентов и так далее. Многие бизнес-организации получают львиную долю своих клиентов благодаря рекламе в интернете и в первую очередь заботятся о создании сайта своей организации. Поэтому создание сайтов стало очень популярным в наше время.

С помощью сайтов-визиток можно значительно увеличить число заказчиков, проследить заинтересованность людей за определенный период времени данными услугами и это как никогда актуально.

Процесс разработки сайта с нуля без использования конструкторов и систем управления содержимым (CMS), создание клиентской и серверной частей веб сайта с использованием HTML5, CSS и JavaScript достаточно трудоемкий.

Цель курсовой работы заключается в верстке сайта-визитки «DETAIL52» без использования различных CMS и иных конструкторов, т. е. разработка кода своими руками.

В связи с этим были поставлены следующие задачи:

1. Рассмотреть способы верстки сайта от самого простого, до самого технологичного.
2. Проанализировать существующие языки для написания сайта.
3. Проанализировать способы анимирования элементов сайта.
4. Произвести установку необходимых инструментов для разработки.
5. Сверстать страницу.
6. Рассмотреть различные анимации и исходя из своих предпочтений обеспечить подвижность некоторых элементов.
7. Добавить скрипт кнопки, которая ведет на верх страницы.
8. Реализовать форму заказа звонка.
9. Загрузить готовый проект в репозиторий github.

Практическая значимостьпроекта заключается в том, чтос разрабатываемый сайт будет размещен в сети интернет и будет приносить реальную пользу заказчику, а именно ИП Софронову М. И., который и заказал разработку этого сайта.

# ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

## 1.1 Способы разработки сайта

Существует множество способов и подходов при разработке сайта. Все они кардинально или не очень отличаются друг от друга. Разберем несколько способов, которые мне известны и имеют значительные отличия в процессе разработки. Способы проранжированны от самого простого, но менее гибкого, к сложному, но максимально гибкому способу.

### 1.1.1 С помощью CMS типа Wordpress

Положительными моментами использования такого способа создания сайта являются то, что:

* к нему можно бесплатно скачать движок и шаблоны;
* WordPress имеет весьма обширный ассортимент шаблонов;
* быстро устанавливается;
* имеет понятную и простую административную панель;
* можно скачать дополнения и расширения (плагины);
* доступна возможность изменять код шаблона.

Главное преимущество WordPress, завоевавшее ему популярность - это его бесплатность.

WordPress удобен как для начинающего, так и для профессионала. Любому пользователю, пожелавшему запустить свой сайт или блог, всегда можно воспользоваться WordPress с большим количеством готовых шаблонов. Контент на сайт можно добавить при помощи визуального редактора, с которым справится любой работавший в MS Word. Администраторская панель интуитивно понятна, поэтому ее можно освоить за небольшой промежуток времени.

Для расширения возможностей шаблона можно установить плагины. Многие из них, подобно и самим шаблонам WordPress, можно скачать бесплатно. При желании отредактировать структуру сайта, отображение или стиль, можно изменив код в файлах шаблона.

А главное, что WordPress, благодаря плодотворной работе программистов и доступной установке официальных обновлений, в достаточной мере защищен от заражений, а появляющиеся прорехи - оперативно исправляются.

Конечно, имеются и минусы:

* довольно сложная структура проекта, необходимо разобраться, где что находится и как работает cms, чтобы делать значительные изменения шаблонов;
* ощутимая на сервер нагрузка;
* иногда низкая скорость для загрузки веб-ресурса;
* небольшой набор базовых возможностей;
* дублирование изображений и страниц;
* огромное количество некачественно сделанных тем и расширений с ошибками;
* предвзятое отношение поисковых систем к сайту на бесплатном движке.

CMS WordPress, а именно масса дополнений, устанавливаемых к нему, существенно нагружают сервер, где размещен сайт. Чтоб избежать такой нагрузки приходится сокращать число дополнительных плагинов до самого возможного минимума, а главное, размещать сайт на качественном хостинге. Эти небольшие тонкости будут способствовать увеличению скорости загрузки сайта.

Еще один недостаток WordPress - автоматическая генерация дублей страниц, что отрицательно влияет на продвижение сайта. Также данный движок при загрузке изображений делает их копии, при этом в разных размерах.

Всеобщее использование WordPress, его достоинство и в то же время - недостаток, так как на его основе попутно с CLK создают и массу ГС или сателлитов. Поэтому поисковые системы настороженно относятся к новым сайтам на бесплатных CMS.

Изначально шаблон имеет ограниченный набор функций, но их можно расширить плагинами. Но помните, чрезмерное их количество может повлиять на скорость работы веб-ресурса.

Во избежание такой ситуации, дополнительно можно редактировать сам код. Однако понадобятся определенные навыки.

Вывод: создание сайта с помощью CMS требует некоторых навыков работы с конкретной CMS, либо на освоение может уйти дополнительное время. На создание собственной темы с уникальным дизайном может уйти немалое время, т.к. существуют определенные структуры, которым должна соответствовать тема, иначе она может просто не заработать или не все функции CMS будут в ней доступны.

### 1.1.2 Разработка собственного кода без использования конструкторов и визуальных редакторов.

Способ для тех, кто хочет, чтобы все работало «по их правилам». Полностью с нуля разрабатывается движок сайта, верстаются страницы и т.д. и т.п. В общем, здесь разработчик практически ничем не ограничен и выбирает инструменты разработки под свои задачи или согласно своим навыкам.

Разрабатывая интернет-магазин или социальную сеть с множеством постоянно изменяющихся данных в режиме реального времени, обычно используют React.js или Vue.js, т.к. эти технологии специально разрабатывались для выполнения именно такой задачи - удобной работы с множеством изменяющихся данных и взаимодействии между ними.

Разрабатывая простой форум или блог можно использовать PHP. Это позволит хранить в базе, добавлять и выводить на страницах данные, а также создать систему учета пользователей.

Если задачей является разработать простой сайт с минимумом функционала, основной целью которого будет предоставление информации, можно обойтись HTML, CSS и JS. Ведь все, что требуется от сайта - показать страницу пользователю и ничего более. Такой вариант может иметь место, если на сайт не планируется внедрение нового, более сложного функционала. Но иметь статичный сайт не есть круто, информация на сайте должна регулярно изменяться, добавляется новая, изменяется старая, это полезно для продвижения сайта в поисковых системах и для того, чтобы конечный пользователь сайта чувствовал, что сайт «живой». При простых статичных сверстанных страницах будет не так просто добавлять новую информацию, придется каждый раз редактировать код, а если эта информация отображается не на одной странице, то необходимо будет редактировать каждую — это значительно усложняет поддержку веб-сайта.

Был выбран второй вариант - написать весь код самостоятельно. От сайта не требуется серьезный функционал, однако некоторые функции будут присутствовать в проекте. В будущем будет проще разобраться в структуре проекта, когда будет создаваться полностью функциональный сайт.

## 1.2 Инструменты разработки

### 1.2.1 Основной минимум инструментов

HTML (от англ. HyperText Markup Language - «язык гипертекстовой разметки») - стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). То, без чего не обойтись при создании веб-сайта. С помощью данного языка мы будем описывать все элементы сайта браузеру.

CSS (от англ. Cascading Style Sheets - каскадные таблицы стилей) - формальный язык описания внешнего вида документа, написанного с использованием языка разметки HTML. После описания элементов, необходимо кастомизировать их внешний вид.

JavaScript (аббр. JS) - мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией стандарта ECMAScript (стандарт ECMA-262).

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. Большинство браузеров (напри. Google Chrome) работают на JS и имеют собственный интерпретатор для работы с данным языком программирования. В браузере Google Chrome если множество методов языка JS, которые позволяют работать непосредственно с самим браузером прямо из консоли.

Все языки программирования из данного списка по дефолту поддерживаются большинством современных веб-браузеров.

### 1.2.2 Добавляемые на сайт расширения (библиотеки, фреймворки)

Фреймворки — это программные продукты, которые упрощают создание и поддержку технически сложных или нагруженных проектов. Фреймворк, как правило, содержит только базовые программные модули, а все специфичные для проекта компоненты реализуются разработчиком на их основе. Тем самым достигается не только высокая скорость разработки, но и большая производительность и надёжность решений.

Веб-фреймворк — это платформа для создания сайтов и веб-приложений, облегчающее разработку и объединение разных компонентов большого программного проекта. За счёт широких возможностей в реализациии бизнес-логики и высокой производительности эта платформа особенно хорошо подходит для создания сложных сайтов, бизнес-приложений и веб-сервисов.

Библиотека — это более простой компонент архитектуры программного обеспечения. Программная библиотека может быть использована просто как набор подсистем близкой функциональности, не влияя на архитектуру основного программного продукта и не накладывая на неё никаких ограничений.

Исходя из собственных соображений об написании кода только своими руками, было принято решение не использовать никаких расширений на сайте.

## 1.3 Разработка макета веб-сайта

Adobe Photoshop - многофункциональный [графический редактор](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80), разрабатываемый и распространяемый компанией [Adobe Systems](https://ru.wikipedia.org/wiki/Adobe). В основном [работает с растровыми изображениями](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80), однако имеет некоторые [векторные](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0) инструменты. Продукт является лидером рынка в области коммерческих средств редактирования растровых изображений и наиболее известной программой разработчика.

Figma — это графический онлайн-редактор для дизайнеров интерфейсов и веб-разработчиков. Сейчас это удобная, бесплатная альтернатива Photoshop. Большое преимущество платформы - возможность работать прямо в браузере. При этом есть и десктопная версия, которая также требует доступ к интернету.

PlatformaLP - профильный конструктор одностраничников. Система обладает качественным визуальным редактором и хорошим набором готовых шаблонов. Средства анализа и увеличения конверсии прилагаются.

Не смотря на простоту работы с конструкторами и иными средствами создания макетов, было принято решение не использовать их и не создавать макет, а попробовать импровизировать.

## 1.4 Выбор способа анимирования объектов

Анимация - один из основных трендов в современном сайтостроении. Элементы анимации в интерфейсах выносят их юзабилити на новый уровень, делая их интуитивно понятными и доступными обычному пользователю. На лендингах или, например, корпоративных сайтах она помогает более наглядно рассказать посетителю о вашем продукте или показать его преимущества.

Существует множество способов для анимации элементов на сайте. В основном элементы приводятся в движение с помощью CSS и JS.

CSS позволяет создавать простые анимации без использования JS.

JavaScript может быть использован для управления такими CSS-анимациями. Это позволяет делать более сложные анимации, используя небольшие кусочки кода.

JS имеет свои библиотеки для более эффективной работы анимации.

Рассмотрим некоторые из них.

[GreenSock Animation Platform](http://greensock.com/gsap) или [GSAP](http://greensock.com/gsap) — это, пожалуй, одна из самых функциональных JavaScript-библиотек для анимирования. Как говорят её создатели, это «новый стандарт для HTML5 анимации». Она имеет низкий порог вхождения, подробную документацию, высокую производительность, гибкость и совместимость со старыми браузерами. Важным преимуществом GSAP является также то, что библиотека умеет «анимировать» не только CSS или Canvas, но и **любую** числовую переменную **любого** JavaScript-объекта.

[Tween.js](http://www.createjs.com/tweenjs) является дополнением к easel.js - одному из самых популярных javascript-фреймворков для удобной и легкой работы с canvas. Тем не менее, tween не имеет никаких зависимостей и может использоваться как вместе, так и отдельно от easel. Основная задача библиотеки tween.js — это анимация в Canvas. И она отлично с ней справляется, делая за вас и упрощая многие вещи.

Move.js - javascript-библиотека, позволяющая делать всё то же, что и tween.js, только с использованием CSS3 и HTML5 вместо Canvas. Она является не самой легкой (как для JS), но одной из самых популярных. Набор заранее заготовленных видов анимации (поворот, наклон, ротация и тд.) делает её также одной из самых легких в использовании.

jQuery- [набор функций JavaScript](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_JavaScript), фокусирующийся на взаимодействии [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) и [HTML](https://ru.wikipedia.org/wiki/HTML). Библиотека jQuery помогает легко получать доступ к любому элементу [DOM](https://ru.wikipedia.org/wiki/Document_Object_Model), обращаться к атрибутам и содержимому элементов [DOM](https://ru.wikipedia.org/wiki/Document_Object_Model), манипулировать ими. Также библиотека jQuery предоставляет удобный [API](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) для работы с [AJAX](https://ru.wikipedia.org/wiki/AJAX). Разработка jQuery ведётся командой добровольцев на пожертвования. По статистике, почти 50% сайтов используют JQuery.

Но у всего есть свои минусы:

* у CSS есть аппаратное ускорение для плавных переходов со встроенной поддержкой в браузере, но правила объявляются в CSS и заставляет хитро изворачиваться в JavaScript, чтобы изменять значения динамически;
* jQuery познакомил с анимацией ряд других разработчиков, но он известен частыми проблемами с производительностью;
* [GreenSock](http://greensock.com/), [Tween.js](http://www.createjs.com/tweenjs) и Move.js улучшают производительность JavaScript, и отточены на множестве тестовых примеров практически до совершенства. Однако, они по-прежнему требуют поддержки и загрузки внешних библиотек.

Сейчас существует Web Animations API. Его цель - соединить мощь производительности CSS с плюсами и гибкостью JavaScript, и заставить сами браузеры следить за тем, чтобы это хорошо работало. Но поведение данной спецификации может измениться в будующих версиях, так как это **экспериментальная технология.**

WAAPI не планируется на замену существующим поведениям, а вместо этого объединяет различные способы и даже позволяет им взаимодействовать. Синтаксис похож на CSS, но добавляет переменные, элементы управления и вызов функций при завершении.

Исходя из вышесказанного было принято решение попробовать на практике данную спецификацию и применить ее в проекте.

# 2. Практический раздел

# 

## **2.1** **настройка программного** **обеспечения**

Visual Studio Code - [редактор исходного кода](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0), разработанный [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) для [Windows](https://ru.wikipedia.org/wiki/Windows), [Linux](https://ru.wikipedia.org/wiki/Linux) и [macOS](https://ru.wikipedia.org/wiki/MacOS). Позиционируется как «лёгкий» редактор кода для [кроссплатформенной](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C) разработки [веб](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5)- и [облачных](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BB%D0%B0%D1%87%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F) приложений. Включает в себя [отладчик](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BB%D0%B0%D0%B4%D1%87%D0%B8%D0%BA), инструменты для работы с [Git](https://ru.wikipedia.org/wiki/Git), [подсветку синтаксиса](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D1%81%D0%B2%D0%B5%D1%82%D0%BA%D0%B0_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81%D0%B0), [IntelliSense](https://ru.wikipedia.org/wiki/IntelliSense) и средства для [рефакторинга](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3). Имеет широкие возможности для [кастомизации](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%81%D1%82%D0%BE%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F): пользовательские темы, [сочетания клавиш](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%87%D0%B5%D1%82%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BA%D0%BB%D0%B0%D0%B2%D0%B8%D1%88) и [файлы конфигурации](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB_%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D0%B8). Распространяется бесплатно, разрабатывается как программное обеспечение с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), но готовые сборки распространяются под [проприетарной лицензией](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BF%D1%80%D0%B8%D0%B5%D1%82%D0%B0%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)

Установка и настройка VS Code:

1. Загружаем последнюю версию VS Code.
2. Запускаем инсталлятор и следуем инструкциям: принимаем, добавляем все галочки, далее, ждем пока программа установится на наш ПК.
3. Запускаем приложение.
4. Заходим в настройки и задаем параметры табуляции.
5. Заходим во вкладку «Расширения» и устанавливаем необходимые нам дополнения для более удобной работы (Live Server, Prettier, Quokka.js и т.д.).
6. Создаем новый файл или новую папку (куда добавляем новые файлы с необходимым расширением и начинаем работу).

## 2.2 Разработка структуры проекта

Многостраничные сайты обычно разделяют на несколько страниц, также на отдельных страницах располагаются одинаковые части страниц, такие как header (шапка сайта) и footer (подвал сайта). В данном проекте такой вариант не используется так как сайт-визитка является одностраничным.

Проект состоит из каталога «css», где хранятся файлы стилейю.

Каталог «img» содержит изображения и иконки.

В каталоге «js» находятся скрипты.

Каталог «fonts» содержит в себе используемые в проекте шрифты.

## 2.3 Верстка страницы

Так как макета не имеется дизайн сайта, будет браться из головы на основе других просмотренных сайтов.

Некоторые дизайнеры разрабатывают анимации которые хотят увидеть в проекте и только потом подстраивают под них контент проекта. В данном проекте будет верстаться сначала контент, а только потом уже анимация элементов.

Не маловажным для дизайна сайта являются цвета. Было принято решение использовать популярную и самую простую комбинацию - белый, красный и черный.

Создается файл index.html и начинается верстка.

Верстаем шапку сайта, примерный внешний вид продемонстрирован на рисунке 1.

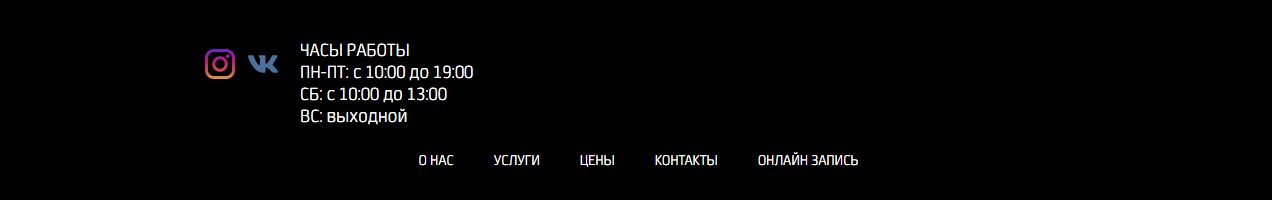


Рисунок 1 - Шапка сайта без логотипа

Для любого сайта нужен логотип. В данном проекте логотип будет достаточно простой: текст «DETAIL52» на который наложен шрифт «CARboned». Итог можно увидеть на рисунке 2.

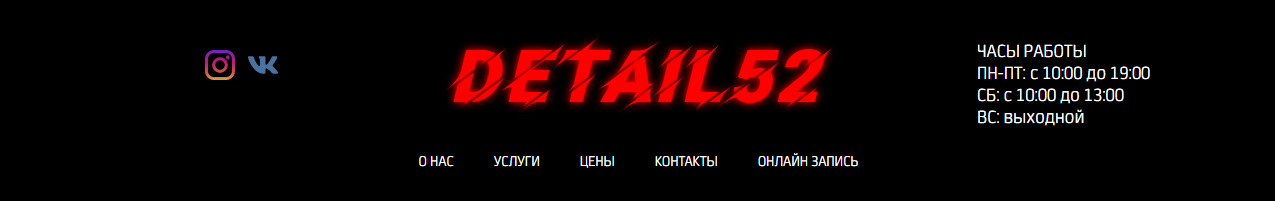


Рисунок 2 - Шапка сайта с логотипом

Далее располагается один из самых главных элементов страницы - паралакс.

HTML код у данного элемента не большой, как показано на рисунке 3. CSS так же не содержит в себе ничего сложного, что представлено на рисунке 4, однако эффект от данного элемента весьма впечатляющий.



Рисунок 3 - HTML код элемента паралакс

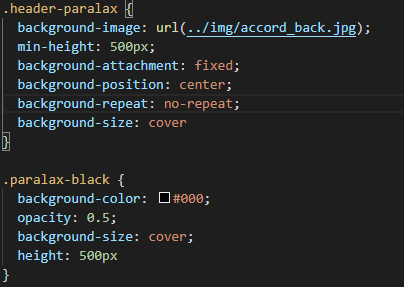


Рисунок 4 - CSS код элемента паралакс

После паралакса располагается блок с вводным текстом.

Далее расположен блок с описанием услуг сервиса, которые продемонстрированы на рисунке 5, для элементов которого есть эффект затемнения при наведении, рисунок 6 и рисунок 7.

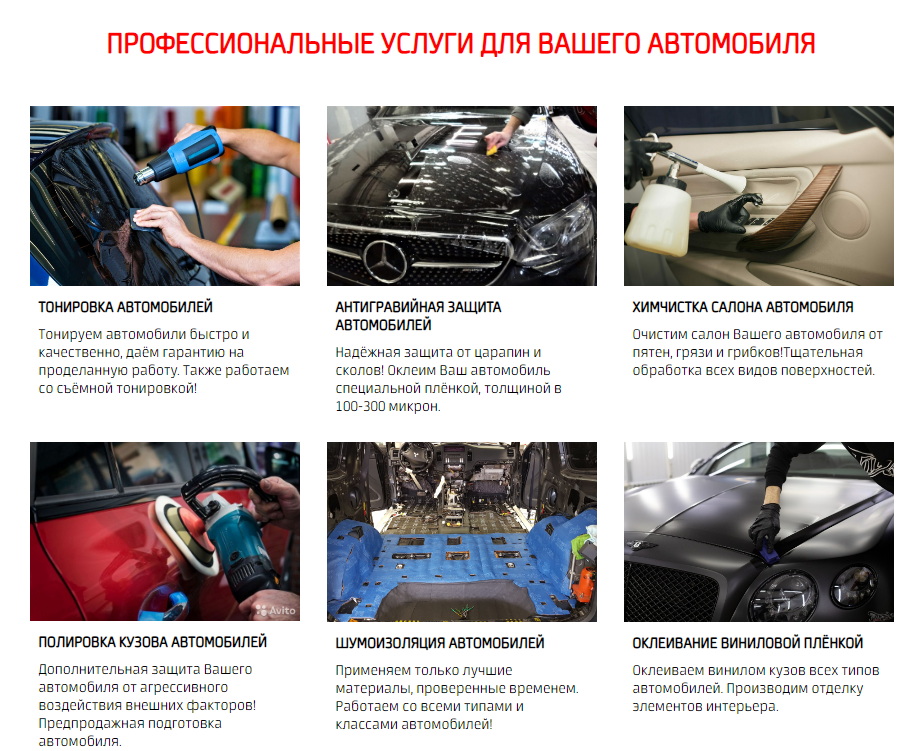


Рисунок 5 - вид блока с услугами сервиса

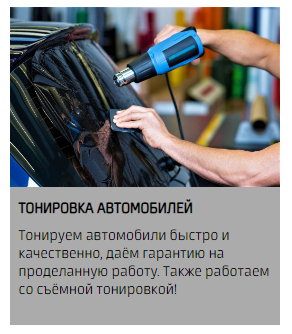


Рисунок 6 - Вид затемненного элемента



Рисунок 7 - CSS код эффекта затемнения

Блок с ценами на услуги был выполнен с помошью тега <details>, где содержится наименование скрытого блока, и <summary>, где лежит сам контент, рассмотренный на рисунке 8 и рисунке 9.

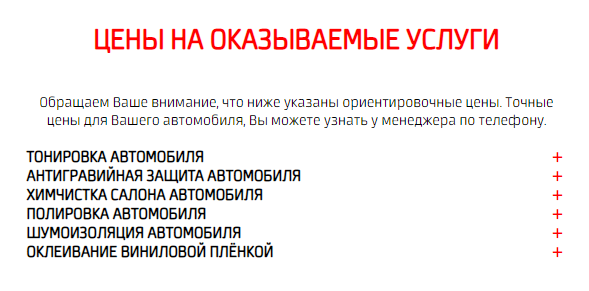


Рисунок 8 - вид блока с ценами при закрытых элементах

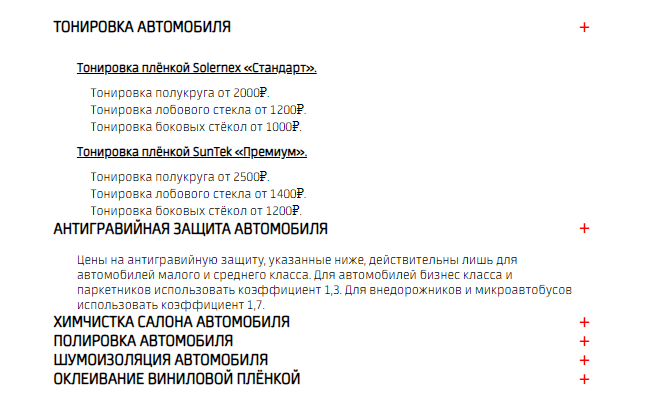


Рисунок 9 - Пример раскрытых блоков

После блока с ценами располагаются контакты, в которые было принято решение вставить скрипт с рабочей картой Яндекса, показанный на рисунке 10. Код был взят из специального конструктора карт от самой компании Яндекс, рисунок 11.

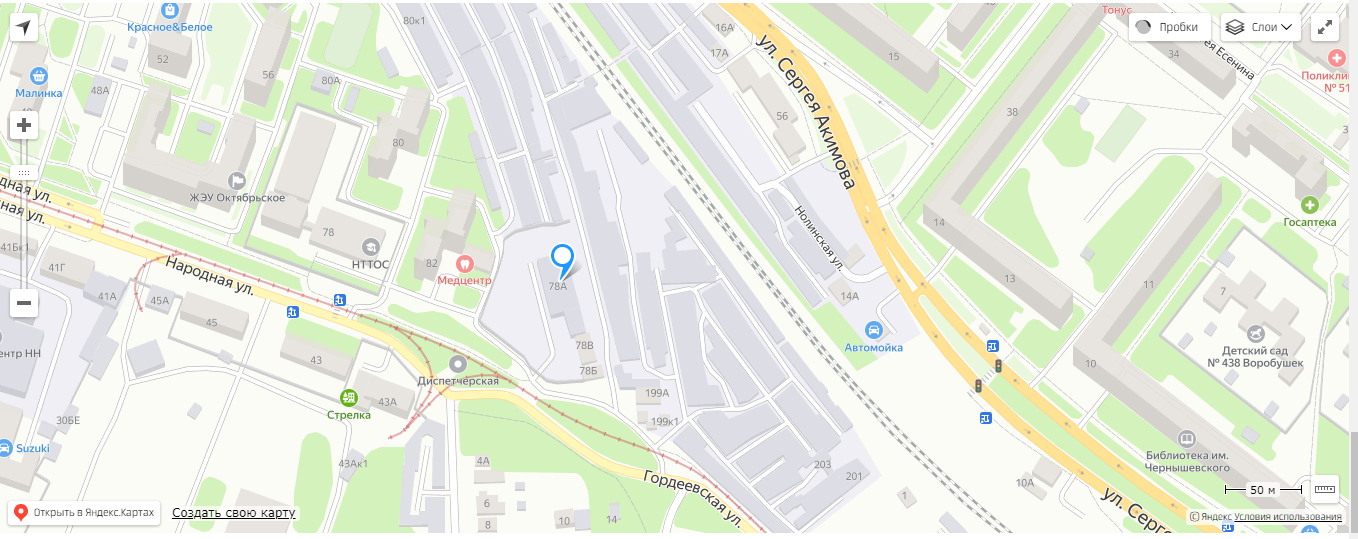


Рисунок 10 - Вид карты

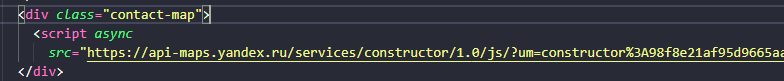


Рисунок 11 - Пример кода карты

Итак, очередь подошла к блоку для записи онлайн. Пишется заголовок, немного пояснительного текста и самое главное - кнопка, которая будет вызывать модальное окно с формой, как на рисунке 12).

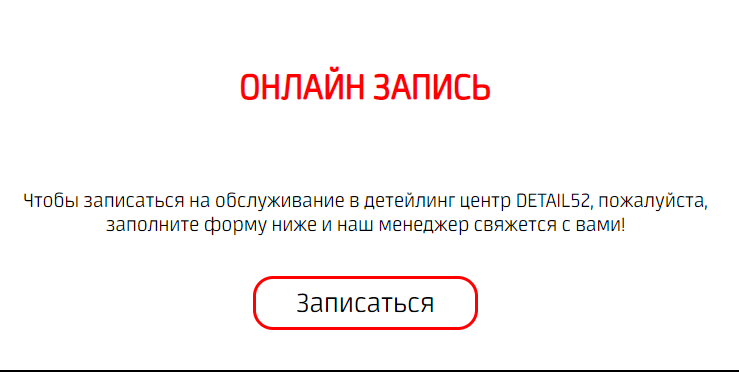


Рисунок 12 - Вид блока для записи онлайн

Теперь необходимо сверстать форму, которая должна будет отправлять данные менеджеру, который в течение некоторого времени обязан осуществить обратную связь с клиентом и записать его на конкретную дату. Форма достаточно простая и требует многих доработок, представлена на рисунке 13 и рисунке 14.



Рисунок 13 - Пример кода модального окна с формой

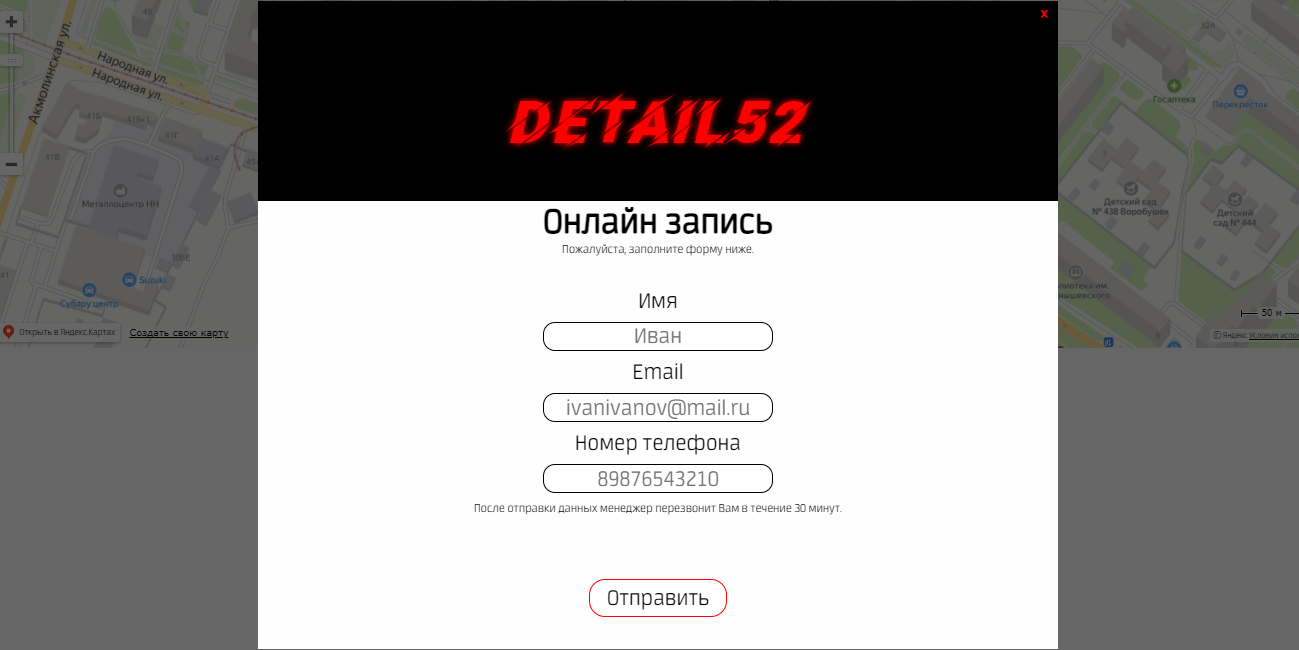


Рисунок 14 - Вид модального окна с формой

Теперь необходимо написать скрипт, который будет осуществлять отображение модального окна при клике на кнопку. Тут все достаточно просто. Окно появляется при клике, закрывается - при клике на специальный крестик или по пустым полям вокруг формы. Пример кода отображен на рисунке 15.

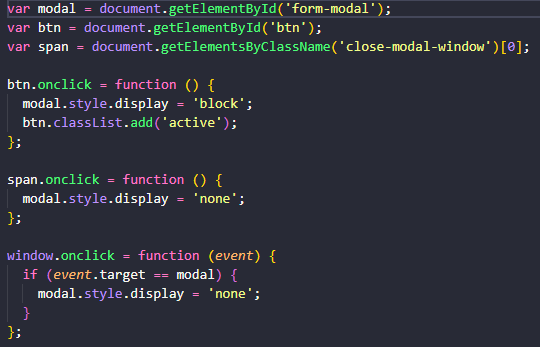


Рисунок 15

Завершаем верстку подвалом, в котором расположится навигационное меню, ведущее к услугам, как показано на рисунке 16.

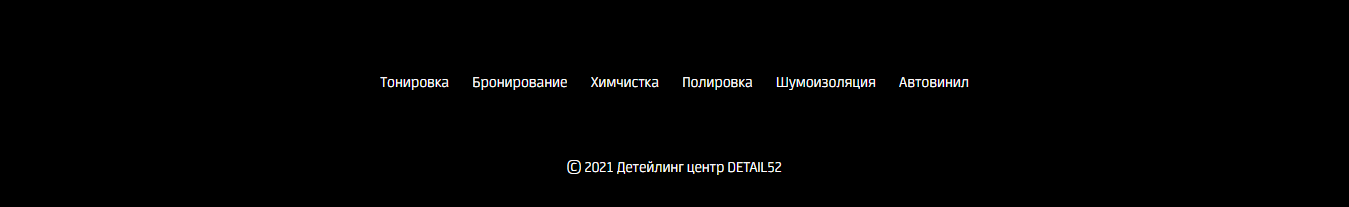


Рисунок 16 - Вид подвала сайта

Далее необходимо обеспечить подвижность элементов на сайте. Было принято решение анимировать элементы блока с ценами, а именно их раскрытие и закрытие. Исходя из сказанного в теоретическом разделе будет использоваться Web Animations API.

Для обеспечения возможности повторного использования кода потребуется класс «Accordion» на рисунке 17. Имея такой класс, открывается возможность создавать экземпляры для любого количества details на странице.

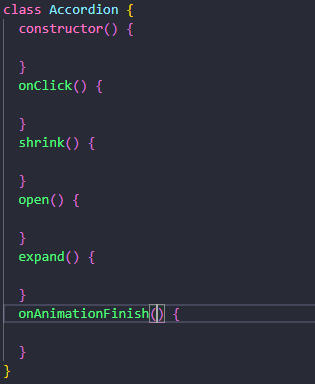


Рисунок 17

Constructor() служит для хранения необходимых аккордеону данных , как на рисунке 18.

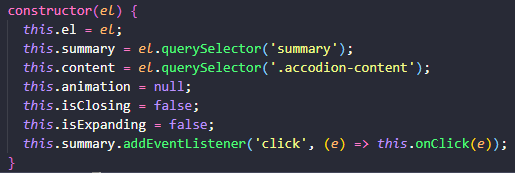


Рисунок 18 - Код конструктора

В функции «onClick» проверяется, находится ли элемент в процессе анимирования. Необходимо это сделать для случая, когда пользователь кликает по аккордеону до завершения анимации. Нельзя, чтобы аккордеон «прыгал» от полностью открытого до полностью закрытого.  
Элемент «details» имеет атрибут «open», добавляемый браузером при открытии элемента. Значение этого атрибута получается посредством this.el.open на рисунке 19.

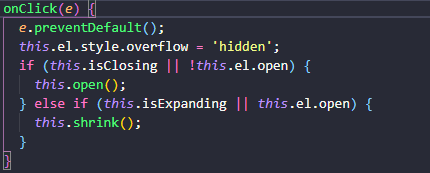


Рисунок 19

Функция «shrink» использует функцию «animate» WAAPI. WAAPI очень похож на инструкцию «keyframes» CSS в том, что необходимо определить ключевые кадры анимации. В данном случае требуется лишь два таких кадра: первый - текущая высота элемента «details» (в открытом состоянии), второй - высота закрытого details (высота summary). Пример кода показан на рисунке 20.

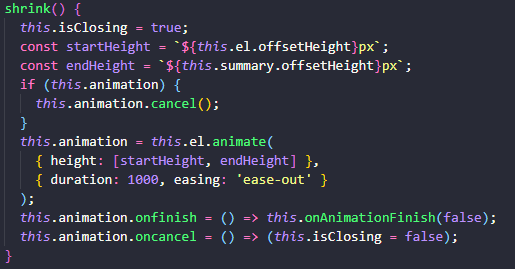


Рисунок 20 - Пример кода

Функция «open» вызывается, когда необходимо раскрыть аккордеон. Данная функция не управляет анимацией аккордеона. Сначала рассчитывается высота элемента «details» и добавляется ему соответствующие встроенные стили. После того, как это сделано, добавляется атрибут «open» для того, чтобы сделать контент видимым, но в тоже время скрытым благодаря overflow: hidden и фиксированной высоте элемента. Далее ожидается следующий кадр для вызова функции «expand» и анимирования элемента. Пример кода можно увидеть на рисунке 21.

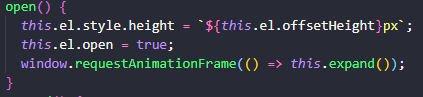


Рисунок 21 - Пример кода

Функция «expand» похожа на функцию «shrink», но вместо анимирования от текущей высоты элемента до его высоты в закрытом состоянии, анимируется от высоты элемента до полной высоты. Полная высота равняется высоте summary плюс высота внутреннего содержимого , как на рисунке 22.

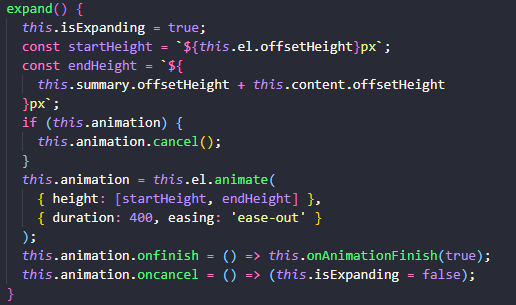


Рисунок 22

onAnimationFinish() вызывается в конце анимации раскрытия и закрытия details. Она принимает один параметр - логическое значение для атрибута «open», который больше не обрабатывается браузером, как на рисунке 23.

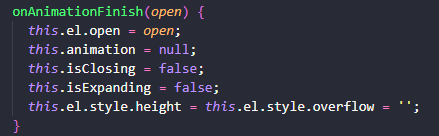


Рисунок 23

Все, что осталось сделать, это создать экземпляр класса «Accordion» для каждого элемента «details» на странице. Пример кода на рисунке 24.



Рисунок 24

Анимация элемента закончена, далее необходимо добавить кнопку, которая отправит пользователя наверх страницы. Скрипт достаточно простой, страница при клике на кнопку скролится вверх на определенное количество пунктов. Сама кнопка появляется тогда, когда пользователь пролистнул страницу на высоту, равную высоте отображения его дисплея рисунок 25 и рисунок 26.

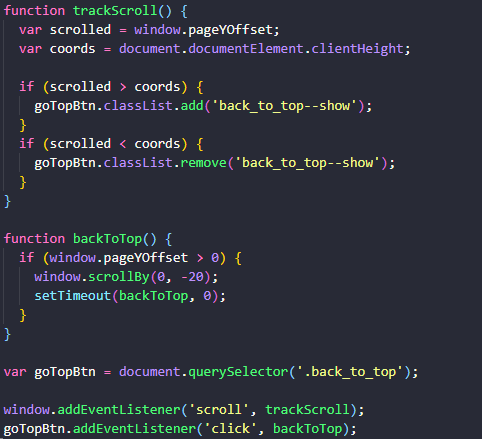


Рисунок 25 - пример кода скрипта кнопки «Наверх»

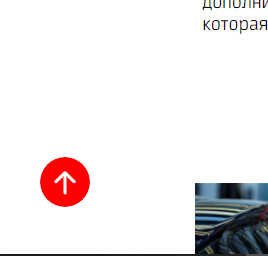


Рисунок 26 - Вид кнопки «Наверх»

По завершении работы проект для общего доступа был залит на веб-сервис GitHub.

# ЗАКЛЮЧЕНИЕ

Несомненно, данный проект требует доработок. В будущем планируется реализовать серверную часть (отправку данных из формы в базу данных, панель администратора). Так же требует улучшений сама форма онлайн записи. Появятся дополнительные страницы, содержащие подробное описание услуг и их технологий.

Но при разработке данного проекта был получен опыт работы с WAAPI и создании курсовых работ. Анимацию можно было сделать на одном CSS, но лучше попробовать себя в чем-то новом.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. <https://ru.wikipedia.org>
2. <https://habr.com>
3. <https://developer.mozilla.org>
4. <https://yandex.ru/map-constructor>
5. <https://css-minifier.com>
6. <https://autoprefixer.github.io/ru>
7. <https://validator.w3.org/nu>
8. <https://ru.stackoverflow.com>
9. <https://www.cyberforum.ru>
10. <https://materializecss.com>
11. <https://code.tutsplus.com/ru>
12. <https://tpverstak.ru>
13. <https://htmlacademy.ru>
14. <https://icon-icons.com>
15. <https://html5book.ru>