МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОМСКОЙ ОБЛАСТИ

БПОУ «Омский автотранспортный колледж»

Специальность: 09.02.07 «Информационные системы и программирование

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

**КУРСОВОЙ ПРОЕКТ**

**по МДК.08.01 «Проектирование и разработка интерфейса пользователей»**

**Тема:** Разработка клиентской части Web-сайта для парка культуры и отдыха

Выполнил студент группы ИСП393

Архипов Артём Александрович

ФИО

Проверил преподаватель:

Рычкова Анна Александровна

Оценка

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

**Омск 2022**

**Содержание**

[ВВЕДЕНИЕ 4](#_Toc102730643)

[ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ 5](#_Toc102730644)

[1.1 Обоснование разработки сайта 5](#_Toc102730645)

[1.2 Характеристика целевой аудитории 5](#_Toc102730646)

[1.3 Техническое задание на проект 6](#_Toc102730647)

[2 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ 11](#_Toc102730648)

[2.1 Логическая структура сайта 11](#_Toc102730649)

[2.2 Описание цветовой модели сайта 11](#_Toc102730650)

[2.3 Организация разметки страниц сайта 12](#_Toc102730651)

[2.4 Формирование стилей страниц 14](#_Toc102730652)

[2.5 Организация веб-сценариев JS 17](#_Toc102730653)

[2.6 Тестирование веб-сайта 18](#_Toc102730654)

[3. Организационный раздел 19](#_Toc102730655)

[3.1 Описание назначения веб-сайта 19](#_Toc102730656)

[3.2 Перечень функциональных возможностей 19](#_Toc102730657)

[3.3 Технические требования для использования продукта 19](#_Toc102730658)

[ЗАКЛЮЧЕНИЕ 20](#_Toc102730659)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 21](#_Toc102730660)

# ВВЕДЕНИЕ

Городской парк - структурный элемент системы озелененных территорий города, выполняющий рекреационные и архитектурно-художественные функции.

Для городских парков большое значение имеет функциональная и композиционная взаимосвязь с городским окружением: их композиция определяется функциональным назначением и местоположением в городской застройке. Основой архитектурно-планировочного решения парка являются природно-ландшафтные компоненты - рельеф, вода и насаждения.

В целях ознакомить людей с парком нужно создать веб-сайт.

Целью курсовой работы является создание и описание клиентской части веб-сайта парка культуры и отдыха.

Для достижения цели проекта нужно выполнить следующие задачи:

1. Исследовать предметную область
2. Охарактеризовать целевую аудиторию;
3. Составить техническое задание на проект;
4. Сформировать логическую структуру сайта;
5. Создать дизайн- макет сайта;
6. Выполнить верстку страниц сайта;
7. Разработать интерфейс пользователя;
8. Организовать обратную связь с клиентами;
9. Сформировать рекомендации для пользования сайтом.

# ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

## Обоснование разработки сайта

Парк культуры и отдыха – природный и культурно-просветительский комплекс, который по размерам, размещению в плане населенного пункта и экологическим характеристикам позволяет обеспечивать наилучшие условия для отдыха населения и проведения просветительских, культурно-массовых, физкультурно-оздоровительных мероприятий, организации игр и развлечений, создание условий для занятий художественным любительским творчеством. В современном городе парк - это самое демократичное и самое экологически чистое учреждение культуры и досуга.

## 1.2 Характеристика целевой аудитории

ХЦА подразделяются на общие и специализированные — в последнем случае они сфокусированы на конкретной тематике и, соответственно, ориентированы не на всю аудиторию в целом, а на её часть. Чёткость определения целевой аудитории в целях предпринимательской деятельности бизнеса обеспечивает средствам массовой коммуникации определённую устойчивость, однако снижает масштабы спроса на их продукцию. Согласно социологическим исследованиям, журналисты не всегда имеют адекватное представление об аудитории, для которой предназначаются их публикации, в результате чего материалы усредняются и обезличиваются, а ХЦА становятся всё более похожи. Целевая аудитория 14-44 лет, в стратегии ведения социальных сетей опираются на молодую аудиторию. Городские жители, которые придерживаются активного образа жизни аудитория считается наиболее активной и платёжеспособной и поэтому ценится у рекламодателей.

## 1.3 Техническое задание на проект

* + 1. **Наименование и реквизиты Заказчика**

**Заказчик:**

Индивидуальный предприниматель Артем Архипов.

ЧР, г.Омск, ул. Гагарина, д. 110, оф. 6

Телефон: 8-666-(666)-66-66

e-mail: ip.helper@goods.com

* + 1. **Перечень документов, на основе которых создается Сайт**

Сайт разрабатывается на основе данного Технического Задания, сопровождающих файлов и документов. Документы, предлагаемые Исполнителем, предварительно согласовываются с Заказчиком.

**Плановые сроки начала и окончания работ**

Определяются в соответствии с согласованными сроками проведения работ,

определенными в Договоре между Заказчиком и Исполнителем.

* + 1. **Назначение и цели создания сайта**

**Назначение сайта**

Сайт должен представлять «Beerplace» в сети интернет,

позволять посетителям выбраться на прогулку в парк.

**Цели создания сайта**

Целью создания сайта является ознакомления с парком, его аттракционами, погодой, местоположением.

**Целевая аудитория**

Парк нацелен на любую аудиторию, которая умеет пользоваться тырнетом.

* + 1. **Этапы создания сайта**

Разработка сайта для «Beerplace» включает в себя

следующие основные этапы работ:

разработку макета-прототипа сайта;

разработку технического задания на создание сайта;

разработку дизайна Веб-сайта;

программирование функциональных модулей сайта;

интеграцию дизайна с программным обеспечением;

первичное наполнение информационными материалами Заказчика**.**

* + 1. **Основные требования к сайту**

**Требования к структуре и функционированию сайта**

Сайт должен состоять из взаимосвязанных разделов с четко разделенными функциями.

**Структура сайта**

Сайт должен состоять из следующих разделов:

Главная страница

Галерея

Контакты

Навигация

Навигация должна присутствовать по страницам

* Главная
* Галерея
* Контакты

В разделе шапки и подвала сайта соответственно. В случае, если пользователь зашел с телефона, меню превращается в бургер, при нажатии на который открывается полное меню.

В подвале меню остается прежним

* + 1. **Требования к дизайну сайта**

**Общие требования**

Основным цветом сайта будет какой-либо оттенок зеленого.   
В качестве заднего фона белый цвет.

Тени и все такое.

**Шрифты**

Шрифт подбирается на усмотрение дизайнера Исполнителя.

Шрифты для сайта должны быть системными и отображаться на всех компью-

терах и во всех браузерах. Размеры шрифтов должны быть в диапазоне 10-100пикселей

размера и легко восприниматься для беглого чтения.

Шрифты должны быть четко различимы, как по цвету, так и по контрасту в сочетании с используемыми фоновыми цветами и изображениями.

**Требования к дизайну элементов сайта**

**Логотип**

Логотип может отсутствовать.

**Шапка сайта**

Шапка сайта должна содержать меню сайта.

Основное и второстепенное меню

Основное меню должно располагаться в верхней части окна и

содержать ссылки на все разделы первого уровня.

**Основное поле контента**

Основное поле контента должно располагаться в центре страницы. В этом

поле отображается основное содержание выбранного раздела. Стилевое

оформление материалов и их элементов (ссылок, заголовков, основного текста,

изображений, форм, таблиц и т.п.) должно быть единым для всех страниц веб-

сайта.

* + 1. **Описание страниц**

1. Главная страница

Главная страница содержит интро, секцию погоды, секцию с аттракционами и формой для связи.

1. Страница «Контакты»

Содержит адрес и номер телефона, форму, карту.

1. Страница «Галерея»

Содержит галерею.

Шапка сайта.

Главное меню.

Ниже заголовок страницы, под ним контактная информация о Заказчике

(адрес, телефоны, адреса электронной почты), информация о

свидетельстве, банковские реквизиты и схема проезда.6.9. Страница

Требования к серверной части

Веб-сайт должен функционировать в стандартной операционной серверной

среде и не требовать приобретения дополнительных программных лицензий для работы сайта и подключения пользователей.

**1.4 Описание ПО для разработки сайта**

*HTML+SASS +JS*

*Редактор: webstorm*

*Также были подключены некоторые node модули, как и сам node.*

Аббревиатура HTML расшифровывается как HyperText Markup Language. То есть HTML – это язык гипертекстовой разметки размещенных в интернете документов. Благодаря его использованию браузеры получают информацию о том, как отображать различные элементы веб-страниц. Некоторые пользователи путают HTML с языками программирования.

Sass (Syntactically Awesome Stylesheets) — модуль, включенный в [Haml](https://ru.wikipedia.org/wiki/Haml). Sass — это [метаязык](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%B0%D1%8F%D0%B7%D1%8B%D0%BA) на основе [CSS](https://ru.wikipedia.org/wiki/CSS), предназначенный для увеличения уровня абстракции CSS-кода и упрощения файлов [каскадных таблиц стилей](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%81%D0%BA%D0%B0%D0%B4%D0%BD%D1%8B%D0%B5_%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D1%8B_%D1%81%D1%82%D0%B8%D0%BB%D0%B5%D0%B9).

Node или Node.js — программная платформа, основанная на движке [V8](https://ru.wikipedia.org/wiki/V8_(%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA_JavaScript)) (транслирующем [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) в [машинный код](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4)), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами [ввода-вывода](https://ru.wikipedia.org/wiki/%D0%92%D0%B2%D0%BE%D0%B4-%D0%B2%D1%8B%D0%B2%D0%BE%D0%B4) через свой [API](https://ru.wikipedia.org/wiki/API), написанный на [C++](https://ru.wikipedia.org/wiki/C%2B%2B), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль [веб-сервера](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), но есть возможность разрабатывать на Node.js и десктопные оконные приложения (при помощи [NW.js](https://ru.wikipedia.org/wiki/NW.js), AppJS или [Electron](https://ru.wikipedia.org/wiki/Electron) для [Linux](https://ru.wikipedia.org/wiki/Linux), [Windows](https://ru.wikipedia.org/wiki/Windows) и [macOS](https://ru.wikipedia.org/wiki/MacOS)) и даже программировать микроконтроллеры (например, tessel, low.js и espruino). В основе Node.js лежит [событийно-ориентированное](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B1%D1%8B%D1%82%D0%B8%D0%B9%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%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%D0%B5) и [асинхронное (или реактивное)](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%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%D0%B5) программирование с [неблокирующим вводом/выводом](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B1%D0%BB%D0%BE%D0%BA%D0%B8%D1%80%D1%83%D1%8E%D1%89%D0%B0%D1%8F_%D1%81%D0%B8%D0%BD%D1%85%D1%80%D0%BE%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F).

Webpack — это сборщик модулей [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) 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). Он создан в первую очередь для JavaScript, но может преобразовывать внешние ресурсы, такие как [HTML](https://ru.wikipedia.org/wiki/HTML), [CSS](https://ru.wikipedia.org/wiki/CSS) и изображения, если включены соответствующие загрузчики. webpack принимает модули с зависимостями и генерирует статические ресурсы, представляющие эти модули.

JavaScript - это язык программирования, как правило, выполняется на стороне клиента. Он используется для взаимодействия с пользователем. Он также используется в разработке игр, настольных и мобильных приложений, в создании документов pdf и виджетов рабочего стола. Веб-браузеры имеют встроенную поддержку для этого языка.

Axios(AJAX) — AJAX – это асинхронный js, с помощью которого можно ждать ответы от сервера прямо в коде и применять данные, пришедшие с сервера, как только они будут готовы. Axios – это библиотека, которая упрощает http запросы (get, post, update etc.).

# 2 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ

## 2.1 Логическая структура сайта

Логическая структура сайта — это совокупность всех страниц на сайте, расположенных с учётом иерархии. То есть, взаимосвязь страниц, в которых прослеживается их принадлежность к разделам, категориям, подкатегориям и другим типам страниц (карточкам товара, тегам, фильтрам и прочее). Структура сайта строится таким образом, чтобы упростить пользователям переход от общих тем к конкретной информации, за которой они пришли. От этого зависит удобство пользователя, время, которое он проведет на вашем сайте, целевые действия, которые он совершит.

Сайт должен содержать в себе три страницы, в каждой из которой можно перейти на другую. Страницы содержат обязательно шапку и подвал, а также основное содержимое страницы.

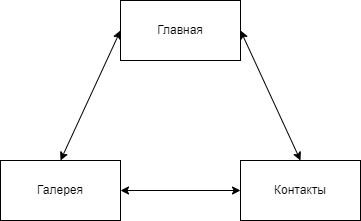


Рисунок 1 - структура сайта

## 2.2 Описание цветовой модели сайта

Дизайн сайта выполняется современными стандартами Веб-дизайна.

Стиль сайта можно описать, как яркий. В качестве цветовой гаммы рекомендуется использовать светлые цвета (желтый, зеленый, небесный, белый) и преимущественно зеленый цвет. Остальные цвета подбираются на усмотрение дизайнера Исполнителя, исходя из ассоциаций с представляемой областью деятельности компании. Все страницы сайта должны: быть выполнены в едином стиле и четко идентифицироваться, как составляющие части веб-сайта компании; быть выполнены с доминирующим присутствием фирменных цветов, содержать логотип и название Заказчика; все страницы сайта должны содержать элементы, дающие посетителю сайта четкое представление о том, в каком из разделов он находится. при добавлении новых элементов на сайт, внешний вид страниц не должен меняться. при увеличении количества информации на одной странице, если она не может в себя вместить полный объем текста или картинок, должен быть осуществлен переход на вторую и последующие страницы с понятным интерфейсом «перехода».

## 2.3 Организация разметки страниц сайта

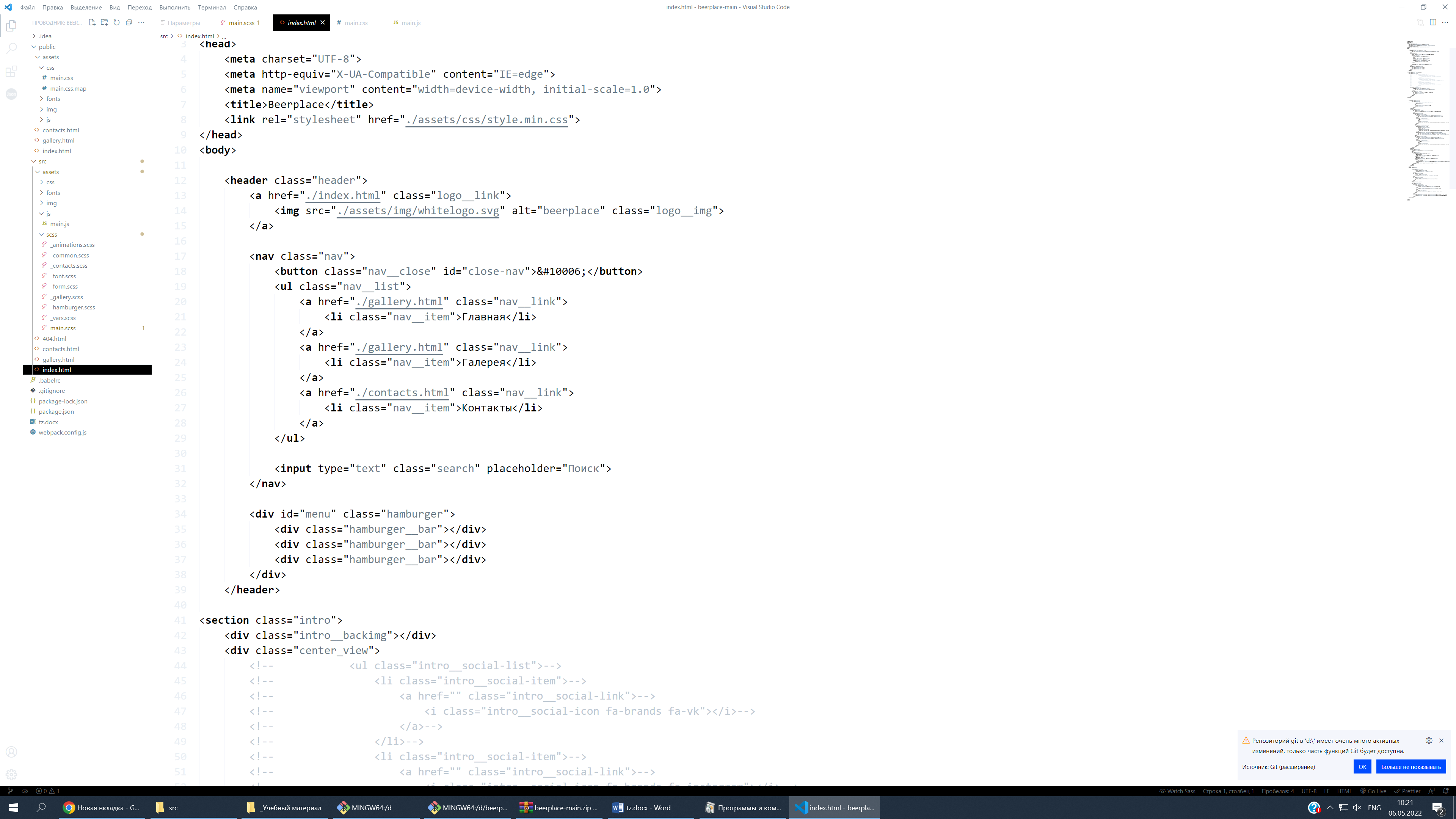
м

Рисунок 2 - Шапка и меню в шапке

На рисунке 2 показан и меню в шапке, которое превращается в гамбургер при определенном разрешении экрана.

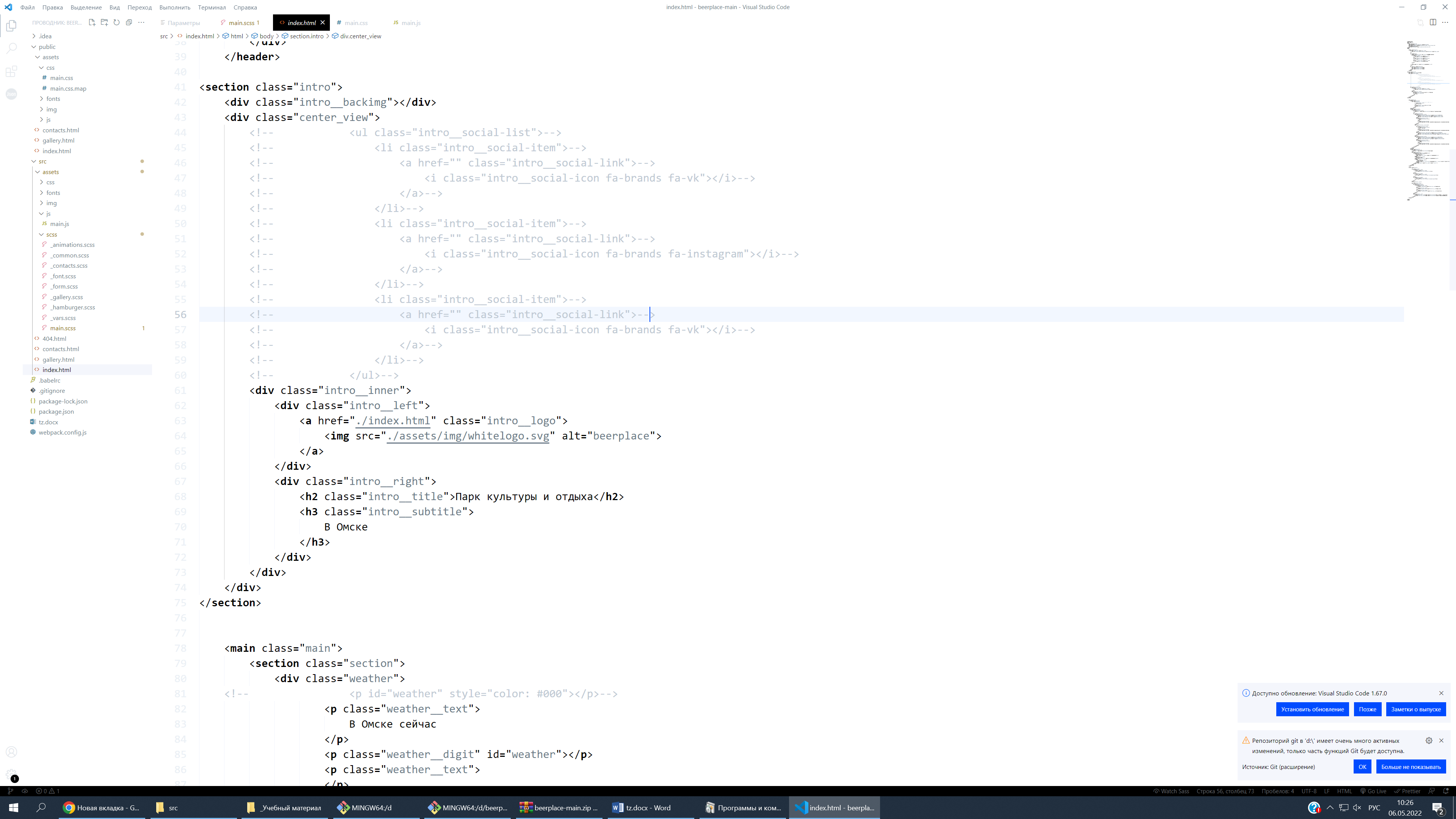


Рисунок 3 - Интро сайта (оно присутствует на всех страницах сайта)

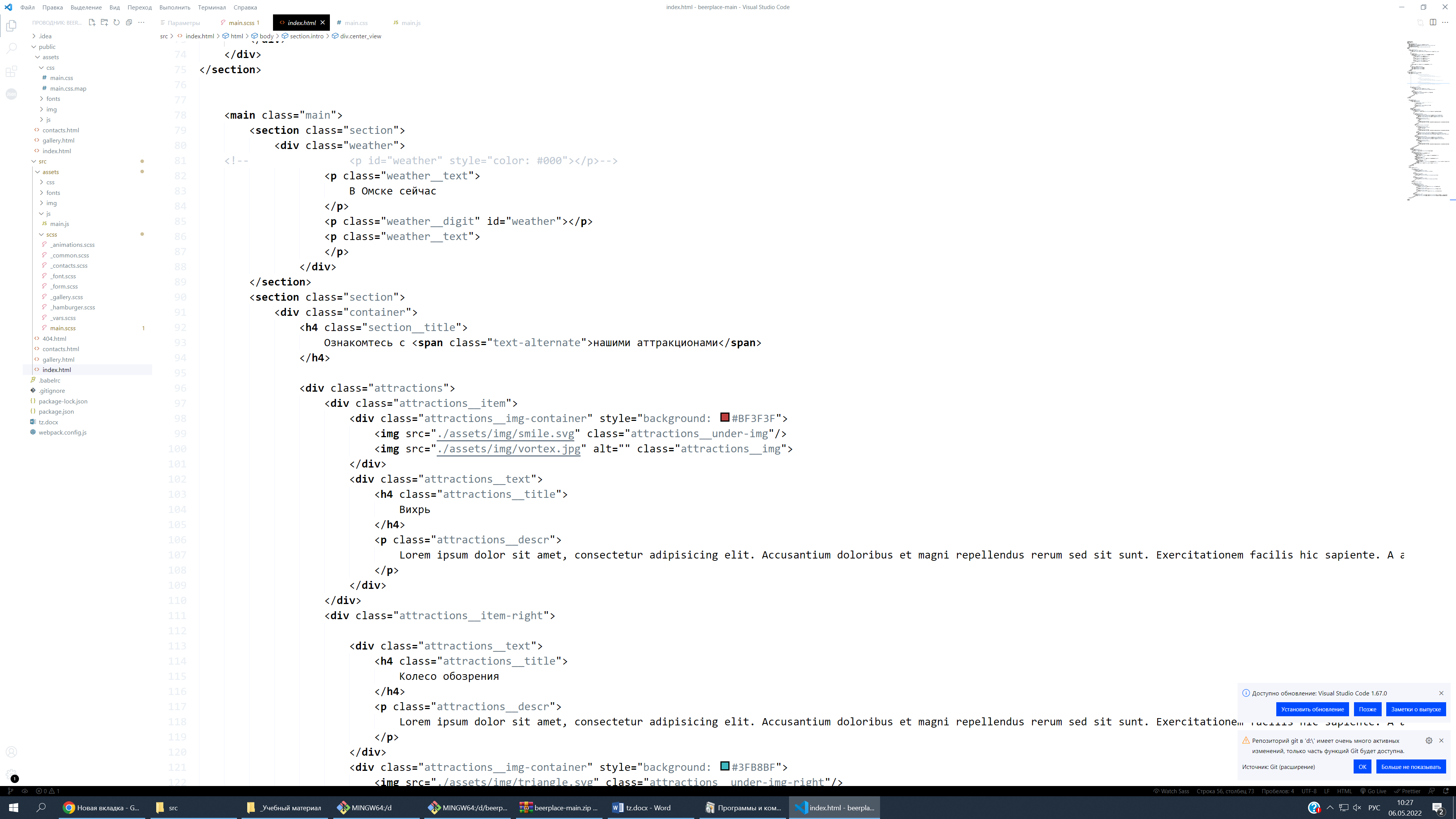


Рисунок 4 - Основная часть сайта

В основной части сайта находятся все секции страницы.

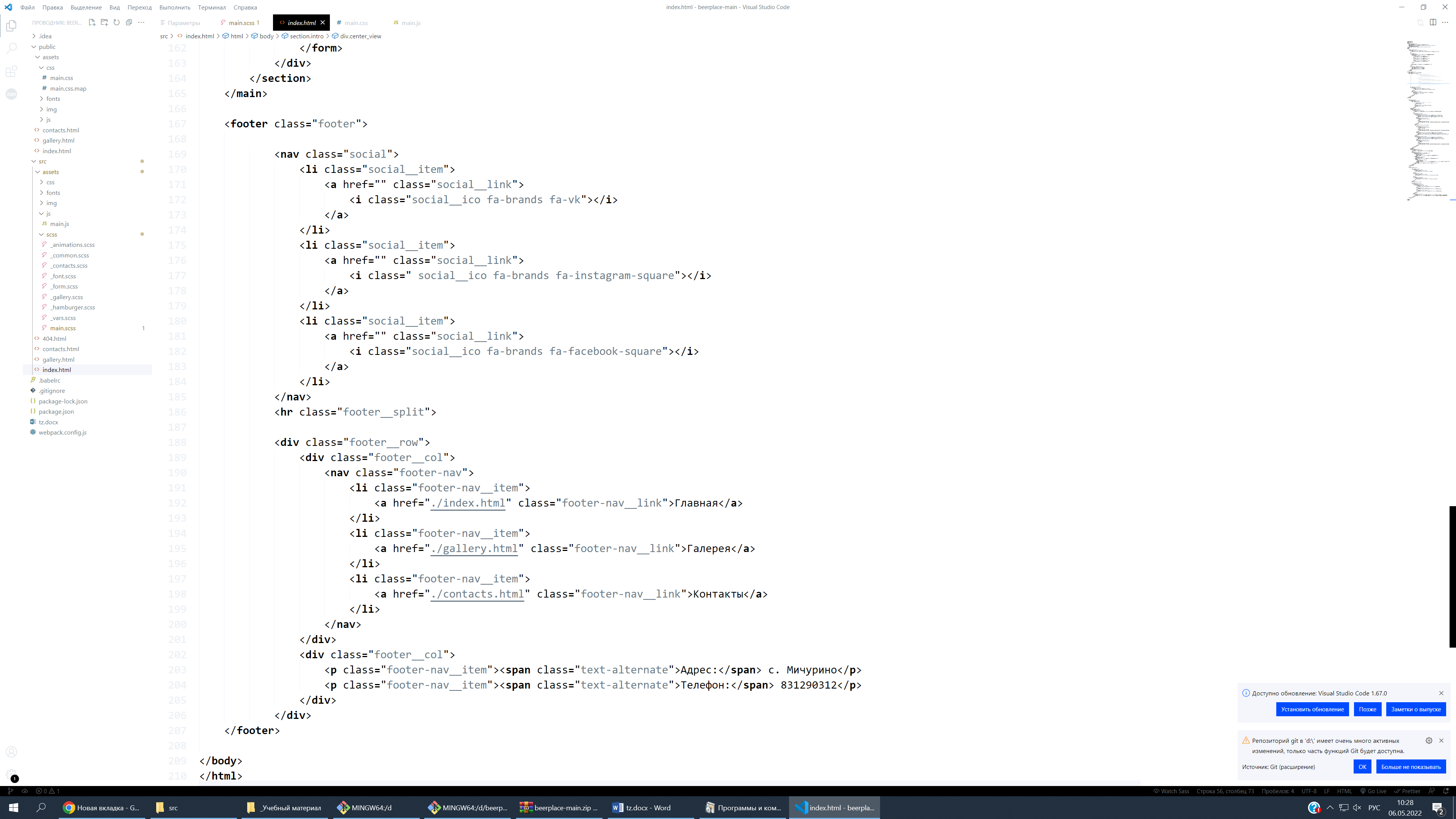


Рисунок 5 - Подвал

# 2.4 Формирование стилей страниц

2.4.1 Реализация цвета, шрифта, фона

При работе над цветовой схемой следует руководствоваться тематикой сайта. Известно, что цвет управляет эмоциями посетителей и определяет отношение к сайту и бренду в целом. Когда мы смотрим на какой-либо цвет, то глаза взаимодействуют с определенной областью головного мозга, известной как гипоталамус. Он посылает сигнал в гипофиз к эндокринной системе, и затем к щитовидной железе. А щитовидная железа дает сигнал на выработку специальных гормонов, которые влияют на настроение, эмоции и поведение.

Для того, чтобы правильно выбрать цветовую схему, необходимо следовать следующими принципам:

правильный цвет для тематики;

правильный цвет для целевой аудитории;

правильный цвет в зависимости от времени;

правильный цвет для каждого целевого действия.

Существуют большие различия между цветовыми предпочтениями у женщин и мужчин. Нейл Патель из компании QuickSprout выявил, какие цвета предпочтительны для каждого пола.

В проведенном им опросе 35% женщин сказали, что синий – их любимый цвет, затем идет фиолетовый (23%) и зеленый (14%). 33% женщин признались, что оранжевый и коричневый их самые нелюбимые цвета, кроме этого к самым нелюбимым женщины отнесли еще и серый (17%).

Существует парадигма, что женщины любят розовый цвет. Но это не так. Розовый цвет в качестве любимого выбрал всего лишь небольшой процент женщин. Поэтому рекомендуется в сайтах электронной коммерции использовать вставки из синего, фиолетового и зеленого цветов. Это улучшит привлекательность вашего сайта для женщин и поднимет конверсию.

Мужчины не любят фиолетовый, оранжевый и коричневый. Им ближе синий, зеленый и черный. Поэтому, если занимается продажами товаров для мужчин, то не нужно использовать фиолетовый, оранжевый и коричневый в качестве основных цветов сайта. Это может оттолкнуть потенциального покупателя. Традиционными цветами для мужской аудитории считаются синий и черный. Именно эти цвета ассоциируются c мужественностью.

Считается, что для того, чтобы вызвать доверие у пользователей, нужно использовать синий цвет. Синий является самым распространенным цветом. Это неудивительно, ведь именно он является наиболее предпочтительным, как для мужчин, так и для женщин. Синий является цветом доверия, мира и порядка, он вызывает в памяти чувства спокойствия и безмятежности. Таким образом, большинство научного сообщества склоняется к мнению, что синий цвет вызывает доверие и спокойствие. Многие известные компании в мире используют именно синий в качестве основного.

Так как при анализе целевой аудитории было выявлено, что ключевой персонаж – это мужчина средних лет, имеющий определенный социальный статус, хотя среди клиентов сайта могут быть также и женщины, то для реализации интерфейса страниц была разработана цветовая схема, в которой преобладающими цветами являются синий – как универсальный и популярный, серый – как символ успешности, благородства и стиля, и белый – как символ открытости и доверия.

Шрифт подбирается на усмотрение дизайнера Исполнителя.

Шрифты для сайта должны быть системными и отображаться на всех компьютерах и во всех браузерах. Размеры шрифтов должны быть в диапазоне 10-24 размера и легко восприниматься для беглого чтения.

Шрифты должны быть четко различимы, как по цвету, так и по контрасту в сочетании с используемыми фоновыми цветами и изображениями.

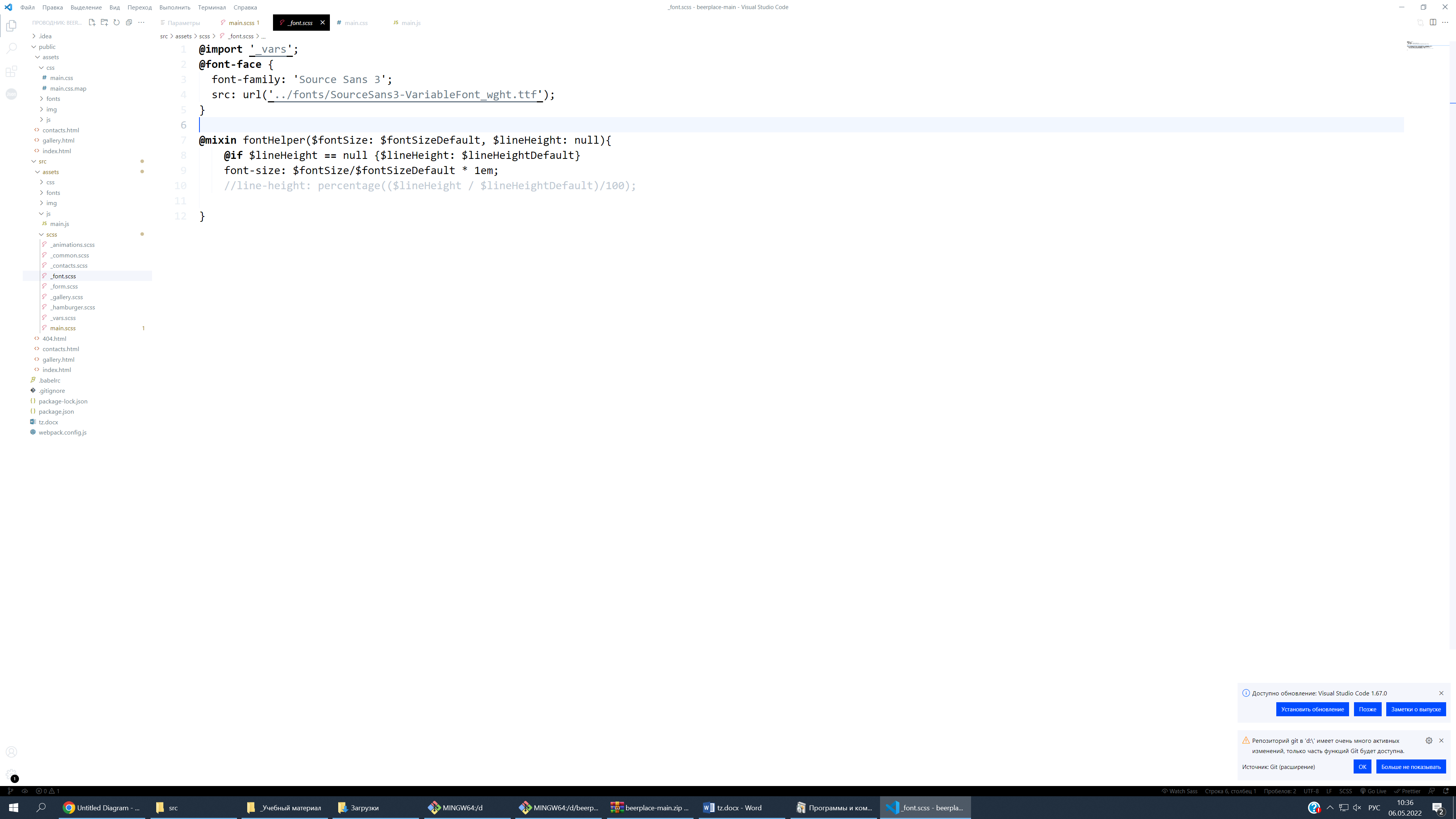


Рисунок 6 – шрифт

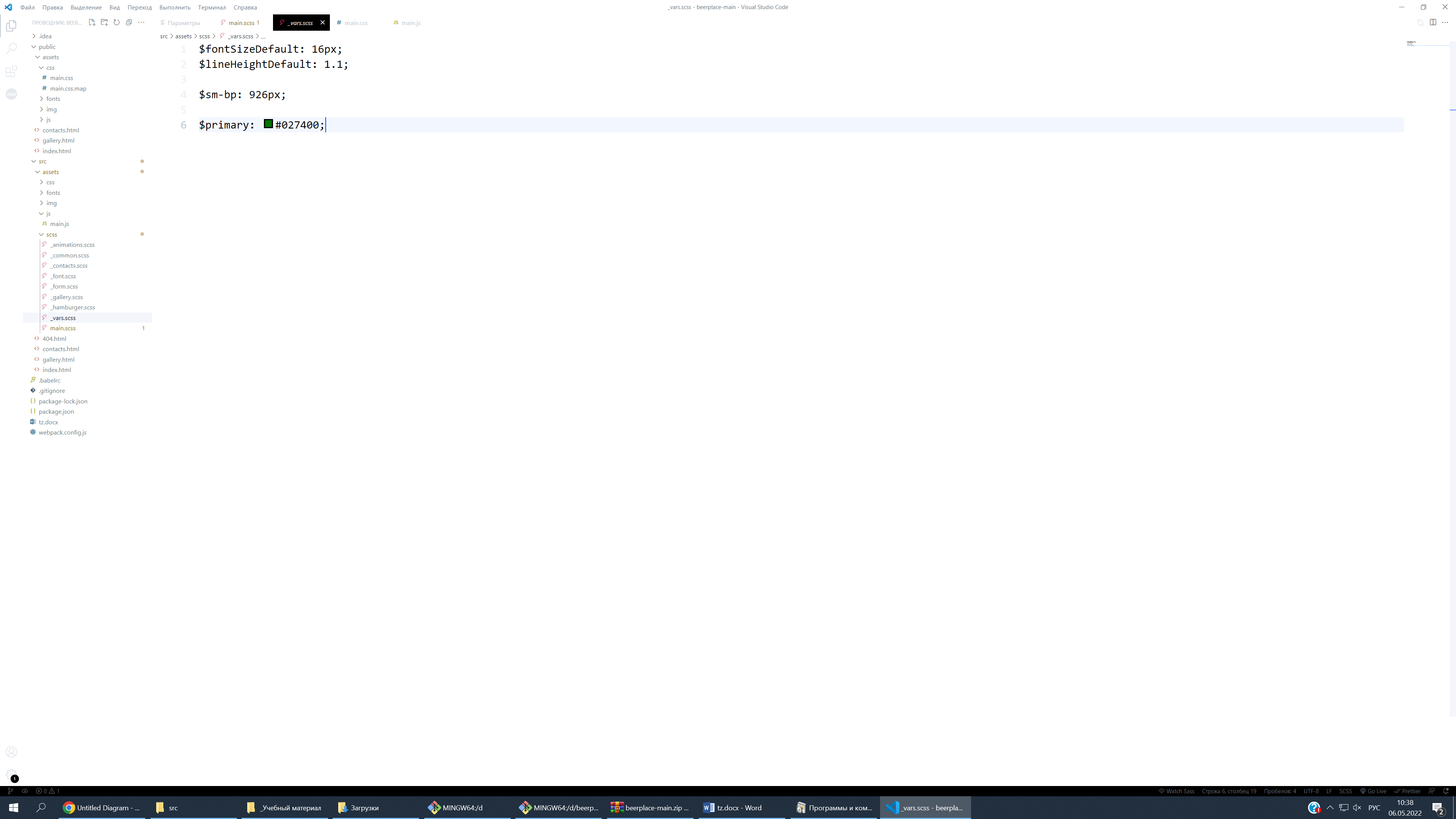


Рисунок 7 – Цвета

Цвет сайта использовался только приоритетный цвет – зеленый, цвет фона в теге body установлен по умолчанию (т.е. белый)

2.4.2 Реализация ссылок



Рисунок 8 - Меню сайта

Главная страница содержит одно меню в шапке и в подвале, которое выглядит как на рисунке 8.

## 2.5 Организация веб-сценариев JS

С помощью AJAX и API apiweathermap js получает нынешнюю погоду в Омске при загрузке страницы. Если элемент weather (т.е. секция, в которой предположительно находится погода) не отсутствует, он дождётся ответа от сервера и вставит погода в id#weather. Также на основе погоды формируется надпись под ней.

При нажатии на бургер убирается/добавляется класс nav\_\_reveal

Слайдер реализован с помощью модального окна.

## 2.6 Тестирование веб-сайта

Тестирование данного продукта производилось на протяжении всего этапа разработки. При этом были выполнены тестовые проверки на валидность кода, проверки функционала сайта в соответствии с заданием, проверки логики построения сайта, корректности ссылок, скорости нахождения нужной информации. Было отработано устранение ошибки 404 путем формирования дополнительной функции по сообщению пользователю о временно неработающих ссылках. Это делается для того, чтобы пользователь видел, что администрация сайта знает о имеющихся проблемах и делает шаги по их устранению Скриншоты проверок

# 3. Организационный раздел

## 3.1 Описание назначения веб-сайта

Городской парк - структурный элемент системы озелененных территорий города, выполняющий рекреационные и архитектурно-художественные функции.

Для городских парков большое значение имеет функциональная и композиционная взаимосвязь с городским окружением: их композиция определяется функциональным назначением и местоположением в городской застройке. Основой архитектурно-планировочного решения парка являются природно-ландшафтные компоненты - рельеф, вода и насаждения.

В целях ознакомить людей с парком нужно создать веб-сайт.

## 3.2 Перечень функциональных возможностей

Сайт может пользоваться api от weathermap и api от yandexmap

## 3.3 Технические требования для использования продукта

На стороне клиента должно поддерживать минимум вот эти бразуеры:

• Edge 12

• Firefox 3.5

• Chrome 4

• Opera 11.5

• Android 2.1

• Blackberry 7

• Samsung Internet 4

Серверная часть:

• Процессор: 2 x Intel Xeon E5-2620v3(12 ядер, 2.4-3.2 GHz, 24 потока)

• Память: 32 Gb DDR4 ECC, 4-х канальная

• Дисковая система: 2 x 240Gb SSD + 1Tb SATA

• Клиентская часть:

• 32(х86) или 64(х64) разрядная система, при этом частота процессора должна быть не менее 1ГГц

• ОЗУ. Для 32-разрядной версии необходим 1ГБ оперативной памяти, а 64-разрядной нужно 2 ГБ

• Память. Для 32-разрядной системы хватит 16 ГБ на жестком диске, а 64-разрядной необходимо 32 ГБ

• Графическое устройство DirectX 9 с драйвером WDDM версии не ниже 1.0

# ЗАКЛЮЧЕНИЕ

В ходе создания данного курсового проекта были выполнены следующие профессиональные задачи:

* Описано функционирование сайта Парка культуры и отдыха.
* Описаны [основные этапы разработки проектируемой системы](#_Toc422300370).
* Выполнено проектирование логической схемы.
* Рассмотрены необходимые функциональные модули.
* Выполнено программирование веб-страниц.
* Организована связь между модулями.
* Разработан дружественный и интуитивно-понятный интерфейс страниц.
* Проведено тестирование продукта.
* Сформулированы способы защиты данных при использовании сайта.

Цель дипломного проекта достигнута.

Сайт разработанный является тестовым, все наработки проявляются аудиторией парка, в дальнейшем есть возможность расширить сайт до полноценного вида, если будет большой охват аудитории на сайте.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Документация Webpack [электронный ресурс]/ - режим доступа: [https://webpack.js.org /](https://developer.mozilla.org/ru/) (06.05.2022)
2. Гарретт, Д. Веб-Дизайн: книга Джесса Гаррета. Элементы опыта взаимодействия / Д. Гарретт. - М.: Символ-Плюс, 2018. - 192 c.
3. Справочник по JavaScript от Mozilla [электронный ресурс]/ - режим доступа: <https://developer.mozilla.org/ru/> (06.05.2022)
4. Дакетт, Д. HTML и CSS. Разработка и дизайн веб-сайтов / Д. Дакетт. - М.: Эксмо, 2018. - 208 c.
5. Справочник по многим языкам программирования и прочему [электронный ресурс]/ - режим доступа: <https://www.w3schools.com/> (06.05.2022)
6. Официальная документация sass [электронный ресурс]/ - режим доступа: <https://sass-lang.com/documentation> (06.05.2022)
7. Киселев, С.В. Веб-дизайн: Учебное пособие / С.В. Киселев. - М.: Академия, 2018. - 416 c.
8. Киселев, С.В. Веб-дизайн / С.В. Киселев. - М.: Academia, 2019. - 285 c.
9. Киселев, С.В. Веб-дизайн: Учебное пособие / С.В. Киселев. - М.: Academia, 2017. - 312 c.
10. Макнейл, П. Веб-дизайн. Книга идей веб-разработчика / П. Макнейл. - СПб.: Питер, 2017. - 480 c.
11. Сырых, Ю. Современный веб-дизайн. Настольный и мобильный / Ю. Сырых. - М.: Диалектика, 2019. - 384 c.
12. Сырых, Ю. Современный веб-дизайн. Эпоха Веб 3.0 / Ю. Сырых. - М.: Вильямс И.Д., 2018. - 368 c.
13. Сырых, Ю.А. Современный веб-дизайн. Настольный и мобильный / Ю.А. Сырых. - М.: Вильямс, 2017. - 384 c.
14. Учебник HTML [электронный ресурс]/ - режим доступа: <http://htmlbook.ru/> (28.04.2022)
15. Лайфхакер [электронный ресурс]/ - режим доступа: <https://lifehacker.ru/kak-sdelat-sajt/> (29.04.2022)