**Содержание**

[Введение 4](#_Toc103843448)

[1. Обзор технических методов и программных средств разработки веб-сайта 5](#_Toc103843449)

[1.1 Обзор аналогичных решений 5](#_Toc103843450)

[1.2 Формирование требований к программному продукту 7](#_Toc103843451)

[1.3 Техническое задание 7](#_Toc103843452)

[1.4 Выбор средств реализации программного средства 7](#_Toc103843453)

[1.5 Вывод 9](#_Toc103843454)

[2. Последовательность разработки содержания и структуры веб-сайта 10](#_Toc103843455)

[2.1 Выбор способа верстки 10](#_Toc103843456)

[2.2 Выбор стилевого оформления 10](#_Toc103843457)

[2.3 Структура сайта 10](#_Toc103843458)

[2.4 Выбор шрифтового оформления 11](#_Toc103843459)

[2.5 Разработка логотипа 11](#_Toc103843460)

[2.6 Разработка пользовательских элементов 12](#_Toc103843461)

[2.7 Вывод 13](#_Toc103843462)

[3. Реализация структуры веб-сайта 15](#_Toc103843463)

[3.1 Структура HTML-документа 15](#_Toc103843464)

[3.2 Добавление таблиц стилей CSS и Sass 17](#_Toc103843465)

[3.3 Использование стандартов XML (SVG) 18](#_Toc103843466)

[3.4 Вывод 18](#_Toc103843467)

[4. Тестирование веб-сайта 19](#_Toc103843468)

[4.1 Адаптивный дизайн веб-сайта 19](#_Toc103843469)

[4.2 Кроссбраузерность веб-сайта 20](#_Toc103843470)

[4.3 Руководство пользователя 20](#_Toc103843471)

[4.4 Выводы 22](#_Toc103843472)

[Заключение 23](#_Toc103843473)

[Список литературы 23](#_Toc103843474)

[Приложение А 24](#_Toc103843475)

[Приложение Б 30](#_Toc103843476)

[Приложение В 37](#_Toc103843477)

Введение

Каждый день люди пользуются интернетом. Каждый пытается найти решение своей проблемы или хотя бы найти что-то полезное. По этой причине цель проекта помочь людям с зависимостью от курения. Все прекрасно знают о вреде сигарет, электронных испарителей, табачных смесей и т. д. Каждый задумывался о том, чтобы прекратить употреблять никотин вышеуказанными способами. Поэтому я решил сделать проект «О вреде курения»

В курсовом проекте основными задачами будет создание сайта, который будет удобным и приятным для пользователя.

Задачи:

- Реализация интуитивно понятного функционала

- Создание формы для оформления консультации

Во время разработки сайта планируется развитие навыков верстки сайта с использованием HTML, CSS и JS.

1. Обзор технических методов и программных средств разработки веб-сайта

1.1 Обзор аналогичных решений

Для реализации сайта нужно учитывать аналоги и попытаться взять из них самое лучшее. Самое важное в сайте это его простота и оформление. В плане оформления и функционала я взял сайт niimid представленный в рисунке.1.1

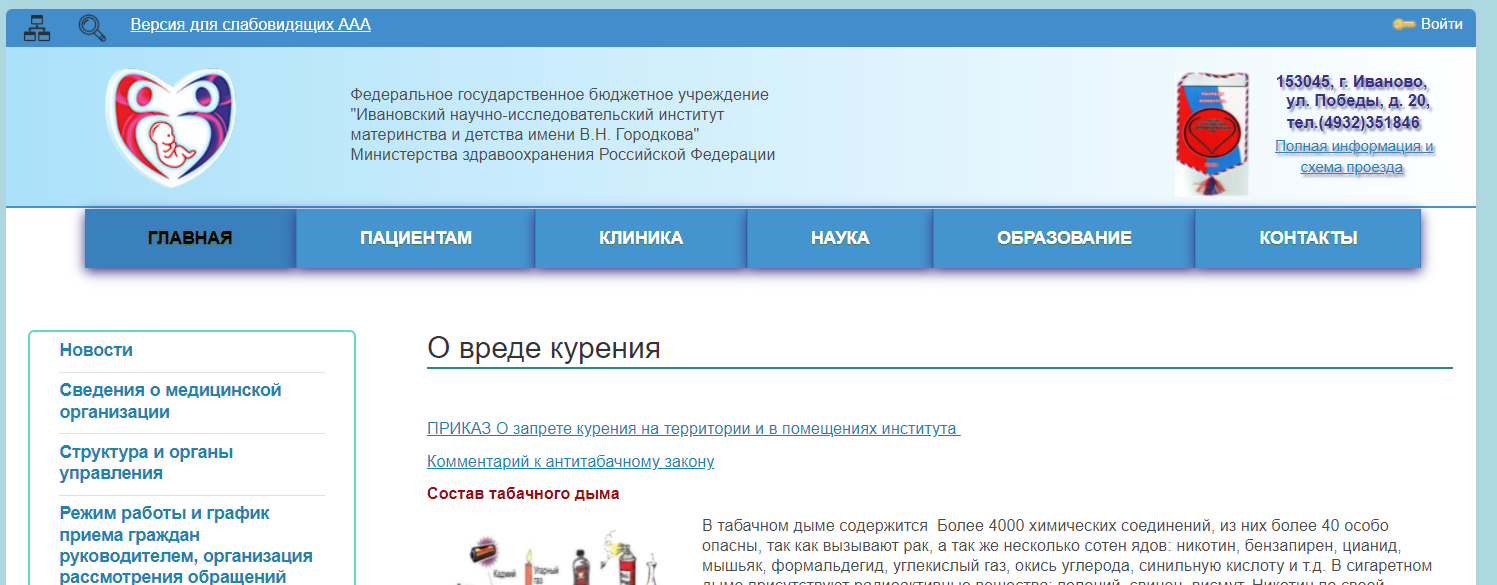


Рисунок 1.1 – Оформление страницы

Основные идеи, взятые отсюда:

- Оформление в виде синей гаммы, которая концентрирует внимание

- Запоминающийся логотип, представленный на рисунке 1.2



Рисунок 1.2 – Логотип сайта

Основная страница с ссылками на другие страницы была оформлена, как и у большинства аналогичных сайтов. Примером для формы был взят сайт 10gpd, представленный на рисунке 1.3. Здесь присутствует вся информация для того, чтобы пользователь мог сразу понять как заполнять форму для записи на консультацию.



Рис.1.3 – Форма для записи на консультацию

Страница с подробными статьями я так же реализовал используя идею аналога niimid, представленный на рисунке 1.4. Тут указана вся информация, касаемо курения. На этой странице подробно описана информация, необходимая для понимая проблемы курения.



Рис.1.4 – Статьи

Статьи являются важной частью любого информационного веб-сайта, по этой причине было решено добавить их в данный проект.

# 1.2 Формирование требований к программному продукту

Основными требованиями к веб-сайту является минимальное количество сложных элементов (gif, анимации и т. д.), которые негативно сказываются на понятности интерфейса сайта, на его оформление.

Сайт должен содержать в себе интуитивно понятный функционал. Так же не менее важной функцией сайта должна быть форма, которая заполняется для оформления заказа консультации. Сайт должен быть оформлен в неброских тонах.

1.3 Техническое задание

Основная задача сайта – передать пользователю информацию о вреде курения, предложить консультация и оказать информативную помощь для борьбы с вредной привычкой.

Сайт будет направлен для людей, которые курят и пытаются бороться с этим.

Основные задачи:

-создание сайта с множеством страниц с использованием программных средств

-создание интерфейса, который бы мог задержать пользователя на сайте как можно дольше.

-создание уникальных иконок и кнопок

-возможность ввода данных для оформления заказа

Основные требования к исполнителю: создание дизайна, проведение анализа существующих решений, создание прототипа, адаптивность и кроссбраузерность сайта (правильное отображение во всех браузерах). В проекте используется гипертекстовый язык разметки HTML, CSS, графические элементы в формате SVG.

# 1.4 Выбор средств реализации программного средства

Для написания сайта была выбрана программа Visual Studio Code. Это удобный редактор кода, разработанный Microsoft, представленный на рисунке 1.4

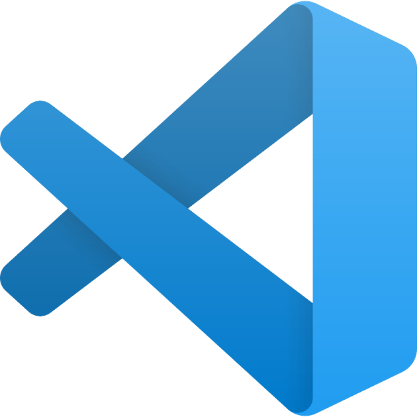


Рисунок 1.4 – VS Code

Эта программа включает в себя отладчик, подсветку синтаксиса и подсказки при написании различных тегов или методов.

Основными языками для создания сайта были HTML5, CSS3, JavaScript.

HTML – язык структурной разметки для создания Web-страниц. Это не язык программирования, это язык разметки. Он отвечает за расположение элементов на странице с помощью тегов. В этом языке отсутствуют логические функции, поэтому вы не сможете в нем произвести арифметические подсчёты. Аббревиатура HTML – Hyper Text Markup Language.

CSS (Cascading Style Sheets – каскадные таблицы стилей) – язык описания внешнего вида документа. CSS предназначен для работы с шрифтами, таблицами, картинками и т.д. Благодаря этому языку мы можем сделать дизайн сайта таким, каким хотим. Все ограничивается лишь фантазией.

JavaScript – язык скриптов. Скрипт представляет собой набор инструкций, которые не требуют компиляции перед запуском. Код JS интерпретируется браузером во время загрузки веб-страницы. Благодаря этому языку мы можем наследовать элементы, управлять объектами на странице, например удалить первый элемент при нажатии на второй элемент. Этот язык позволяет создавать анимации, которые нельзя сделать в CSS.

Для создание макетов было выбрано приложение Figma, который представлен на рисунке 1.5

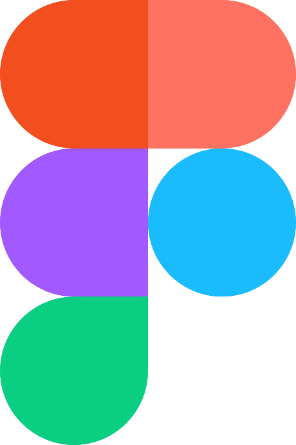


Рисунок 1.5 - Figma

Это приложение позволяет создавать предварительный вид сайта, которому мы будем следовать на протяжении всей курсовой работы. Основными функциями этого приложения являются предварительный показ сайта, работа с текстом и картинками, создание векторных изображений. В Figm’е очень удобное расположение кнопок, что позволяет наслаждаться процессом создания макета. Так же здесь можно создавать кнопки, которые буду переключать вас между фреймами в предварительном показе. Удобно еще то, что в Figm’е присутствует возможность просматривания CSS свойств элементов. Это сильно облегчает процесс.

# 1.5 Вывод

При разработке макета использовалась программа Figma. С помощью HTML создана структура сайта. CSS позволил создать красивый и адаптивный дизайн. Java Script позволил нам создать функциональные элементы. В итоге должен получиться сайт с адаптивной версткой. Сайт должен быть простым, удобным, интуитивно понятным.

2. Последовательность разработки содержания и структуры веб-сайта

2.1 Выбор способа верстки

Сайт будет верстаться прямым способом. Такая верстка выглядит строго и используется чаще всего в верстке литературно-художественных изданий, учебников, некоторых видах газет и журналов. Прямая верстка спокойнее и проще ломаной в восприятии читателями, имеет сдержанный характер.

2.2 Выбор стилевого оформления

Сайт будет стилизован по идее минимализма – убрать все лишнее и выразить самое нужное. Для цветовой палитры я выбрал 2 цвета: синий(#3B5EA3), белый(#FFFFFF). Эти цвета хорошо сочетаются. Акцентный цвет на странице делает его запоминающимся и привлекательным. Задний фон белый – это поспособствует лучшему выделению основной информации.

2.3 Структура сайта

Структура сайта будет представлена на рисунке 2.1

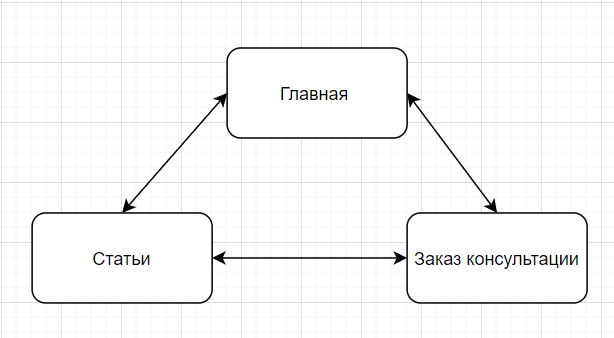


Рисунок 2.1 – Структура сайта

На странице «Главная» можно будет увидеть главную часть, ссылки на другие страницы, а именно «Заказ консультации» и «Статьи»

В блоке «Статьи» можно будет прочитать около 5 страниц со статьями на тему курение, последствия, состав табака и т. д.

На странице «Заказ консультации» можно будет заполнить форму и заказать консультацию-звонок специалиста.

Все страницы имеют в себе ссылки друг на друга, что удобно, так как можно в любой момент переключиться на необходимую страницу.

2.4 Выбор шрифтового оформления

Выбор упал на шрифт «Comfortaa». На данный момент это один из самых популярных шрифтов. Его можно рассмотреть на рисунке 2.2



Рисунок 2.2 – Шрифтовое оформление

Его можно увидеть практически на любом сайте. Он добавляет капельку загадочности.

2.5 Разработка логотипа

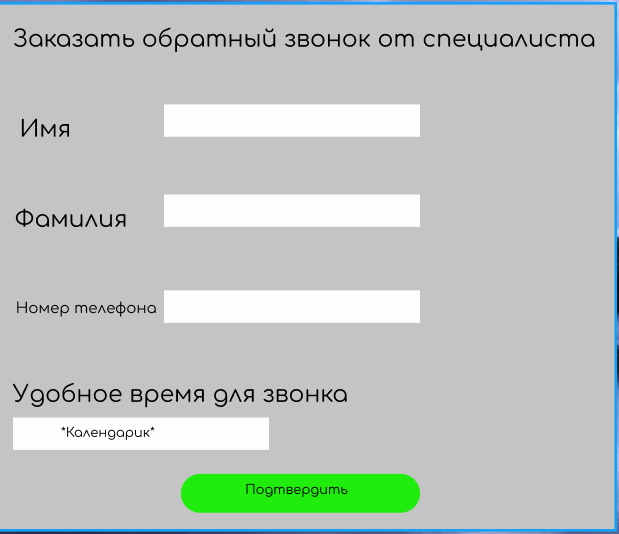
Логотип сразу дает понять пользователям, о чем будет сайт. Этот логотип можно будет заметить везде, тогда взглянув на него у пользователя будет прямая ассоциация с сайтом. Его пример на рисунке 2.3

  
Рисунок 2.3 – Логотип

Только один логотип показывает, что курить очень нежелательно. Цвета были подобраны очень удачно, так как черный и красный хорошо гармонируют, поэтому ничего не сливается и можно заметить все детали.

2.6 Разработка пользовательских элементов

На третьей странице присутствует форма, показанная на рисунке 2.а. С ее помощью можно оформить обратный звонок от специалиста.

Рисунок 2.5(а) – форма заказа

Четыре строки будут являться формами для ввода имени, фамилии, номера телефона и удобного времени для звонка.

Кнопка подтвердить будет перенаправлять на главную страницу, но перед этом будет окно с благодарностью оставления заявки.

На всех страницах будет предусмотрено выпадающее меню, которое предложит нам определенные функции. Пример на рисунке 2.б

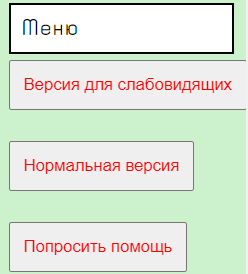


Рисунок 2.5(б) – выпадающее меню

С помощью данного меню можно будет использовать все функции сайта, так же меню добавит некую изюминку в сайт.

2.7 Вывод

Данный курсовой проект нацелен на разработку сайта о вреде курения. Я думаю, что это проблема очень актуальна для нас. Я надеюсь, что моя работа поможет кому-то взять себя в руки и бросить данное занятие. Для этого на моем сайте будет присутствовать все необходимое для этого, а именно – статьи на тему вреда, статьи на тему борьбы с этим, заказ консультации от специалиста

Для начала был создан макет сайта на Figma, который помогает представить будущее сайта и облегчает верстку благодаря свойству программы показывать CSS. Основной задачей макета является создания плана сайта, как он выглядит, какая у него цветовая палитра, как расположены элементы и т.д. Это последний этап перед началом верстки сайта, с него мы начнем копировать элементы в HTML.

Основной задачей сайта является его адаптивность и удобность в использовании, а так же его производительность. Поэтому элементы будут не статическими, а динамическими, чтобы они могли изменяться под размеры устройства.

3. Реализация структуры веб-сайта

3.1 Структура HTML-документа

Структура сайта была написана на языке HTML5. HTML5 является последним стандартом для браузеров для отображения и взаимодействия с веб-страницами. Утвержденный в 2014 году, это первое обновление для HTML за 14 лет.

Структура тела документа состоит из тегов header, main и section.

В теге header располагается логотип, кнопки для перехода на другие страницы сайта и вложенное «бургер-меню», это можно заметить на листинге 3.a

|  |
| --- |
| <header class="header">  <div class="wrapper">  <div class="logo">  <a href="index.html"> <img src="IMG/logo.svg" alt="logo"></a>  </div>  <nav class="nav">  <ul>  <li>  <a href="index.html">Главная</a>  </li>  <li>  <a href="HTML/order.html">  Заказ консультации  </a>  </li>  <li>  <a href="HTML/article.html">  Статьи  </a>  </ul>  </nav>  <div class="burger">  Меню  </div>  </div>  </header> |

Листинг 3.а – Тэг header

В данном листинге мы можем увидеть код нашей страницы. Header является «шапкой» нашего сайта. В нем расположены тэги div, li, nav, a, img. В div заключен логотип (с помощью тэга img), при нажатии на который (с помощью тэга а), можно переместиться на главную страницу. Так же в тэге nav расположены кнопки, которые отвечают за перемещение на другие страницы веб-сайта.

В теге main так же находится кнопка для перехода на другую страницу и главный экран. Это мы можем увидеть в Листинге 3.б

|  |
| --- |
| <main class=”main">  <div class="wrapper">  <div class="description">  Сайт, благододаря которому бросишь курить  </div>  <div class="block\_down">  <div class="quote">  Вред курения очевиден. От курения тупеешь.  Оно несовместимо с творческой работой  <div class="author">В. Гете </div>  </div>  <div class="btn\_block">  <button class="btn">  <a href="HTML/article.html"> Перейти к статьям</a>  </button>  </div>  </div>  </div>  </main> |

Листинг 3.б – Тэг main

В данном листинге можно рассмотреть код нашей страницы, в котором реализован тэг main. Здесь реализована главная страница веб-сайта. Тут расположена сама кнопка (тэг button), которая реагируют на нажатие и переносят на другую страницу. Так же тут находятся тэги div, которые отвечают за оформление с помощью class, которые будут изменяться через CSS.

В теге section находятся сами ссылки на статьи, которые привязаны к кнопкам. Листинг 3.в

|  |
| --- |
| <section class="navs">  <div class="wrapper">  <nav class="nav">  <ul>  <li>  <a href="HTML/order.html">  Заказ консультации  </a>  </li>  <li>  <a href="HTML/article.html ">  Статьи  </a>  </ul>  </nav>  </div>  </section> |

Листинг 3.в – Тэг section

В данном листинге мы можем рассмотреть код нашей страницы, в котором реализован тэг section. В нем расположены кнопки

3.2 Добавление таблиц стилей CSS и Sass

В процессе разработки веб-сайта использовалась такая таблица стилей как CSS. CSS – это язык, с помощью которого описывается внешний вид документа HTML, XML, XHTML. Название означает «каскадная таблица стилей», или Cascading Style Sheets. CSS-стили незаменимы при оформлении страниц сайтов: в одном файле содержатся сведения об отображении всех элементов документа.

Использование CSS направлено на то, чтобы задавать визуальное оформление web-страницы. При выполнении разметки элементов языком HTML документ получит основную структуру страницы. Это заголовки различного уровня, параграфы, маркированные и нумерованные списки, отступы, картинки и прочее. Но, если все оставить как есть, то пользователю на странице будет выведен безликий текст, который будет плохо восприниматься

|  |
| --- |
| <link rel="stylesheet" href="CSS/main.css"> |

Листинг 3.2 – Подключение CSS

Данный код показывает, как можно подключить CSS.

CSS, в отличие от HTML, может менять настройки текстовых блоков. Пользователь задает шрифт, размер этого шрифта, цвет текста и фона. Подобных свойств очень много, а современные версии CSS позволяют даже работать с анимацией. При создании страницы на сайте пользователь сначала прописывает HTML-кодом, что именно там будет отображаться. А при помощи кода CSS можно задать визуальные настройки текстовой основы.

3.3 Использование стандартов XML (SVG)

В процессе разработки веб-сайта, XML использовался для подключения текста и фото к веб-сайту.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <root>  <article>  <description>  Text  </description> |

Листинг 3.3 – использование XML

В данном коде мы можем наглядно увидеть использование XML.

SVG — XML язык разметки на основе векторной графики. По сути это текстовый файл, который является открытым веб-стандартом для описания двухмерных векторных изображений без потери качества при масштабировании. Также поддерживает интерактивность и анимацию. Разрабатывается с 1999 года и используется для того, чтобы описать с помощью языка программирования графические элементы сайта.

3.4 Вывод

На данном этапе была реализована структура на HTML, а также созданы таблицы стилей CSS/Sass. Было продемонстрировано использование стандартов XML (SVG).

4. Тестирование веб-сайта

4.1 Адаптивный дизайн веб-сайта

Сегодня количество пользователей, которые заходят в Интернет с мобильных устройств очень велико, и с каждым днем их становится все больше. А потому адаптивный дизайн веб-сайта является неотъемлемой частью каждого проекта.

CSS3-медиазапросы используются для создания адаптивного дизайна. Он имеет множество параметров, такие как max-width (отвечающий за максимальную ширину) или min-width (минимальную ширину). Рассмотрим на рисунках 4.а, 4.б

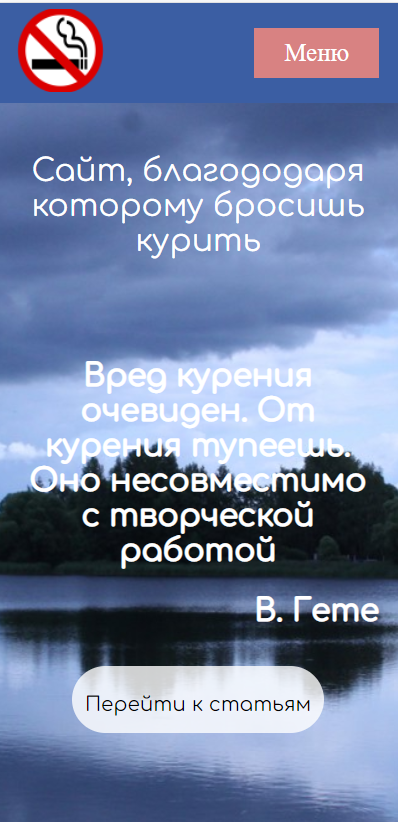


Рисунок 4.а – Размер экрана 320px

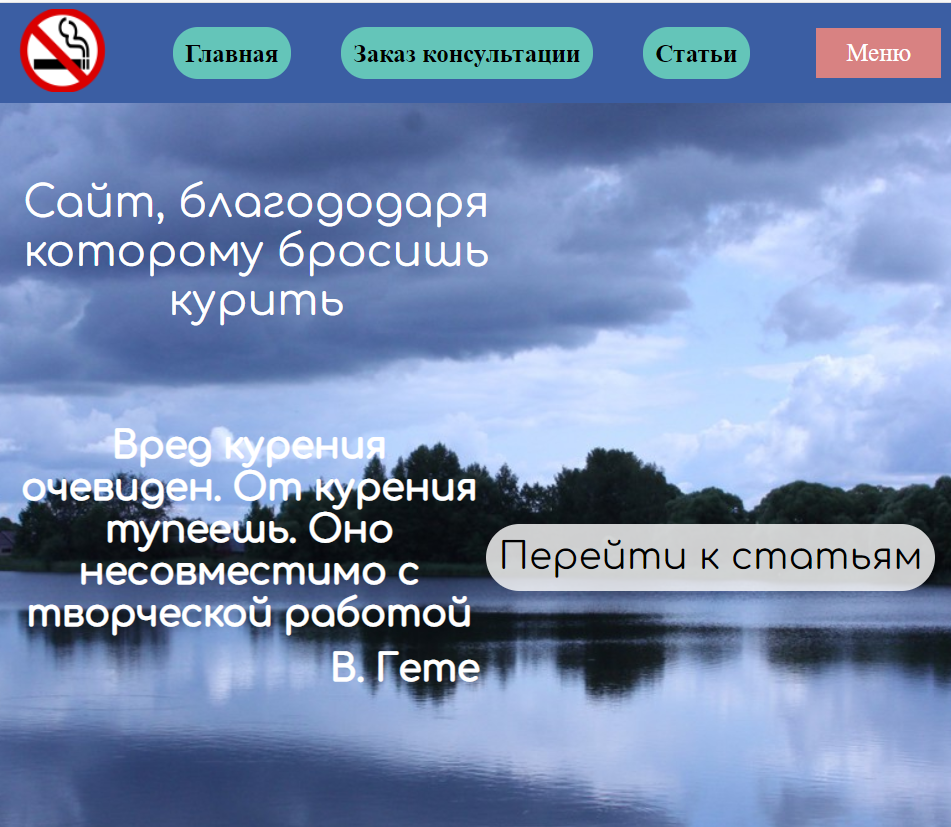


Рисунок 4.б – Размер экрана 728px

Как можно заметить документ будет изменён при различных размерах экрана.

4.2 Кроссбраузерность веб-сайта

Кроссбраузерность – это способность веб-ресурса отображаться одинаково и работать во всех популярных браузерах, без перебоев в функционировании и ошибок в верстке, а также с одинаково корректной читабельностью контента.

У каждого браузера есть свои определенные свойства, функции. При проверке веб-сайта в браузерах Microsoft Edge и Opera GX не было выявлено каких либо нарушений, сбоев.

4.3 Руководство пользователя

На всех страницах пользователь может увидеть вложенное «меню-бургер», при открытии которого перед ним открывается ряд кнопок с полезными функциями. Так же посетитель видит большие кнопки с надписью, при нажатии на которые можно перейти на другие страницы.

В разделе «Статьи» пользователь видит содержимое статей. Так же он видит кнопки. При нажатии на них посетитель перейдёт на главную страницу и на форму заказа консультации (рис. 4.1(а), 4.1(б)).

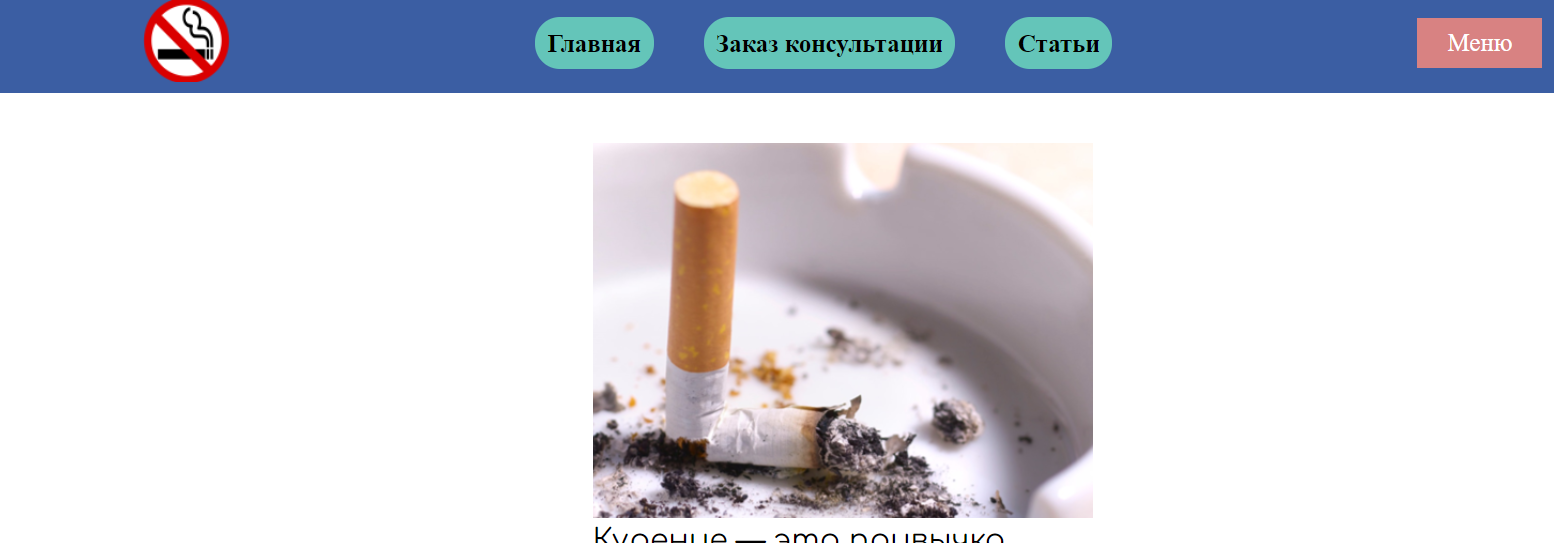


Рисунок 4.1(а) – Страница «Статьи»

На данном рисунке мы можем увидеть страницу «Статьи», на которой расположены кнопки.

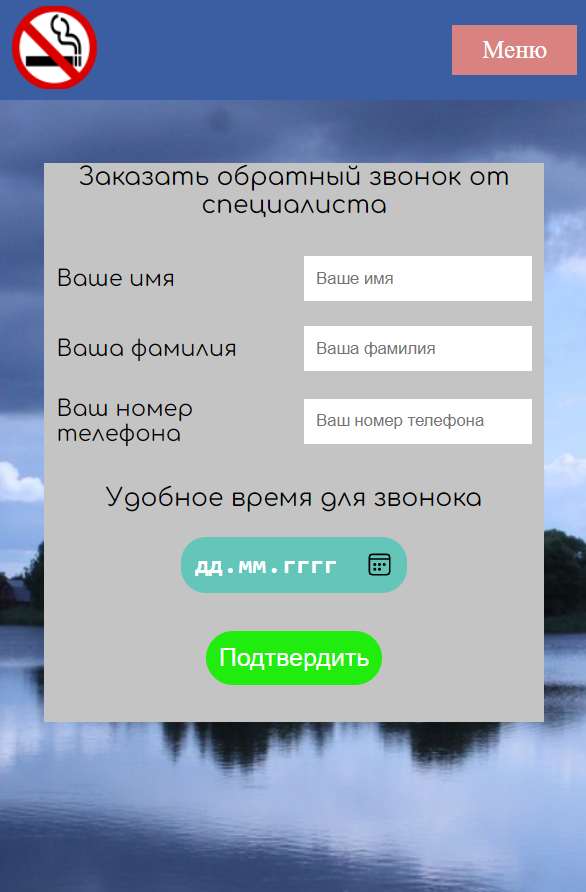


Рисунок 4.1(б) – Страница «Заказ консультации»

На данном рисунке мы можем заметить страницу «Заказ консультации», на которой есть форма для заказа консультации.

В «Подвале» каждой страницы при использовании версии для телефона, пользователь видит кнопки, которые при нажатии откроют страницы «Главная», «Заказ консультации», «Статьи».

4.4 Выводы

В результате работы над данным разделом мы научились пользоваться адаптивной вёрсткой, благодаря чему наш веб-сайт стал доступен на разных устройствах. В дополнение к этому мы проверили кроссбраузерность и написали руководство для пользователя.

Заключение

Целью курсового проекта была разработка веб-сайта «О вреде курения». Который предоставляет информацию о курении и позволяет заказать консультацию от специалиста.

Для создания макетов и прототипов сайта использовалась программа «Figma». Figma — онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени.

Результатом курсового проекта стал готовый к работе веб-сайт, в котором размещена информация о вреде курения. В дальнейшем возможно расширение информации, функционала.

Список литературы

1. Тэги, CSS [Электронный ресурс] / Справочный Режим доступа: **<http://htmlbook.ru/css/font-face>**

2. XML, JavaScript [Электронный ресурс] / Справочный Режим доступа: http://www.codenet.ru/webmast/xml/xslt/6.php

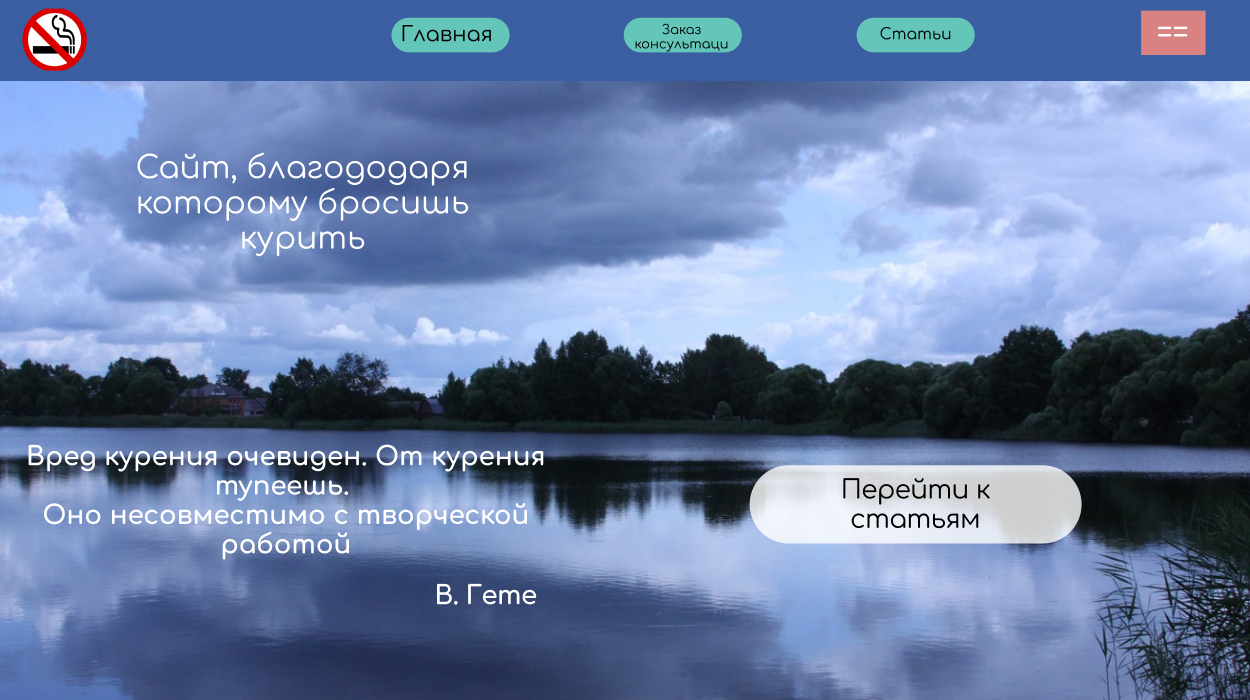
3. Figma [Электронный ресурс] / Справочный Режим доступа: <https://medium.com/slashdesigner/figma-guide-5235b8a8ab4f>

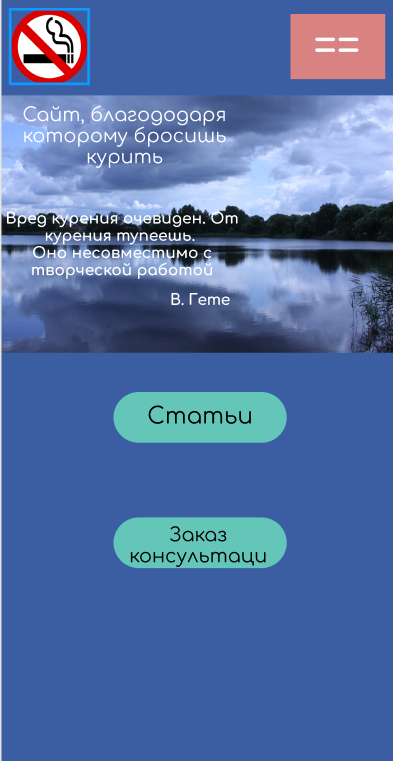
4. Аналог – niimed.ru [Электронный ресурс] – <https://niimed.ru>

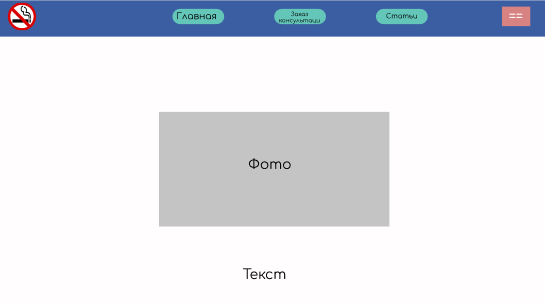
5. Аналог – 10gpd.by [Электронный ресурс] - https://www.10gdp.by

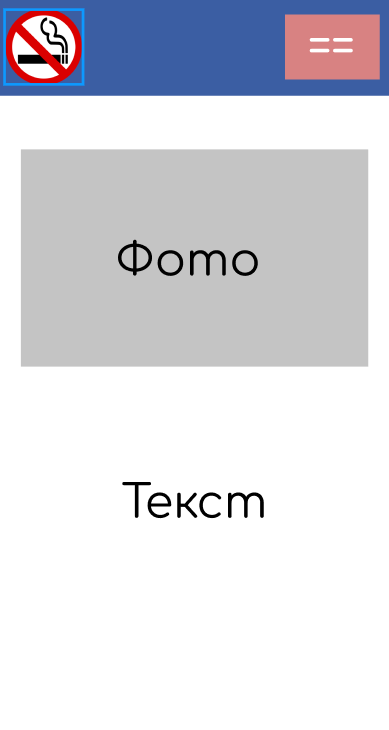
Приложение А

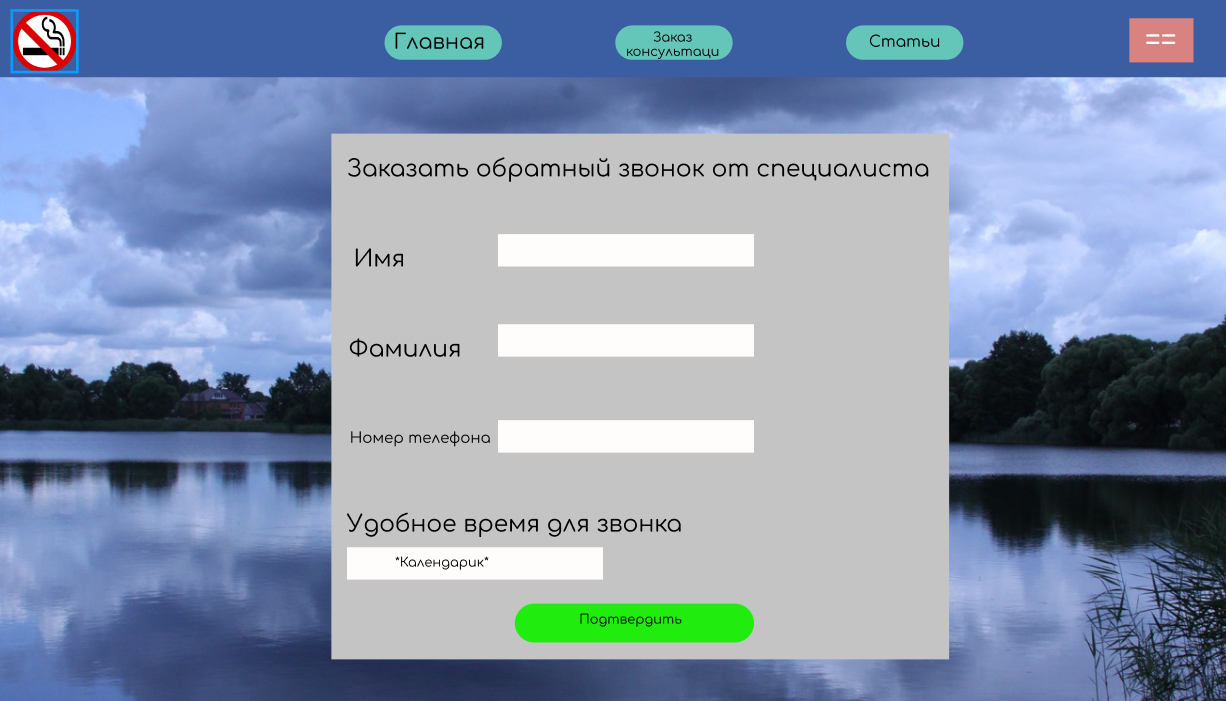
Прототипы страниц и макет

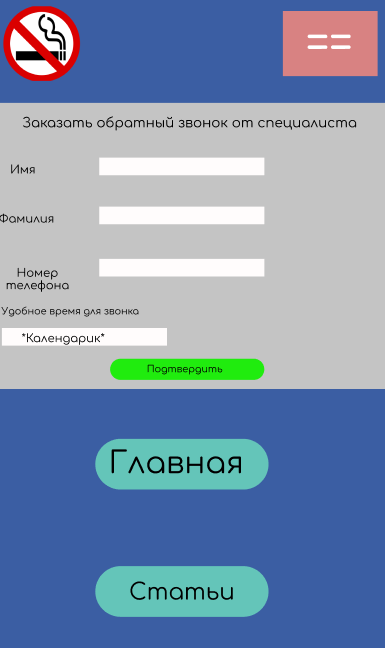


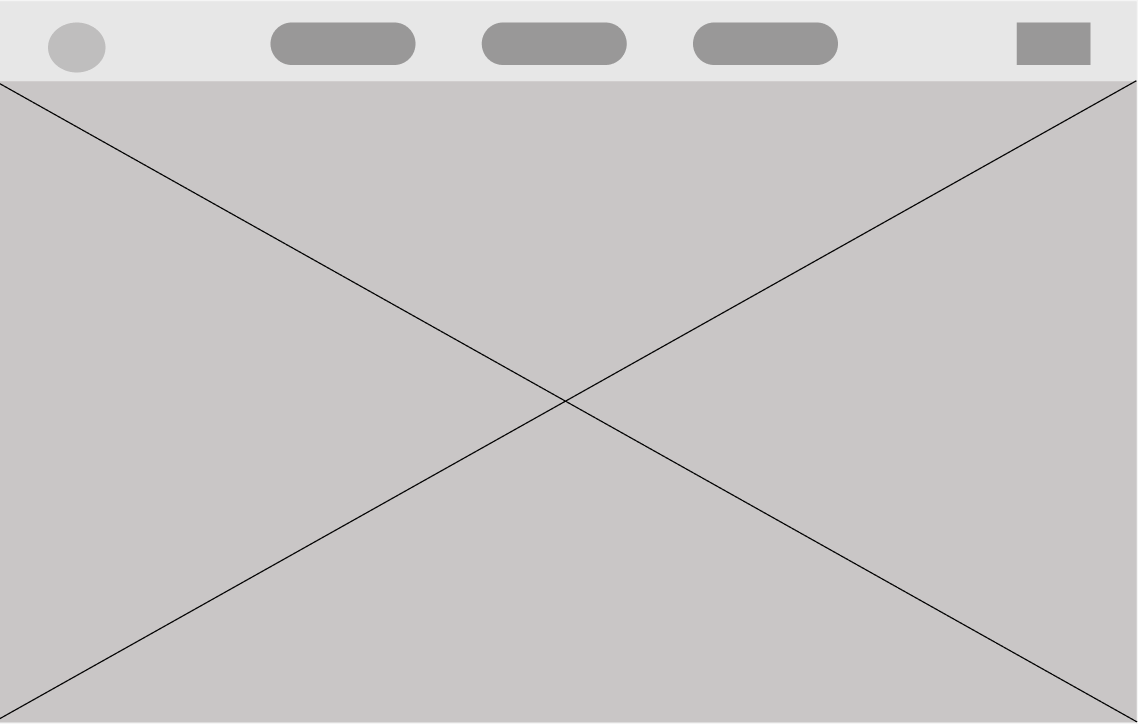
















Приложение Б

Листинг CSS

|  |
| --- |
| @font-face {  font-family: "Comfortaa";  src: url("Font/Comfortaa-VariableFont\_wght.ttf");  }  \* {  margin: 0;  padding: 0;  box-sizing: border-box;  }  #black{  color: red;  }  .wrapper {  max-width: 1150px;  margin: 0 auto;  padding: 0 15px;  }  ul {  list-style: none;  }  a {  text-decoration: none;  color: white;  }  .header {  background: #3b5ea3;  }  .header .wrapper {  height: 80px;  display: flex;  align-items: center;  justify-content: space-between;  }  .header .wrapper .burger {  font-size: 20px;  cursor: pointer;  background: #d88282;  height: 40px;  width: 100px;  z-index: 100;  display: flex;  align-items: center;  justify-content: center;  padding: 4px;  color: white;  }  .header .wrapper .nav ul {  display: flex;  gap: 40px;  }  .header .wrapper .nav ul li a {  border-radius: 20px;  color: black;  background-color: #64c5b9;  padding: 10px;  font-size: 20px;  font-weight: bold;  transition: all 0.3s ease-in-out;  }  .header .wrapper .nav ul li a:hover {  color: white;  }  .header .wrapper .logo {  width: 70px;  height: 70px;  }  .header .wrapper .logo a img {  max-width: 100px;  width: 100%;  }  .main {  background-image: url("../IMG/main.svg");  background-repeat: no-repeat;  background-size: cover;  background-position: center;  height: 100vh;  }  .main .wrapper {  display: flex;  gap: 80px;  flex-direction: column;  }  .main .wrapper .description {  margin-top: 60px;  max-width: 380px;  align-items: center;  font-family: "Comfortaa";  font-style: normal;  font-weight: 400;  font-size: 35px;  text-align: center;  color: #ffffff;  }  .main .wrapper .block\_down {  display: flex;  }  .main .wrapper .block\_down .quote {  font-family: "Comfortaa";  font-style: normal;  font-weight: 700;  font-size: 30px;  text-align: center;  color: #ffffff;  max-width: 380px;  flex: 1;  gap: 10px;  display: flex;  flex-direction: column;  }  .main .wrapper .block\_down .quote .author {  margin-left: auto;  }  .main .wrapper .block\_down .btn\_block {  flex: 1;  display: flex;  justify-content: center;  align-items: center;  }  .main .wrapper .block\_down .btn\_block .btn {  padding: 10px;  background: rgba(255, 255, 255, 0.8);  border-radius: 60px;  font-family: "Comfortaa";  font-style: normal;  font-weight: 400;  font-size: 30px;  border: none;  outline: none;  cursor: pointer;  }  .main .wrapper .block\_down .btn\_block .btn a {  color: black;  }  .menu {  background-color: #3b5ea3;  width: 100%;  height: 100vh;  position: fixed;  top: 0;  left: 0;  transform: translate(100%, 0);  transition: transform 0.3s ease-in-out;  }  .menu.active {  transform: translate(0, 0);  }  .menu ul {  margin-top: 90px;  display: flex;  gap: 40px;  flex-direction: column;  text-align: center;  }  .menu ul li {  margin: 0 auto;  font-size: 20px;  cursor: pointer;  width: 100%;  max-width: 300px;  background: #fff2f2;  padding: 10px;  }  .navs {  display: none;  padding: 30px 0;  }  .navs .wrapper .nav {  margin: 0 auto;  text-align: center;  }  .navs .wrapper .nav ul {  display: flex;  flex-direction: column;  gap: 40px;  }  .navs .wrapper .nav ul li a {  border-radius: 20px;  color: black;  background-color: #64c5b9;  padding: 10px;  font-size: 20px;  font-weight: bold;  transition: all 0.3s ease-in-out;  }  .navs .wrapper .nav ul li a:hover {  color: white;  }  @media (max-width: 725px) {  .wrapper {  max-width: 600px;  }  .main .wrapper .description {  margin-top: 60px;  max-width: 100%;  align-items: center;  font-family: "Comfortaa";  font-style: normal;  font-weight: 400;  font-size: 35px;  text-align: center;  color: #ffffff;  }  .main .wrapper .block\_down {  display: flex;  flex-direction: column;  gap: 30px;  }  .main .wrapper .block\_down .quote {  font-family: "Comfortaa";  font-style: normal;  font-weight: 700;  font-size: 30px;  text-align: center;  color: #ffffff;  max-width: 380px;  flex: 1;  gap: 20px;  display: flex;  flex-direction: column;  margin: 0 auto;  }  .header .wrapper .nav {  display: none;  }  .navs {  background: #3b5ea3;  display: block;  }  }  @media (max-width: 420px) {  .main .wrapper .block\_down .btn\_block .btn a {  font-size: 16px;  }  .main .wrapper .description {  margin-top: 40px;  font-size: 25px;  }  .main .wrapper .block\_down .quote {  font-family: "Comfortaa";  font-style: normal;  font-weight: 700;  font-size: 25px;  text-align: center;  color: #ffffff;  max-width: 380px;  flex: 1;  gap: 20px;  display: flex;  flex-direction: column;  margin: 0 auto;  }  }/\*# sourceMappingURL=main.css.map \*/ |

Приложение В

Листинг Java Script

|  |
| --- |
| let burger = document.querySelector('.burger');  let menu = document.querySelector('.menu');  burger.addEventListener('click', () => {  menu.classList.toggle('active');  }); |

Листинг подключения XML

|  |
| --- |
| var xmlhttp = new XMLHttpRequest();  var path = "../XML/data.xml";  xmlhttp.open("GET", path, true);  xmlhttp.send();  xmlhttp.onreadystatechange = function () {  if (this.readyState == 4 && this.status == 200) {  var xmlDoc = this.responseXML;  var x = xmlDoc.getElementsByTagName("article");  for (i = 0; i < x.length; i++) {  let description = x[i].getElementsByTagName("description")[0].childNodes[0].nodeValue;  let img = x[i].getElementsByTagName("img")[0].childNodes[0].nodeValue;  var div = `  <div class="block\_article">  <div class="block\_img">  <img src=${img} alt="main\_img">  </div>  <div class="block\_text">  ${description}  </div>  </div>`  document.querySelector(".articles").innerHTML += div;  }  }  } |