Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Пояснительная записка**

по курсовой работе:

«Web-сайт Personal Library»

Выполнил:

студент 2 курса 7 группы

специальности ПОИБМС

Комарова Е.И.

Руководитель:

доцент кафедры ИСИТ

Жиляк Н.А.

Оценка:

Дата защиты: 15.12.2018

Минск 2018

Содержание

[Введение 3](#_Toc532357677)

[1. Теоретическая часть 4](#_Toc532357678)

[1.1 Современные веб-технологии 4](#_Toc532357679)

[1.2 Выбор метода решения 5](#_Toc532357680)

[2. Практическая часть 6](#_Toc532357681)

[2.1 Техническое задание 6](#_Toc532357682)

[2.1.1 Назначение, цель и задачи создания сайта 6](#_Toc532357683)

[2.1.2 Структура и описание сайта 6](#_Toc532357684)

[2.1.3 Навигация 7](#_Toc532357685)

[2.1.4 Оформление 8](#_Toc532357686)

[2.2. Верстка 9](#_Toc532357687)

[2.3 Программирование 11](#_Toc532357688)

[3. Описание разработки программного продукта 12](#_Toc532357689)

[Заключение 14](#_Toc532357690)

# **Введение**

Веб-проект представляет собой процесс разработки и создания веб-сайта, деятельности в сети, что направлены на заранее определенную цель. Сеть может быть как доступной для всех, например, в Интернете, так и только для определенных людей, например, интранет. Целью веб-проектов является передача статического и динамического контента – как напрямую конечным пользователям, так и косвенно через различные интерфейсы. Веб-проекты основаны на технологии TCP/IP (Transfer Control Protocol / Internet Protocol) и касаются передачи статического и динамического содержимого.

Веб-проект включает в себя множество аспектов, включая программирование и сопутствующую разработку программного обеспечения, веб-бизнес, веб-сервер и сетевое администрирование, хостинг, графику/дизайн, разработку и администрирование баз данных, создание интерфейсов, управление проектами и обеспечение качества, поисковую оптимизацию, ведение данных в системах управления / редактирования контента и многое другое.

Программирование для веб-проекта может осуществляться с использованием одного или нескольких языков разметки (таких как HTML или XML), языков сценариев (например, JavaScript, Perl, PHP) или даже более сложных языков программирования (таких как C / C ++, C # или Java).

При размещении веб-проекта основными задачами являются предоставление необходимой аппаратной и программной инфраструктуры и обеспечение максимально возможного уровня доступности и надежности. Графический/веб-дизайн для веб-проектов должен обеспечивать высокое качество использования для лиц, взаимодействующих с веб-сайтом.

Определение целей создания сайта, круга функций, которые он должен выполнять, необходимо для лучшего понимания истинных задач, которые собираются решать с помощью сайта его владельцы. Правильно, конкретно поставленная цель позволяет сделать сайт эффективным инструментом для заказчика, минимизировать затраты на создание сайта. От продуманности формулировки этой задачи зависит эффективность работы сайта в долгосрочной перспективе.

Основной моей целью является размещение полезной для саморазвития коллекции книг и статей в сети Интернет. Для достижения этой цели я ставлю перед собой следующие задачи: разработка структуры будущего веб-сайта, его создание с помощью языка разметки HTML и языка сценариев JavaScript, заполнение созданного сайта контентом.

# **Теоретическая часть**

## **1.1 Современные веб-технологии**

Для того, чтобы стать полноценным веб-разработчиком, мы должны овладеть многими навыками. Начинающим часто бывает нелегко найти правильный путь обучения и получить быстрый результат. На первый взгляд, может быть непросто понять, что нужно выучить и как все сходится в конце.

Этот раздел даст частичный обзор технологий, которые можно изучить, чтобы стать полноценным разработчиком в 2018 году и далее. Я начну рассматривать основы и языки программирования, а затем перейду к более сложным темам, таким как фреймворки и дополнительные инструменты.

Лучший способ начать свою карьеру в веб-разработке - хорошо освоить HTML и CSS, которые являются основой всего остального.

Язык гипертекстовой разметки (HTML) необходим для каждого сайта. Используя HTML, мы определяем и структурируем контент сайта, используя простой синтаксис разметки.

Каскадные таблицы стилей (CSS) – это простой способ добавить стили к нашим сайтам (например, добавить шрифты, цвета, макет и т. д.).

Поняв HTML, мы обязательно должны перейти к CSS и убедиться, что понимаем основы. HTML и CSS – это две основы, которые должен знать каждый веб-разработчик.

JavaScript является необходимым строительным блоком для добавления динамических функций на наш сайт. Язык JavaScript является реализацией базовой спецификации ECMAScript. Изучение JavaScript с нуля является важной вехой в начале карьеры веб-разработки. Для многих сред на внешнем интерфейсе (Angular, React, Vue.js) и на серверной стороне (Node.js) знание JavaScript является обязательным условием.

Основы языка JavaScript просты в освоении. Однако по мере развития спецификации ECMAScript в язык добавляются все новые и новые расширенные функции. Например, в 2017 году были добавлены такие функции, как асинхронные функции, общая память и атомарные операции.

Bootstrap сегодня является бесспорным лидером среди доступных интерфейсных сред. Учитывая его огромную популярность, которая продолжает расти с каждым днем, можно быть уверенным, что этот замечательный инструментарий не подведет на пути к созданию успешных веб-сайтов.

Поддерживает последние версии Firefox, Chrome, Safari, IE810-11-Microsoft Edge. Главная сила Bootstrap – его огромная популярность. Технически он не обязательно лучше, но Bootstrap предлагает гораздо больше ресурсов (статей и учебных пособий, сторонних плагинов и расширений).

## **1.2 Выбор метода решения**

Заменив устройства ПК, мобильные телефоны становятся самыми популярными устройствами для доступа в Интернет. Таким образом, веб-дизайнеры столкнулись с проблемой, как представить одну и ту же веб-страницу на устройствах разных размеров? Решением является создание адаптивного сайта. В этом разделе будут представлены некоторые базовые знания для реализации адаптивного веб-дизайна.

Bootstrap – это самая популярная в мире платформа для разработки веб-сайтов для мобильных устройств на HTML, CSS и JavaScript. Поэтому мы будем его использовать для курсового проекта – веб-сайта, который работает на разных устройствах и браузерах.

Также необходимо разработать композиционное решение страниц сайта

Основные законы композиции: цельность и единство, равновесие, соподчинение.

1) Цельность. Благодаря соблюдению этого закона произведение воспринимается как единое неделимое целое, а не как сумма разрозненных элементов. Композиция выступает как система внутренних связей, объединяющая все компоненты форм и содержаний в единое целое. В композиции все элементы приводятся к гармоничной упорядоченности.

2) Равновесие. Это такое состояние композиции, при котором все элементы сбалансированы между собой. Уравновешенные части целого приобретают зрительную устойчивость

3) Соподчинение и равноценность элементов. Соподчинение -это выделение центра композиции (доминанты), которому подчиняются все остальные элементы (причем, не просто подчиняются, а усиливают его значимость), т.е в композиции возникает иерархия.

# **Практическая часть**

## **2.1 Техническое задание**

### **2.1.1 Назначение, цель и задачи создания сайта**

Основным назначением сайта является размещение полезной для саморазвития коллекции книг и статей в сети Интернет.

Целевая и возрастная аудитория сайта – лица, которые увлечены литературой, а также студенты БГТУ. Предполагаемое возрастное ядро аудитории от 16 лет.

Основной задачей сайта является предоставление пользователям доступа к книгам:

• Художественная литература

• Научная литература

• Издания БГТУ.

А также к различным статья, связанных с темой литературы.

### **2.1.2 Структура и описание сайта**

Описание разделов сайта.

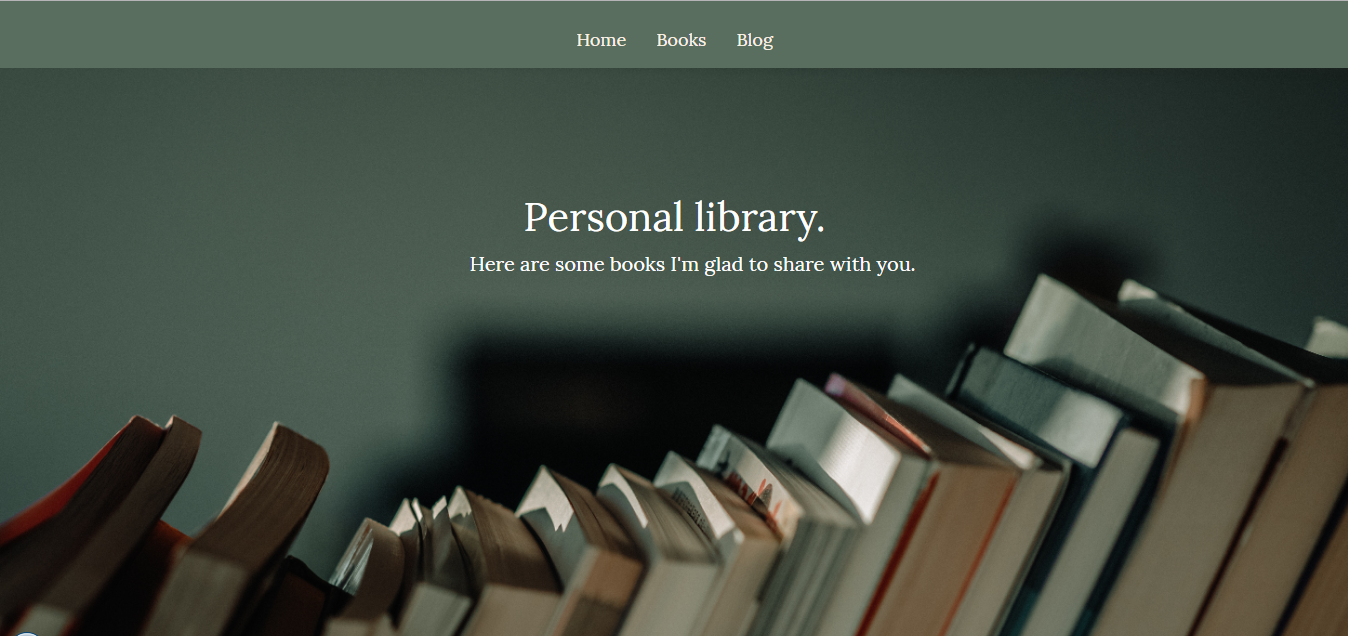
«Home»: этот раздел является основной точкой входа на сайт. (*Рисунок 1. Раздел )*

Рисунок 1. Раздел Home

«Books»: раздел, содержащий книги в формате pdf (*Рисунок 2. Раздел )*.

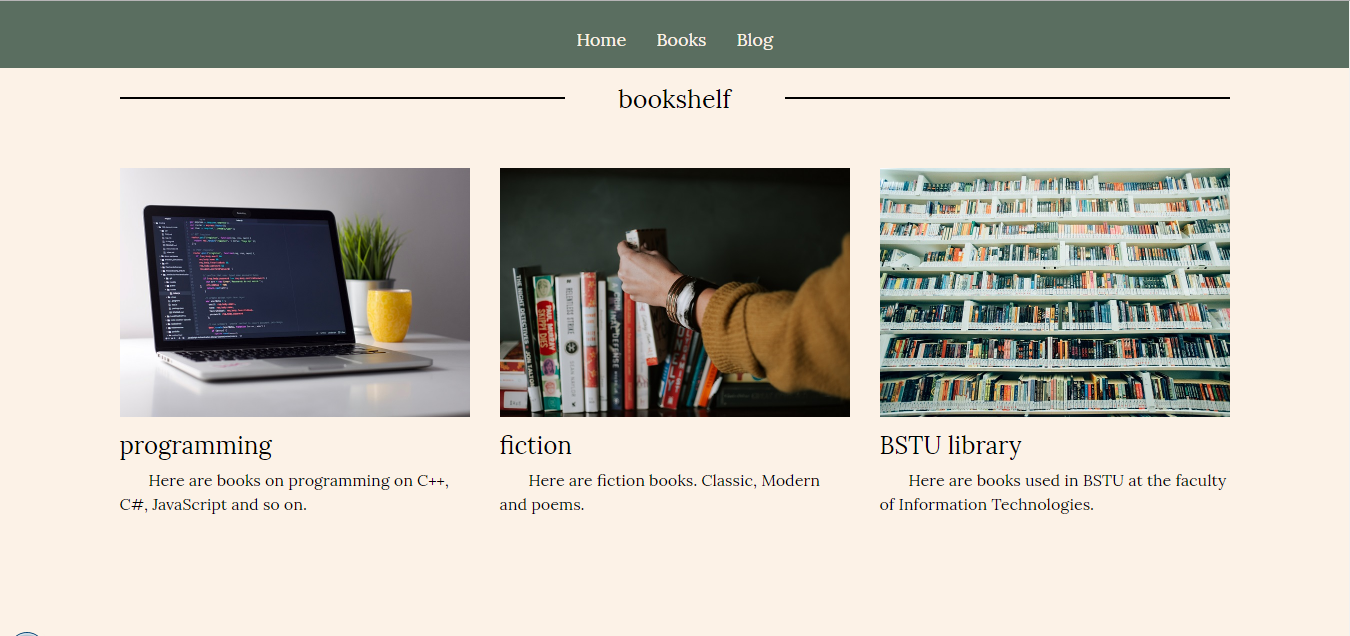


Рисунок 2. Раздел Books

«Blog»: раздел, содержащий информацию о различных сферах литературы *(Рисунок 3. Раздел Blog*)

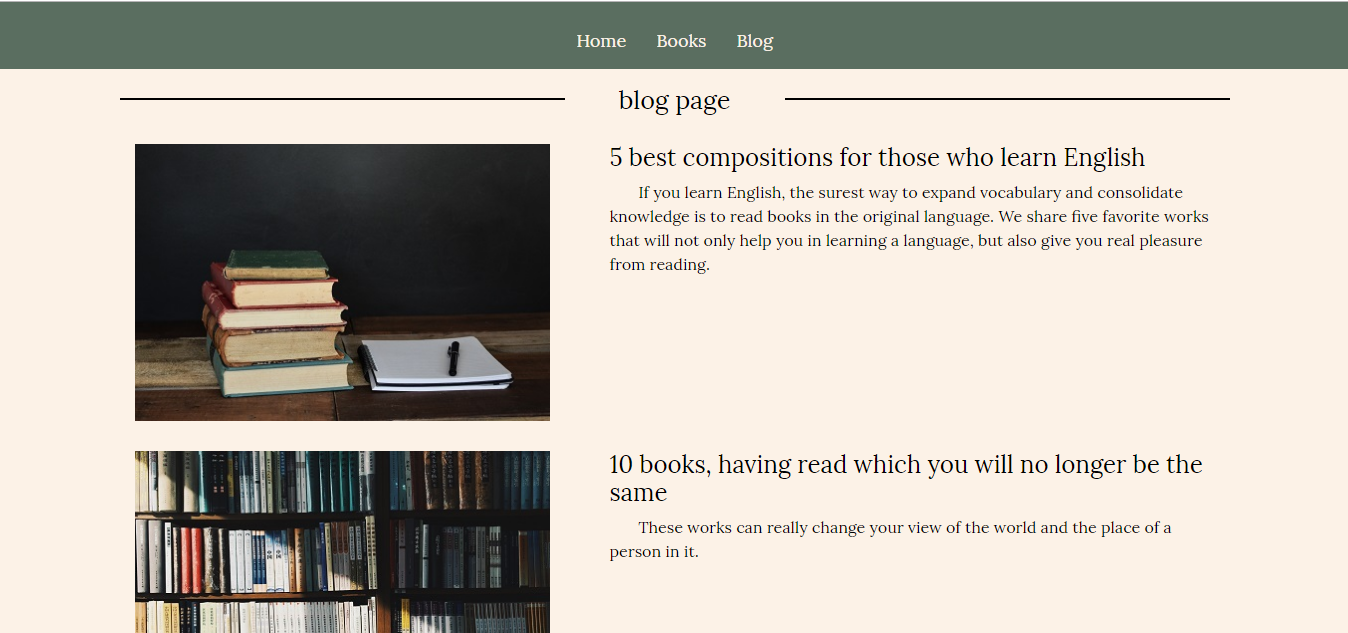


Рисунок 3. Раздел Blog

### **2.1.3 Навигация**

Все средства навигации делятся по отношению к сайту на внешние и внутренние. Внешние средства реализованы в виде кнопок браузера, управляющих переходами по сайту.

Внутреннюю навигацию можно разделить на основную и вспомогательную. Центральный элемент основной навигации - это панель или меню переходов. Основная навигация обычно представлена структурными ссылками, ведущими на разделы сайта. К вспомогательной навигации относят ссылки типа «на уровень вверх», «на стартовую» и т.д.

Желательно, чтобы структурные ссылки отличались от вспомогательных и перекрестных ссылок, размещенных в тексте. На большинстве сайтов они выглядят одинаково. Поэтому часто бывает непонятно, ведет ссылка на целый раздел или на единственную страницу. С проблемой определения содержимого таких ссылок часто сталкиваются пользователи download-менеджеров, когда необходимо указывать, до какого уровня вложенности необходимо скачивать целый сайт или отдельные его разделы.

Качественная вспомогательная навигация должна включать ссылки, ведущие к предыдущему разделу, к странице с оглавлением раздела и на главную страницу сайта.

Навигация сайта будет устроена таким образом, чтобы с каждого радела сайта можно было попасть в любой другой раздел (что поможет пользователем не запутаться в навигации).

### **2.1.4 Оформление**

Сайт будет выполнен в спокойном неброском фоне, который не будет отвлекать посетителей от просмотра содержания сайта.

Все картинки взяты с сайта Unsplash — ресурс с фото под лицензией CCO.

При создании предполагается использовать шрифты от Google Fonts такие как Lora, иконки от Fonts Awesome.

## **2.2. Верстка**

Делать вёрстку под мобильные телефоны и планшеты как было сказано ранее помогает Bootstrap. С блочной структурой Bootstrap мы можем существенно сэкономить время разработки.

Блочная система (Grid System) Bootstrap позволяет нам горизонтально разделить страницу на 12 столбцов. Если использовать все 12 частей нам нет необходимости, такие блоки можно легко группировать, создавая более широкие столбцы. Они будут состоять из двух, трех и так далее частей страницы, разделённой на 12 частей. Это около 8.3% на столбец.

Разделить таким образом можно как страницу, так и любой блочный элемент, размеры блоков будут рассчитаны как часть родительского элемента.

Вот пример того, как можно разделить страницу при помощи Bootstrap. *(Рисунок 4. 12 частей страницы)*

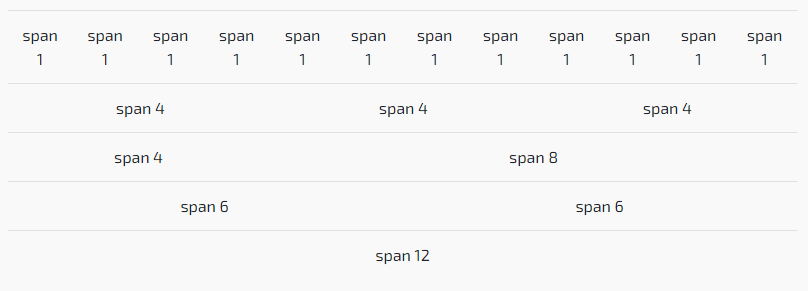


Рисунок 4. 12 частей страницы

Блочная структура в Bootstrap рассчитана также на мобильные устройства и планшеты. Блоки будут автоматически перестраиваться в зависимости от размера экрана.

В блочной верстке Bootstrap есть четыре основных класса ячеек:

* xs (extra small) — для мобильных телефонов, до 767 пикс. включительно
* sm (small) — для планшетов, размер экрана 768—991 пикс.
* md (middle) — для десктопов, 992—1199 пикс.
* lg (large) — для больших экранов, от 1200 пикс.

Прелесть верстки на Bootstrap в том, что эти классы мы можем комбинировать, чтобы создавать адаптивную верстку и точно знать, как макет будет отображаться на том или ином размере экрана.

Базовая структура макета представлена на рассмотрение *(Рисунок 5. Базовая структура макета)*

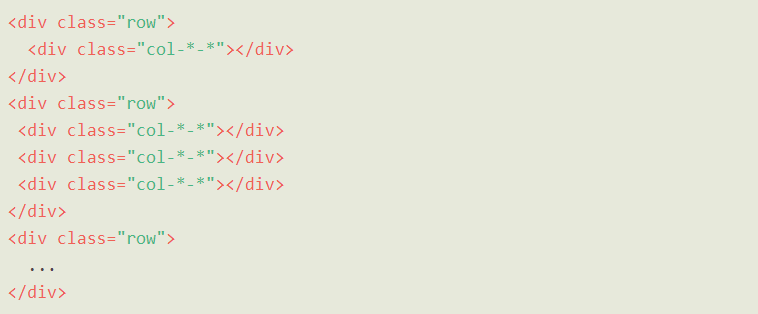
**

Рисунок 5. Базовая структура макета

Нужно обратить внимание на то, что сумма ячеек для каждого типа (sm, xs, md, lg) в ячейке .col-\*-\* не должна превышать 12 частей для каждой строки.

## **2.3 Программирование**

JavaScript является наиболее широко распространенным языком в мире. Я использую его для плавного перехода после нажатия на якорную ссылку.

Как известно, по нажатию на якорную ссылку нас перекидывает на ту часть страницы с которой эта ссылка связана. Сам переход происходит мгновенно и этот нюанс я исправила.

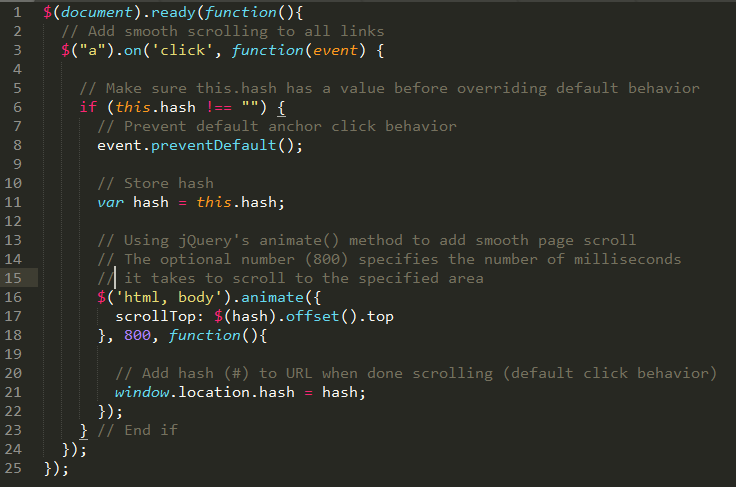
Для того чтобы реализовать плавный переход после нажатия по ссылке, необходимо было написать небольшой скрипт, который отследил бы момент нажатия, выяснил на какой блок ссылается якорь и плавно прокрутил страницу до нужного места (*Рисунок 6. Код JavaScript*)

Рисунок 6. Код JavaScript

Для начала используем метод jQuery .ready() для того, чтобы сделать функцию доступной для выполнения после того, как документ загружен.

Затем отменяем стандартную обработку нажатия по ссылке

event.preventDefault();

После объявляем переменную, в которой будет храниться значение ссылки.

Обозначаем два селектора (head, body) и используем функцию animate, а затем добавляем ссылку в адрес страницы в окне браузера.

# **3. Описание разработки программного продукта**

При открытии сайта отображается главная страница, имеющая название index.html и заголовок Personal Library *(Рисунок 7. Открытие сайта)*

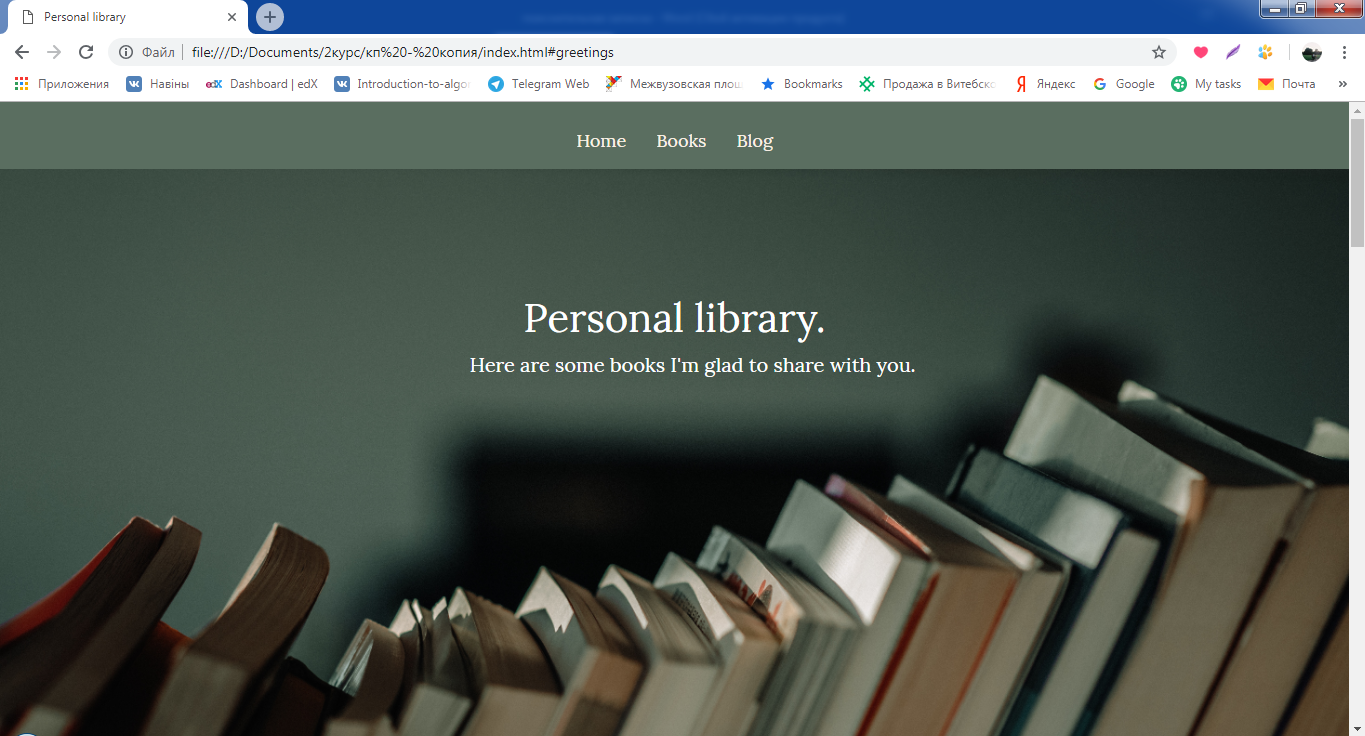


Рисунок 7. Открытие сайта

Шапка сайта является фиксированной, поэтому навигация удобна и доступна из любого местоположения.

В секции Books содержится 3 раздела, содержащих книги в формате pdf, доступных для скачивания. *(Рисунок 8. Секция Books)*

Рисунок 8. Секция Books

Содержится иконка, ведущая к предыдущему разделу.

В секции Blog содержаться различные статьи, доступ к предыдущему разделу также доступен *(Рисунок 9. Секция Blog)*

Рисунок 9. Секция Blog

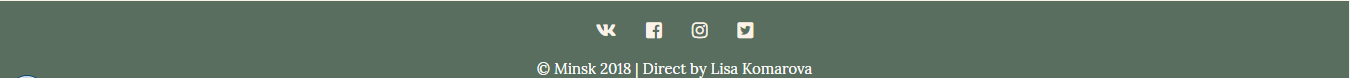
Теперь рассмотрим footer *(Рисунок 10. Раздел footer)*

Рисунок 10. Раздел footer

Данный раздел содержит информацию об авторе сайта, а также ссылки на социальные сети автора.

# **Заключение**

После проведения работы я создала сайт, который отвечает тем требованиям, которые передо мной поставили. Основная задача при создании сайта выполнена - я донесла информацию до пользователя.

В процессе создания сайта использовался язык разметки гипертекста HTML, язык программирования JavaScript, библиотека JavaScript – jQuery, фреймворк Bootstrap. Менялся дизайн, расположение контента, но в итоге работа получилась.

# **Список использованных источников**

1. Черепанова И. uCoz. Создание сайтов.- – Москва: Эксмо-Пресс, 2010.

2. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. Бен Фрейн. 2017.

3. HTML5 для веб-дизайнеров. Автор: Джереми Кит/ 2015.

4. JavaScript. Подробное руководство/Дэвид Флэнаган. 2013. Издательство : Символ-Плюс.

5. jQuery. Подробное руководство по продвинутому JavaScript. Беэр Бибо, Иегуда Кац. 2011. Символ-Плюс

6. Большая книга CSS3. Дэвид Сойер Макфарланд. 2016. Издательство: Питер.