Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Пояснительная записка**

по курсовой работе:

«Транспорт Минска»

Выполнил: студент 2 курса

8 группы

специальности ПОИБМС

Никонович А.В.

Руководитель: ассистент кафедры ИСИТ

Пушило А.Н.

Оценка:

Дата защиты:

Минск 2018

**Содержание**

[Введение 3](#_Toc438085192)

[1 Технические средства разработки веб-сайта 4](#_Toc438085193)

[1.1 Техническое задание](#_Toc438085194) 6

[2 Описание разработки программного продукта](#_Toc438085195) 9

[2.1 Структура веб-страниц 12](#_Toc438085196)

[Заключение 1](#_Toc438085197)4

[Список использованных источников 1](#_Toc438085198)5

[Приложение 1 1](#_Toc438085199)6

[Приложение 2](#_Toc438085200) 17

# Введение

Всемирная паутина (англ. World Wide Web) — распределённая система, предоставляющая доступ к связанным между собой документам, расположенным на различных компьютерах, подключённых к Интернету. Для обозначения Всемирной паутины также используют слово веб (англ. web «паутина») и аббревиатуру WWW.

Всемирную паутину образуют сотни миллионов веб-серверов. Большинство ресурсов Всемирной паутины основаны на технологии гипертекста. Гипертекстовые документы, размещаемые во Всемирной паутине, называются веб-страницами. Несколько веб-страниц, объединённых общей темой, дизайном, а также связанных между собой ссылками и обычно находящихся на одном и том же веб-сервере, называются веб-сайтом. Для загрузки и просмотра веб-страниц используются специальные программы — браузеры (англ. browser).

Всемирная паутина вызвала настоящую революцию в информационных технологиях и взрыв в развитии Интернета. В повседневной речи, говоря об Интернете, часто имеют в виду именно Всемирную паутину. Однако важно понимать, что это не одно и то же.

Задача данной курсовой работы состоит в создании web-сайта с использованием гипертекстового языка разметки HTML, каскадных таблицей стилей CSS, языка XML и языка программирования JavaScript.

Тема курсовой работы — информационный сайт об стране Норвегия

Можно выделить несколько задач моей работы:

* обеспечение удобного и эффективного интерфейса пользователя;
* разработка дизайна, делающего нахождение пользователя на сайте максимально комфортным;
* обеспечение удобного просмотра всех локаций;
* обеспечение удобного и эффективного интерфейса перемещения по локациям;
* интуитивно-понятный интерфейс для пользователей.
* обобщение многочисленной информации, сведение её к максимально простому виду.

Таким образом, главная задача — привлечь посетителей удобством, простотой и доступностью, информативностью.

**1. Технические средства разработки веб-сайта**

Сайт разрабатывался в текстовом редакторе Sublime Text 3. Страницы сайта могут просматриваться в таких браузерах как Internet Explorer, Microsoft Edge, Opera, Mozilla Firefox, Google Chrome.

Sublime Text 3 — свободный текстовый редактор с открытым исходным кодом с подсветкой синтаксиса большого количества языков программирования и разметки. Его особенностями так же являются автозавершение тегов, удобная навигация, возможность скачать модификации и настроить редактор под себя. Базовая функциональность программы может быть расширена как за счёт плагинов, так и сторонних модулей, таких как компиляторы и препроцессоры. Вносим какие-либо изменения в код в редакторе — в окне браузера автоматически отображаются изменения;

* Подсветка синтаксиса;
* подсказки при редактировании CSS, JS и HTML-файлов;
* Сворачивание кода;
* Автодополнение и автоматическое закрытие скобок и тэгов (если активировано);
* Закладки;
* Регулярные выражения для поиска и замены;
* Сравнение файлов;
* Менеджер проектов;
* Карта документа;
* Командная палитра;
* API плагинов на Python;
* Резервное копирование сохраняемых файлов (включается в настройках);
* Поддержка и конвертирование кодировок ANSI, UTF-8 и UCS-2;

Именно огромное количество плагинов позволяет превратить данный текстовый редактор в мощный комбайн для WEB-разработки.

Задача данной курсовой работы состоит в создании web-сайта с использованием гипертекстового языка разметки HTML, каскадных таблицей стилей CSS, языка XML и языка программирования JavaScript.

Подробнее про каждый из них:

Язык HTML позволяет размечать в тексте:

* смысловую роль текстового блока (например: логическое ударение, заголовок (от первого до шестого уровня), параграф, пункт списка и др.), который обрабатывается браузером в соответствии со смыслом (например, в голосовых браузерах — изменение интонации , в графических — выделением курсивом, и т. п.) или настройками пользователя;
* гипертекстовые ссылки, которые значительно упрощают чтение множества связанных документов, ибо позволяют запросить документ с адресом, указанным в коде ссылки, простым щелчком мыши;
* гарнитуру, кегль, начертание, цвет шрифта для визуального вывода.
* специальные символы (выходящие за рамки ASCII символы пунктуации, математические символы, греческие и готические буквы, стрелки и т. п.);
* формы для введения пользователем данных, которые позднее подвергаются обработке. Формы и другую информацию можно обрабатывать с помощью специальных серверных программ (например, на языках PHP или Perl);
* открытие мультимедийных файлов, выводимых как непосредственно браузером (например, изображения в форматах JPEG, GIF или PNG; аудиофайлы MIDI и др.), так и внешними приложениями, «встраиваемыми» в окно браузера (Flash-анимация, Java-апплеты и прочее);

JavaScript – это [динамический язык программирования](https://developer.mozilla.org/ru/docs/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C/Dynamic_programming_language), который применяется к [HTML](https://developer.mozilla.org/ru/docs/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C/HTML) документу, и может обеспечить динамическую интерактивность на веб-сайтах. Так же его называют языком сценариев или языком скриптов. Сценарии JavaScript выполняются на компьютере пользователя, поэтому в браузерах предусмотрена возможность отключения выполнения сценариев JavaScript. JavaScript не предназначен для создания автономных приложений. Программа на JavaScript встраивается непосредственно в исходный текст HTML-документа и интерпретируется браузером по мере загрузки этого документа. С помощью JavaScript можно динамически изменять текст загружаемого HTML-документа и реагировать на события, связанные с действиями посетителя или изменениями состояния документа, или окна.

JQuery – это библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript, CSS и HTML. Данная библиотека позволяет изменять содержимое HTML-документов путем манипулирования объектами модели, создаваемой браузерами в процессе обработки HTML-кода.

CSS (Cascading Style Sheets, каскадные таблицы стилей) – это язык описания внешнего вида документа, написанного с использованием языка разметки. CSS позволяет представлять один и тот же документ в различных стилях. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

XML (англ. Extensible Markup Language) — расширяемый язык разметки. Разрабатывался как язык с максимально простым формальным синтаксисом, удобный для создания и обработки документов как программами, так и человеком, с подчёркиванием нацеленности на использование в сети Интернет. Язык называется расширяемый, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка. Сочетание простого формального синтаксиса, удобства для человека, расширяемости, а также базирования на кодировках Юникод для представления содержания документов привело к широкому использованию XML.

## 1.1 Техническое задание

**1.1.1 Цель проекта**

Цель проекта — создание лаконичного, удобного сайта для получения всей необходимой информации. Привлечение посетителей простотой, информативностью. Познакомить пользователя с транспортом города Минска, его историей, маршрутной сеткой, подвижным составом, заинтересовать посетителя данной тематикой. Сделать поездку в транспорте более увлекательной.

**1.1.2 Дизайн сайта**

Дизайн включает в себя цветовое оформление, наличие или отсутствие логотипа. Целевой аудиторией сайта будут являться потенциальные посетители, которые уже пользовались тем или иными сайтами, которые носит информационный характер и хотят узнать еще больше.

Дизайн сайта играет немаловажную роль в создании самого сайта.

Оформление должно быть тематическим, чтобы посетитель, попав на сайт, сразу понял, что представляет из себя предоставляемый ресурс.

Для оформления моего сайта я выбрал блочный стиль, т.к. он набирает популярность и является очень удобным в понимании. Сайт состоит из разнообразных цветов, чтобы поднять пользователю настроение и сделать информацию запоминающейся.  Фон должен быть не сильно ярким, слегка бледным и подходить тематике сайта. Именно поэтому я решил в качестве цвета фона выбрать небесно-голубой, потому что он благотворно влияет на психику и является цветом герба города Минска.

**1.1.4 Архитектура сайта**

Архитектура сайта — структура страниц и программной части сайта. Это понятие, которое неразрывно связано с контентом и оптимизацией. Под архитектурой понимают систему организации файлов, страниц и просто информации, содержащейся на сайте. Если такой системы нет или она была составлена неправильно еще при создании ресурса, в дальнейшем это может серьезно усложнить жизнь посетителям сайта.

Главная цель архитектуры сайта — это создание ресурса, который бы позволил пользователю быстро найти необходимую информацию. Даже если посетитель впервые заглянул на сайт, он должен в течение нескольких минут освоиться и найти нужное. Больше времени на ваш ресурс посетитель вряд ли потратит: он не будет загружать десятки страниц, чтобы найти нужную информацию, и просто-напросто уйдет на другой сайт. Ведь гораздо проще и быстрее набрать интересующий запрос в поисковой системе, чем искать что-то на одном сайте.

Без грамотно выстроенной архитектуры сайта невозможна скорая и полная индексация поисковыми роботами. Архитектурой сайта необходимо заниматься еще на этапе его проектирования, т.к. существующий ресурс с уже имеющейся неграмотной архитектурой переделывать гораздо сложнее и дольше.

Создавать сайт без создания архитектуры, что строить дом без чертежа. Прежде, чем начать писать код, необходимо понять, из чего будет состоять сайт и как лучше распределить информацию.

**1.1.5 Навигация**

Навигация — это возможность прохода куда-либо. Применительно к сайтам, это означает способ перехода от одной страницы к другой и все промежуточные действия, включая мыслительные процессы в это время. Идеальная навигация по сайту должна быть элементарно простой и моментально быстрой.

Удобство и простота использования сайта заключается в его незамысловатой навигации. Система навигации сайта — система программных и визуальных средств, с помощью которых Посетитель может перемещаться по данному сайту и находить нужную ему информацию. Поскольку веб-сайт представляет собой, упрощенно говоря, набор документов, связанных гиперссылками, то под термином «навигация» мы понимаем в основном то, как организована на страницах сайта работа с гиперссылками.

Каждый сайт может содержать несколько видов навигации. При этом нет обязательных требований к месту расположения, но есть устоявшиеся представления о том, где обычно находится тот или иной блок навигации. Расположение блоков управления как правило подчинено обычной логике — подобное с подобным, близкое по смыслу — близко по месту.

На главной странице размещена информация о контенте сайта и его создателе, оформленная в виде меню, реализованном с помощью CSS:

* Главная;
* История.

На данной странице расположен своеобразный слайдер, сделанный на CSS3, представляет фотографии транспорта различных эпох размером 720х360:

* Подвижной состав.

Здесь расположено подробное описание транспортных средств с фотографиями.

* Маршруты

Тут находится информация о маршрутной сетке, реализованная в табличной форме, а также анимация из SVG-картинок.

* Тарифы

Переход входом на эту страницу пользователя попросят оплатить проезд, а далее он сможет рассчитать стоимость проездного документа при помощи JS-формы.

.

На каждой странице есть переход на главную страницу и на другие разделы, представленные на сайте.

**1.1.6 Технические требования**

**Сайт реализован при помощи языка HTML,** каскадных таблицей стилей CSS, языка программирования JavaScript.

Язык сайта: русский.

Разрешение экрана: 1920x1080 пкс.

Сайт просматривается в браузерах: Internet Explorer 7.0 и выше, или Firefox 3.5 и выше, или Opera 9.5 и выше, или Chrome 2 и выше.

Время загрузки каждой страницы до 2 секунд.

Размер шрифтов должен обеспечивать удобство восприятия текста при минимально допустимом размере экрана.

# 2 Описание разработки программного продукта

## 2.1 Структура веб-страниц

Структура сайта:

Сайт состоит из следующих разделов:

1. Главная страница;
2. История;
3. Подвижной состав;
4. Маршруты;
5. Тарифы

Пользовательский интерфейс сайта должен обеспечивать понятное представление размещенной на нем информации, быстрый и логичный переход к разделам и страницам. Система должна обеспечивать навигацию по всем доступным пользователю ресурсам и отображать соответствующую информацию.

При выборе какого-либо из пунктов меню пользователем должна загружаться соответствующая ему информационная страница. Графический материал представлен в формате JPG, PNG. Тестовая информация соответствует разделам сайта.

**2.1.1 Веб-страница «Главная»**

Документ начинается тегом <html>,  заканчивается тегом</html>.

Далее следует элемент <head> </head> (рисунок 1.1).

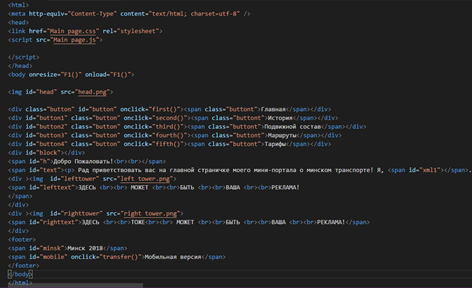


Рисунок 2.1 - Содержимое главной страницы

Элемент head является заголовком, в котором содержится служебная информация о документе: название (<title> </title> ― обязательный элемент тега <head>, текст между этими тегами отображается в строке браузера), кодировка документа (<meta/>), ссылка на таблицу стилей и на JavaScript. Вся эта информация не отображается на странице и на внешний вид не влияет. Нужна она браузеру для предоставления сведений о документе.[[1](#_Список_использованных_источников)]

Данный блок находится на всех страницах сайта и остается неизменным.

В теге <body> </body> содержится:

1. Меню сайта (рисунок 1.2);

Рисунок 2.2 – Меню.

1. Приветственное послание с выводом ФИО из xml и анимация[[2](#_Список_использованных_источников)]

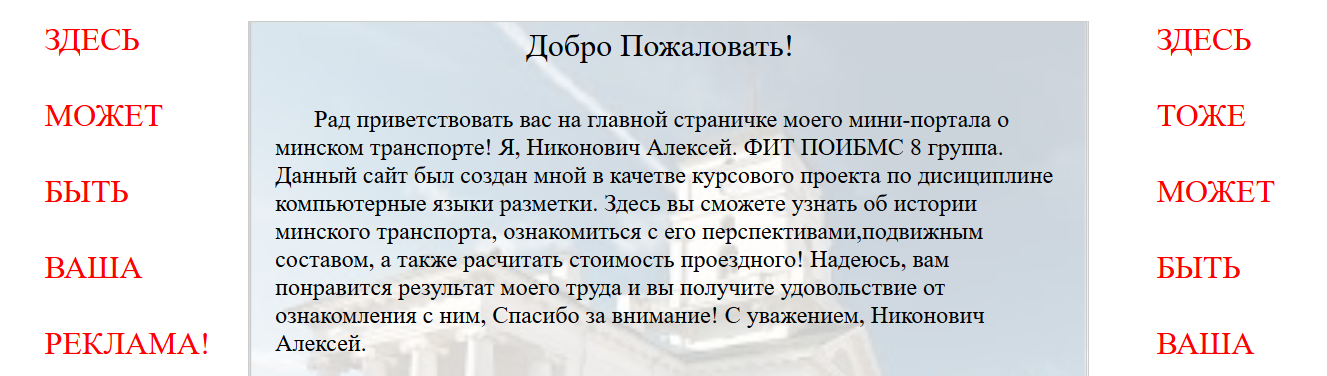


Рисунок 2.3 – Приветственное послание

1. Картинка над меню



Рисунок 2.4 – Картинка над меню

**2.1.2 Веб-страница «История»**

1. Данная страница содержит в себе слайдер, написанный с помощью CSS[[3](#_Список_использованных_источников)]

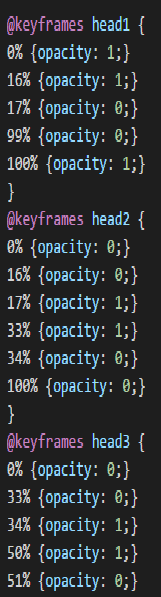


Рисунок 2.5 —код нужный для работы слайдера

1. Подробную информацию об истории транспорта

Рисунок 2.6 —слайдер

**2.1.3 Веб-страница «Подвижной состав»**

На этой странице находится подробная информация о подвижном составе транспорта Минска[[4](#_Список_использованных_источников)]



Рисунок 2.7 – Информация о подвижном составе

* + 1. **Веб-страница «Маршруты»**

1. Веб-страница содержит SVG-Анимацией

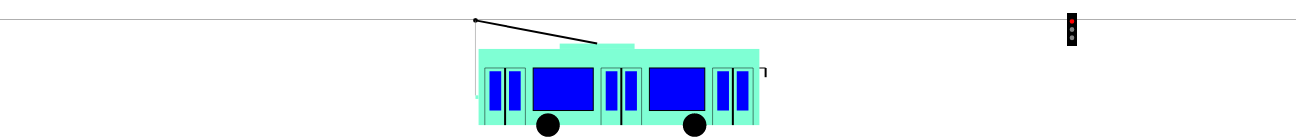


Рисунок 2.8 – SVG-Анимация

1. Таблицу с маршрутами



Рисунок 2.9 – Таблица с маршрутами

* + 1. **Веб-страница «Тарифы»**

На данной странице размещается JS-калькулятор расчета стоимости проездных документов.[[5](#_Список_использованных_источников)]



Рисунок 2.10 – JS-калькулятор

# Заключение

В ходе данной курсовой работы были изучены такие языка разметки веб – сайта, как HTML5, JavaScript, CSS3. Обоснована актуальность выбранной темы, разработана структура страниц сайта и продемонстрирована в виде иллюстраций, создано описание этапов разработки веб – сайта с описанием всех элементов и их содержимого. Была изучена литература и источники, включающие в себя информацию о средствах создания веб – сайта. В частности, были изучены и использованы такие возможности языков разметки, как адаптивность, svg-анимация, блочная вёрстка. Активно использовались CSS и JS анимации. Также в процессе создания курсового проекта на одной из страниц были применены формы. Также в ходе выполнеия проекта я научился логически соединять элементы вэб-cтраницы.

Были закреплены теоретические знания и практические навыки по дисциплине «Компьютерные языки разметки».

# Список использованных источников

1. Техника разметки текста с помощью HTML // [Электронный ресурс]. –2017. – Режим доступа: <http://htmlbook.ru/>.
2. Хуан Диего Гоше. HTML5. Для профессионалов / Хуан Диего Гоше. – Санкт-Петербург: Питер, 2014. – 306–360 с.
3. Дэвид Сойер Макфарланд. Большая книга CSS3 / Дэвид Сойер Макфарланд. – Санкт-Петербург: Питер, 2015. – 106–200 с.
4. Беэр Бибо, Иегуда Кац jQuery. Подробное руководство по продвинутому JavaScript / Беэр Бибо, Иегуда Кац. – Москва: Символ-Плюс, 2011. – 104 с.
5. JavaScript // [Электронный ресурс]. – 2007 – 2017. – Режим доступа: https://learn.javascript.ru/.
6. Google maps API // [Электронный ресурс].– Режим доступа: <https://developers.google.com/maps/?hl=ru>
7. Дэвид Флэнаган. JavaScript. Подробное руководство / Дэвид Флэнаган. – Москва: Символ-Плюс, 2013. – 208–247 с.

# Приложение 1. Стили меню на всех страницах

.button:hover{

opacity: 0.5;

}

.button {

border-style: double;

border-color: rgb(68, 0, 255);

background-color: rgb(0, 110, 255);

color: white;

text-align: center;

position: absolute;

}

.buttont {

color: white;

font-size: x-large;

}

#button {

border-color: rgb(255, 0, 0);

background-color: rgb(255, 72, 0);

}

# Приложение 2. Стили анимации для рекламы

#lefttower {

position: absolute;

animation-name: leftA;

animation-duration: 10s;

animation-iteration-count: infinite;

}

#righttower {

position: absolute;

animation-name: rightA;

animation-duration: 10s;

animation-iteration-count: infinite;

}

#lefttext {

position: absolute;

animation-name: leftT;

animation-duration: 10s;

animation-iteration-count: infinite;

font-size: xx-large;

color: red;

}

#righttext {

position: absolute;

font-size: xx-large;

animation-name: rightT;

animation-duration: 10s;

animation-iteration-count: infinite;

color: red;

}

@keyframes leftA {

0% {opacity: 1;}

33% {opacity: 1;}

34% {opacity: 0;}

100% {opacity: 0;}

}

@keyframes leftT {

0% {opacity: 0;}

33% {opacity: 0;}

34% {opacity: 1;}

100% {opacity: 1;}

}

@keyframes rightA {

0% {opacity: 1;}

67% {opacity: 1;}

68% {opacity: 0;}

100% {opacity: 0;}

}

@keyframes rightT {

0% {opacity: 0;}

67% {opacity: 0;}

68% {opacity: 1;}

100% {opacity: 1;}

}

# Приложение 3. Стили слайдера

#head1 {

position: absolute;

animation-name: head1;

animation-duration: 30s;

animation-iteration-count: infinite;

}

#head2 {

position: absolute;

animation-name: head2;

animation-duration: 30s;

animation-iteration-count: infinite;

}

#head3 {

position: absolute;

animation-name: head3;

animation-duration: 30s;

animation-iteration-count: infinite;

}

#head4 {

position: absolute;

animation-name: head4;

animation-duration: 30s;

animation-iteration-count: infinite;

}

#head5 {

position: absolute;

animation-name: head5;

animation-duration: 30s;

animation-iteration-count: infinite;

}

#head6 {

position: absolute;

animation-name: head6;

animation-duration: 30s;

animation-iteration-count: infinite;

}

@keyframes head1 {

0% {opacity: 1;}

16% {opacity: 1;}

17% {opacity: 0;}

99% {opacity: 0;}

100% {opacity: 1;}

}

@keyframes head2 {

0% {opacity: 0;}

16% {opacity: 0;}

17% {opacity: 1;}

33% {opacity: 1;}

34% {opacity: 0;}

100% {opacity: 0;}

}

@keyframes head3 {

0% {opacity: 0;}

33% {opacity: 0;}

34% {opacity: 1;}

50% {opacity: 1;}

51% {opacity: 0;}

100% {opacity: 0;}

}

@keyframes head4 {

0% {opacity: 0;}

50% {opacity: 0;}

51% {opacity: 1;}

67% {opacity: 1;}

68% {opacity: 0;}

100% {opacity: 0;}

}

@keyframes head5 {

0% {opacity: 0;}

67% {opacity: 0;}

68% {opacity: 1;}

85% {opacity: 1;}

86% {opacity: 0;}

100% {opacity: 0;}

}

@keyframes head6 {

0% {opacity: 0;}

85% {opacity: 0;}

86% {opacity: 1;}

100% {opacity: 1;}

}