

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

ОТЧЕТ

По лабораторной работе №1

«ОСНОВЫ *HTML* И *CSS*»

Выполнил

ст. гр. 220601
М. Ю. Шухман

Проверил

А. Л. Гончаревич

Минск 2025

СОДЕРЖАНИЕ

Введение	3
1 Постановка задачи.....	4
2 Теоретическая часть	5
3 Ход работы	7
Заключение	9

ВВЕДЕНИЕ

В рамках лабораторной работы исследуются основы языка гипертекстовой разметки *HTML* и языка описания внешнего вида *HTML*-документа *CSS*. Язык *HTML*, разработанный Тимом Бернерсом-Ли в 1986-1991 годах, продолжает развиваться и по сей день. Актуальной версией на настоящее время является *HTML5*, представленная в 2014 году. *HTML* представляет собой теговый язык разметки: начало и конец его элементов обозначаются специальными пометками, которые и называются теги. Благодаря своей стандартизации, *HTML* поддерживается большинством браузеров, что позволяет разработчикам разрабатывать одну версию сайта для всех устройств.

Как и *HTML*, *CSS* не является языком программирования. Он представляет собой язык таблицы стилей и позволяет применять стили выборочно к элементам в документах *HTML*. *CSS* взаимодействует с браузером, чтобы придать документу оформление. 1-я версия стандарта *CSS* опубликована 17 декабря 1996 года, и основным посылом являлось добавление оформления документа без программирования или сложной логики. *CSS* является одним из основных языков свободной веб-разработки, который стандартизован спецификацией *W3C*. Стандарт делится на уровни: *CSS1* в настоящее время устарел, *CSS2.1* рекомендован для применения, а *CSS3*, разбитый на более мелкие модули, развивается на пути стандартизации.

1 ПОСТАНОВКА ЗАДАЧИ

При создании веб-сайта необходимо использовать и освоить базовые навыки написания сайтов с применением языков разметки и стилизации, таких как *HTML* и *CSS*. Проектируемый сайт должен являться многостраничным, содержать не менее трёх полноценных страниц, каждая из которых должна иметь чёткую структуру, логическую взаимосвязь с другими страницами и удобную навигацию для пользователей.

Перед началом работы необходимо разработать концепцию и тематику веб-сайта, продумать его структуру, макет и элементы интерфейса. Затем приступить к созданию страниц, их разметке с помощью *HTML* и стилизации с применением *CSS*, обеспечивая привлекательный внешний вид, удобство использования и адаптивность на различных устройствах.

После завершения процесса разработки сайта его необходимо тщательно протестировать, чтобы убедиться в корректности работы всех функциональных элементов. Важно проверить правильность отображения контента, удобство навигации, соответствие дизайна заданной концепции, а также адаптивность сайта на экранах различных размеров, включая мобильные устройства и планшеты. После этого следует провести оптимизацию кода и проверить его на валидность, устранив возможные ошибки и недочёты.

В ходе разработки необходимо использовать систему управления версиями *Git*, что позволит вести контроль над процессом разработки, отслеживать изменения и эффективно работать в команде. Репозиторий проекта нужно разместить на платформе *GitHub* для хранения кода и удобного взаимодействия с другими разработчиками. После завершения всех этапов сайт необходимо опубликовать в сети, используя сервис *GitHub Pages*, чтобы обеспечить к нему доступ пользователей и продемонстрировать результат проделанной работы.

2 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Семантическая разметка в *HTML* помогает более точно описать контент, что улучшает его восприятие и обработку. Более конкретные теги помогают распознавать содержимое. Например, это такие теги как *<header>*, *<footer>*, *<nav>*.

Важно тестировать отображение страницы на различных видах устройств для того, чтобы убедиться, что сайт остается приспособленным и удобным для взаимодействия.

Адаптивность на различных видах устройств достигается разработкой различных стилей для одних и тех же элементов разметки при различных параметрах экрана. В этом помогают медиазапросы.

Медиазапросы – это инструмент в *CSS*, который позволяет изменять стили в зависимости от характеристик устройства, а именно размера экрана, плотности пикселей, ориентации и других. Они используются для создания адаптивного дизайна, обеспечивая корректное отображение на разных устройствах.

В *HTML* все элементы делятся на три основных типа:

- блочные (*block*);
- строчные (*inline*);
- строчно-блочные (*inline-block*).

Они различаются поведением в потоке документа и способом отображения.

Блочные элементы всегда начинаются с новой строки и занимают всю ширину родительского контейнера, им можно задавать ширину и высоту, поддерживают все свойства отступов (*margin*, *padding*).

Строчные элементы не начинаются с новой строки, а располагаются в одной строке с другими элементами, занимая столько места, сколько необходимо для содержимого, отступы применяются только по горизонтали.

Строчно-блочные элементы ведут себя как строчные при размещении, имеют задаваемые размеры и им можно задавать ширину и высоту.

Для реакции на действия пользователя можно использовать псевдоклассы. Псевдоклассы в *CSS* – это специальные ключевые слова, которые добавляются к селекторам и позволяют применять стили к элементам в зависимости от их состояния, позиции или взаимодействия с пользователем.

Относительное позиционирование — это метод размещения элементов на веб-странице, при котором элемент остается в обычном потоке документа, но может быть сдвинут относительно своего исходного положения с помощью свойств *top*, *right*, *bottom*, *left*. Его можно использовать для перемещения

элементов без упрощения структуры, создания вложенных абсолютно позиционированных элементов.

Для интеграции в проект изображений необходимо, чтобы они располагались в одной папке с проектом. Далее можно использовать изображения, указывая в качестве источника путь к файлу. Данный путь может быть абсолютным или относительным. Рекомендовано использовать относительные пути, чтобы ссылки не сбивались при переносе сайта в другое место.

В веб-разработке браузеры по умолчанию применяют свои встроенные стили к *HTML*-элементам. Это приводит к тому, что один и тот же сайт может выглядеть по-разному в разных браузерах. Чтобы избежать этих различий, используются сброс и нормализация стилей.

Сброс стилей заключается в сбросе всех основных встроенных стилей. Это позволяет полностью контролировать то, как будет выглядеть страница в различных браузерах. К примеру, можно убрать все отступы.

Нормализация не сбрасывает, а унифицирует стили, чтобы элементы выглядели одинаково во всех браузерах. Это позволяет поддерживать кроссбраузерные стили, но не удалять полезных свойств. Исправляет различия в стилях заголовков.

3 ХОД РАБОТЫ

Согласно заданию лабораторной работы, необходимо создать сайт для онлайн-курсов. В качестве тематики курсов выбраны курсы норвежского языка, предоставляемые онлайн-школой. Для ориентира при разработке выбран макет, структура которого ляжет в основу сайта. Данный шаблон предложен конструктором сайтов *uKit*. Шаблон располагается по ссылке: <https://ukit.com/websites/ru/education/website/norsk-new>. Главная страница веб-сайта имеет вид, представленный на рисунке 3.1.

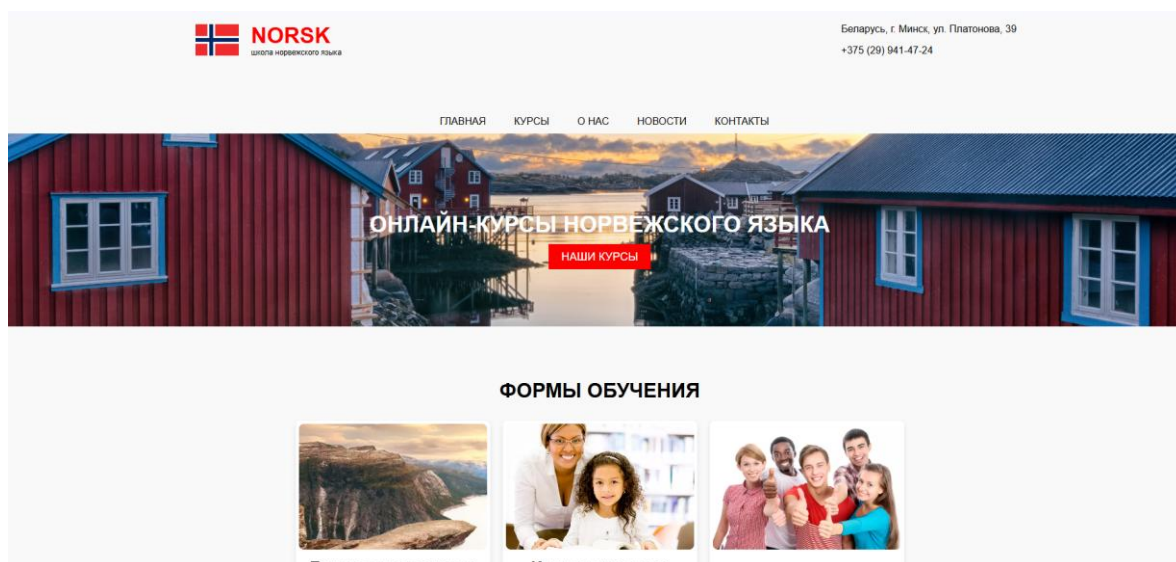


Рисунок 3.1 – Главная страница сайта

В структуре данной страницы можно выделить заголовок, секцию навигации, основное содержимое документа и подвал. Для заголовка, секции навигации и подвала использованы специализированные теги, введенные в *HTML5*.

Для размещения элементов в основном содержимом, а также для элементов навигации, использовался метод *flexbox*, предназначенный для размещения элементов в строках или столбцах. *Flexbox* позволяет удобно управлять расположением, порядком, размерами и отступами между элементами веб-страницы. Это и отличает его от другого компоновщика *Grid Layout*.

Flexbox подход упрощает вёрстку сайта, а также значительно увеличивает его адаптивность. Данный метод применён, например, для размещения карточек с формами обучения. При этом сами карточки являются *flex*-контейнерами, в которых элементы располагаются в столбец. Данный пример использования *flexbox* представлен на рисунке 3.2.

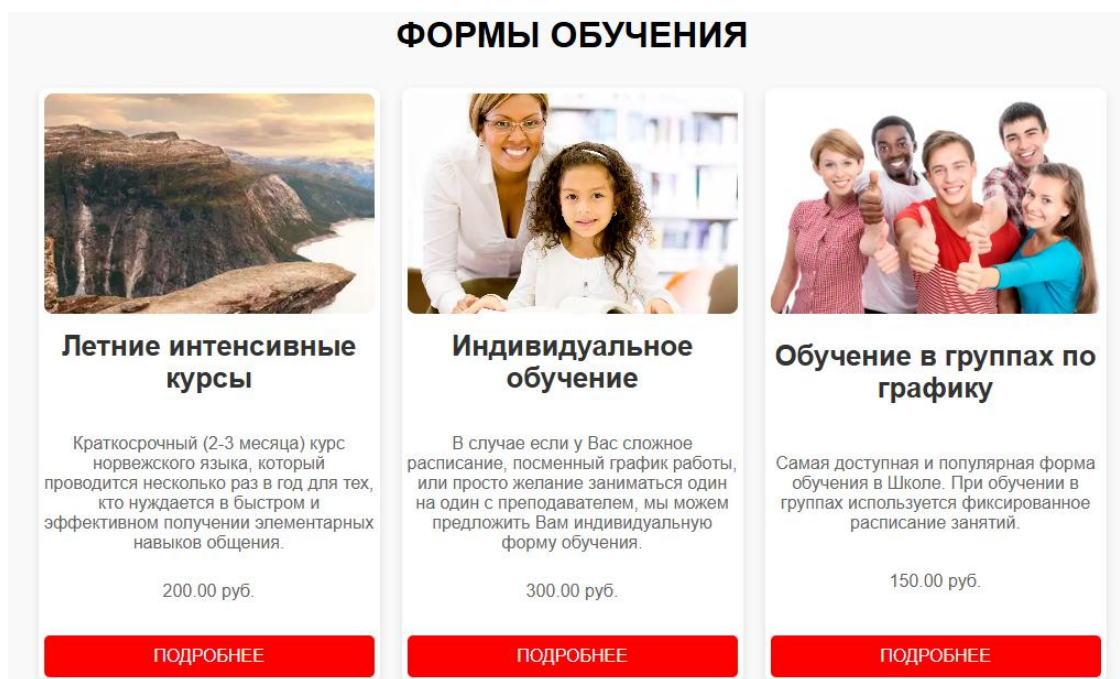


Рисунок 3.2 – Пример использования *flexbox*

Один из этапов разработки – это реализация адаптивности сайта для устройств с другими размерами экранов. При недостатке пространства для размещения элементов в строку они размещаются в несколько строк. Пример такого размещения элементов представлен на рисунке 3.3.

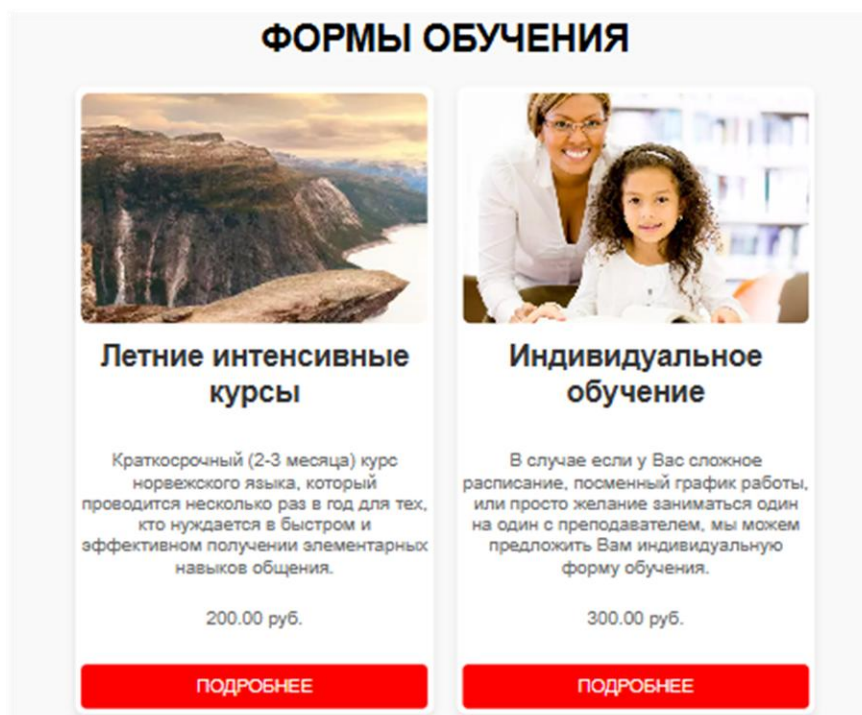


Рисунок 3.3 – Пример размещения элементов в несколько строк

Для устройств с ещё меньшим размером экрана, например, для небольших мобильных устройств, панель навигации скрывается в выпадающем меню и выводится по нажатию, при этом элементы навигации располагаются в столбец. Панель навигации в развёрнутом виде представлена на рисунке 3.4.

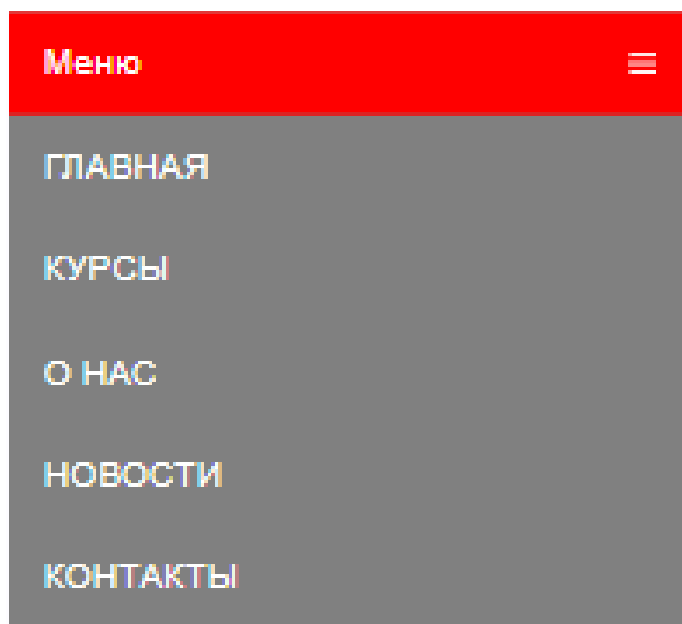


Рисунок 3.4 – Отображение страницы на устройстве с меньшим размером экрана

Сайт также содержит страницы «Контакты», «Курсы», «О нас», «Новости», «Пользовательское соглашение» и «Политика конфиденциальности». Страница «Контакты» представлена на рисунке 3.5.

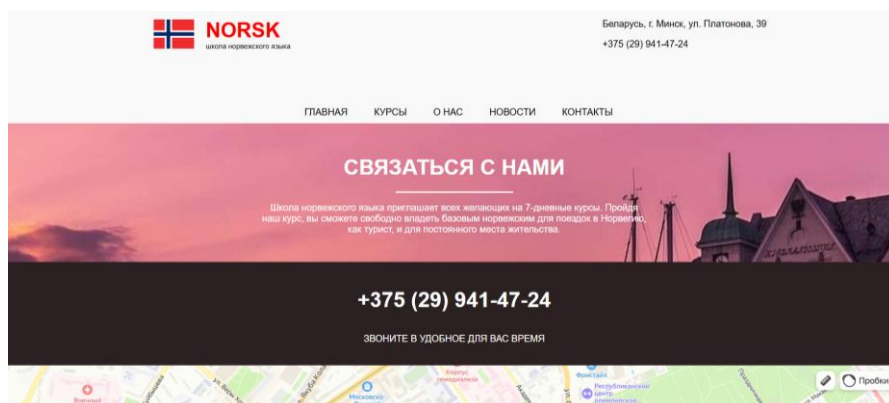


Рисунок 3.5 – Страница «Контакты»

На страницу «Контакты» добавлен виджет Яндекс-карт. Данный элемент более информативен, чем указание адреса текстом. Посетитель сайта

при взаимодействии с картой быстрее понимает, где находится данное место, что избавляет от необходимости отдельно заходить на сайт карт и искать указанный адрес там. Внешний вид виджета представлен на рисунке 3.6.

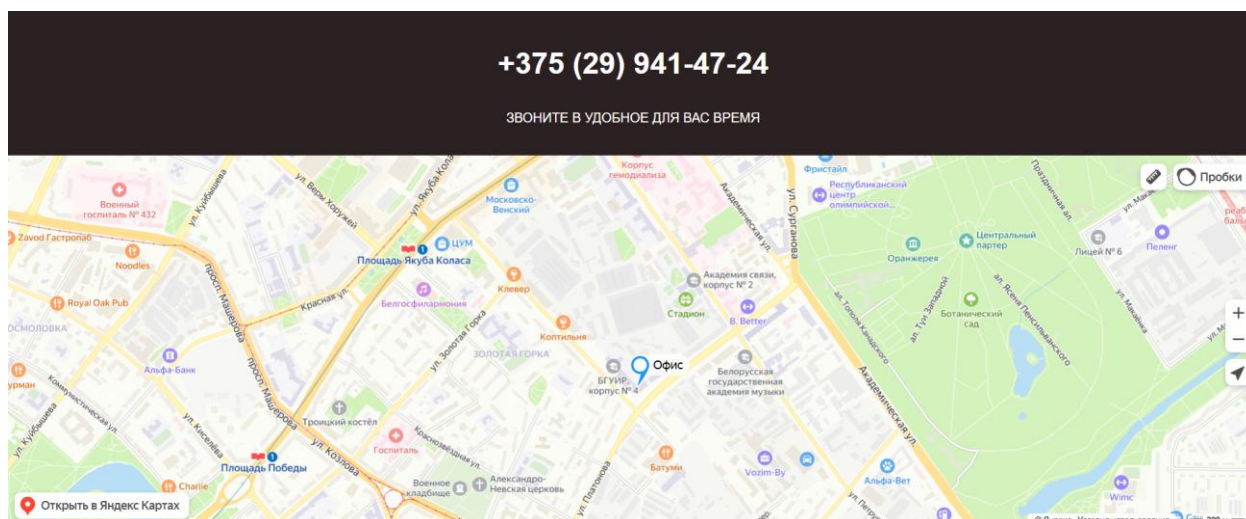


Рисунок 3.6 – Виджет карт

Страница «Курсы» содержит расширенную информацию о курсах, предоставляемых онлайн-школой и формах обучения. Данная страница представлена на рисунке 3.7.

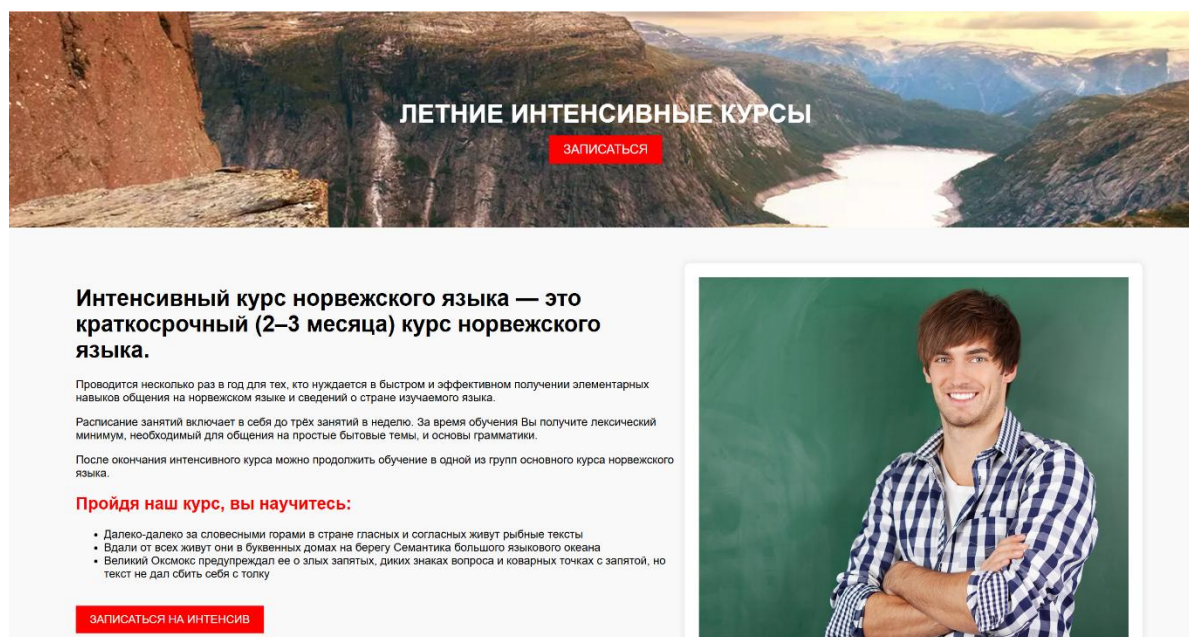


Рисунок 3.7 – Страница «Курсы»

Страница «О нас» содержит справочную информацию об онлайн-школе. Данная страница изображена на рисунке 3.8.

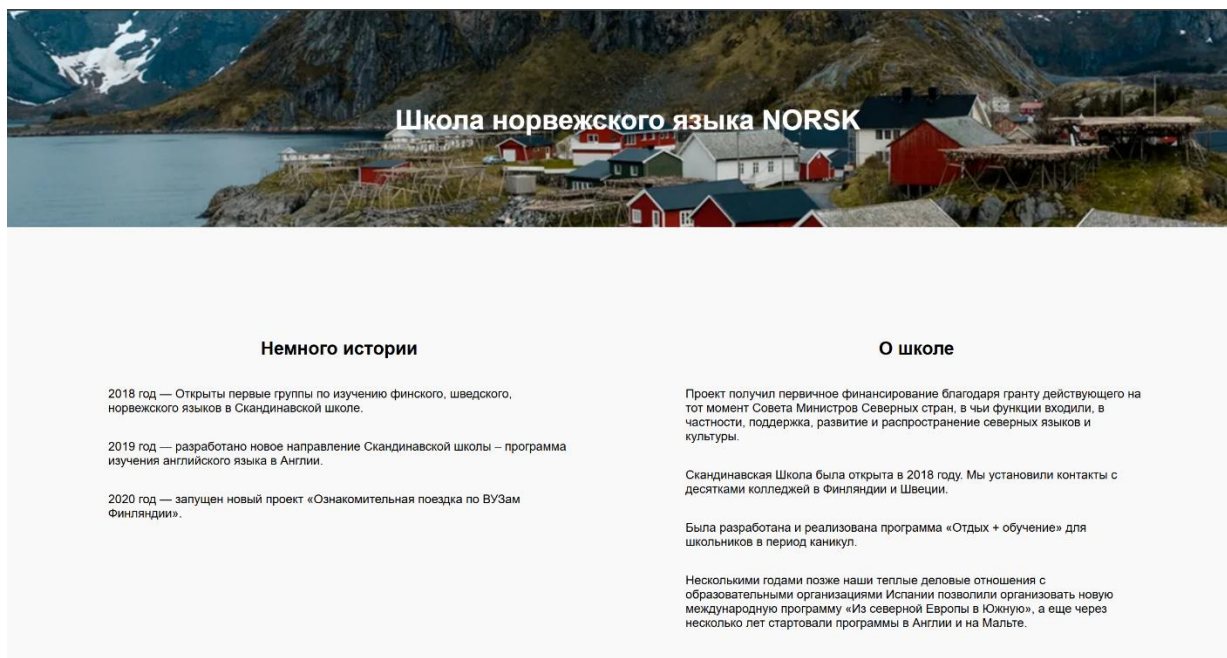


Рисунок 3.8 – Страница «О нас»

На странице «Новости» размещаются новости онлайн-школы. Данная страница предоставляет пользователям актуальную информацию о нововведениях и изменениях в программе обучения. Страница представлена на рисунке 3.9.

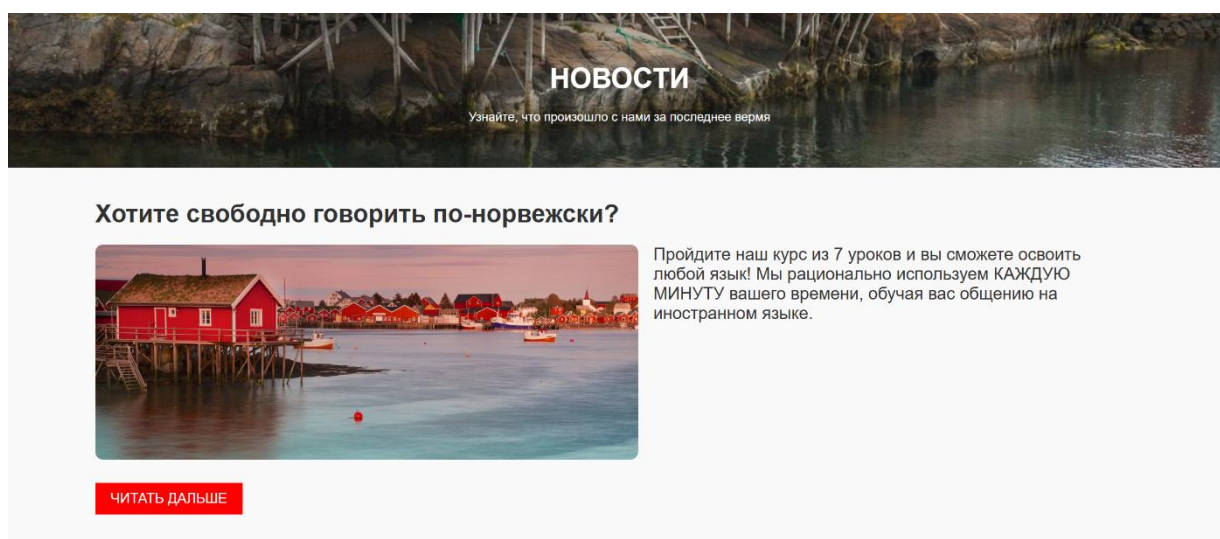


Рисунок 3.9 – Страница «Новости»

Страница «Пользовательское соглашение» содержит текст пользовательского соглашения, которое регулирует порядок работы сайта и определяет условия использования материалов и сервисов, размещённых на сайте. Данная страница представлена на рисунке 3.10.

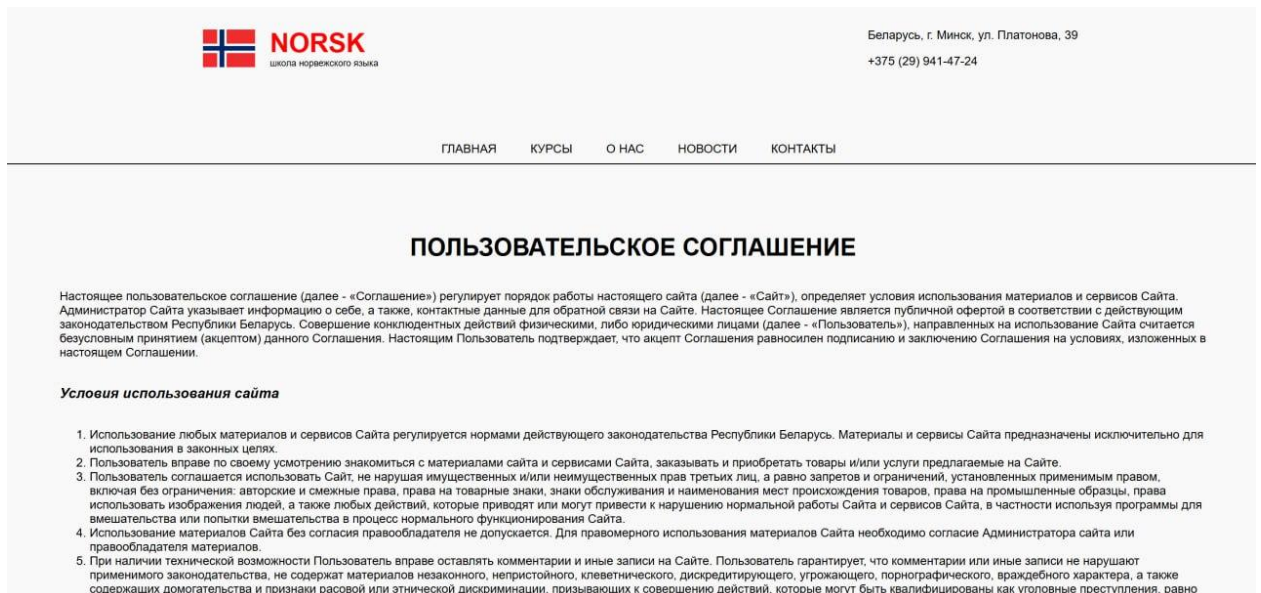


Рисунок 3.10 – Страница «Пользовательское соглашение»

Страница «Политика конфиденциальности» содержит текст политики конфиденциальности сайта. Страница представлена на рисунке 3.11.

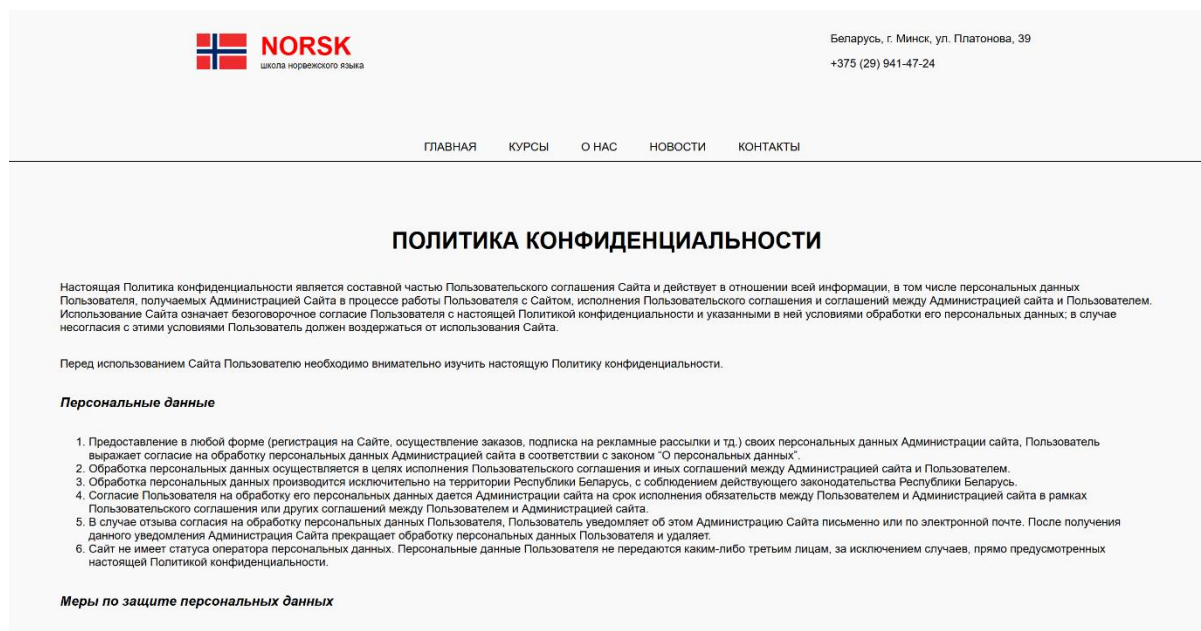


Рисунок 3.11 – Страница «Политика конфиденциальности»

Код оптимизирован, для чего удалены лишние пробелы, а также минимизирован CSS с использованием *CSS Minifier*.

Создан новый репозиторий на *GitHub*, в который проект в последующем сохранен. Репозиторий располагается по ссылке <https://github.com/ShuhMen/ITiVP>. Сайт развёрнут на *GitHub Pages* и располагается по адресу <https://shuhmen.github.io/ITiVP/1%20lab/index.html>.

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы разработан веб-сайт по предложенной тематике «Сайт для онлайн-курсов».

В итоге сайт содержит следующие веб-страницы:

- главная страница;
- страница «Курсы»;
- страница «О нас»;
- страница «Новости»;
- страница «Контакты»;
- страница с пользовательским соглашением;
- страница с политикой конфиденциальности.

При создании веб-страниц с различными макетами освоены базовые навыки владения *HTML* и *CSS*. Изучено, как получать доступ к растровым и векторным ресурсам проекта как. Разработанный веб-сайт обладает дружелюбным и понятным интерфейсом, что сделает его использование удобным и приятным для посетителей.

Разработка веб-сайта велась с использованием *GitHub*. Сервис значительно упростил совместную разработку проекта. В процессе создавались ветки для каждого разработчика, что позволило без конфликтов заниматься параллельной разработкой без конфликтов. В последствии изменения в ветвях переданы в основную ветвь.