**Проект**

**«Веб-приложение для анализа семантики HTML кода»**

Автор: Мелков Илья Андреевич

Ученик ЦДНИТТ «Наследники Ползунова» 11 класса

Научный руководитель: Андреева Ангелика Юрьевна

2021г.

**0 слайд**

**Вступление**

**Здравствуйте, меня зовут Мелков Илья Андреевич. Я учусь в** ЦДНИТТ «Наследники Ползунова» города Барнаула в 11 классе. Я представляю свой проект под названием «Веб-приложение для анализа семантики HTMl-кода».

**1 слайд**

**Актуальность:** есть группа пользователей, которые по каким-либо причинам не могут видеть графический интерфейс веб-сайтов. Многие сайты сейчас не адаптивны для них и это большая проблема, но я хочу сделать так чтобы разработчикам легче было определять, на сколько адаптивен их сайт для такой группы пользователей. Так как технологии стремительно развиваются уже нельзя игнорировать доступность контента своего сайта. Даже слепе люди могут взаимодействовать с веб-сайтами с помощью Screenreader-ов. Screenreader – это приложение, которое может воспроизводить текст на экране в аудио формат.

**2 слайд**

**Цель:**

* Раскрыть проблему доступности веб-страниц для группы пользователей, которые не видят графический интерфейс.
* Упростить работу разработчикам и сделать веб-приложение, которые будет анализировать их страницу, показывать ошибки и как их страницу будет видеть screenreader.

**3 слайд**

**Задачи:**

Поставленные задачи приведены на слайде.

**4 слайд**

**Аналоги**

Изучая проблему я не нашёл сервисов, которые помогут разработчику показать семантическую вёрстку его сайта. Из похожий сервисов – это Валидатор (<https://validator.w3.org/>) и pr.cy (https://a.pr-cy.ru/fipi.ru/). Проблема в том, что эти сервисы позволяют очень строго проверить сайт на многие аспекты веб-разработки, но совсем не затрагивают сематическую вёрстку и адаптивность контента. Я хочу указать на проблемы адаптивности контента сайта разработчику. Показать, что надо добавить разработчику для того чтобы его сайт был доступен для людей которые не могут видеть интерфейс сайта или их сайт легче взаимодействовал с вспомогательными приложениями.

**5 слайд**

**Целевая аудитория:**

1. Разработчики, которые хотят проверить свой сайт адаптивность контента
2. Заказчик, который хочет проверить работу разработчика.

**6-7 слайд**

**Стек технологий:**

Стек технологий приведён на слайде. Среда разработки Visual Studio Code.

**8 слайд**

**Теория**

**Семантическая вёрстка** — подход к разметке, который опирается не на содержание сайта, а на смысловое предназначение каждого блока и логическую структуру документа. Так как на странице сайта — читатели могут быть немного другими.

**9 слайд**

**Почему семантика важна?**

Чтобы сделать сайт доступным**.** Зрячие пользователи могут без проблем с первого взгляда понять, где какая часть страницы находится — где заголовок, списки или изображения. Для незрячих или частично незрячих всё сложнее. Основной инструмент для просмотра сайтов не браузер, который отрисовывает страницу, а скринридер, который читает текст со страницы вслух. Этот инструмент «зачитывает» содержимое страницы, и семантическая структура помогает ему лучше определять, какой сейчас блок, а пользователю понимать, о чём идёт речь. У скринридеров есть функция навигации по заголовкам, что ускоряет знакомство с информацией на сайте. Также семантика помогает правильно выводить контент на устройствах которые не приспособлены к отображению графического интерфейса такие, как электронные часы и другие. Ещё помогает в SEO-оптимизации. Браузеру легче взаимодействовать с определёнными тегами и выводить ваш сайт в поиске выше на похожие запросы.

**Моя работа делится на 3 главных части:**

1. Скрипт проверки файла
2. Графический дизайн
3. Загрузка сайта на хостинг для общего доступа по ссылки

**12 слайд**

Скрипт запускается по нажатию кнопки при загруженном файле в специальном поле. Скрипт представляет собой большую работу с html-файлом, который преобразуется в текстовую переменную. Её прогоняют по определённым правилам семантики, правильности написания кода и выводит контент основных тегов на сайте (как бы screenreader прочитал вашу страницу). На примере этой части кода я покажу как в целом работает мой скрипт т.к. остальные проверки файла работают по такому же принципу. Данный кусок кода выполняет проверку на наличие тегах nav. Первая часть кода находить начало и конец тега nav и вычленяет все атрибуты которые есть в нём. Далее он оставляет контент, который находится в этом теге и удаляет этот элемент. Потом он проверяет все остальные циклом for, так как тегов nav может быть несколько на странице.

**13 слайд**

Для простоты представления проекта я приобрёл хостинг. Сделал компиляцию своего приложения и загрузил на его на сервер. Он доступен по ссылке: <https://u113718.test-handyhost.ru/>. Или QR-коду.

**14 слайд**

Дизайн представляет собой лэндинг, рассчитанный на ознакомление пользователя с семантикой и не отвлекать его от процесса проверки файла. Задача была создать интуитивно понятный дизайн для пользователя. Выполнен в черно-белых цветах. Сайт адаптивный для телефонов.

**15 слайд**

**Дальнейшее взаимодействие и улучшение проекта**

1. Сейчас при анализе некоторых файлов происходит небольшое нарушение разметки у некоторых блоков. Это никак не влияет на вывод информации. Это связано с HTML-кодом который проходит через сервис. Нарушение разметки вызывает код написанный на сторонних фреймворках.
2. При входе на сайт выводится уведомление об опасности. Это больше не проблема, а особенность хостинга, которым я пользуюсь. Чтобы это исправить надо либо приобрести на хостинге ssl-сертификат.

**16 слайд**

**Библиография**

Вдохновлялся лекцией Вадима Макеева: <https://www.youtube.com/watch?v=W5DgVWtitjQ&t=1852s>

Спецификация html5:

<https://www.w3.org/TR/2011/WD-html5-20110405/>

Пояснение: Я вдохновлялся этой лекцией, а при написании приложения я не пользовался сторонней литературой. Со всеми возникающими ошибками я разобрался сам иногда прибегая к видео-урокам по js для решения оптимизации или поиска более подходящего метода. Поэтому всем чем я пользовался это лекция и спецификацией html5.