

Содержание

| | |
|---|-----------|
| АББРЕВИАТУРЫ | 3 |
| ВВЕДЕНИЕ | 4 |
| ГЛАВА 1. Доступность web-приложений для людей с ограниченными возможностями | 6 |
| 1.1 Электронная доступность | 6 |
| 1.2 Общие проблемы доступности | 8 |
| 1.3 Принципы доступности | 9 |
| 1.4 Как сделать доступными электронные ИКТ-продукты? | 10 |
| 1.5 Предоставление пользователям возможности настраивать продукт под себя | 10 |
| 1.6 Рекомендации по доступности веб-контента | 11 |
| 1.7 Базовые принципы | 12 |
| 1.8 Идея сайта | 15 |
| 1.9 Особенности сайта | 16 |
| 1.9.1 Цветовой контраст | 16 |
| 1.9.2 Логическая кнопка TAB | 17 |
| 1.9.3 Преобразование текста в звук | 18 |
| 1.9.4 Простота | 19 |
| ГЛАВА 2. Технический аспект | 20 |
| 2.1 Внешний интерфейс | 20 |
| 2.2 Серверная сторона | 23 |
| 2.2.1 Среда выполнения Node.js | 23 |

| | | |
|----------------------------|--|-----------|
| 2.2.2 | Фреймворк Express | 24 |
| 2.2.3 | Как работает серверная часть | 25 |
| ЗАКЛЮЧЕНИЕ | | 30 |
| ЛИТЕРАТУРНЫЕ ССЫЛКИ | | 32 |

АББРЕВИАТУРЫ

1. WCAG: Правила доступности веб-контента (Web Content Accessibility Guidelines);
2. ИКТ: информационные и коммуникационные технологии;
3. HTML: Язык гипертекстовой разметки (Hypertext Markup Language);
4. REST: передача состояния представления (Representational state transfer);
5. API: программный интерфейс приложения (Application programming interface).

ВВЕДЕНИЕ

С начала нового тысячелетия интернет стал важной частью жизни: люди постоянно просматривают веб-сайты, пользуются разнообразными приложениями и переписываются текстовыми сообщениями. Интернет настолько вошел в нашу повседневность, что мы не можем представить себе жизнь без него. Одни используют его в качестве средства общения, другие - в бизнесе, третьи для отслеживания новостей. Но так же следует отметить: именно интернет позволяет людям с ограниченными возможностями быть самими собой без всяческих ограничений.

Согласно данным Организации Объединенных Наций, инвалиды составляют самое многочисленное меньшинство в мире. По оценкам Всемирной Организации Здравоохранения, один миллиард человек имеет инвалидность, причем 80% из них проживают в развивающихся странах. Инвалиды могут в равной степени участвовать в жизни общества и вносить существенный вклад в экономику, если имеются соответствующие инструменты Интернета.

Для инвалидов доступность означает возможность использовать продукт или услугу также эффективно, как и для человека без инвалидности. Это означает использование всех возможностей, предоставляемых и здоровым людям. В некоторых случаях это невозможно, и для заполнения этого пробела могут потребоваться вспомогательные технологии. Инвалиды сталкиваются со столькими же различными препятствиями, сколько существует видов и степеней инвалидности. Например, люди с нарушением зрения, которые ис-

пользуют программное обеспечение для чтения с экрана, могут столкнуться с веб-сайтами, которые имеют запутанную навигацию или не имеют описания изображений; в то время как люди с нарушением слуха могут быть не в состоянии участвовать в онлайн-конференциях из-за отсутствия субтитров.

Таким образом можно выделить объект исследования - программы для людей с ограниченными возможностями. Предметом исследования является корреляция поведения инвалидов в Интернете благодаря этим программам. Задачи и цели данного исследования:

- Выяснить качество предоставляемых возможностей для инвалидов;
- Установить улучшение или ухудшение качества программ;
- Облегчение жизни для людей с ограниченными возможностями;
- Обратить внимание общественности к данной проблеме.

ГЛАВА 1. Доступность web-приложений для людей с ограниченными возможностями

1.1 Электронная доступность

Электронная доступность - это показатель того, как продукт или услуга используется человеком. Этот показатель должен быть так же эффективен для людей с инвалидностью, как и для обычных людей.

Например, если незрячий человек может использовать все функции банкомата так же легко, как и зрячий, то этот банкомат можно считать полностью используемым и незрячими людьми. Если же человек в инвалидном кресле попытается воспользоваться обычным автомобилем, то у него возникнут определенные проблемы. Скорее всего, он не сможет воспользоваться этим автомобилем, тогда про этот вид транспорта можно сказать, что он «недоступен для людей в инвалидных креслах». В некоторых случаях это может быть возможно для некоторых людей, но пользоваться такой машиной все равно будет очень сложно, или можно будет использовать лишь некоторые ее функции, но не все. Тогда автомобиль будет описан, как «частично доступный» или даже «относительно недоступный» для этой категории людей. Точное описание будет зависеть от степени проблемы, с которыми сталкиваются различные типы пользователей[3]. Таким образом, понятие доступности связано с разнообразными потребностями и возможностями различных

категорий пользователей - людей с ограниченными возможностями - и выражается в степенях: от “полной доступности” или “частичной доступности” до “полной недоступности”, для определенной группы пользователей[2].

Стандарты доступа для конкретных типов продуктов часто пытаются количественно оценить удобство и простоту использования измеримыми способами, указывая необходимые атрибуты, объективные тесты и критерии прохождения. Хорошим примером является руководство по доступности веб-содержимого, которое указывает проверяемые критерии успеха для трех уровней соответствия (А, АА или ААА)[3, 7]. По этому можно объективно указать, доступна ли та или иная веб-страница для конкретного уровня. На вопрос о том, является ли сайт свободно посещаемым, можно ответить, указав, соответствует ли он согласованному уровню руководящих принципов WCAG. Однако, для многих видов ИКТ не существует согласованных на международном уровне полных стандартов доступности. Таким образом, соблюдение стандартов зачастую не может рассматриваться как полная доступность[1, 3].

На практике, возможность действий связана с опытом людей с ограниченными возможностями для выполнения задач, для которых продукт предназначен. Чем больше людей могут использовать продукт, тем больше задач они могут выполнять с ним, и чем легче им выполнять эти задачи, тем более доступным является продукт. По сути, продукт или услуга доступны в той мере, в какой их функциональные возможности отвечают потребностям людей с ограниченными возможностями.

1.2 Общие проблемы доступности

Веб-сайты обычно содержат смесь текста, изображений, ссылок, кнопок, таблиц, интерактивных форм и другого контента. Многие их свойства и функции могут быть недоступны, включая следующие[2]:

- Экранные кнопки реагируют только на щелчок мыши, поэтому человек с ограниченными физическими возможностями, который не может использовать мышь, не может “щелкнуть” их, как обычно. Такой человек должен иметь возможность нажать клавишу Tab и Enter на клавиатуре[2].
- В платежной форме метки для полей ввода и элементов управления (например, имя, выбор способа оплаты) отображаются таким образом, что они не могут быть прочитаны программой преобразования текста в речь, используемой слепым человеком, поэтому этот человек не знает назначение каждого поля или элемента управления[2].
- Визуальный дизайн и макет могут различаться от страницы к странице, что делает веб-сайт запутанным и трудным для людей с познавательной дисфункцией[2].
- В онлайн-видео зачастую нет субтитров, звуковых дорожек описания или текстовых транскриптов. Таким образом, глухие, слабослышащие и слепые пользователи не имеют доступа к полному контенту[2].

1.3 Принципы доступности

Для обеспечения полного доступа все пользователи должны иметь возможность выполнять четыре действия для каждого элемента управления, инструкции или вывода[1]:

- Восприимчивость;
- Понятность;
- Работоспособность;
- Надежность.

То есть, человек должен иметь возможность воспринимать все элементы информации на экране и иметь простой доступ к ним.

Например, глухой человек может не знать о существовании звукового сигнала на странице, а слепой - визуального сигнала. Человек с ослабленным зрением может видеть текст на дисплее банкомата, но имеет такие трудности при чтении, что не может получить доступ к информации, которую он предоставляет[3].

Понятность - это возможность определить назначение того или иного элемента, даже если человек сталкивается с ним впервые.

Человек с инвалидностью не может следовать сложной или плохо написанной инструкции в банкомате. Ему может быть трудно понять, где найти нужную информацию на непоследовательно структурированном веб-сайте.

Человек в инвалидной коляске может быть не в состоянии добраться до слота для карты в банкомате. Так же проблемы с банкоматом могут возникать у людей с нарушением моторики рук, так

как им сложно нажимать на маленькие кнопки на клавиатуре для ввода пин-кода[3].

1.4 Как сделать доступными электронные ИКТ-продукты?

Продукт ИКТ доступен для инвалидов, если они могут использовать его как есть, с незначительной корректировкой или конфигурированием, или в сочетании с некоторыми вспомогательными технологиями.

Это может оказаться трудной или даже невозможной задачей: разработать конкретный продукт таким образом, что все люди смогут использовать его полностью, как есть, без каких-либо корректировок или дополнений. Однако, большинство продуктов могут быть доступны большинству пользователей, обеспечивая максимальную базовую доступность, позволяя настраивать интерфейс, использовать различные методы взаимодействия, таким образом выдавая выходные данные в нескольких формах и обеспечивая совместимость с вспомогательными технологиями.

1.5 Предоставление пользователям возможности настраивать продукт под себя

Предоставление пользователям возможности настраивать продукт в соответствии с их потребностями часто является хорошим способом удовлетворения различных нужд. Например, банкоматы, веб-сайты, мобильные телефоны и приложения для ПК могут предо-

ставлять различные текстовые форматы и цветовые схемы для людей с различными визуальными способностями. Также должна быть возможность увеличить время, в течение которого им разрешено вводить код или пароль. Некоторые сайты могут иметь две версии, что дает выбор между полностью функциональным видом или более простым видом с увеличенными кнопками, покрытием не всех, а только самых важных функций и т.п[3].

1.6 Рекомендации по доступности веб-контента

Данные рекомендации охватывают широкий спектр советов по улучшению доступности веб-контента. Следование этим рекомендациям сделает контент доступным для более широкого круга людей, в том числе и с ограниченными возможностями, которые включают слепоту и слабовидение, глухоту и частичную потерю слуха, ограниченные физические возможности, нарушения речи, светочувствительность и их комбинации, а также некоторые нарушения когнитивных способностей. Но это не означает, что сайт будет абсолютно доступным для людей ограниченными возможностями. Эти рекомендации касаются доступности веб-контента на настольных компьютерах, ноутбуках, планшетах и мобильных устройствах. Следование этим рекомендациям также часто делает веб-контент более удобным для пользователя в целом[8].

1.7 Базовые принципы

Уровень соответствия “А” это минимальный уровень требований, которым должны соответствовать все веб-сайты, приложения и электронное содержимое, например документы.

Уровень соответствия “АА” является приемлемым уровнем доступности для многих онлайн-сервисов, которые должны работать с большинством вспомогательных технологий, которые в настоящее время широко доступны как на настольных, так и на мобильных устройствах[4].

Уровень соответствия требованиям “ААА” это самый высокий уровень доступности, который обеспечивает все для полного доступного контента.

Базовые принципы сгруппированы по 4 категориям:

1. Воспринимаемость: понятен для слабовидящего человека (с помощью программы чтения с экрана, экранной лупы или другой вспомогательной технологии) или глухого (с использованием подписей, письменных транскриптов и т. д.)[4]. Это означает, что для не текстового контента должны быть предусмотрены текстовые альтернативы (“альтернативный текст”). Кроме того, следует добавить транскрипцию аудио- и видеоматериалов, включая подписи к видео. Содержание должно быть логически структурировано и должно быть удобно в перемещении по нему с помощью программы чтения с экрана. Обл Ссылки должны при наведении мыши выделяться не только цветом, но также подчеркиваться. Должен быть высо-

кий контраст между цветом фона и цветом текста и не забывать о возможности увеличения размера текста, не нарушая функции сайта[7].

2. Работоспособность: пользователь с ограниченными возможностями может управлять функциями, например, веб-сайт может перемещаться по сочетаниям клавиш для тех, кто не может использовать мышь. Пользователи должны иметь возможность приостанавливать или останавливать перемещение контента, избегать мигающих объектов или, по крайней мере, иметь возможность их отключить. Все страницы и рамки должны быть описаны и подписаны. Должны быть добавлены описания к ссылкам, чтобы пользователи могли понять, куда они ведут. Должны использоваться осмысленные заголовки и метки, все доступные метки должны совпадать с используемыми в интерфейсе. Пользователи клавиатуры должны легко видеть элемент, на котором в настоящее время сосредоточены их клавиатура или вспомогательные технологии - это называется "активным фокусом". События мыши или динамические взаимодействия (например, скрипты, используемые для декора сайта) следует использовать только при крайней необходимости, или пользователь должен иметь возможность отключить их и взаимодействовать с интерфейсом по-другому[1, 4].

3. Понятность: технология может быть использована пользователями с различными когнитивными способностями. Следует использовать простой язык и короткие тексты. Сложные слова и фразы не могут быть использованы; если их нельзя

избежать, должно быть представлено объяснение. Все аббревиатуры и сокращения должны быть разъяснены, если они неизвестны или широко не используются. Должно быть ясно, на каком языке написано содержание, и указано, если оно было изменено. Функции должны выглядеть согласованными и вести себя предсказуемым образом. Убедитесь, что все поля формы имеют видимые и значимые метки, и они должны быть правильно помечены[1, 7].

4. Надежность: технология должна быть совместима с существующими вспомогательными технологиями и готова к обновлению для будущих версий вспомогательных технологий. Необходимо использовать допустимый HTML, чтобы клиентская сторона, включая вспомогательные технологии, могли точно интерпретировать и анализировать содержимое. Код позволяет ассистивным технологиям узнать, для чего предназначен каждый компонент пользовательского интерфейса, в каком состоянии он находится в данный момент и меняется ли он[1, 7]. Важные статусные сообщения или модальные диалоги должны быть размечены таким образом, чтобы информировать пользователей об их присутствии и целях, а также позволять им взаимодействовать с ними с помощью вспомогательных технологий. Пользователь должен иметь возможность вернуться к тому, что он делал после взаимодействия с сообщением о состоянии или модальным окном. Разработчик может предусмотреть возможные трудности для людей с ограниченными возможностями. Например, пользователь мо-

жет выполнять следующие действия:

- использовать клавиатуру вместо мыши;
- изменить интерфейс сайта (удалить CSS и стили), чтобы контент был более читабельным;
- использовать программы чтения с экрана для озвучивания содержимого вслух;
- использовать экранную лупу, чтобы увеличить часть экрана или весь экран;
- использовать голосовые команды для навигации по сайту.

1.8 Идея сайта

Разрабатываемый в данном дипломном проекте программный продукт - это полностью доступный словарный ресурс, который представляет результаты из нескольких словарей (Оксфордский словарь, Яндекс словарь, словарь Lexicala) доступным способом для людей с особыми потребностями, где пользователь может ввести слово и получить определение того же слова из нескольких словарей в доступной форме.

Сайт обслуживает пользователей с любыми особенностями здоровья и отличается следующими характеристиками:

- Высокая контрастность и большие шрифты для людей с когнитивными нарушениями и дислексией;

- Полностью доступный TAB для людей с ограниченными физическими возможностями, что означает, что они могут полностью использовать клавиатуру для доступа ко всем функциям веб-сайта с помощью кнопки TAB;
- Поддерживает преобразование текста в речь для людей со слепотой или частичным зрением;
- Простой дизайн и структура для людей с когнитивными проблемами, так что им легче сосредоточиться и сконцентрироваться в минимальной, прямой и свободной от помех среде.

1.9 Особенности сайта

Как было сказано ранее, этот веб-сайт представляет собой онлайн-словарь, предназначенный специально для людей с особыми потребностями, но вопрос в том, что делает его таким?

1.9.1 Цветовой контраст

Цвета для веб-сайта, выбранные специально для обеспечения высокого контраста между элементами веб-страницы, включая, что наиболее важно, контраст между фоном и текстом, например:

Используя анализатор цветового контраста, который является инструментом веб-разработки для оценки уровня цветового контраста и его уровня соответствия стандартам, можно увидеть, что кнопка отправки, например, имеет уровень соответствия AAA для всех видов дальтонизм[8].

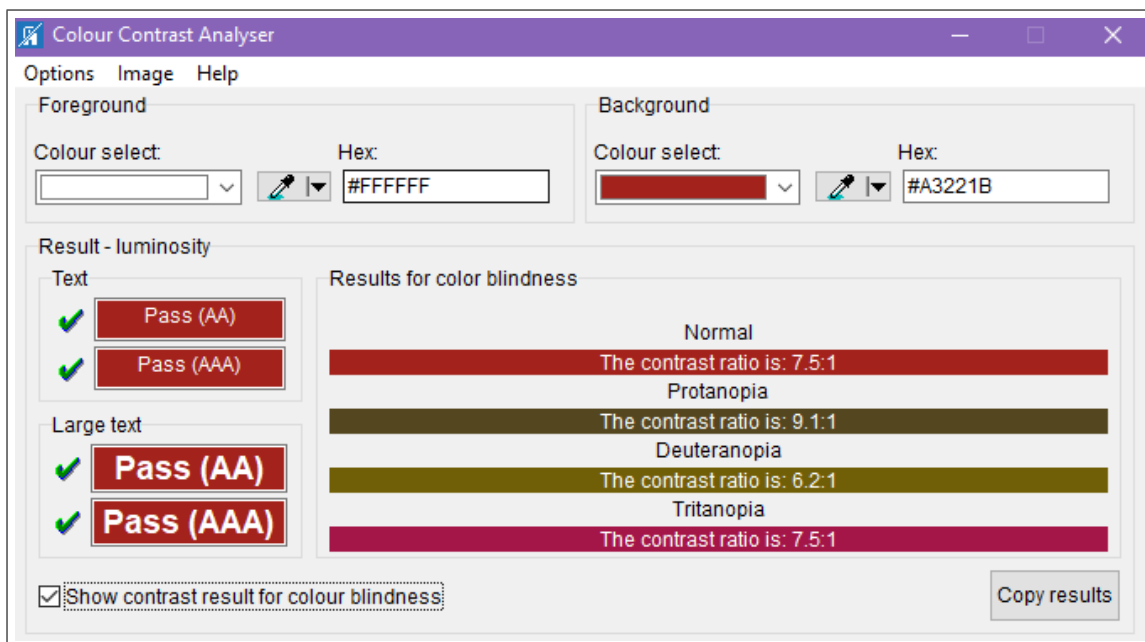


Рис. 1: Результаты цветного контрастного анализатора

1.9.2 Логическая кнопка TAB

Одним из наиболее важных аспектов, которые следует учитывать при создании веб-сайта, доступного в Интернете, является кнопка TAB, которая является одним из наиболее важных инструментов для пользователей с ограниченными физическими возможностями, которые не могут использовать компьютерную мышь, она позволяет пользователю перемещаться между элементами полностью полагаясь на клавиатуру. Некоторые веб-элементы уже имеют встроенную функциональность TAB по умолчанию, так что web-dev не нужно беспокоиться об этом, однако существует большой кусок веб-элементов, которые обычно не видны даже при использовании кнопки TAB, поэтому, когда пользователь щелкает элемент, который полностью пропускается, что делает большую часть сайта скрытой от пользователей с особыми потребностями.

Для этого есть специальные атрибуты, которые необходимо записать в HTML-код веб-сайта, чтобы сделать эти элементы видимыми.

Это также касается не только видимости элементов, но и логического пути, который необходимо учитывать, когда пользователь начинает использовать эту кнопку, элементы должны быть сфокусированы в логическом смысле, например, в онлайн-словаре, текстовом поле отправки. Сначала необходимо сосредоточиться на кнопке отправки, а затем, когда пользователь просматривает результаты, необходимо сосредоточиться на именах источников словаря, прежде чем переходить к определениям каждого источника, эти вещи должны быть рассмотрены веб-разработчик, чтобы сделать сайт доступным[8].

1.9.3 Преобразование текста в звук

Другим важным аспектом веб-сайта, доступного через Интернет, является обслуживание слепых пользователей, пользователей с частичным зрением и людей с дислексией и другими нарушениями познавательной способности и обучения, которым необходимо слышать и видеть текст, чтобы лучше понять содержание сайта.

Большинство операционных систем имеют встроенные приложения для преобразования речи в текст, которые считывают то, что написано на экране, а также многие приложения, загружаемые из Интернета, служат той же цели[6].

Имея это в виду, веб-сайт должен быть полностью стабильным при использовании этих приложений, кнопка не может быть про-

сто названа кнопкой, для слепого пользователя, ведь это не будет иметь смысла, и не даст ему достаточно информации: чтобы понять, как работает кнопка, ей должно быть присвоено полное имя «Кнопка отправки слова» Например, в онлайн-словаре использовалась семантическая разметка HTML для таких структур, как заголовки, абзацы, списки, формы и таблицы. текстовые альтернативы для изображений, значков и другого нетекстового контента.

1.9.4 Простота

Для пользователей с когнитивными проблемами могут возникнуть проблемы с концентрацией внимания и сосредоточенностью на чрезмерно загруженном и сложном веб-сайте.

Для предотвращения этого веб-сайт онлайн-словаря упрощен простыми цветами, минимальными узорами и кнопками, со всем прямым, только с абсолютно необходимой информацией, представленной пользователю[7].

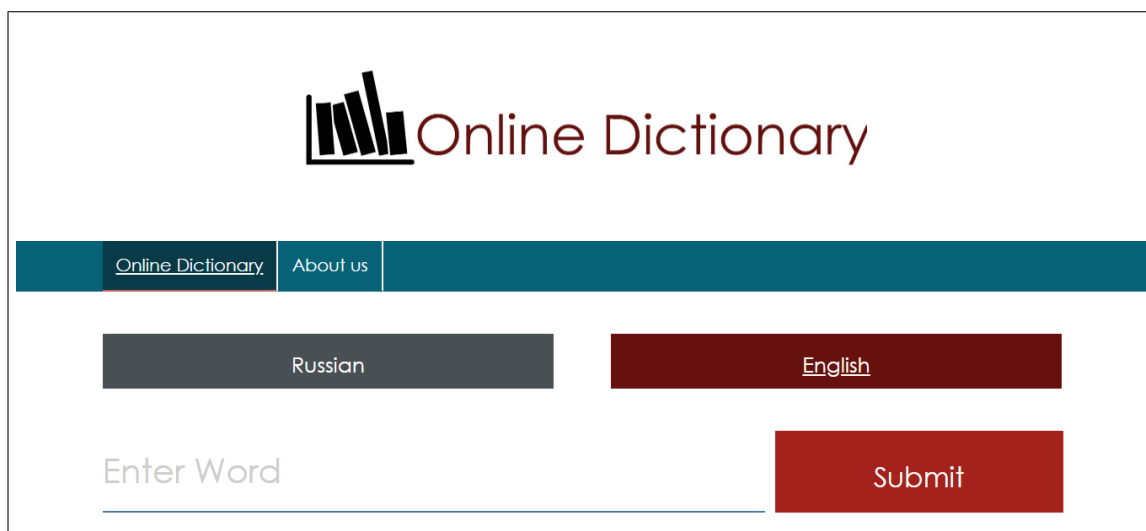


Рис. 2: Главная страница сайта

ГЛАВА 2. Технический аспект

Веб-сайт состоит из двух основных частей: серверной части и веб-интерфейса, серверная часть отвечает за вызовы сервера и API и поддерживает функциональность веб-сайта. Веб-интерфейс - это часть, которая взаимодействует с пользователем и которая соединяет пользователя с сервер. Файловая иерархия состоит из файла `server.js`, который содержит весь основной код на стороне сервера и вызовов API, и файла `package.json`, Одним из наиболее важных файлов является файлом `Package.json`, который содержит информацию о приложении, зависимости и скриптах.

2.1 Внешний интерфейс

До сих пор мы работали с английской версией API, по сути, то же самое делается с российской версией, за исключением того, что она использует разные вызовы API для российских источников.

Код интерфейса написан на Pug.js, который является посредником. Pug - это шаблонизатор для Node.js. Шаблонный движок позволяет вводить данные и затем генерировать HTML. Вкратце: во время выполнения Pug заменяет переменные в файле действительными значениями, а затем отправляет полученную строку HTML клиенту.

Первые две важные кнопки на веб-сайте - это кнопки изменения языка, которые переключаются между русской и английской версиями веб-сайта, в файле сервера пути уже заданы для двух страниц, а код pug просто указывает браузеру правильный путь.

```

1 .btn-group.btn-group-lg (style='width: 100%; padding-bottom:10px; ')
2   a.button.btn.btn-block.btnstyle (type='button', style='border-
   radius: 10px 10px 10px 10px;', tabindex='3' href="/English")
   English
3   a.button.btn.btn-block.btnstyle (type='button', style='border-
   radius: 10px 10px 10px 10px;', tabindex='4' href="/Russian")
   Russian

```

Листинг 1: Код для кнопок смены языка в Pug.js

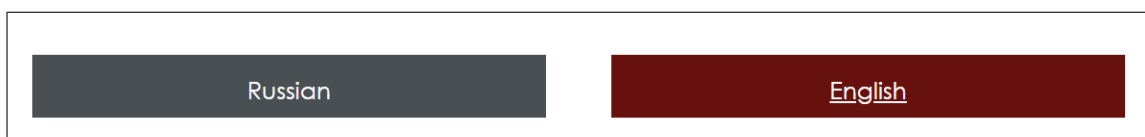


Рис. 3: Кнопки смены языка

Далее идет область, включающая в себя два основных компонента: текстовое поле и кнопку отправки. Первое включает в себя заданные команды, которые не позволяют пользователю использовать несколько слов даже при копировании и вставке в текстовое поле, чтобы избежать осложнений, поскольку веб-сайт создан для того, чтобы давать определения отдельным словам. Когда пользователь нажимает кнопку «Отправить», используя POST, слово отправляется на внутренний сервер для соответствующей обработки[5].

```

1 form (action='searchEn', method='POST', style='height:97px; width:100%;
   float:right; ')
2   input.btn.btn-lg.btn-rounded.searchbutton (type='submit', tabindex=
   '6' onclick="func1 ()")
3   input#nospace (type='text', name='oxford', style='padding: 29px 32
   px; width:69%; height:100%; float:left; display:inline-block;
   font-size:40px', tabindex='5')

```

Листинг 2: Форма для поиска слова

Рис. 4: Кнопка отправки

После этого веб-сайт разделяется на столбцы для каждого источника словаря. Pug.js поддерживает интерполяцию, которая позволяет серверу отправлять данные, включая строку и HTML-код, во внешний интерфейс, используя «!» Для строки HTML и «» для обычной строки, используя Функция `res.render` на сервере, например столбец Оксфордского словаря:

```

1 h2(tabindex='7') Oxford Dictionary
2   button.btn.btn-sm.copybtn(onclick="copyToClipboard('#OxDic')",
3   tabindex='8') Copy text
4   h3#OxDic(tabindex='9')
5   p#Oxformat #{idword}
6   p !{oxdefinition}
7   p !{oxexample}

```

Листинг 3: Колонка для Оксфордского словаря

Также для простоты использования кнопка копирования была добавлена к каждому источнику для быстрого копирования информации.

```

1 function copyToClipboard(element) {
2   var $temp = $("<input>");
3   $("body").append($temp);
4   $temp.val($(element).text()).select();
5   document.execCommand("copy");
6   $temp.remove();
7 }

```

Листинг 4: Функция копирования

2.2 Серверная сторона

Веб-сайт написан с использованием трех основных технологий: Node.js для серверной части, Express.js, который представляет собой инфраструктуру Node.js, и Pug.js, который является механизмом шаблонов. Далее представлены краткие определения этих технологий.

2.2.1 Среда выполнения Node.js

Node.js - это межплатформенная среда выполнения JavaScript, которая может использоваться как для серверов, так и для настольных приложений. Она может использоваться для создания различных типов приложений, таких как приложение командной строки, веб-приложение, приложение чата в реальном времени, Сервер API REST и т.д. Однако он в основном используется для создания сетевых программ, таких как веб-серверы.

Node.js предназначен для работы на выделенном HTTP-сервере и использования одного потока с одним процессом за раз. Приложения Node.js основаны на событиях и работают асинхронно. Код, построенный на платформе Node, не соответствует традиционной модели приема, обработки, отправки, ожидания, получения. Вместо этого Node обрабатывает входящие запросы в постоянном стеке событий и отправляет небольшие запросы один за другим, не ожидая ответов.

Следует отметить, что Node.js не является JavaScript сам по себе, это программа для запуска JavaScript. JavaScript не справляется с сокетами уровня операционной системы и проблемами с сетевым

подключением. Программа Node написана на языке C, который способен выполнять мощные функции. JavaScript идеально подходит для отправки инструкций в C-программу, которая может быть выполнена в глубоких слоях ОС.

Это отступление от базовых моделей, которые запускают более крупные и более сложные процессы и одновременно запускают несколько потоков, каждый из которых ожидает своего соответствующего ответа, прежде чем двигаться дальше.

2.2.2 Фреймворк Express

Express - это веб-инфраструктура маршрутизации и промежуточного программного обеспечения, которая обладает минимальной собственной функциональностью: приложение Express - это, по сути, серия вызовов функций промежуточного программного обеспечения, действующая только как тонкий слой основных функций веб-приложений. В Express нет встроенного механизма реляционного сопоставления объектов или шаблонов. Он не построен на конкретных компонентах, он стремится передать управление в руки разработчика и упростить разработку веб-приложений для Node.js. Express наиболее популярен среди стартапов, которые хотят создать продукт как можно быстрее и не имеют большого количества унаследованного кода.

2.2.3 Как работает серверная часть

Основным файлом, который запускает серверную часть веб-сайта, является `server.js`, который начинается с объявления всех используемых модулей и зависимостей:

```
1 var express = require('express');
2 var app = express();
3 var request = require('request');
4 var https = require('https');
5 var Dictionary = require("oxford-dictionary-api");
6 var path = require('path');
```

Листинг 5: Инициализация библиотек Node.js

Включая `express.js`, который является используемой платформой. Модуль запроса, который позволяет серверу отправлять вызовы API, а также собственный Оксфордский API `node.js`. Затем мы начнем отображать пути к файлам для веб-сайта, у которого есть две стороны - русская и английская - учитывая, что веб-сайт будет работать на двух языках, которые требуют разной компоновки и разных вызовов API. Такой метод оказался намного быстрее, чем создание двух отдельных страниц для каждой версии.

По умолчанию интерфейс веб-сайтов написан на HTML, так как онлайн-словарь будет использовать `Pug.js` для рендеринга в HTML, поэтому для Механизм просмотра (View engine) необходимо установить `Pug`, и установить путь к общедоступной папке, которая содержит ресурсы веб-сайта, включая изображения, файлы CSS и сценарии.

```
1 app.set('view engine', 'pug');
2 app.use(express.static(
3 path.join(__dirname, 'public')));
```

Листинг 6: Настройка сервера

Команда `app.get` используется для установки путей с целью навигации по веб-сайту, поэтому на интерфейсных кнопках нужно просто вызывать эти пути, чтобы отправить пользователя на нужную страницу.

```
1 app.get('/', function(req, res) {
2     res.render('MainEn');
3 });
4 app.get('/Russian', function(req, res) {
5     res.render('MainRus');
6 });
7 app.get('/English', function(req, res) {
8     res.render('MainEn');
9 });
```

Листинг 7: Настройка путей для страниц сайта

А теперь та часть, где пользователь нажимает кнопку отправки после заполнения текстового поля: сначала он извлекает слово, которое было отправлено using the POST method.

```

1 app.post( '/searchEn' ,
2 function( searchReq , searchRes ) {
3   var searchInput = "";
4   var searchInputVal = "";
5   searchReq.on( 'data' , function( data ) {
6     searchInput += data;
7     console.log( "searchInput " + searchInput );
8     searchInput = searchInput.split( "=" );
9     searchInputVal = searchInput[1];
10    console.log( "searchInputVal: " + searchInputVal );

```

Листинг 8: Получение слова со стороны пользователя

Т.к. Node.js написан на Javascript, это означает что он асинхронизировано, то есть текущая выполняемая функция не блокирует поток, а программа не ожидает результата, а переходит к следующему шагу во время выполнения предыдущей функции, чтобы убедиться, что все три вызова API были сделаны и завершены до работы с результатами. В данном проекте был использован модуль под названием «Async», который позволяет нескольким функциям работать параллельно и позволяет программе двигаться вперед, только если все они были завершены.

```

1 async.parallel({
2   one: function(cb) {
3     \\
4   },
5   two: function(cb) {
6     \\
7   },

```

Листинг 9: Функция Async.Parallel

После этого каждый вызов возвращает объект JSON, содержащий требуемые данные. Следующим шагом является их обработка и форматирование и проверка того, что обрабатываются ситуации, когда результат не найден или ввод недействителен.

```

1 if (yantr.includes('"def":[]')) {
2     yantr=" No word Available";
3 } else {
4     \\format string }
5 if (oxstr.includes("No entry available for")
6 | oxstr == undefined) {
7     oxdic=" No word Available";
8 } else {
9     \\format string }
10

```

Листинг 10: Форматирование строки JSON

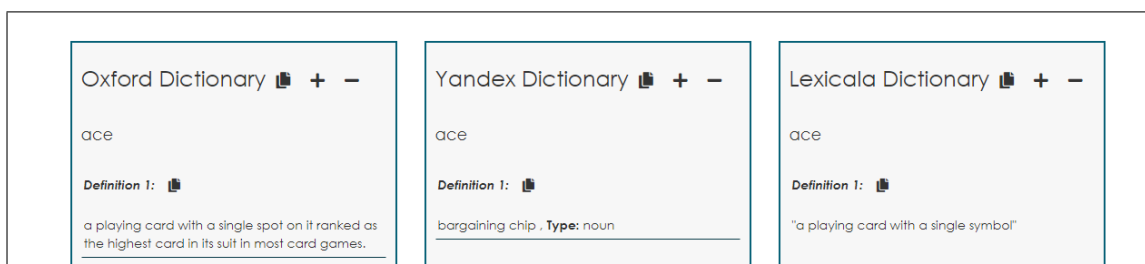


Рис. 5: Результаты представления

До сих пор мы работали с английской версией API, но по сути, то же самое делается с российской версией, за исключением того, что она использует разные вызовы API для русских источников.

```
1 searchRes.render("MainEn", {  
2     yantitle: yantxt,  
3     yanposition: yanpos,  
4     yanspeech: yants,  
5     yantrans: yantr,  
6     idword: oxdic });
```

Листинг 11: Отправка результатов на сайт

ЗАКЛЮЧЕНИЕ

В результате проведенного исследования можно сделать выводы о технологиях доступности и о том, как создать веб-сайт.

Есть много способов, которые позволяют людям с ограниченными возможностями использовать Интернет, так как есть много различных инвалидов для рассмотрения. Глухие нуждаются в подписях, но они также полезны для людей без динамиков на их компьютере. Слепые будут использовать считыватели экрана, такие как JAWS. Люди, чьи проблемы касаются зрения, могут использовать дополнительные инструменты для увеличения размера шрифта. Людям с когнитивными нарушениями может потребоваться больше времени для обработки информации на одной веб-странице, поэтому четко организованный контент очень помогает. Хотя доступность необходима некоторым группам для использования интернета, она выгодна почти всем. Доступность способствует удобству использования в целом: все извлекают выгоду из четких инструкций, возможностей исправить ошибки формы, простых визуальных макетов, высокого цветового контраста и возможности прочитать транскрипт или подписи к видео или аудиозаписи.

Результатом исследования является актуальный и доступный онлайн-словарь, который обслуживает слабовидящих, слабослышащих, пользователей с клавиатурой и людей с когнитивными нарушениями, потому что каждый заслуживает доступа ко всей информации в интернете независимо от ограничений.

Наконец, веб-сайты и приложения, закодированные для обеспе-

чения доступности, в большей степени соответствуют веб-техническим стандартам, что означает, что они с большей вероятностью будут работать на разных платформах и продолжать работать на разных устройствах по мере изменения технологий. Многие разработчики считают, что кодирование доступности ускоряет время разработки и уменьшает ошибки всех видов после запуска программного обеспечения.

ЛИТЕРАТУРНЫЕ ССЫЛКИ

1. Nielsen, Jakob. Designing web usability: The practice of simplicity. New Riders Publishing, 1999.
2. Harper, Simon, and Yeliz Yesilada, eds. Web accessibility: a foundation for research. Springer Science & Business Media, 2008.
3. Paciello, Mike. Web accessibility for people with disabilities. CRC Press, 2000.
4. <https://www.w3.org/WAI/fundamentals/accessibility-principles/>
5. Seeman, L., Cooper, M., Schwerdtfeger, R., Pappas, L. Accessible Rich Internet Applications (WAI-ARIA) Version 1.0 . W3C Working Draft, 4 February 2008.
6. Файзрахманов Руслан Рустамович. "Новый подход к обеспечению веб-доступности для незрячих пользователей на основе улучшения навигационных характеристик веб-страниц" Инженерный вестник Дона, vol. 23, no. 4-2, 2012, pp. 87.
7. Фельке-Моррис, Терри. Большая книга веб-дизайна. Litres, 2017.
8. Кобелев, А. Р. Разработка Web-ресурса для людей с ограниченными возможностями по зрению. Diss. 2017.