

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



Основная профессиональная образовательная программа
Направление подготовки 09.03.01 Информатика и вычислительная техника
Направленность (профиль) «Технологии разработки программного обеспечения»

Выпускная квалификационная работа

Разработка веб-приложения для настройки гитары

Студента 4 курса
Кузнецова Антона Денисовича

Научный руководитель
Кандидат физ-мат. Наук
Жуков Николай Николаевич

Санкт - Петербург
2022

Содержание

Введение	3
Глава 1. Теоретическая часть и выбор средств разработки веб-приложения	6
1.1 Анализ предметной области	6
1.1.1 Типы веб-приложений, сайтов и их различия	6
1.1.2 Сущность гитарного тюнера	8
1.2 Устройство программного тюнера	9
1.2.1 Общие сведения об устройстве тюнера	9
1.2.2 Метод декодирования цифрового сигнала	10
1.3 Выбор среды и инструментов для разработки клиентского веб-приложения	11
1.3.1 Общие сведения о рабочей среде	11
1.3.2 Инструменты разработки	12
1.3.3 Обзор веб-хостинга GitHub pages	14
1.4 Этапы разработки клиентского веб-приложения “Тюнер”	15
Выводы по первой главе	22
Глава 2. Проектирование и разработка веб-программы по настройке гитары.	23
2.1 Установка программной среды Visual Studio Code	23
2.1.1 Этапы установки	23
2.1.2 Настройка среды и поиск необходимых расширений	24
2.2 Разработка интерфейса сайта на основе его структуры и наполнение контентом	25
2.2.1 Файловая структура проекта	25
2.2.2 Код интерфейса сайта	26
2.2.3 Стилизация интерфейса и его наполнение	28
2.3 Разработка программы по настройке гитары	30
2.4 Размещение сайта на GitHub Pages	32
Выводы по второй главе	34
ЗАКЛЮЧЕНИЕ	35
Список литературы	36
Приложения	38

Введение

Музыка — неотъемлемая часть истории, культуры и повседневной жизни человека. Этот вид искусства уникален тем, что рассчитан на слуховое восприятие и отличается от других видов искусств прямым и активным влиянием на чувства людей. Выразительным средством здесь выступает звук и различные компоненты музыкальной гаммы: мелодия, ритм, гармония, композиция. В музыке создаются звуки с особыми новыми свойствами, которых нет в природе, а также которые не существуют вне музыки. Голос был одним из первых музыкальных инструментов человека. В основе музыки лежит ритм и гармония — что в совокупности дает мелодию.

На нынешнем этапе развития инструментальной музыки гитара занимает в ней особое положение, связанное с её популярностью. Популярность этого инструмента обусловлена прежде всего тем, что её часто используют как инструмент в профессиональном творчестве, а также и в домашнем в качестве хобби. Использование гитары приходится на подавляющее число жанров музыки из-за широкого спектра звучания и возможностей данного инструмента. Чтобы от гитары получить правильное звучание, её нужно уметь настраивать. Раньше настройка производилась на слух, это было недоступно обычным любителям из-за неопытности. Однако в настоящее время, достаточно воспользоваться вспомогательными устройствами, упрощающими настройку гитар для обычных людей.

Одним из таких устройств является электронный тюнер, определяющий высоту звука и сравнивающий её с эталонной частотой, после чего выводится результат сравнения. Благодаря цифровизации в быту процесс настройки гитары упростился в разы, теперь не нужно приобретать отдельно тюнер, а с помощью компьютера, микрофона и интернета достаточно воспользоваться программой, которая реализовывает функции тюнера в онлайн-режиме.

Следовательно, **актуальная задача** – разработать онлайн-приложение, которое упрощает процесс настройки гитары, а конкретно — реализация альтернативной существующим продуктам онлайн-версии тюнера.

Объектом исследования — является онлайн-тюнер для настройки гитары.

Предметом исследования выступает реализация алгоритма Гёрцеля, позволяющего эффективно вычислять значение частоты одного частотного компонента, с применением веб-технологий.

Целью проводимой работы является разработка доступного веб-приложения с альтернативной визуализацией процесса настройки гитары и составлением справочника по отладке музыкального инструмента.

Для достижения поставленной **цели** необходимо решить следующие задачи:

1. Проанализировать предметную область и существующий алгоритм Гёрцеля для детектирования и декодирования тональных сигналов;
2. Рассмотреть языки программирования и инструменты, предназначенные для создания веб-приложений;
3. Спроектировать и разработать веб-приложение, содержащее в себе справочник теории по настройке гитары и саму программу на основе выбранного алгоритма;
4. Изучить доступный сервис для хостинга разрабатываемого сайта GitHub Pages;
5. Разместить разработанный электронный ресурс на хостинге.

Результатом бакалаврской выпускной квалификационной работы является готовое веб-приложение, представляющее из себя альтернативную визуализацию тюнера на основе алгоритма Гёрцеля и справочника по настройке гитары.

Структура бакалаврской выпускной квалификационной работы.

Выпускная квалификационная работа состоит из введения, двух глав, заключения, списка литературы и приложения. Содержит 38 страниц, 20 изображений и 3 таблицы.

Глава 1. Теоретическая часть и выбор средств разработки веб-приложения

1.1 Анализ предметной области

1.1.1 Типы веб-приложений, сайтов и их различия

Веб-приложение — представляет из себя программное обеспечение прикладного характера, работающего на веб-сервере, чем отличается от компьютерных программ, запуск которых происходит локально на устройстве в операционной системе.

Сайт или веб-сайт — это место расположения контента, расположенного на сервере, в виде одной или нескольких связанных логически между собой веб-страниц. В большинстве своем сайт в сети интернет является массивом связанных данных, которые имеют единственный адрес и работающий как единое целое. Название «веб-сайт» обусловлено тем, что доступ к ним организован по протоколу HTTP. В своей совокупности сайты формируют всемирную паутину, где информационные сегменты мирового сообщества объединяются в единое целое, формируя базу данных и все-мировую коммуникацию.

Виды сайтов:

1. **Несложные по структуре сайты**, содержащие несколько страничек — сайты-визитки. На таких сайтах обычно немного информации о владельце и небольшой объем содержательной информации.
2. **Тематические сайты**

Данные сайты заточены под определенный вопрос, предоставляя пользователям интернета возможность получить нужную информацию по необходимой теме.

3. Сайты с большим функционалом

Сайты-гиганты представляют из себя огромный ресурс разнообразного функционала и различной информации. Такие ресурсы, как Яндекс, Mail.ru или Google обладают большим потоком посещаемости.

Виды веб-приложений [1]:

1. Серверные веб-приложения

Работа данных приложений осуществляется на удаленных компьютерах. Веб-приложения обычно разрабатываются на таких языках программирования, как C#, Python, PHP, Java, JavaScript и др. Переход между страницами вызывает генерацию нового контента, который отображается у клиента.

В пример можно привести серверное приложение – push-уведомления от почтовых сервисов, мессенджеров и т. п. в смартфонах.

2. Клиентские веб-приложения

В клиентских приложениях в чистом виде нет надобности в серверной части, так как можно обойтись возможностями JavaScript, благодаря использованию браузера пользователя в качестве оболочки. Результат своей работы не сохраняется дольше одной сессии.

Примером можно привести: простенькие игры, фоторедакторы и утилиты по настройке музыкального инструмента.

Разрабатываемый гитарный тюнер является клиентским веб-приложением.

3. Single page application (SPA) веб-приложения

Single page application — одностраничные веб-приложения с реализацией сложного функционала в одном сеансе браузера без перезагрузки. Динамическое обновление содержимого страницы достигается технологией AJAX, асинхронный JavaScript и XML.

В совокупности с JavaScript-фреймворками: React, Angular, Vue, работа таких программ становится максимально эффективной.

Почти все почтовые сервисы разработаны как SPA.

1.1.2 Сущность гитарного тюнера

Тюнер является электронным устройством небольшого размера. С его помощью можно выполнить настройку инструмента при отсутствии идеального слуха или наличии посторонних шумов вокруг. Приборы классифицируются исходя из технических характеристик [2].

Виды тюнеров:

Тюнеры для настройки гитары делятся на виды в зависимости от способа получения входного бита.

1. Портативные

Универсальное устройство с несколькими вариантами настройки, что позволяет применять его в работе с акустической и электрогитарой. Подключение к инструменту происходит при помощи специального разъема, расположенного на корпусе.

В процессе настройки на дисплее отображается нота, соответствующая той струне, из которой извлечен звук. При помощи специальной стрелки и светодиодного индикатора показываются погрешности тонов.

2. Прищепки

Этот вариант применяется в работе с акустическим музыкальным инструментом. Прибор крепится на головку грифа при помощи прищепки. Принцип работы основан на улавливании частоты вибрации в момент касания струн. Это происходит при помощи чувствительного микрофона и датчика, на который выводится уровень настройки. Посторонние шумы не препятствуют работе.

3. Программы

Принцип работы гитарных тюнеров онлайн заключается в настройке инструмента с использованием микрофона и специального приложения, которое устанавливается на компьютер или мобильный телефон.

Доступный способ регулировки звучания, подходящий для всех видов гитар, включая семиструнную. Единственным недостатком является микрофон, предназначенный для распознавания голоса, который может давать погрешности при воспроизведении нот.

1.2 Устройство программного тюнера

1.2.1 Общие сведения об устройстве тюнера

Программный тюнер улавливает сигнал гитары, анализирует его, сверяет с эталонным сигналом и показывает при помощи индикатора величину расхождения. Тюнеры быстро завоевали популярность среди музыкантов, так как позволили очень быстро настраивать гитару даже в очень шумном помещении.

Компьютер может захватывать живой звук/музыку с помощью микрофона, подключенного к звуковой плате. Современные звуковые платы могут записывать цифровые сигналы [3].

Цифровой сигнал — это набор дискретных звуковых значений, которые равномерно расположены в звуковом интервале. Цифровой сигнал не предоставляет никакой информации о частотах, которые присутствуют в звуке. Чтобы определить их, данные должны быть проанализированы.

1.2.2 Метод декодирования цифрового сигнала

Для реализации программного тюнера был выбран алгоритм Гёрцеля.

Алгоритм Гёрцеля — это специальная реализация дискретного преобразования Фурье (ДПФ) [4] в форме рекурсивного фильтра. Данный алгоритм был предложен Джеральдом Гёрцелем в 1958 году. В отличие от быстрого преобразования Фурье, вычисляющего все частотные компоненты ДПФ, алгоритм Гёрцеля позволяет эффективно вычислить значение одного частотного компонента. Алгоритм Гёрцеля является популярным алгоритмом для решения задачи детектирования и декодирования тональных сигналов в телефонии [5].

Одной из характеристик музыкальной ноты является высота звука. Традиционно музыкальный алфавит частот делится на октавы, а затем на полутона. В октаве 12 именованных полей: C, C#, D, D#, E, F, F#, G, G#, A, A#. Октавы также имеют названия: большая, малая, первая, вторая и т.д. Стандартная высота звука (А в первой октаве или А4) имеет частоту 440 Гц. Частоты двух соседних нот отличаются в $2^{1/12}$ раза, а частоты нот с тем же именем в двух соседних октавах различаются в 2 раза [6].

Частоты нот в герцах:

	Название октав			
	Большая	Малая	Первая	Вторая
<i>C</i>	65.4064	130.8128	261.6256	523.2511
<i>C#</i>	69.2957	138.5913	277.1826	554.3653
<i>D</i>	73.4162	146.8324	293.6648	587.3295
<i>D#</i>	77.7817	155.5635	311.1270	622.2540
<i>E</i>	82.4069	164.8138	329.6276	659.2551
<i>F</i>	87.3071	174.6141	349.2282	698.4565
<i>F#</i>	92.4986	184.9972	369.9944	739.9888
<i>G</i>	97.9989	195.9977	391.9954	783.9909
<i>G#</i>	103.8262	207.6523	415.3047	830.6094
<i>A</i>	110.0000	220.0000	440.0000	880.0000
<i>A#</i>	116.5409	233.0819	466.1638	932.3275
<i>B</i>	123.4708	246.9417	493.8833	987.7666

Рис. 1 – Таблица нот и их частот

1.3 Выбор среды и инструментов для разработки клиентского веб-приложения

1.3.1 Общие сведения о рабочей среде

Для реализации проекта был выбран редактор кода Visual Studio Code.

Visual Studio Code — это редактор исходного кода, который можно использовать с различными языками программирования, включая Java, JavaScript, Node.js, Python, C++. Он основан на платформе Electron, которая используется для разработки веб-приложений Node.js, работающих на механизме компоновки Blink [7].

В Visual Studio Code используется тот же компонент редактора (под кодовым названием «Монако»), что и в Azure DevOps (ранее называвшиеся Visual Studio Online и Visual Studio Team Services).

По умолчанию VS Code включает базовую поддержку большинства распространенных языков программирования. Эта базовая поддержка включает в себя подсветку синтаксиса, сопоставление скобок, свертывание кода и настраиваемые фрагменты. Visual Studio Code также поставляется с IntelliSense для JavaScript, TypeScript, JSON, CSS и HTML, а также с поддержкой отладки для Node.js. Поддержка дополнительных языков может быть обеспечена свободно доступными расширениями на VS Code каталоге.

Вместо системы проектов он позволяет пользователям открывать один или несколько каталогов, которые затем можно сохранить в рабочих областях для повторного использования в будущем. Это позволяет ему работать как независимый от языка редактор кода для любого языка. Он поддерживает множество языков программирования и набор функций, которые различаются в зависимости от языка. Нежелательные файлы и папки можно исключить из дерева проекта через настройки. Многие функции Visual Studio Code не отображаются через меню или пользовательский интерфейс, но к ним можно получить доступ через палитру команд.

Код Visual Studio можно расширить с помощью расширений, доступных в центральном репозитории [20]. Сюда входят дополнения к редактору и языковая поддержка. Примечательной особенностью является возможность создавать расширения, добавляющие поддержку новых языков, тем и отладчиков, выполняющие статический анализ кода и добавляющие линтеры кода с использованием протокола Language Server.

1.3.2 Инструменты разработки

Как было указано ранее, для разрабатываемого веб-приложения не потребуется серверной части, так как весь функционал будет работать на

клиентской стороне. Поэтому для разработки приложения были выбраны данные инструменты: HTML5, CSS, JavaScript.

HTML — язык разметки гипертекста, как следует из названия, язык разметки, который используется для представления данных в Интернете. Язык разметки — это язык, который используется для выделения основного текста и придания ему структуры. Здесь разметка используется для указания гипертекста, т.е. текста, который является ссылкой на другой ресурс, такой как страница или документ.

CSS — каскадные таблицы стилей, как следует из названия, таблицы стилей, которые обеспечивают стилизацию уже структурированной структуры HTML. Они называются каскадными, потому что указанные стили могут наследоваться и переопределяться на основе структуры HTML-документа [8].

JavaScript — это клиентский язык сценариев, который используется для создания веб-страниц. Это отдельный язык, разработанный в Netscape. Он используется, когда веб-страницу нужно сделать динамической и добавить специальные эффекты на страницы, такие как прокрутка, развертывание и многие типы графики.

Страницы можно генерировать как на стороне клиента, так и на стороне сервера [16]. Последнее реализуется через механизм подстановок на стороне сервера (Server Site Includes). Компания Netscape распространила в 1995 году механизм управления страницами и на клиента, разработав язык программирования JavaScript. Таким образом, JavaScript — это язык управления сценариями просмотра гипертекстовых страниц Web на стороне клиента. Если быть более точным, то JavaScript — это не только язык программирования на стороне клиента. Liveware, прародитель JavaScript, является средством подстановок на стороне сервера Netscape. Однако, наибольшую популярность JavaScript обеспечило программирование на стороне клиента.

К возможностям JavaScript относятся:

- отображать изменяющиеся данные, такие как текущее время или дата;
- программировать переменное содержание в зависимости от даты, браузера пользователя или других условий;
- изменять внешний вид элементов страницы, если пользователь щелк-нул мышью или провел курсор мыши над элементом.
- Для языка высокого уровня JavaScript обладает сильными возможностями. Он не позволяет работать на уровне машинных кодов, но предоставляет доступ ко многим возможностям браузеров, веб-страниц, а иногда и системы, в которой работает браузер. В отличие от Java или C, программы на JavaScript обходятся без компиляции, и браузеру не придется загружать виртуальную машину для выполнения программного кода [9].

1.3.3 Обзор веб-хостинга GitHub Pages

Для размещения разрабатываемого клиентского веб-приложения гитарный тюнер, состоящего из статических HTML-страниц, был выбран бесплатный хостинг GitHub Pages.

Веб-хостинг — это услуга по предоставлению места для хранения веб-сайта или приложения на сервере в Интернете. Как только веб-сайт будет доступен в Интернете, к нему смогут получить доступ другие компьютеры, подключенные к Интернету. Как только компания размещает веб-сайт, пользователи смогут получить к нему доступ, введя данный веб-адрес (доменное имя) в своем веб-браузере. Когда они это делают, их компьютер подключается к серверу, на котором размещен веб-сайт. Сервер, в свою очередь, обслуживает (отправляет файлы, которые вы сохранили в хранилище, для отображения) веб-сайт вашему веб-посетителю в его веб-браузере.

Сервис GitHub Pages является бесплатным доступным хостингом для статических файлов. Обычный хостинг размещает файлы сайта на своих серверах и скрывает всю техническую часть от посторонних пользователей. Поскольку GitHub Pages является некоммерческим продуктом, его спецификой является открытость файлов для любого пользователя.

Преимущества данного сервиса:

- Это полноценный сайт с HTTPS. В логику простого сайта отлично укладывается множество простых страниц: небольшой сайт компании, личная страничка, простой блог, лендинг или сайт-визитка на GitHub.
- За статическим сайтом не нужно следить, а значит не нужно отслеживать версию CMS и другого ПО, которое легче взломать.
- Возможность деплоя через Git. Благодаря тому, что это децентрализованная система, всегда есть резервная локальная версия сайта [10].

1.4 Этапы разработки клиентского веб-приложения “Тюнер”

Процесс разработки веб-приложения состоит из нескольких определенных этапов.

Основными этапами являются:

- Определение целей и задач проекта.
- Разработка структуры сайта.
- Создание макетов.
- HTML-верстка.
- Программирование.

Целью разработки приложения выступает:

- Разработка визуализации процесса настройки гитары;
- Создание справочника с информацией по настройке инструмента;

Структура сайта

С помощью UML диаграммы, представленной на рисунке 2, проиллюстрирована внутренняя структура веб-приложения

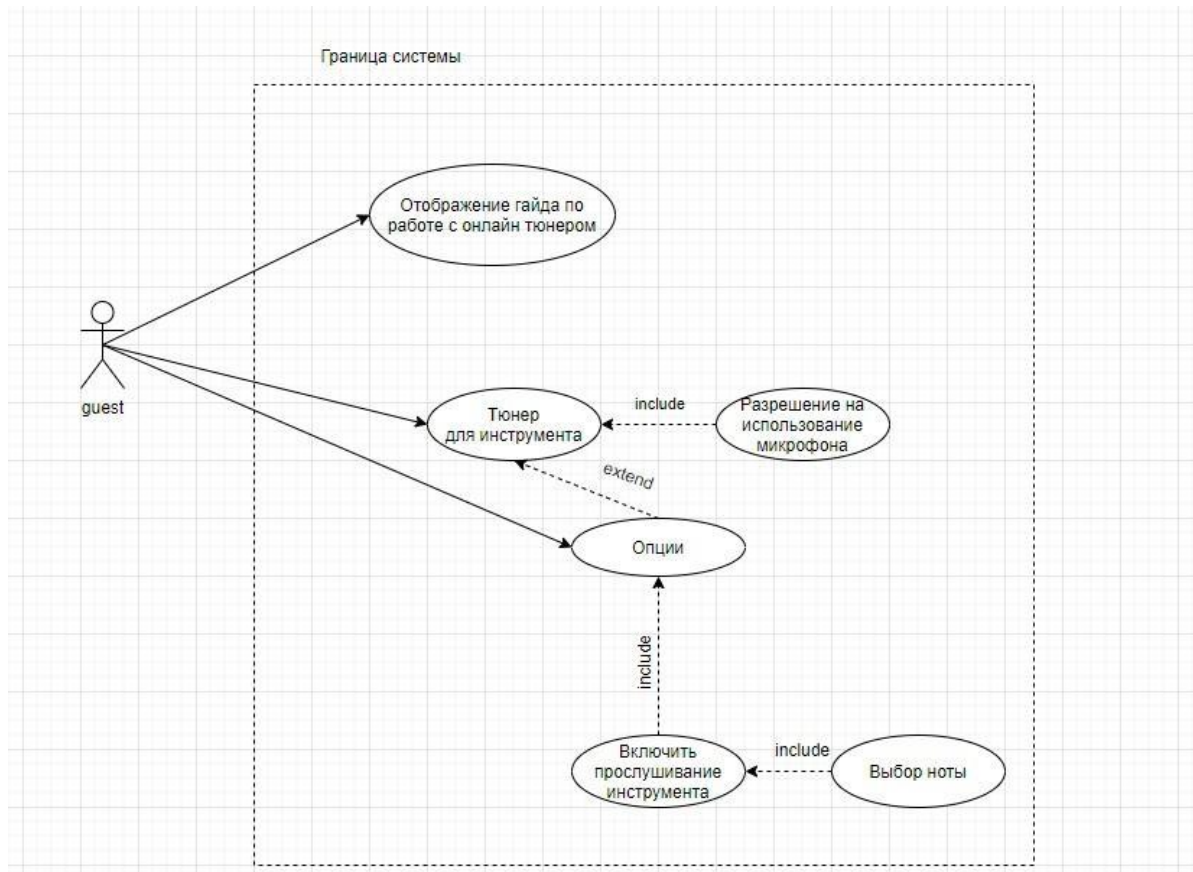


Рисунок 2 - Структура сайта UML

Пользователю не нужна регистрация, чтобы пользоваться всем функционалом онлайн-тюнера. Для начала настройки инструмента, потребуется разрешить доступ браузеру к микрофону, после чего пользователь может запустить основное приложения и провести отладку инструмента. Также у сайта есть вторая страница с справочником, где пользователю предоставлена информация по работе и настройке гитары.

Разработка макета страниц

Перед разработкой фронтэнд-составляющей была разработана макет страниц веб-приложения.

Главная страница веб-приложения “Tuner” состоит из двух частей: первая — окно с приложением, вторая часть содержит в себе инструкцию по использованию. Макет первой страницы представлен на рисунке 3.

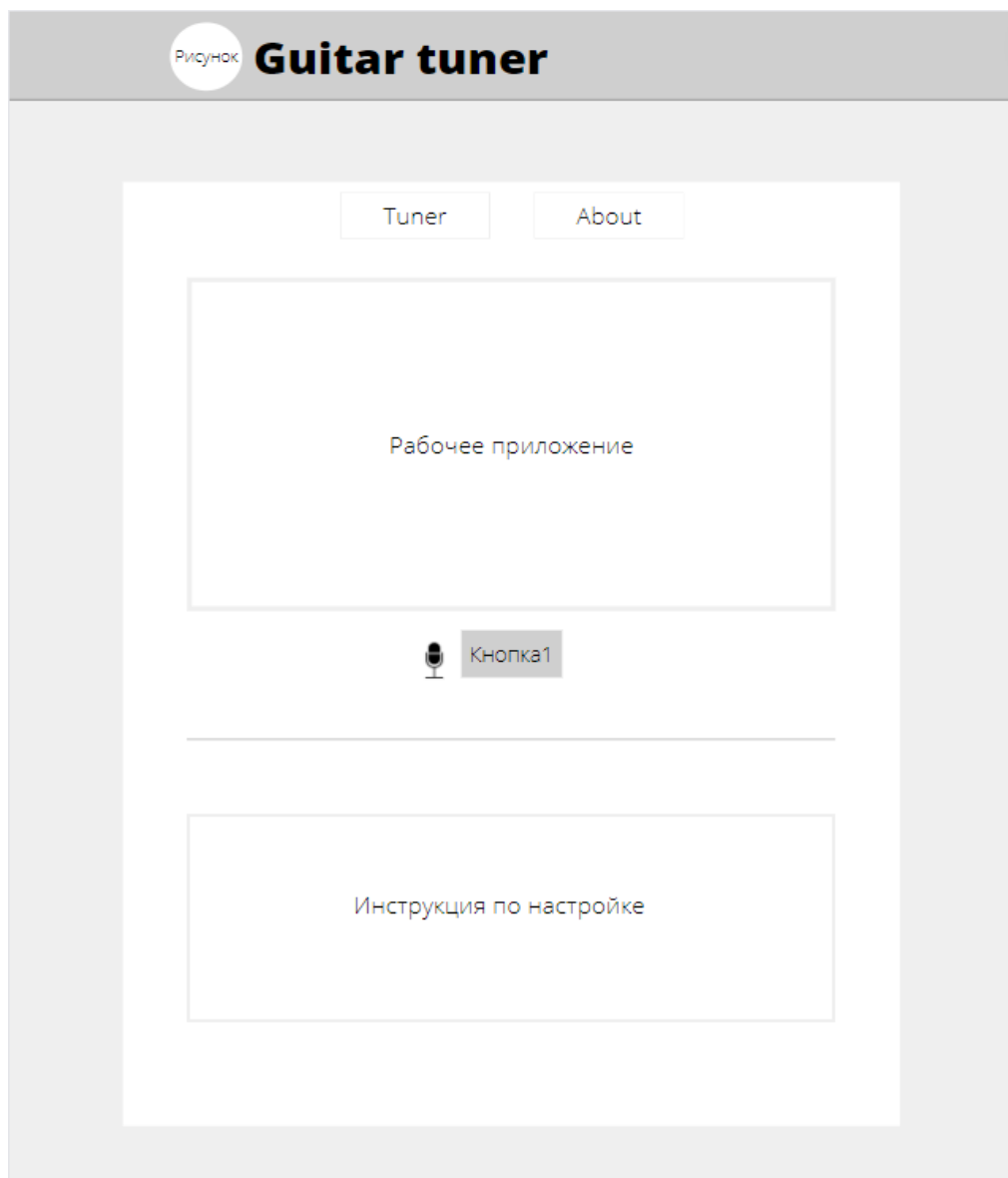


Рис. 3 – Главная страница “Tuner”

Вторая страница — “About”, в ней содержится справочник с информацией о устройстве струн и дополнительная справка по работе самого приложения.

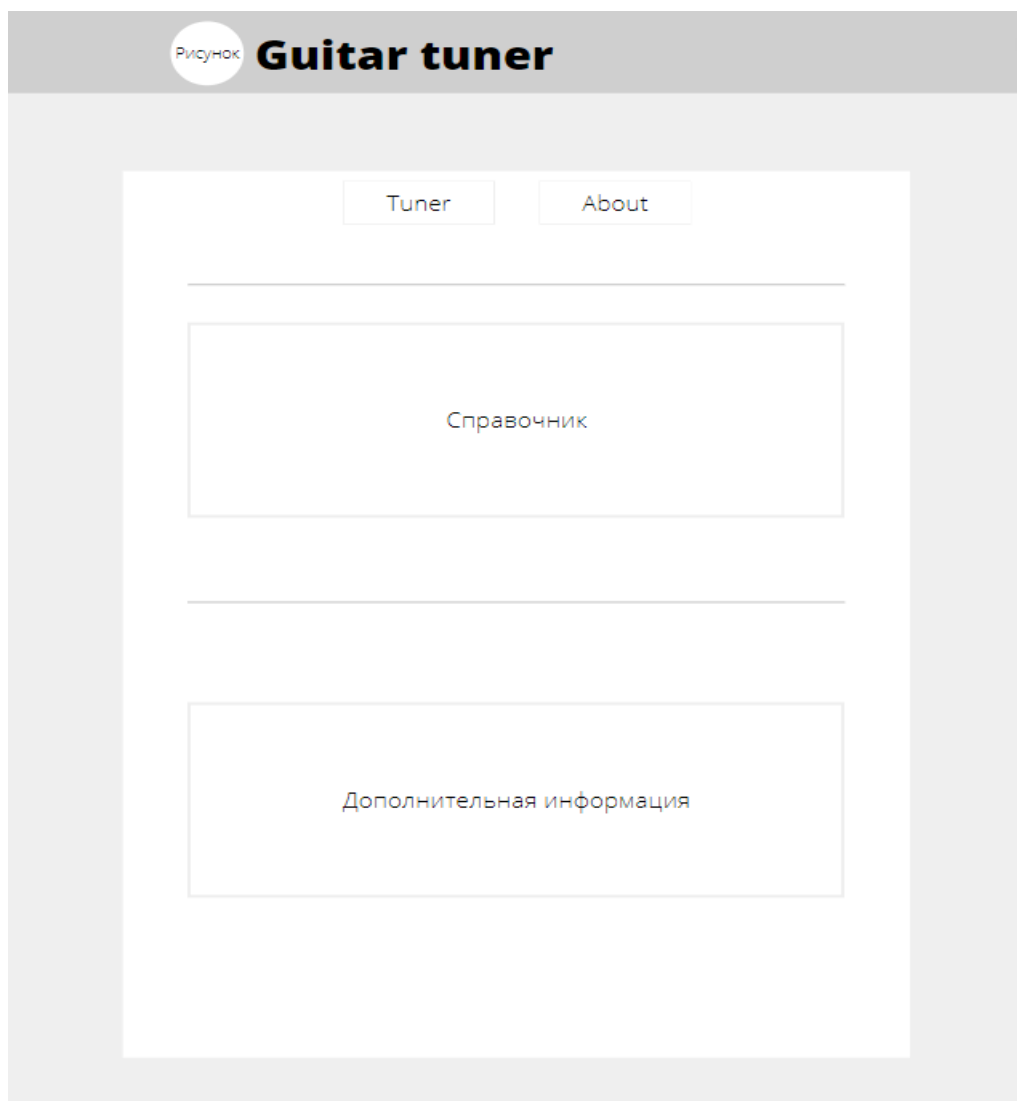


Рис. 4 – Страница “Справочник”

HTML верстка

Следующий шаг заключается в верстке макета.

Вёрстка — это преобразование созданных графических макетов страниц в html-код, который должен отображаться в Internet-браузере в точном соответствии с исходным макетом [18-19].

В данной работе была выбрана блочная модель верстки сайта с помощью тега `div`. Блоком является прямоугольная область, обладающая рядом свойств, таких как: `margin`, `padding` и `border` (рис. 5).

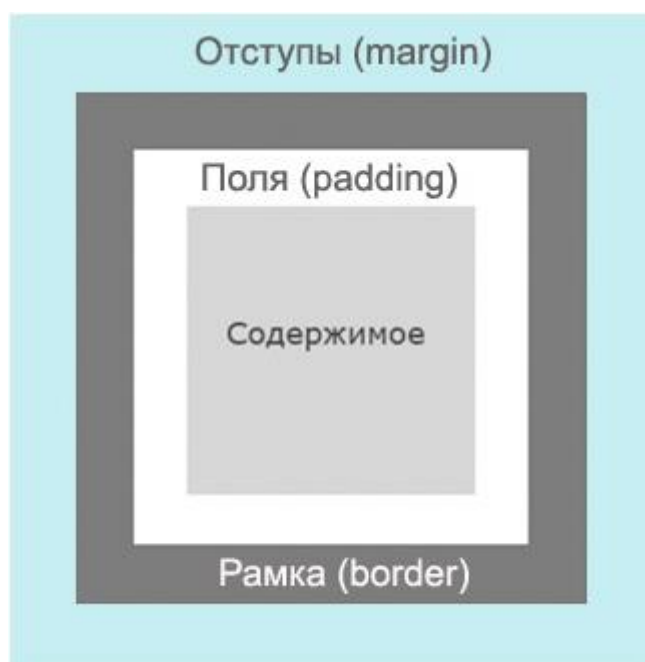


Рис. 5 - Блок

Страницы, сверстанные с помощью блочной верстки, значительно меньше весят по сравнению с табличной версткой [11].

Схема страницы, которая включает в себя шапку, блок контента и поле навигации представлены на рисунке 6.

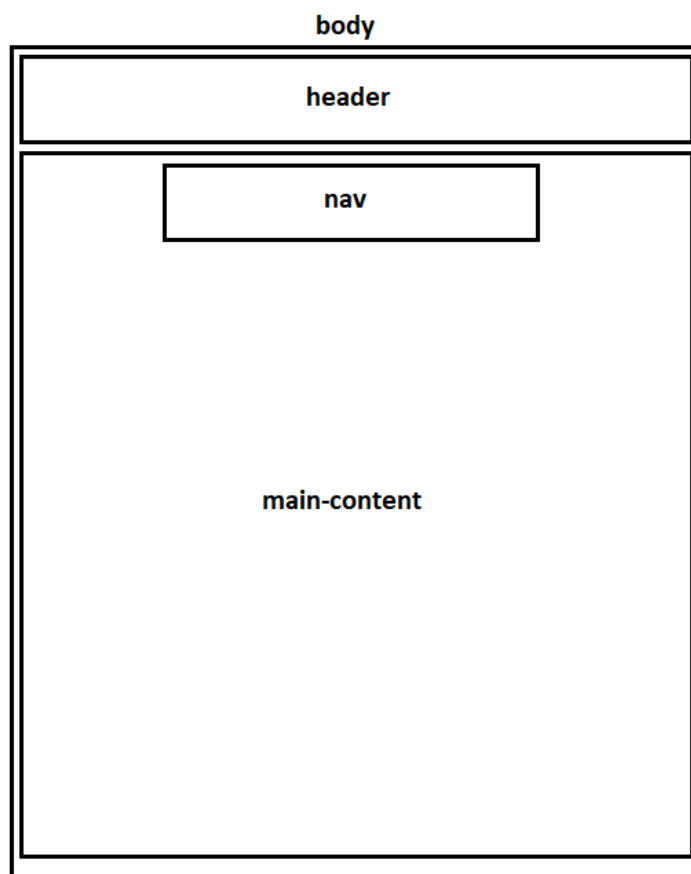


Рис. 6 – Макет структуры HTML

В шапке сайта будет располагаться логотип, в навигационном блоке ссылка на другую страницу. Основной блок используется для размещения приложения и вывода информации.

Для верстки макета используются следующие основные блоки: `body`, `header`, `main-content`, `nav content`

- **body** используется как контейнер для всего содержимого.
- **header**: шапка сайта.
- **nav**: блок навигации.
- **main-content**: основная информация и контент.

Каждый из этих блоков является контейнером `DIV`, блоки должны идти друг за другом.

Программирование

Для реализации программного тюнера будет использоваться язык программирования JavaScript. Визуализация процесса настройки будет осуществляться благодаря графическому элементу в HTML <canvas>, данный элемент предназначен для создания растровой графики с помощью JavaScript [12].

Программный тюнер должен улавливать сигнал гитары с поступающего звука из микрофона, анализировать его, после чего сверять с эталонным сигналом и показывать при помощи отрисованного индикатора на холсте величину расхождения.

Рабочий холст будет отображать фортепианную шкалу с шестью открытыми гитарными струнами: E2, A2, D3, G3, B3, E4 (см. Рисунок 7).

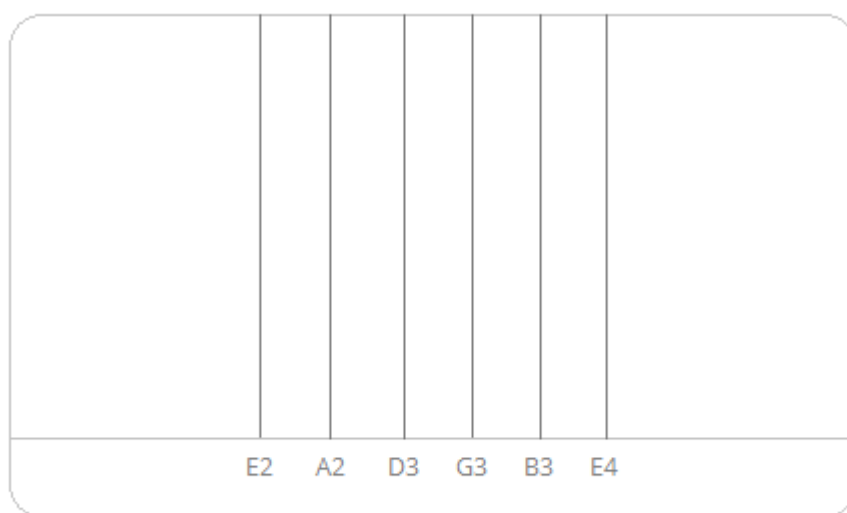


Рис. 7 Макет рабочего холста

Написанный код будет обрабатывать поступающий звук с микрофона и выводить амплитудно-частотную характеристику фильтра Гёрцеля.

Выводы по первой главе

1. Проведен анализ предметной области, где дана характеристика веб-приложению и сущности гитарного тюнера, определен и выбран вид программного продукта.
2. Описано устройство программного тюнера, определена математическая модель на основе алгоритма Гёрцеля.
3. Проанализированы и описаны необходимые инструменты по разработке веб-приложения.
4. Выведены и составлены этапы по реализации программного продукта.

Глава 2. Проектирование и разработка веб-программы по настройке гитары.

2.1 Установка программной среды Visual Studio Code

2.1.1 Этапы установки

Установочные материалы находятся на официальном сайте разработчика и проводится стандартная процедура установки:

1. Скачивание исполняемого файла с официального сайта (см. Рисунок 8).

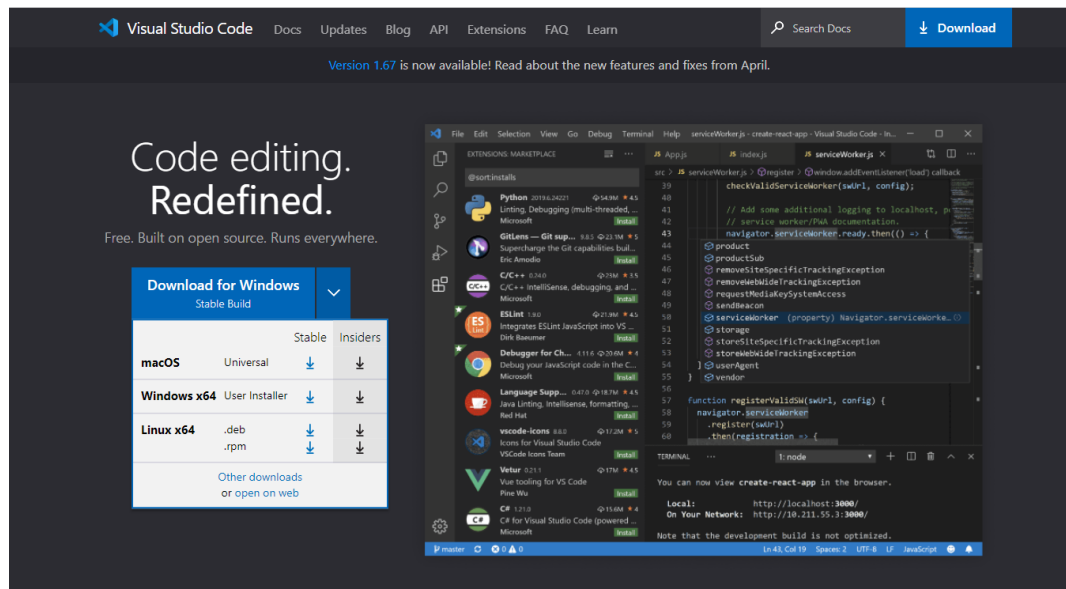


Рис. 8 установка среды

2. Запуск исполняемого файла для начала процесса установки.
3. Ознакомление и согласие с лицензией, в мастере установок, производится выбор необходимых инструментов, после чего начинается процесс установки.

2.1.2 Настройка среды и поиск необходимых расширений

После установки производится настройка среды. Необходимо осуществить отладку среды для корректной работы с кодом.

Для разработки сайта используются следующие расширения:

- HTML Snippets
- HTML CSS Support
- HTML Boilerplate
- JavaScript (ES6) code snippets
- Open in browser

HTML Snippets

Это расширение добавляет расширенную языковую поддержку для HTML-разметки в VS Code:

- Полные теги HTML5
- Маркировка тэгов
- Фрагменты
- Краткую информацию
- Описание, если тег устарел

HTML CSS Support

Расширение добавляет:

- HTML id и class завершения атрибутов.
- Поддержку связанных и встроенных таблиц стилей.
- Поддержку наследований шаблонов.
- Поддержку дополнительных таблиц стилей.
- Поддержку других HTML-подобных языков.
- Проверку селекторов CSS по запросу.

HTML Boilerplate

Это расширение предоставляет стандартный шаблон HTML-кода, используемый во всех веб-приложениях.

JavaScript (ES6) code snippets

Это расширение содержит фрагменты кода для JavaScript в синтаксисе ES6.

Open in browser

Данное расширение упрощает запуск проекта в необходимом браузере.

2.2 Разработка интерфейса сайта на основе его структуры и наполнение контентом

2.2.1 Файловая структура проекта

В корневой папке проекта содержится 3 каталога: `css`, `images`, `scripts` и 2 html файла страниц: `index`, `about`, (см. Рисунок 9).

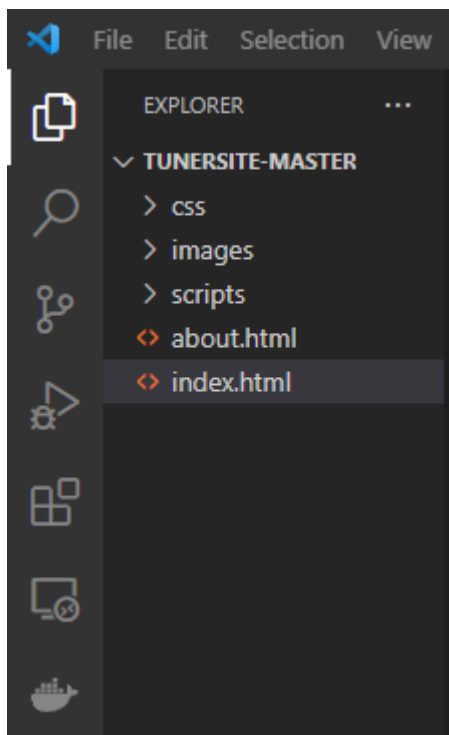


Рис. 9 – файловая структура.

В `css` каталоге содержатся стили, в `images` картинки сайта, в `scripts` код основной программы.

2.2.2 Код интерфейса сайта

На основе разработанного макета, составлен скелет html кода в блочном стиле написания. Сайт состоит из двух страниц “Tune” и “About”

Структура страницы “Tune”

В тэге head на обеих страницах прописаны необходимые параметры для работы страницы, где указаны шрифты, ссылки на компоненты и титульное название страницы, (см. Рисунок 10).

```
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Guitar tuner</title>
8   <link rel="stylesheet" href="css/style.css">
9   <link rel="preconnect" href="https://fonts.googleapis.com">
10  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
11  <link href="https://fonts.googleapis.com/css2?family=Asap:ital@1&display=swap" rel="stylesheet">
12
13 </head>
```

Рис. 10 Параметры сайта

В теле страницы body создан скелет интерфейса страницы “Tune”, состоящий из шапки, навигационного меню и основного контента (см. Рисунок 11).

```
14 <body >
15
16   <header class="header">
17     <div class="logo">
18       <div class="spike-01">
19 >         <div class="logo-content">...
24       </div>
25       <div class="spike-02">
26 >         </div>
27     </div>
28   </div>
29 </header>
30 <main class="main-content" >
31   <div class="menu-section">
32     <nav>
33       <div class="top-nav">
34         <a class="active" href="index.html"> TUNE </a>
35         <a href="about.html"> ABOUT </a>
36       </div>
37     </nav>
38   </div>
39   <div class="tuner_container">
40     <canvas id="myCanvas" ></canvas>
41   </div>
42 >   <div class="buttons_container">...
49 </div>
50 >   <div class="instruction_container">...
85 </div>
86 </main>
87
88 <script src="scripts/tuner.js">
89 </script>
90 </body>
```

Рис. 11 код интерфейса “Tune”

В тэге класса header расположен элемент div лога сайта с классом logo, где размещается иконка и название сайта.

Тэг main класса main-content содержит в себе основное содержание страниц, в таблице приведены внутренние элементы.

Класс элемента div	Назначение
menu-section	отвечает за навигацию по сайту
tuner__container	содержит в себе рабочий холст программы тюнер
buttons_container	содержит две кнопки микрофон и запуск
instruction__container	содержит инструкцию по запуску программы

Структура страницы “About”

Вторая страница “About” содержит справочник по гитарной теории и дополнительные сведения по работе с приложением.




Тег main класса main-content отличается содержанием, в нем находятся иные элементы, приведенные в таблице.

Класс элемента div	Назначение
menu-section	отвечает за навигацию по сайту
manual__container	содержит в себе блок div класса manual-text, отвечающий за текст и упорядоченный список с дополнительной информацией

2.2.3 Стилизация интерфейса и его наполнение

В качестве шрифта для текста лого был выбран “Asap, sans-serif”, для основного текста используется шрифт "Times New Roman, Georgia, Serif"

Цветовая палитра сайта состоит из трех цветов [17]:

Цвет	hex Код	Элемент
	#d49a58e0	Цвет используется для шапки сайта
	#ffffff	Для основного контента
	#a7a7a78c	Свободные зоны

Для лого была разработана картинка с миниатюрной гитарой (см. Рисунок 12)



Рис. 12 – Лого

Следующий шаг, стилизация и размещение элементов на двух страницах, прописывание текста в блоках информации, (см. Рисунок 13).

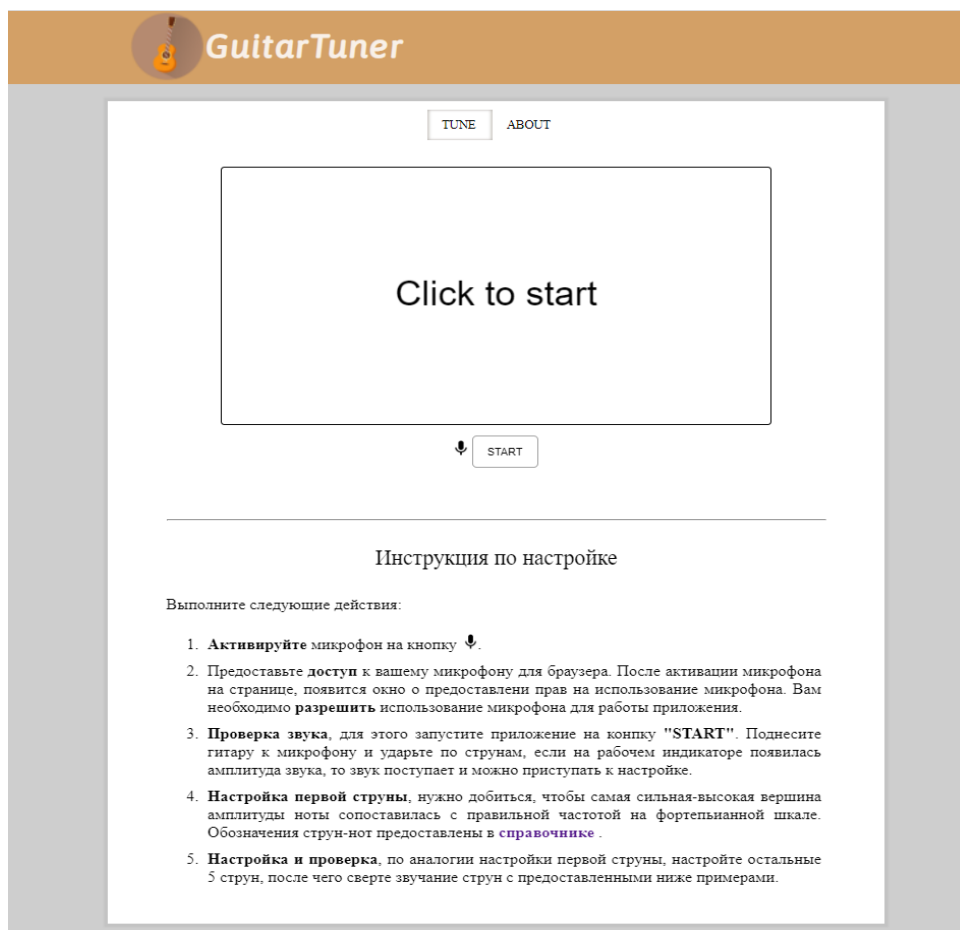


Рис. 13 – Наполнение страницы “Tune”.

На странице “About” справочник заполняется дорожками звучаний шести струн гитары и сводкой по наименованию струн (см. Рисунок 14).

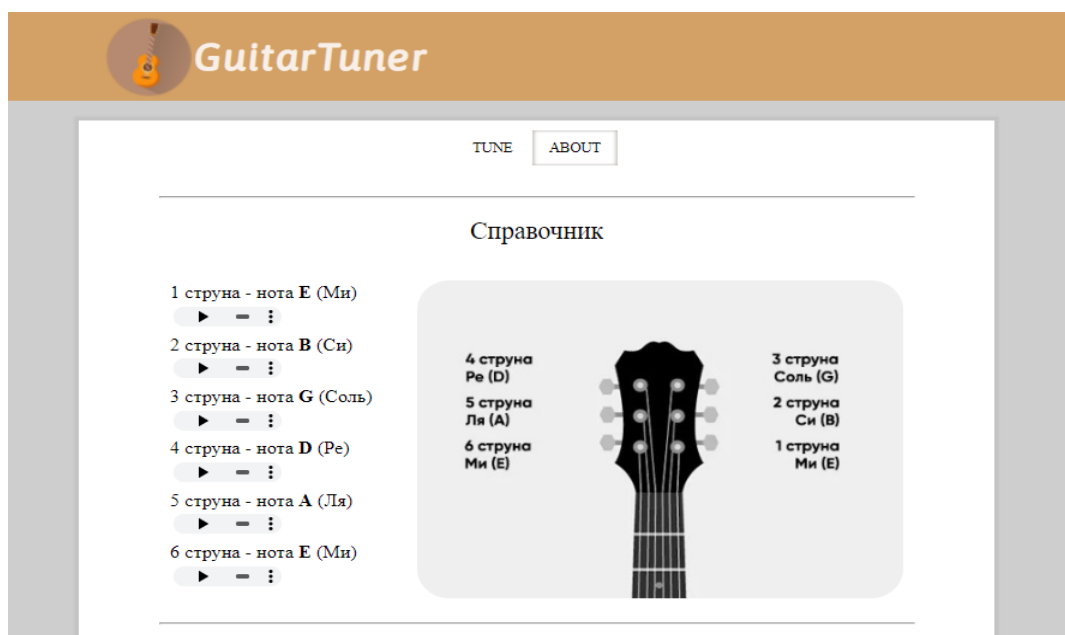


Рис. 14 – Справочник.

В разделе дополнительные сведения по работе с приложением подробно описывается принцип работы тюнера.

2.3 Разработка программы по настройке гитары

Программа будет получать звук с микрофона пользователя, обрабатывать его и выводить граф фильтра для определенной ноты на рабочий холст, пользователю останется только подстроить звучание определенной струны под эталонную частоту ноты.

Данный алгоритм Гёрцеля реализованный в функции, фильтрует полученный спектр звука [13-15], после чего возвращает самую сильную ноту входного звука (см. рис. 15).

```
17 function Goertzel(buffer, sampleRate, freq)
18 {
19     var w = 2 * 3.141592654 * freq / sampleRate;
20     var cr = Math.cos(w);
21     var ci = Math.sin(w);
22     var coeff = 2 * cr;
23
24     var sprev = 0;
25     var sprev2 = 0;
26
27     for(var n=0; n < buffer.length; n++)
28     {
29         var s = buffer[n] + coeff * sprev - sprev2;
30         sprev2 = sprev;
31         sprev = s;
32     }
33
34     var power = sprev2 * sprev2 + sprev * sprev - coeff * sprev * sprev2;
35
36     return power;
37 }
```

Рис. 15 — Функция алгоритма Гёрцеля

С помощью данной функции происходит отрисовка фортепианной шкалы, с отфильтрованным транслируемым звуком, на рабочий холст выводится отфильтрованная амплитуда звука, по которой пользователь может определить ноту (см. Рисунок 16).

```
110 function DrawPiano(buffer, sampleRate)
111 {
112     var noteFreqs = [ 329, 246, 196, 146, 110, 82];
113     var noteNumber = [ 20, 25, 30, 35, 39, 44 ];
114
115     canvasCtx.font="14px ti92pluspc";
116     canvasCtx.textAlign="center";
117     canvasCtx.beginPath();
118     canvasCtx.strokeStyle = 'rgb(255, 0, 0)';
119
120     for(var i=0; i < noteNumber.length; i++)
121     {
122         note = noteNumber[i]
123         var x = ( myCanvas.width / 66 ) * (note+1); // 88 66
124         canvasCtx.moveTo(x, 270);
125         canvasCtx.lineTo(x, 0);
126         canvasCtx.fillText(NoteNameFromNote(note),x,290);
127     }
128     canvasCtx.stroke();
129
130     canvasCtx.beginPath();
131     canvasCtx.strokeStyle = 'rgb(0, 0, 0)';
132     for(var note=0; note < 88; note++)
133     {
134         var x = (myCanvas.width / 66)*(note+1); // 88
135
136         var power = Math.sqrt(Goertzel(buffer, sampleRate, FreqFromNote(note)));
137
138         canvasCtx.moveTo(x, 270);
139         canvasCtx.lineTo(x, 270-power);
140     }
141
142     canvasCtx.moveTo(0, 271);
143     canvasCtx.lineTo(600, 271);
144
145     canvasCtx.stroke();
146 }
```

Рис. 16 — Функция фортепианная шкалы

Вспомогательная функция вычисления частоты ноты (см. Рисунок 17).

```
105 function FreqFromNote(note)
106 {
107     return 440.0 * Math.pow(2, (note-49) / 12);
108 }
```

Рис. 17 — Функция частоты ноты

Отрисовка эталонной частоты открытых нот струн гитары (см. Рисунок 18).

```
69 function drawGuitar(buffer, sampleRate)
70 {
71     canvasCtx.beginPath();
72     for(var s=0; s<6 ;s++)
73     {
74         var x = (myCanvas.width / 7 )*(s+1);
75         canvasCtx.moveTo(x, 300);
76         canvasCtx.lineTo(x, 0);
77     }
78     canvasCtx.stroke();
79
80     var freqs = [ 329, 246, 196, 146, 110, 82];
81     canvasCtx.beginPath();
82
83     for(var s=0;s<6;s++)
84     {
85         var x = (myCanvas.width/7)*(s+1);
86         for(var f =-10; f < 10 ; f++)
87         {
88             var power = Math.sqrt(Goertzel(buffer, sampleRate, freqs[s] + f*2));
89
90             canvasCtx.moveTo( x + f*3 , 300);
91             canvasCtx.lineTo( x + f*3 , 300-power);
92         }
93     }
94     canvasCtx.stroke();
95 };
```

Рис. 18 – Функция отрисовки эталонной частоты.

2.4 Размещение сайта на GitHub Pages

Чтобы разместить разработанное веб-приложение на хостинг, был создан аккаунт на GitHub, после чего был создан репозиторий под названием GuitarTuner. В созданный репозиторий был размещен разработанный проект, содержащий в себе исходный код веб-сайта (см. Рисунок 19).

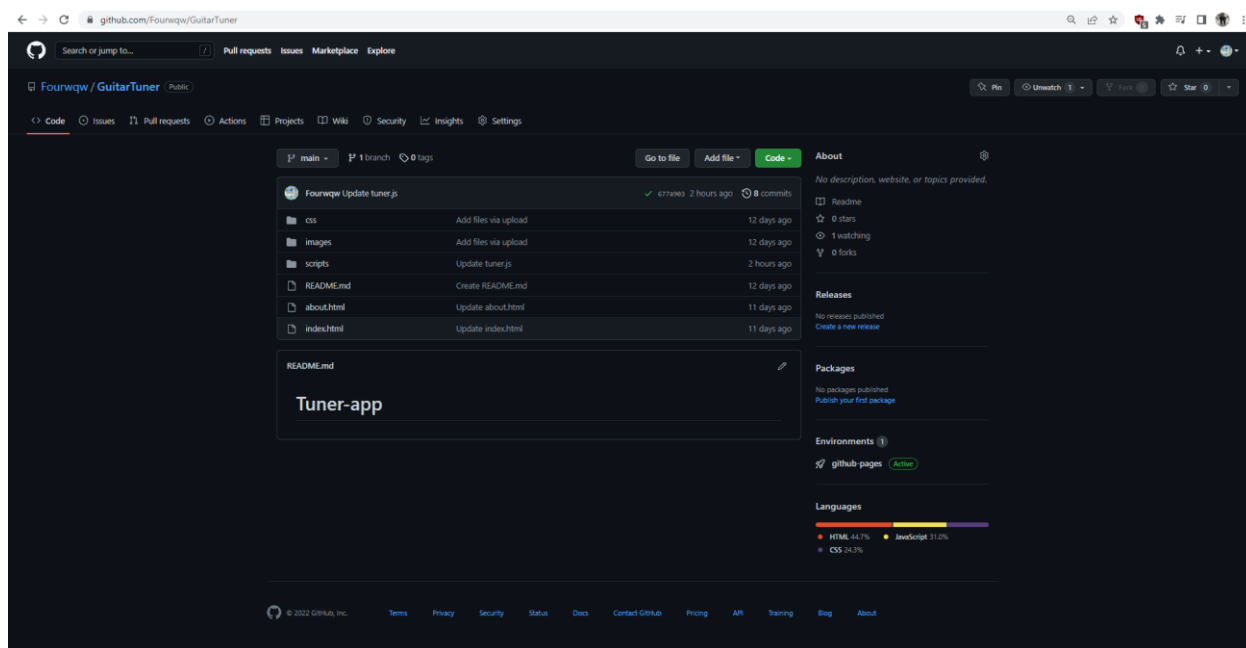


Рис. 19 – Размещенный проект в GitHub

Далее в настройках репозитория во вкладке GitHub Pages, разработанный проект был опубликован в сеть под доменом <https://fourwqw.github.io/GuitarTuner/index.html>. Размещенное готовое приложение на рисунке 20 в приложении 1.

Выводы по второй главе

1. Изучена и отлажена рабочая программная среда Visual Studio Code.
2. Спроектирована клиентская часть и разработана структура проекта.
3. Разработан рабочий холст для визуализации отфильтрованного звукового спектра, составлен справочник по настройке гитары.
4. Реализован алгоритм Гёрцеля для получения отфильтрованного звукового сигнала нот.
5. Проведено размещение готового веб-приложения на хостинг GitHub Pages.

ЗАКЛЮЧЕНИЕ

Цель проводимой работы заключалась в разработке доступного веб-приложения с альтернативной визуализацией процесса настройки гитары и составлением справочника по отладке музыкального инструмента.

Для достижения поставленной цели было сделано:

1. Проанализирована предметная область и существующий алгоритм Гёрцеля для детектирования и декодирования тональных сигналов;
2. Рассмотрены языки программирования и инструменты, предназначенные для создания веб-приложений;
3. Спроектировано и разработано веб-приложение, содержащее в себе справочник теории по настройке гитары и саму программу на основе выбранного алгоритма;
4. Изучен бесплатный сервис для хостинга GitHub Pages разработанного приложения;
5. Разработанный электронный ресурс был размещен на хостинге.

Таким образом проведена реализация всех задач данной ВКР.

Список литературы

1. Web-приложение: понятие, компоненты и принципы работы. — Текст: электронный // smartika : [сайт]. — URL: <https://smartika.ru/courses/web/> (дата обращения: 13.02.2022).
2. Гитарный тюнер виды и настройка. — Текст: электронный // Уроки гитары Одесса: [сайт]. — URL: <http://guitars.od.ua/vidy-gitarnykh-tyuneroi.html> (дата обращения: 14.02.2022).
3. Гитарный тюнер на языке C#. — Текст: электронный // vscode.ru : [сайт]. — URL: <https://vscode.ru/prog-lessons/gitarnyy-tyuner-na-yazyike-c.html> (дата обращения: 23.04.2022).
4. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов = Fast Algorithms for Digital Signal Processing. — М.: Мир, 1989. — 448 с.
5. Эммануил С. Айфичер, Барри У. Джервис. Цифровая обработка сигналов: практический подход = Digital Signal Processing: A Practical Approach. — 2-е издание. — М.: Издательский дом «Вильямс», 2004. — 992 с.
6. Агеев, Д. В. Гитара Уроки мастера для начинающих / Д. В. Агеев. — Санкт-Петербург: ОАО "Печатный двор" им. А. М. Горького., 2009. — 133 с.
7. Николай, Прохоренок Редактор Visual Studio Code. / Прохоренок Николай. — Текст: электронный // Прохоренок РФ. : [сайт]. — URL: <http://xn--e1akicaccic2c.xn--p1ai/pdf/vscode/> (дата обращения: 23.04.2022).
8. CSS — Текст: электронный // mdn web docs: [сайт]. — URL: <https://developer.mozilla.org/ru/docs/Web/CSS> (дата обращения: 29.11.2021).
9. Современный учебник — Текст: электронный // javascript.ru: [сайт]. — URL: <https://learn.javascript.ru/> (дата обращения: 24.11.2021).

10. Пошаговая инструкция: как сделать бесплатный сайт на GitHub Pages.
— Текст: электронный // reg.ru: [сайт]. — URL:
<https://www.reg.ru/blog/poshagovaya-instruktsiya-kak-sdelat-besplatnyj-sajt-na-github-pages/> (дата обращения: 10.05.2022).
11. CSS. Карманный справочник, 4-е изд.: Пер. с англ. - М. : ООО "И.Д. Вильямс 2016. - 288 с.: ил. - Парал. тит. англ. ISBN 978-5-8459-2081-2
12. HTML Canvas Graphics — Текст: электронный // w3schools: [сайт]. — URL: https://www.w3schools.com/html/html5_canvas.asp (дата обращения: 24.11.2021).
13. Леонидов, В. В. Конспект лекции «Алгоритм Гёрцеля» / В. В. Леонидов.
// Учебно-методический комплект по дисциплине «Цифровая обработка сигналов» . — : МГТУ имени Н.Э. Баумана, — 7 с.
14. Оппенгейм А., Шаффер Р. Цифровая обработка сигналов. Москва, Техносфера, 2012. 1048 с.
15. Алгоритм Гёрцеля. — Текст: электронный // DSPLIB: [сайт]. — URL: <https://ru.dsplib.org/content/goertzel/goertzel.html> (дата обращения: 21.05.2022).
16. Введение в CGI: Информация. — Текст: электронный // Интуит: [сайт]. — URL: <https://intuit.ru/studies/courses/36/36/info> (дата обращения: 10.05.2022).
17. «Иттен И. Искусство цвета»: Издатель Д. Аронов; М.; 2004 ISBN 5-94056-008-3
18. Основы web-дизайна, Методика проектирования, Учебное пособие, Нагаева И.А., Фролов А.Б., Кузнецов И.А., 2021. — 236 с.
19. Гото К., Котлер Э. Веб-редизайн: книга Келли Гото и Эмили Котлер: Пер. с англ. — СПб.: Символ-Плюс, 2007. — 376 с.
20. Управление расширениями для Visual Studio. — Текст : электронный // Microsoft : [сайт]. — URL: <https://docs.microsoft.com/ru-ru/visualstudio/ide/finding-and-using-visual-studio-extensions?view=vs-2019> (дата обращения: 10.01.2021).

Приложения

Приложение 1

Готовое и размещенное веб-приложение “гитарный тюнер”

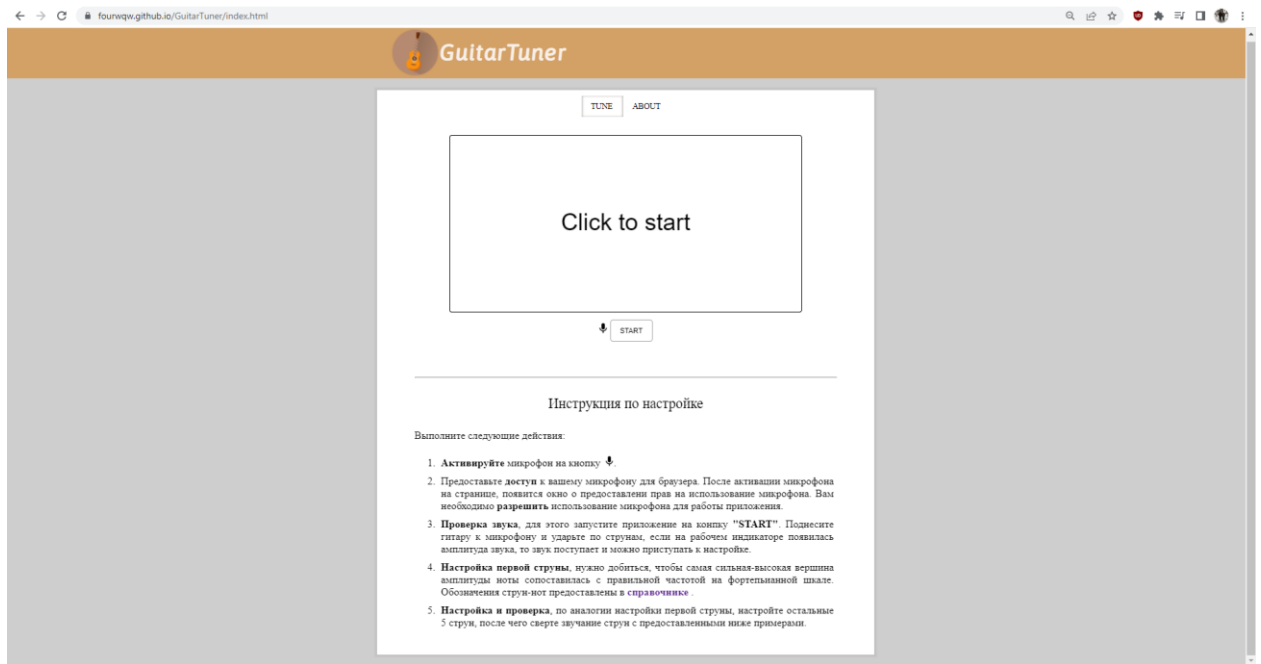


Рис. 20 – размещенное веб-приложение