

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждениевысшего образования

**«МИРЭА — Российский технологический университет»**

**РТУ МИРЭА**

|  |  |  |
| --- | --- | --- |
| **Институт искусственного интеллекта**  **Базовая кафедра № 252 «Информационная безопасность»** | | |
|  |

**КУРСОВАЯ РАБОТА**

по дисциплине «*Методы программирования*»

**Тема курсовой работы: Приложение медиа плеер**

**Студент группы: ККСО-01-20 КлювиткинА.Р. КлювиткинА.Р.**

*(учебная группа, фамилия, имя, отчество студента) (подпись студента)*

**Руководитель курсовой работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_**

*должность, звание, ученая степень (подпись руководителя)*

**Рецензент** (при наличии) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*должность, звание, ученая степень ( подпись рецензента)*

Работа представлена к защите «24» декабря 2022 г.

Допущен к защите «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 20\_\_г.

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Cодержание**

[Введение 3](#_Toc122457145)

[1. Теоретическая часть 4](#_Toc122457146)

[*1.1 Словарь терминов и сокращений 4*](#_Toc122457147)

[*1.2 Постановка задачи 5*](#_Toc122457148)

[*1.3 Выбор среды программирования 6*](#_Toc122457149)

[1.4 html5 7](#_Toc122457150)

[*1.5 JavaScript 9*](#_Toc122457151)

[2. Практическая часть 12](#_Toc122457152)

[2.1 Воспроизведение видео на странице 12](#_Toc122457153)

[2.2 Изменение громкости 13](#_Toc122457154)

[2.3 Функция перемотки 13](#_Toc122457155)

[2.4 Синхронизация видеопотоков 14](#_Toc122457156)

[2.5 Синхронная переход по таймкоду 15](#_Toc122457157)

[2.6 Загрузка видео 15](#_Toc122457158)

[2.7 Стили 16](#_Toc122457159)

[Заключение 17](#_Toc122457160)

[Приложение 18](#_Toc122457161)

# Введение

В данной курсовой работе рассмотрим web-приложение медиа плеер, реализованный на JavaScript и HTML5. Будет рассмотрена необходимость и полезность такого приложения в реальном мире. Также будут изучены технологии, которые применяем для работоспособности такого приложения и рассмотрен программный код.

Курсовая работа состоит из двух глав, которые разбирают теоретическую часть разрабатываемого приложения и практическую, в котором будет показана его работа. Итоговой целью является создать медиа плеер с функцией воспроизведения до 3 видео одновременно.

# Теоретическая часть

## Словарь терминов и сокращений

|  |  |
| --- | --- |
| Таймкод | Последовательность числовых кодов, генерируемых через регулярные промежутки времени системой синхронизации времени |
| JS | JavaScript |
| HTML | HyperText Markup Language |
| Парадигма | Совокупность идей и понятий, определяющих стиль написания компьютерных программ (подход к программированию). |
| Archimate 3.0 | Самый современный и самый популярный стандарт архитектурного подхода и языка описания архитектуры. |
| Фрейм | Элемент ([виджет](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81%D0%B0" \o "Элемент интерфейса)) [графического интерфейса пользователя](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F), который является контейнером для других объектов. |
| Cookie | Небольшой фрагмент данных, отправленный [веб-сервером](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) и хранимый на [компьютере](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) пользователя. |
| User-friendly | Удобный для пользователя |
| Прогрессбар | Индикатор, который отображает прогресс выполнения задачи |

*Таблица 1. Словарь терминов и сокращений*

## Постановка задачи

Прежде чем перейти непосредственно к разработке приложения, нужно корректно сформулировать постановку задачи, разбить её на подзадачи. Для грамотного функционирования медиа плеера, понадобиться реализовать несколько основных функций:

1)Воспроизведение видео из файла на web-странице

2)Добавление дополнительно до двух видеофрагментов на страницу

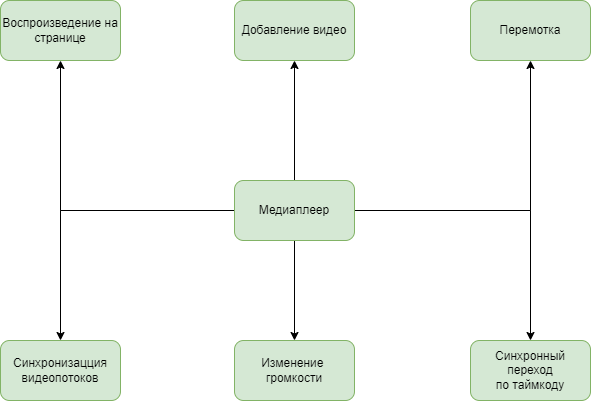
3)Синхронизация видеопотоков

4)Реализация функции перемотки

5)Реализация функции изменения громкости

6)Реализация функции синхронного перехода по таймкоду

Далее во всей работе для визуализации диаграмм использовано приложение draw.io, в частности стандартная нотация и нотация archimate 3.0.



*Рисунок 1.* [*Декомпозиция*](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%B7%D0%B8%D1%86%D0%B8%D1%8F) *задачи*

## Выбор среды программирования

В процессе выбора языка программирования для разработки медиа плеера, было решено использовать JavaScript, соответственно было принято решение, что плеер будет реализован как web-приложение с использованием языка разметки HTML5. Данный выбор обусловлен большим количеством уже существующих классов и методов, которые помогут в разработке приложения. Так же сильным аргументом в пользу выбора JavaScript был мой опыт работы с этим языком программирования и опыт в web-разработке.

# 1.4 html5

HTML(HyperText Markup Language)5 – язык разметки web-страницы. Веб-браузеры получают HTML документ от сервера по протоколам [HTTP](https://ru.wikipedia.org/wiki/HTTP)/[HTTPS](https://ru.wikipedia.org/wiki/HTTPS) или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора. [Элементы HTML](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D1%8B_HTML) являются строительными блоками HTML страниц. С помощью HTML разные конструкции, изображения и другие объекты, такие как [интерактивная веб-форма](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0_(HTML)), могут быть встроены в отображаемую страницу. HTML предоставляет средства для создания заголовков, абзацев, списков, ссылок, цитат и других элементов. Элементы HTML выделяются тегами, записанными с использованием угловых скобок. В HTML можно встроить программный код на языке программирования [JavaScript](https://ru.wikipedia.org/wiki/JavaScript), для управления поведением и содержанием веб-страниц. Также включение [CSS](https://ru.wikipedia.org/wiki/CSS) в HTML описывает внешний вид и макет страницы.

HTML5 – последняя и самая мощная на данный момент версия стандарта HTML. В HTML5 реализовано множество новых синтаксических особенностей. Например, элементы <video>, <audio> и <canvas>, а также возможность использования [SVG](https://ru.wikipedia.org/wiki/SVG) и [математических формул](https://ru.wikipedia.org/wiki/MathML). Эти новшества разработаны для упрощения создания и управления графическими и мультимедийными объектами в сети без необходимости использования сторонних [API](https://ru.wikipedia.org/wiki/API) и [плагинов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%B3%D0%B8%D0%BD). HTML5 вводит несколько новых [элементов](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D1%8B_HTML) и атрибутов, которые отражают типичное использование разметки на современных [веб-сайтах](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82). Некоторые из них — семантические замены для использования универсальных блочных (<div>) и строчных (<span>) элементов, например, <nav> (блок навигации по сайту), <footer> (обычно относится к нижней части страницы или последней строке HTML кода) или <audio> и [<video>](https://ru.wikipedia.org/wiki/HTML5_video) вместо <object>. Некоторые устаревшие элементы, которые можно было использовать в [HTML 4.01](https://ru.wikipedia.org/wiki/HTML), были исключены, включая чисто оформительские элементы, такие как <font> и <center>, чьи эффекты выполняются с помощью [каскадных таблиц стилей](https://ru.wikipedia.org/wiki/CSS). Также в поведении гипертекстовых страниц снова заострено внимание на важности скриптов [DOM](https://ru.wikipedia.org/wiki/Document_Object_Model) (например, [Javascript](https://ru.wikipedia.org/wiki/Javascript" \o "Javascript)).

## JavaScript

Javascript – динамический скриптовый язык программирования высокого уровня. Он отличается мультипарадигменностью. Речь идет о поддержке функционального, императивного, событийно-ориентированного стилей. Чаще всего язык используется для создания интерактивных веб-страниц и приложений. Современный Java скрипт это «безопасный» язык программирования. Он не предоставляет низкоуровневых средств работы с памятью, процессором из-за того, что изначально был ориентирован на браузеры, в которых это не требуется. Основные архитектурные черты: [динамическая типизация](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), [слабая типизация](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%B0%D0%B1%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), [автоматическое управление памятью](https://ru.wikipedia.org/wiki/%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0_%D0%BC%D1%83%D1%81%D0%BE%D1%80%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [прототипное программирование](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5" \o "Прототипное программирование), функции как [объекты первого класса](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_%D0%BF%D0%B5%D1%80%D0%B2%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0). На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java. Структурно JavaScript можно представить в виде объединения трёх чётко различимых друг от друга частей:

ядро (ECMAScript),

объектная модель браузера ([Browser Object Model или BOM](https://ru.wikipedia.org/w/index.php?title=Browser_Object_Model&action=edit&redlink=1" \o "Browser Object Model (страница отсутствует))[[en]](https://en.wikipedia.org/wiki/Browser_Object_Model)),

объектная модель документа ([Document Object Model или DOM](https://ru.wikipedia.org/wiki/Document_Object_Model" \o "Document Object Model)).

Объектную модель документа иногда рассматривают как отдельную от JavaScript сущность, что согласуется с определением как независимого от языка интерфейса документа.

**Ядро** [**ECMAScript**](https://ru.wikipedia.org/wiki/ECMAScript)

ECMAScript не является браузерным языком и в нём не определяются методы ввода и вывода информации. Это, скорее, основа для построения скриптовых языков. Спецификация ECMAScript описывает типы данных, инструкции, ключевые и [зарезервированные](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D1%80%D0%B5%D0%B7%D0%B5%D1%80%D0%B2%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D1%81%D0%BB%D0%BE%D0%B2%D0%BE) слова, [операторы](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), объекты, [регулярные выражения](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F), не ограничивая авторов производных языков в расширении их новыми составляющими.

**Объектная модель браузера**

Объектная модель браузера — браузер-специфичная часть языка, являющаяся прослойкой между ядром и объектной моделью документа. Основное предназначение объектной модели браузера — управление окнами браузера и обеспечение их взаимодействия. Каждое из окон браузера представляется объектом window.

Помимо управления окнами, в рамках объектной модели браузера, браузерами обычно обеспечивается поддержка следующих сущностей:

* управление [фреймами](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC_(HTML)),
* поддержка задержки в исполнении кода и зацикливания с задержкой,
* системные диалоги,
* управление адресом открытой страницы,
* управление информацией о браузере,
* управление информацией о параметрах [монитора](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%80_(%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE)),
* ограниченное управление историей просмотра страниц,
* поддержка работы с [HTTP cookie](https://ru.wikipedia.org/wiki/HTTP_cookie).

Объектная модель документа

Объектная модель документа — [интерфейс программирования приложений](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) для HTML и [XML](https://ru.wikipedia.org/wiki/XML)-документов. Документ (например, веб-страница) может быть представлен в виде дерева объектов, обладающих рядом свойств, которые позволяют производить с ним различные манипуляции:

* генерация и добавление узлов,
* получение узлов,
* изменение узлов,
* изменение связей между узлами,
* удаление узлов.

Встраивание в веб-страницы

Для добавления JavaScript-кода на страницу можно использовать теги <script></script>, которые рекомендуется, но не обязательно, помещать внутри контейнера <head>. Контейнеров <script> в одном документе может быть сколько угодно.

Скрипт, выводящий [модальное окно](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%BE%D0%BA%D0%BD%D0%BE) с классической надписью «Hello, World!» внутри браузера:

<**script** type="application/javascript">

alert('Hello, World!');

</**script**>

# 2. Практическая часть

В данном разделе курсовой работы будут описаны решения задач, полученных в ходе декомпозиции задачи в пункте 1.2

Многие функции которые требуют синхронизации видео являются асинхронными, чтобы уменьшить задержку между переключениями. Можно сказать что мы «отслаиваем» функцию в другой поток.

# 2.1 Воспроизведение видео на странице

Для отображения окна видео плеера в браузере использован div video-player, в котором в последующем будут находится нужные кнопки . Функция воспроизведения и паузы реализованы с помощью встроенных в JS методов play и pause. Для добавления еще двух макетов видеоплеера были добавлены клоны данного div.

Посмотреть наполнение файла index.html можно перейдя по ссылке в приложении к данной курсовой работе.

Были добавлены кнопки play и pause для каждого окна плеера.

async function toggelPlay(video) {

  videosPlayState ? video.play() : video.pause();

}

*Листинг 1*

# 2.2 Изменение громкости

Для изменения громкости используем обработчик событий(callback). Сначала мы отобразим прогрессбар, после чего будем обрабатывать каждое его изменение и соответственно менять громкость с помощью встроенных методов. Данная функция была реализована сразу для трех макетов плеера.

for (let volume\_slider of volume\_sliders) {

  volume\_slider.addEventListener("change", () => {

    volume\_slider.closest(".video-player").children[0].volume = volume\_slider.value;

  });

};

*Листинг 2*

# 2.3 Функция перемотки

Аналогично с изменением громкости, был добавлен прогресс бар и добавлена функция визуализации прогресса воспроизведения видео.



*Рисунок 2. Панель навигации плеера*

for (let video of videos) {

  video.addEventListener("timeupdate", () => {

    const progressPercentage = (video.currentTime / video.duration) \* 100;

    video.closest(".video-player").children[1].children[0].children[0].style.flexBasis = `${progressPercentage}%`;

  });

};

*Листинг 3*

Так же реализован переход на таймкод по клику на прогрессбар.

for (let progressBar of progressBars) {

  progressBar.addEventListener("click", (e) => {

    const scrubTime = (e.offsetX / progressBar.offsetWidth) \* progressBar.closest(".video-player").children[0].duration;

    progressBar.closest(".video-player").children[0].currentTime = scrubTime;

  });

}

*Листинг 4*

# 2.4 Синхронизация видеопотоков

Для синхронного воспроизведения видео была реализована функция, которая меня состояние видео для каждого окна плеера, затем другая функция(из пункта 2.1) при смене этого состояния воспроизводит и ставить на паузу видео соответственно.

function toggleAllPlay() {

  videosPlayState = !videosPlayState;

  for (let video of videos) {

    toggelPlay(video);

    changePlayIcons();

  };

}

*Листинг 5*

Привязка кнопок к данной функции

for (let playButton of playButtons) {

  playButton.addEventListener("click", toggleAllPlay);

  mainPlayButton.addEventListener("click", toggleAllPlay);

}

*Листинг 6*

# 2.5 Синхронный переход по таймкоду

Для реализации синхронной перемотки было принято решение создать отдельный объект для ввода таймкода.

<input style="margin-left: 10px; width: 80%;" type="time" id="syncTime" step="1" value="00:00:00">

*Листинг 7*

Далее обрабатываем нажатие на кнопку Sync, переводим всё время в секунды и перематываем все видео по этому таймкод.

syncButton.addEventListener("click", () => {

  for (let video of videos) {

    let [hours, minutes, seconds] = syncTimeInput.value.split(":");

    video.currentTime = (+hours \* 60 \* 60) + (+minutes \* 60) + (+seconds);

  };

});

*Листинг 8*

# 2.6 Загрузка видео

В качестве расширения функционала приложения была добавлена функция выбора видео прямо из него. Теперь чтобы воспроизвести желаемое видео не нужно указывать путь до нужного файла внутри кода программы, а просто выбрать его в всплывающем меню, что делает наше web-приложение более user-friendly.

[file1, file2, file3].forEach((file, index) => {

  file.addEventListener("change", (e) => {

    const file = e.target.files[0];

    const video = videos[index];

    const url = URL.createObjectURL(file);

    video.children[0].src = url;

    video.load();

  });

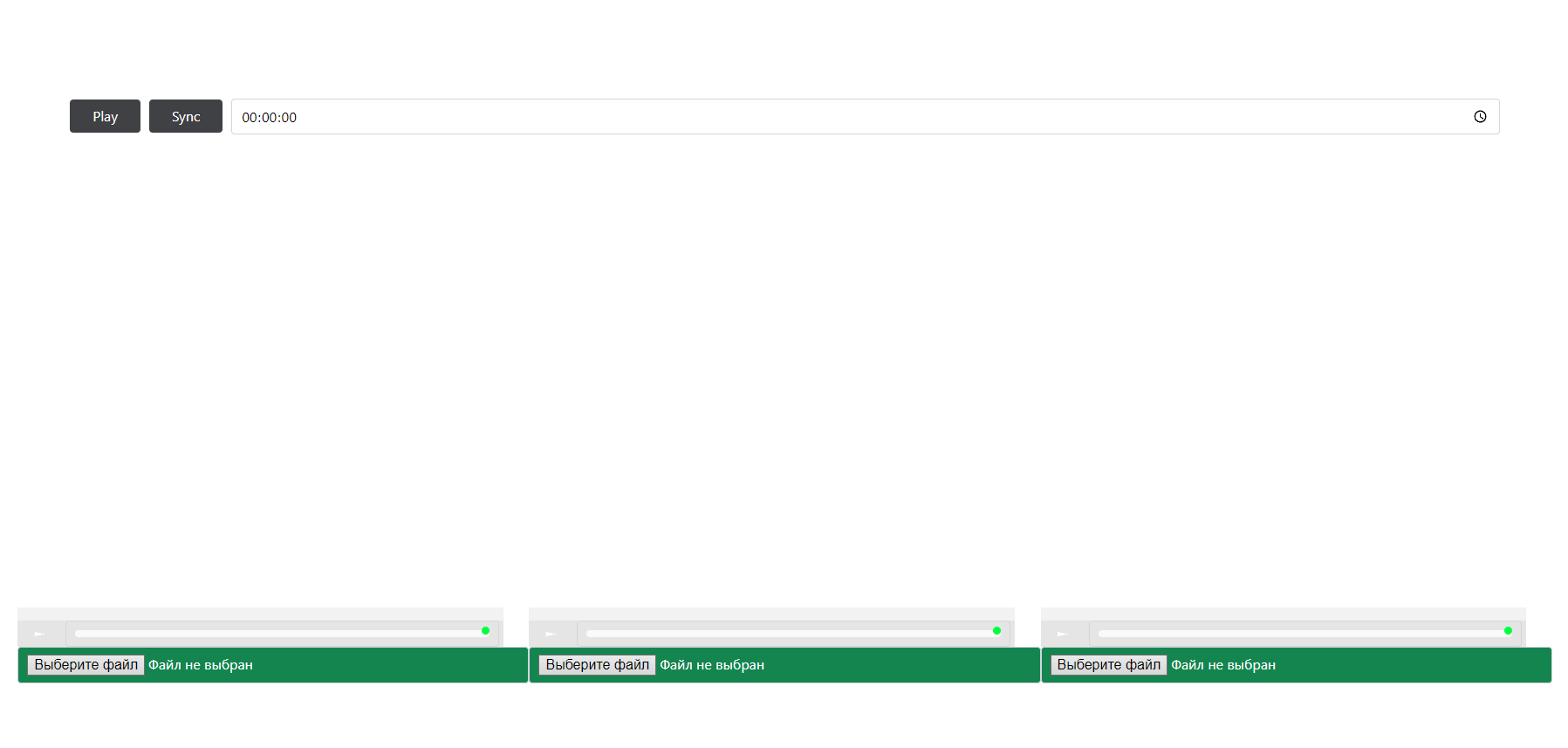
});

*Листинг 9*

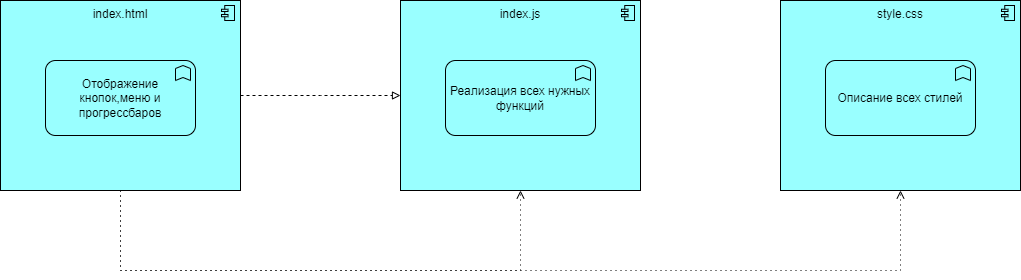
# 2.7 Стили

Все стили описаны в файле style.css , так же был использован css framework chota, а конкретно были взяты несколько стилей для кнопок и слайдеров.

Финальный визуал приложения выглядит данным образом:



*Рисунок 3. Финальный вид приложения*

* Рисунок 4. Компонентная-функциональная диаграмма*

# Заключение

В ходе работы было рассмотрено web-приложение медиа плеер, основные его функции. Так же были изучены новые методы JS и более детально изучен язык разметки html в частности html5.

# Приложение

С исходным кодом можно ознакомиться по ссылке: <https://github.com/Kl-Alex/mediaplayer>

Приложение доступно по ссылке:

<https://kl-alex.github.io/mediaplayer/>