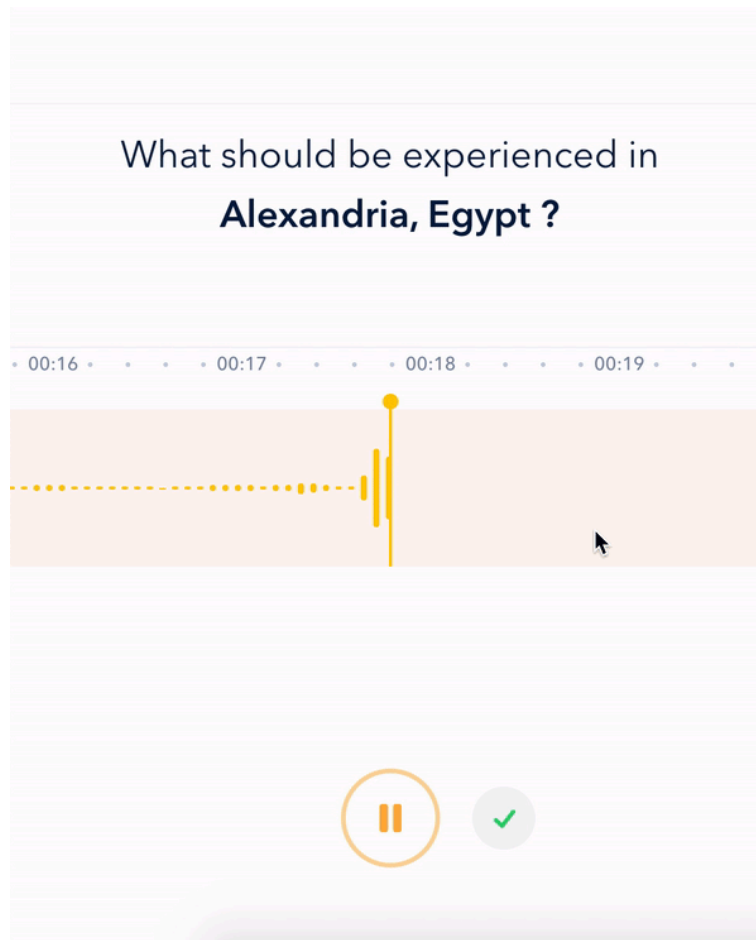


Requirements:

- Milestones 1-4 should be made on local blank React.js project
 - The code for every milestone should be well written and well commented.
 - The code should be well optimized and be well working on the audios with duration > 10 minutes
-

MILESTONE 1**1. Audio recording without images**

Gif 1.

1.1. Visualize [MediaStream](#) produced by users microphone according to Gif 1. (with the track where audio waves appear, timeline above that track, line in the center of the track and moving effect)

```
//Code snippet
const recordedChunks = [];
navigator.mediaDevices.getUserMedia({ audio: true }).then(mediaStream => {
const mediaRecorder = new MediaRecorder(mediaStream);

mediaRecorder.addEventListener('dataavailable', function(e) {
  if (e.data.size > 0) {
    recordedChunks.push(e.data);
  }
});

mediaRecorder.start();

// Do something with mediaStream to visualize it.
});
```

1.2. When MediaStream is stopped (mediaRecorder.stop()) or paused (mediaRecorder.pause()) or resumed (mediaRecorder.resume()) visualization should stop / resume accordingly.

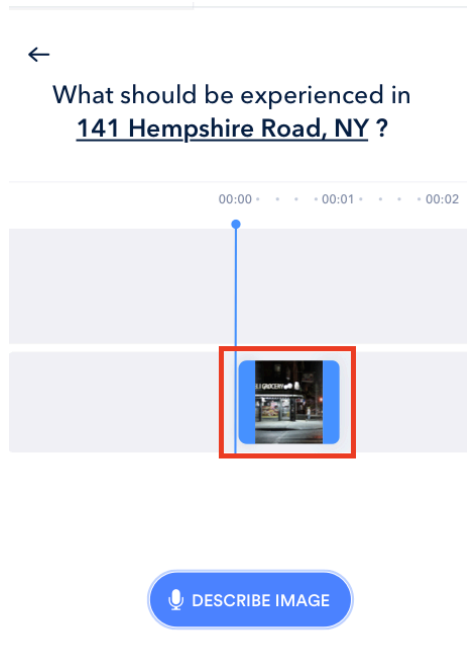
1.3 Time on the timeline should be well synchronized with actual audio captured by mediaRecorder. (therefore time should be measured by getting actual duration of an audio, not by using setInterval or some other method)

MILESTONE 2

2. Recording audio with images

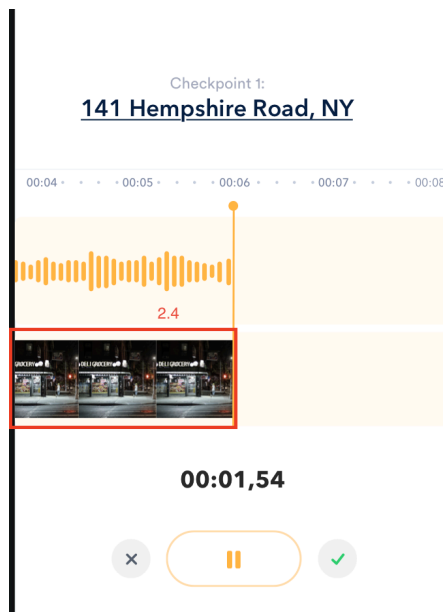
2.1. Providing the user with the ability to upload images (one at a time). Images should be represented by objects with the following keys: base64, startTime (audio timepoint in milliseconds at which image display begins), endTime (audio timepoint in milliseconds at which image display stops).

2.2. When the image is uploaded, it should be displayed on the track below the audio waves track, on the right side of the line. (see Screenshot 1)



Screenshot 1.

2.3 When recording starts image appears on the left side and starts to repeat itself until the recording is stopped (see Screenshot 2)



Screenshot 2.

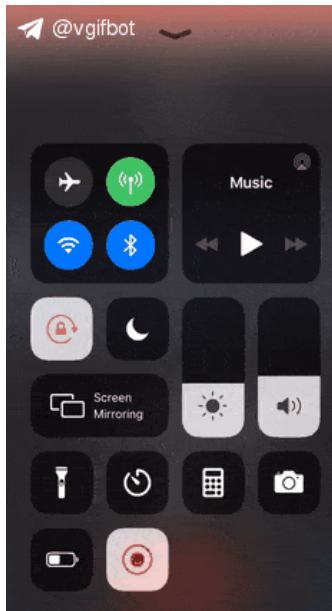
MILESTONE 3

3. Audio recording isn't active

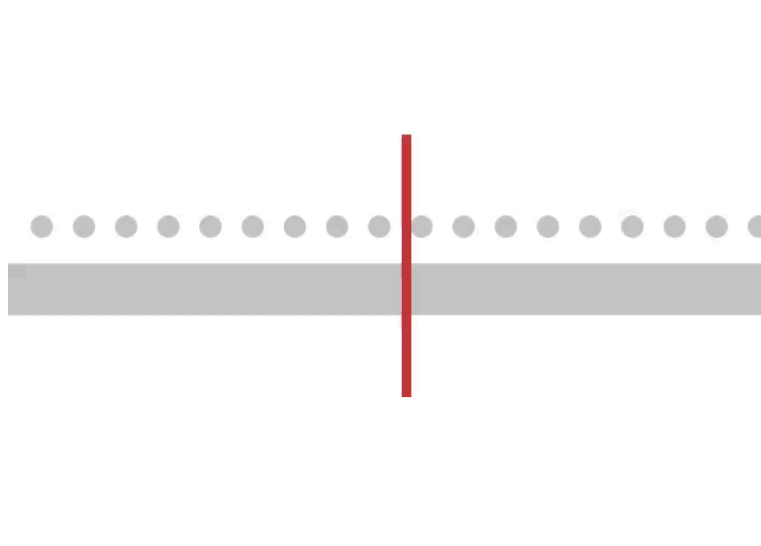
3.1. Both tracks (audio wave track and image track) and audio waves themselves become gray.

3.2. Cursor (line in the center) becomes movable (see Gif 2)

3.3. Both tracks become movable (see Gif 3)

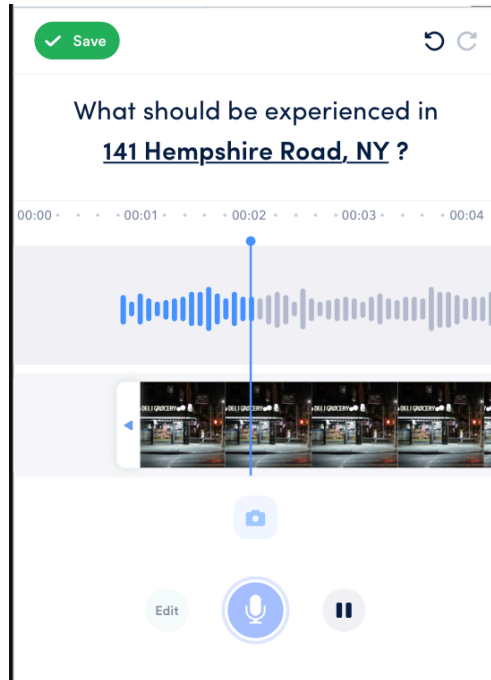


Gif 2.



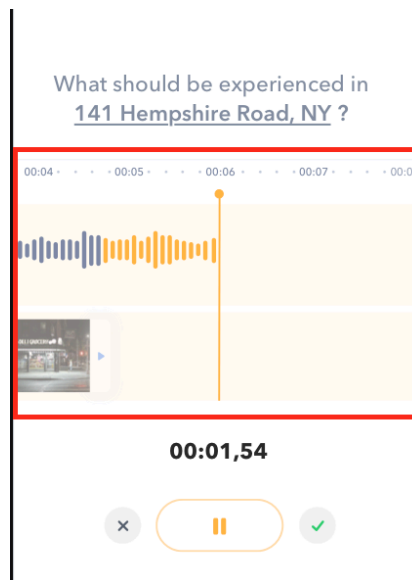
Gif 3.

3.4. Audio ability to play the audio is introduced. Audio waves related to the already played a part should become blue, remaining audio waves should stay gray with 50% opacity. (see Screenshot 3)



Screenshot 3.

3.5. When the user continues recording, new audio waves should be yellow, old waves remain gray (see Screenshot 4)



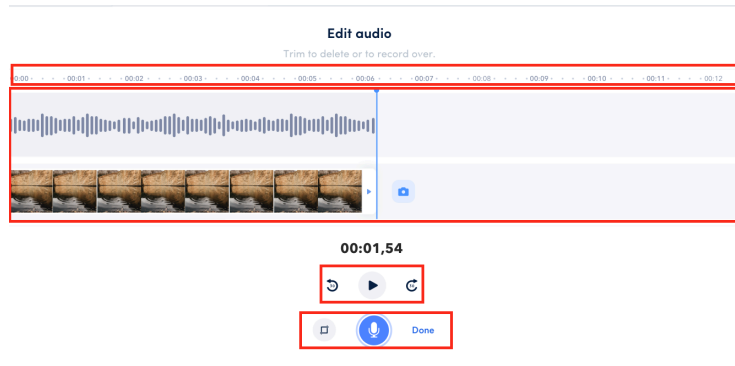
Screenshot 4.

MILESTONE 4

4. Editor

4.1. Implement the function that receives an [AudioBuffer](#) and visualizes it. (see Screenshot 5)

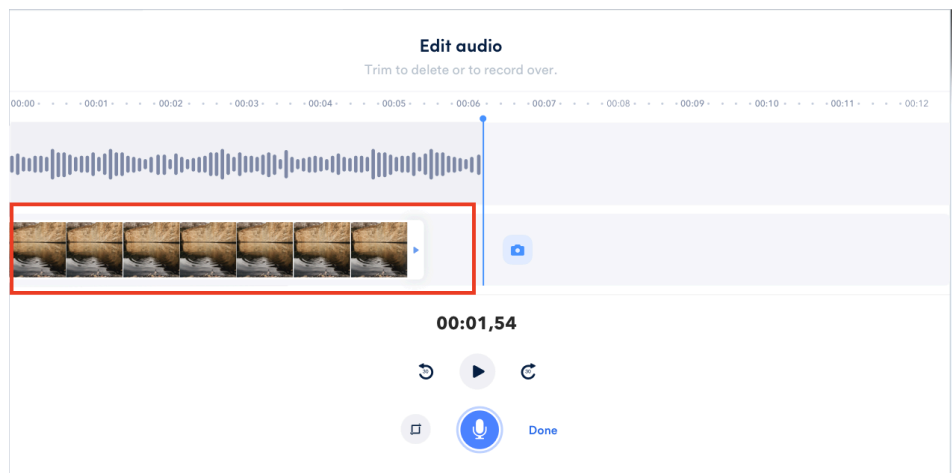
4.2. Implement the function that receives an array of image objects and visualizes it (see Screenshot 5)



Screenshot 5.

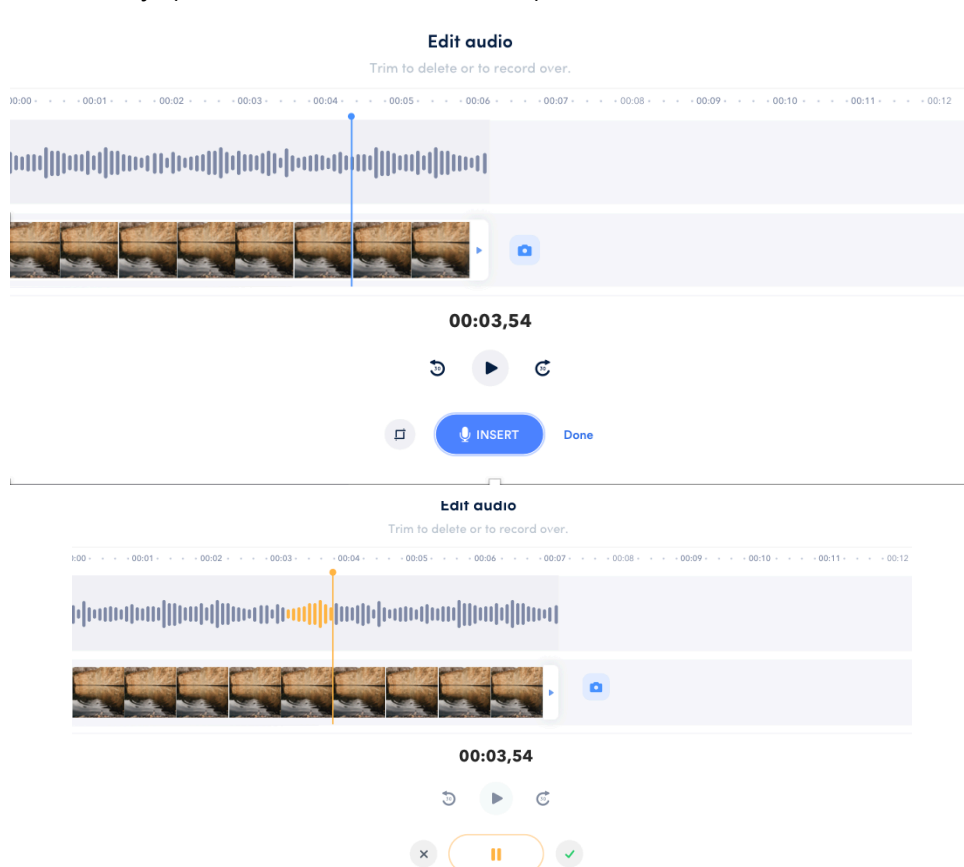
4.3. Each image on the image track has its own sliders (function from 4.2 should be able to render images both with and without these sliders) on both sides which can be manipulated. (see Screenshot 6).

- On drag to inner/outer side:
 - Increase/Decrease repeated images
- On drag to inner/outer side is finished:
 - Update start/end time of the image
- The image cannot:
 - Overlap already occupied space by another image
 - Go over the single line border



Screenshot 6.

4.4. When recording is activated, newly recorded audio waves should be drawn in yellow, placed where the cursor (line) is currently at. The image in the image track should also start repeating if necessary. (see Screenshots 10 and 11)

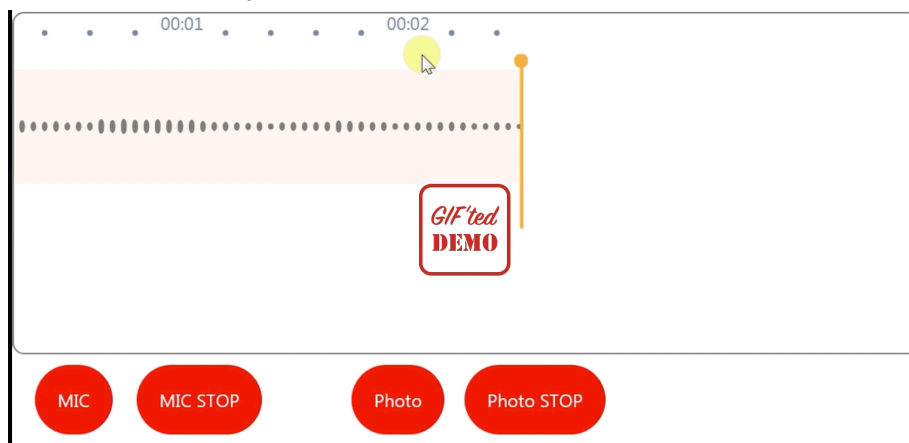


Screenshots 10. and 11.

Примечание к пункту 4.4:

Сейчас:

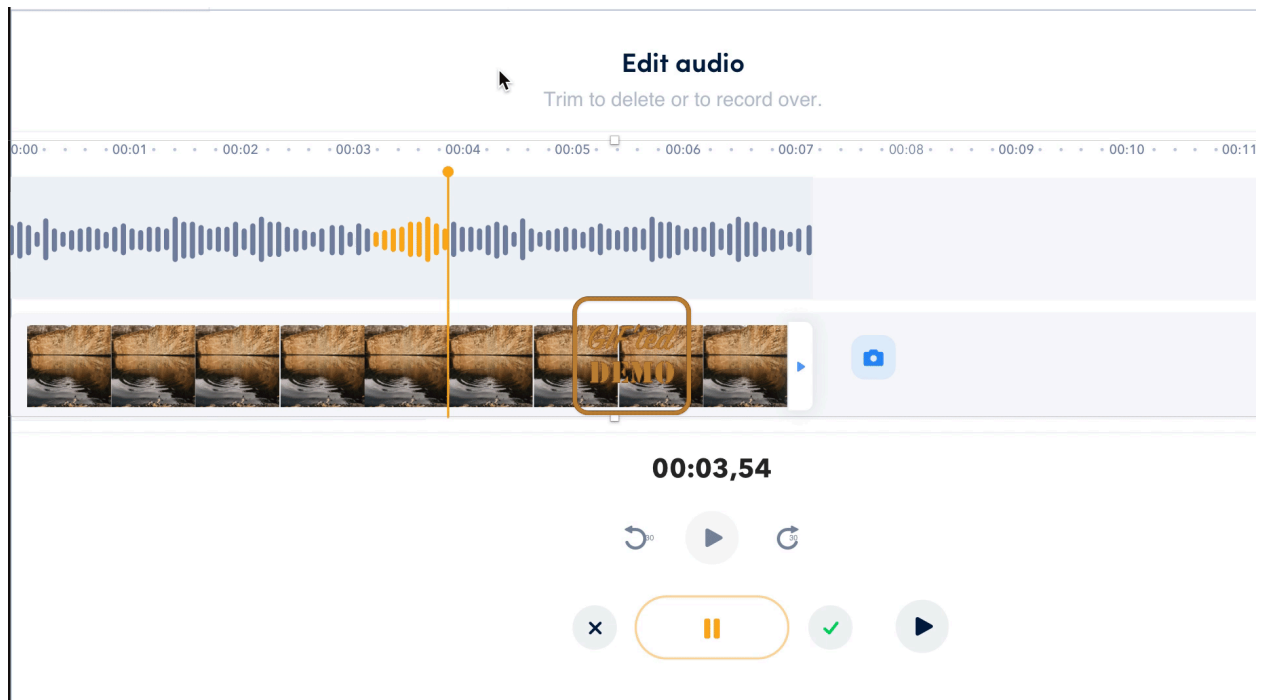
Время на старом участке записи не обновляется.(см.гиф 5)



Gif 5.

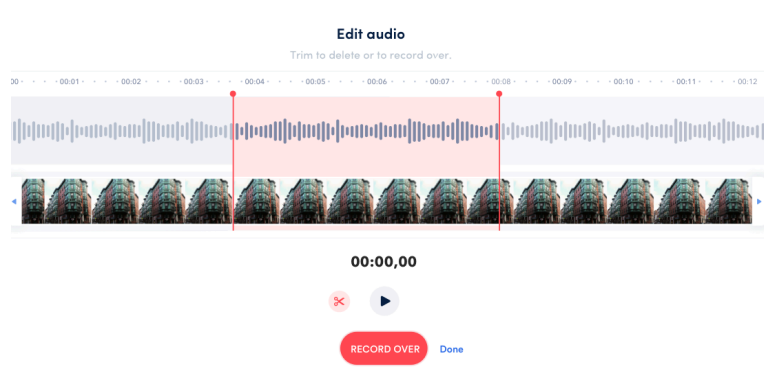
Ожидается:

Планировалось, чтобы таймлайн продолжил привычное ему поведение - двигался справа на лево не упираясь под текущее положение линии. См. Гиф 6



Gif 6.

4.5. When the trim button is clicked, single line is replaced with two lines, creating a selection. (see Screenshot 12) When record over button is clicked, audio waves related to selected part should be removed from audio wave track and newly recorded audio waves should be drawn in yellow. Images related to selected part should also be removed (if such images exist), image repeating under newly recorded part should continue.



Screenshot 12.

MILESTONE 5

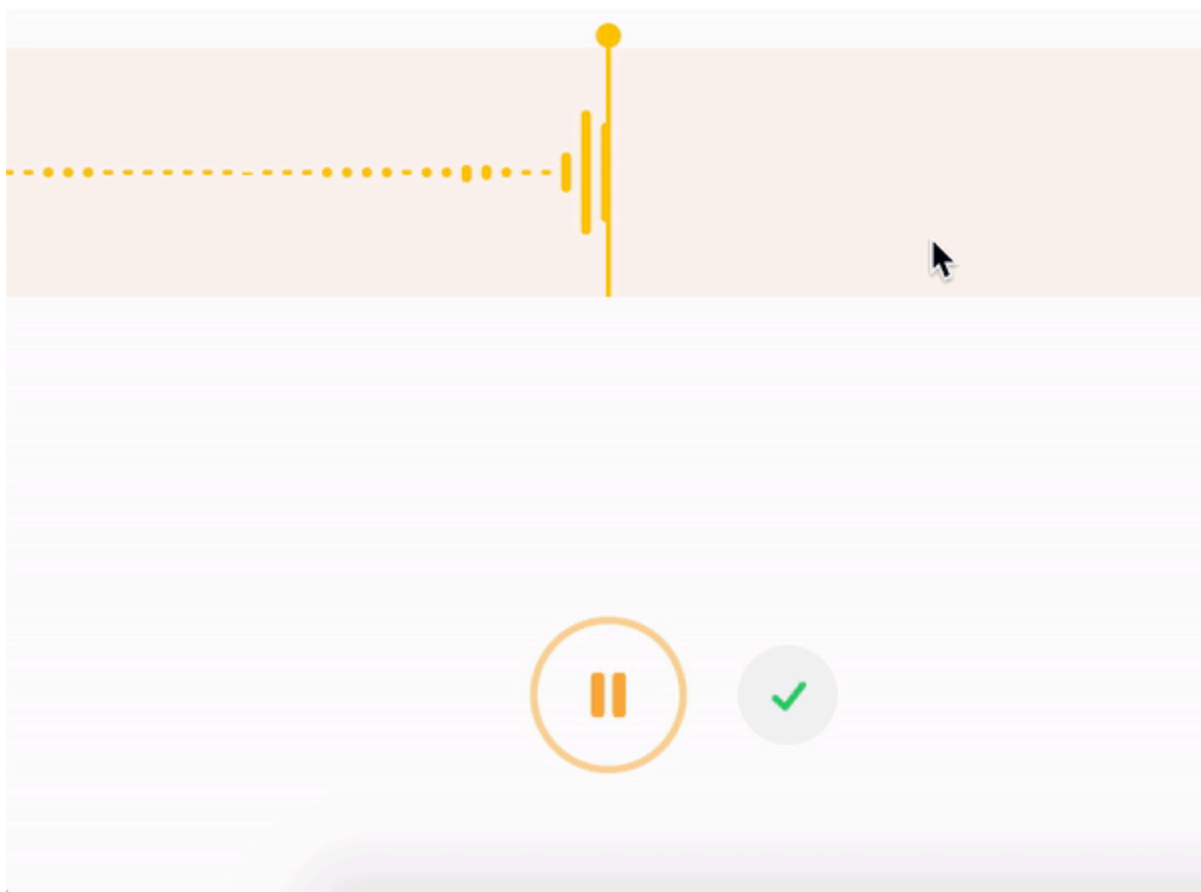
5. Moving implemented functionality into existing React application

5.1. Everything related to actual audio and image uploading will already be implemented in that application, therefore audio recording and image upload for milestones 1-4 can be as simple as needed. The same is true for buttons.

1. Визуализация активной записи.

Во время, когда идет запись аудио, в компоненте `asideRecordingNew.jsx` (`src/components/asideComponents/asideRecordingNew.jsx`), каждые 50 миллисекунд должны прорисовываться рисинки без таймлайна и без возможности перемещаться по дорожке. Одновременно на экране должно быть не более 50 таких рисинок. (т.е. последние 2.5 секунды).

Приращение/убывание длины рисинок должно происходить более плавно и менее чувствительно, чем реализовано в вашем проекте.



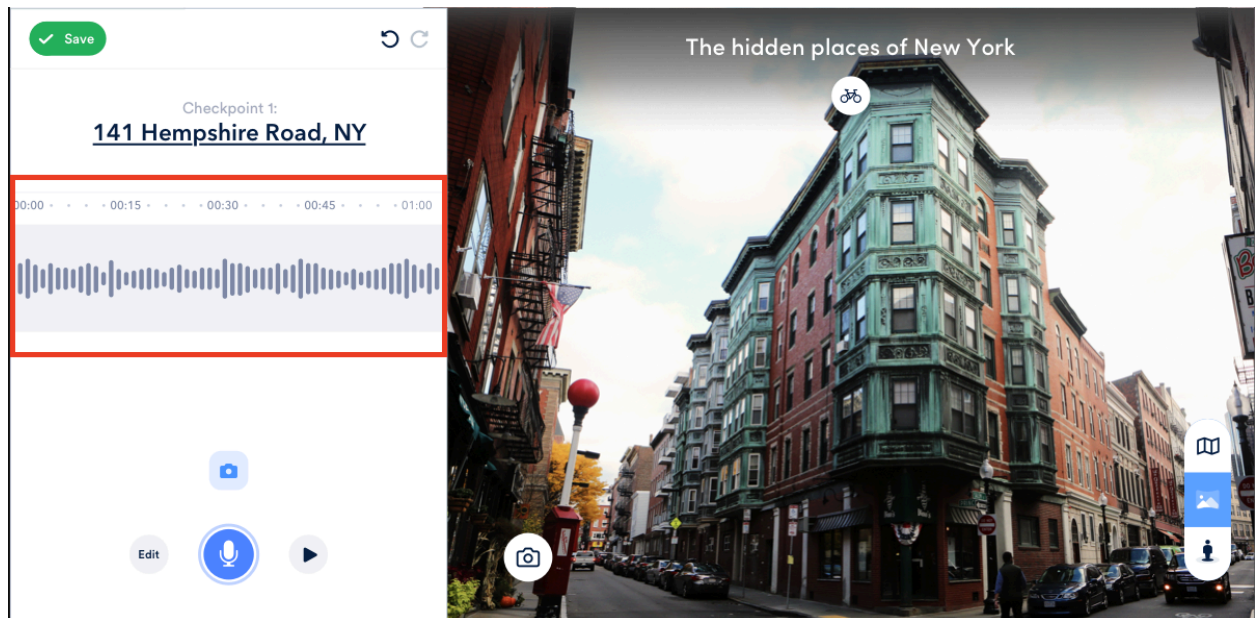
Гиф 1 (Визуализация живой записи в минимальном виде)

2. Визуализация аудио, при выключенной записи.

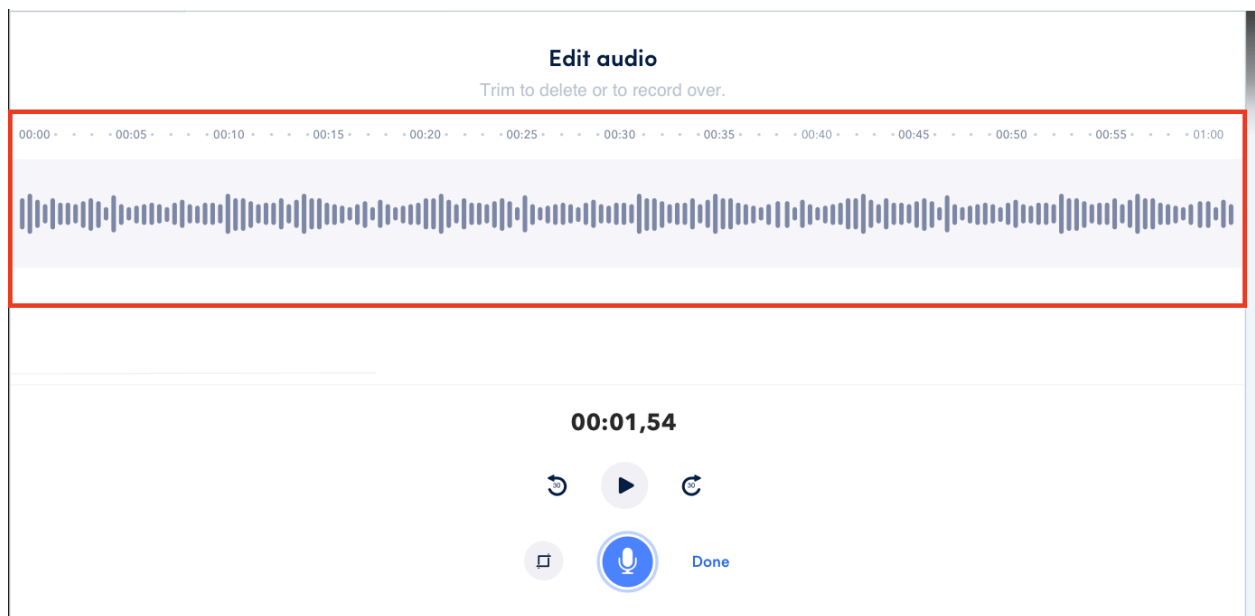
Как только запись выключается, записанное аудио конвертируется в `AudioBuffer`, который нужно также визуализировать в компоненте `asideRecordingNew` во всю длину с таймлайном над ним. (визуализация должна помещаться в ширину компонента. В проекте есть два расширения - Минимальный - скрин 1 и Расширенный - скрин 2).

Получить этот `AudioBuffer` на странице `EnterAddressNew` (`src/pages/EnterAddressNew/index.jsx`) можно следующим образом:

```
const audioBuffer = getLastItemFromUndoStack('audio', this.props);
```



Скрин 1 (Минимальный вид)



Скрин 2 (Расширенный вид)

3. Проигрывание аудио.

После визуализации `AudioBuffer`, его можно проиграть, нажав на `play`. При проигрывании рисинки относящиеся к проигранному участку окрашиваются в другой цвет.

Если при активном проигрывании, кликнуть на какую-нибудь из рисинок, проигрывание продолжается с участка к которому относится эта рисинка.

Проигрывание `AudioBuffer` уже реализовано. В файле `AudioActionsNew` (`src/actions/AudioActionsNew.js`) за `play` отвечают строки 45-72 (остальная часть функции `startPlaying` на данный момент не релевантна).

За `stop` отвечают строки 159-178 (остальная часть функции `stopPlaying` тоже не релевантна)

Возможно, в рамках данной имплементации требуемый функционал не реализуем, тогда ее нужно будет переделать.