
README.md



A multimedia web application developed for the [Multimedia Services and Applications](#) course @ feup

Description

Inclusive is a web application that allows the user to take photos and record videos with real time annotations that may constitute drawings on the screen, speech to text or even the creation of a virtual background with an imported image or a blur effect.

It aims to be a valid solution for those who want, for example, the *zoom* functionalities without the need to start a meeting and save the captures or recordings for future analysis.

A demo of the development application is available [here](#).

Functional Specification

The following user stories illustrate the functionalites of the developed web application:

Core functionalities

- As a user, I want to be able to take pictures with my web cam, so I can view them after.
- As a user, I want to be able to record videos with my web cam, so I can view them after.

Audio manipulation

- As a user, I want to be able to record a video with sound, so that others can not only see the recorded frames but hear the correspondent audio as well.
- As a user, I want to be able to see my audio converted to text at the time I am speaking, so I can see in the video what I just said.

Drawing annotations

- As a user, I want to be able to take notes on a white board, so I can show others my notes while they see me.

- As a user, I want to be able to take notes within my own picture, so I can not only show the captured web cam frames, but also show some notes/drawings.
- As a user, I want to be able to erase my own drawings and notes, so I can correct some mistakes I could possibly have done.
- As a user, I want to be able to record the animated annotation independently of the recording mode I am in (camera, white board, screen sharing), so that I can save them to watch later.

Screen share

- As a user, I want to be able to share my screen, so I can show others what I'm seeing/doing on my computer.
- As a user, I want to be able to take notes within my screen share, so I can not only show my screen to others, but also show some notes/drawings.
- As a user, I want to have the possibility of showing or not my web cam while annotating on the white board or sharing my screen, so that I have a more customizable environment.
- As a user, I want to be able to detect faces within the range of my camera, so I can hide everything around me but my face.

Low-fidelity prototype

We started by making a few sketches to give a visual identity to our application.

We wanted three main modules: only camera, camera and whiteboard, camera and screen. The user would be able to select each one of these modes in the top left corner.

We also wanted the user to be able to take a photo of the current state of the screen, to record it, to annotate it and to erase the present annotations. The button that would trigger these interactions are present in the bottom left corner.



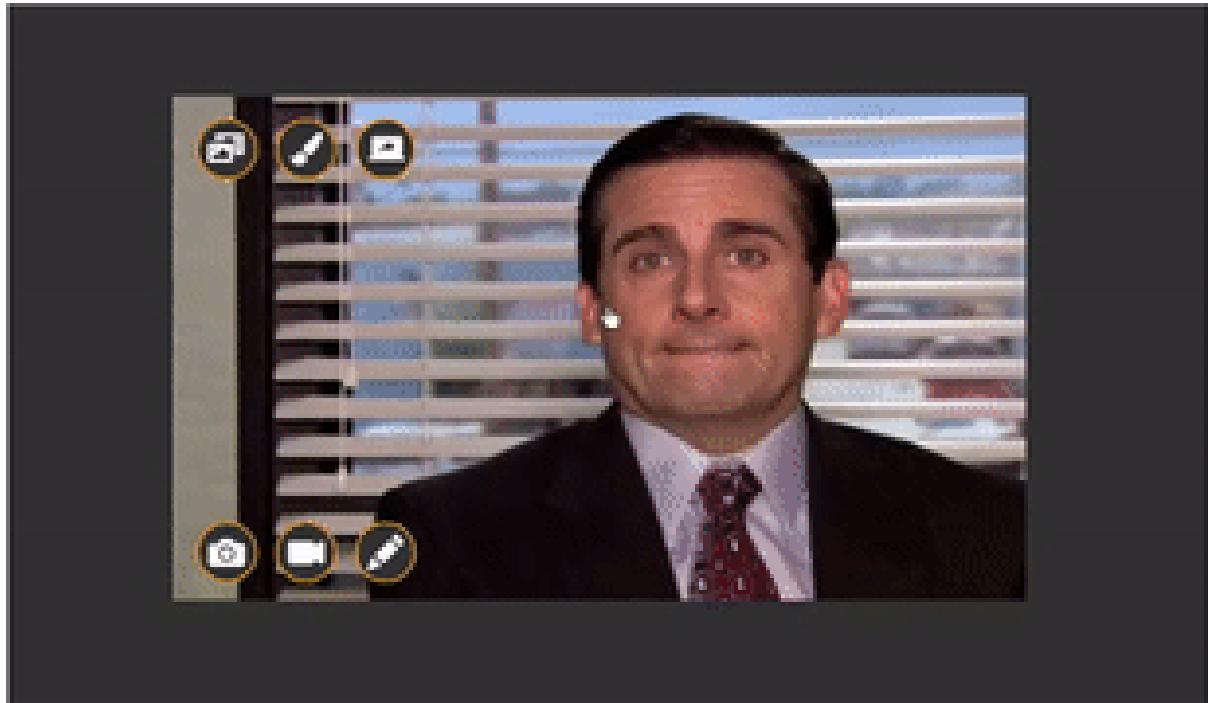
Here we show the main interactions that were proposed in an initial stage of development. There are some interactions missing as we only thought about implementing them in later stages of development.



High-fidelity prototype

The next step was the development of a high-fidelity prototype where the interactions could be tested with user input (in this case, clicks).

The following prototype is accessible to the public at the following url: <https://sam-prototype.herokuapp.com/>



Tools used

- **Figma:** mockups
- **Invision:** wireframes with interactions
- **Protoshare:** development of the high-fidelity prototype
- **React:** chosen framework to develop the web application
- **body-pix:** a *tensorflow* node module with models for the human body segmentation that can be applied on the web
- **Media recorder API:** for capturing frames from the web cam and the audio input
- **Web speech API:** for traducing the recorded audio into text in real time

How to use

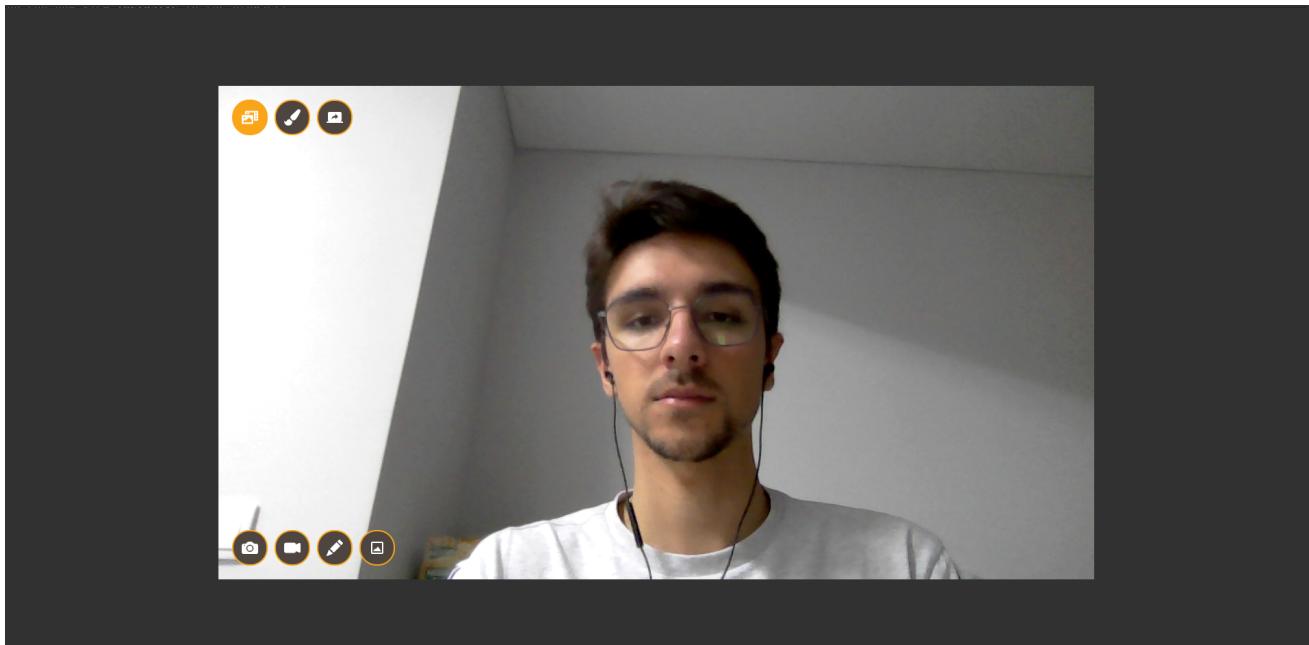
Set up

To run the app, the user only has to certify he has `npm` installed and run the following commands (the first command may only be needed if the app is being run for the first time):

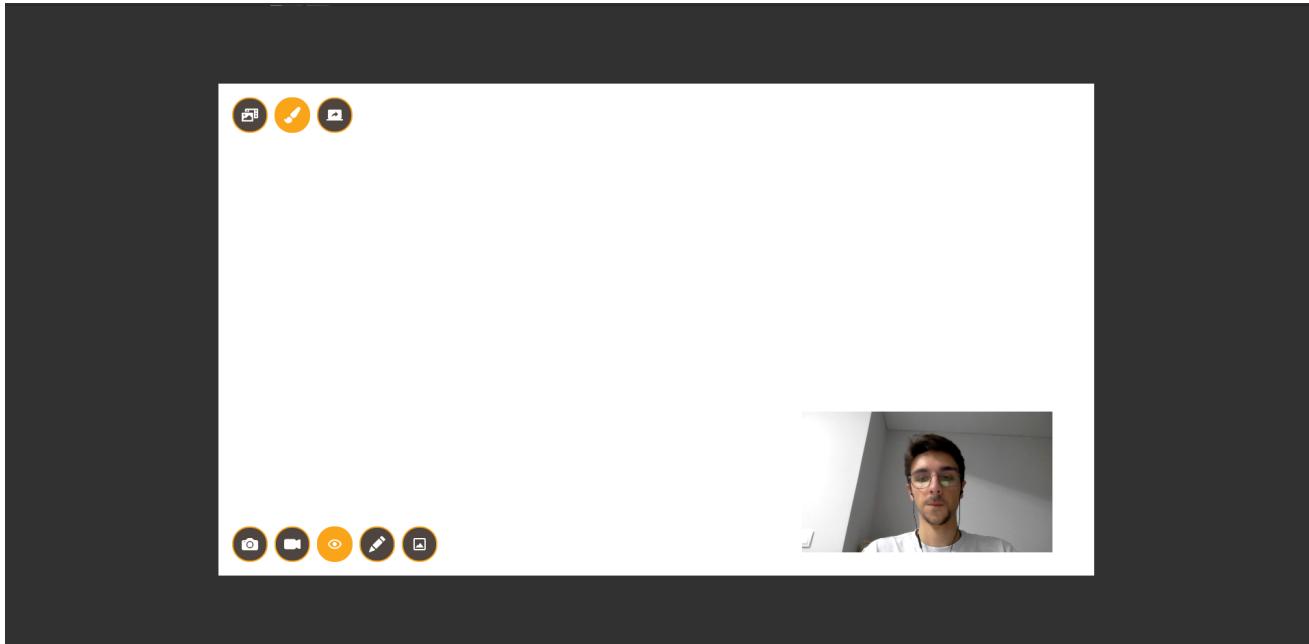
```
npm install  npm start
```

Using modes

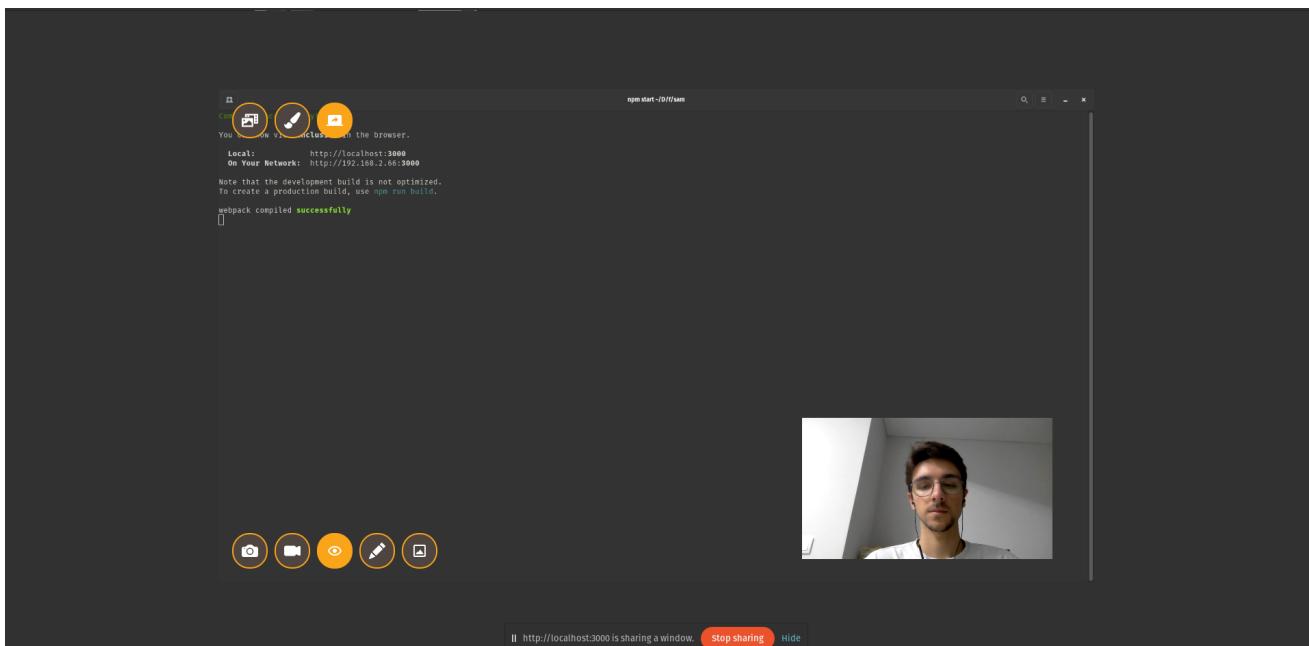
The application has three main modes: camera only, white board and screen sharing. When accessing the web application the user is directed to the camera only page:



The user can click on the second button on the top left corner to access the white board mode:



And by clicking on the third button he is redirected to the screen sharing mode:

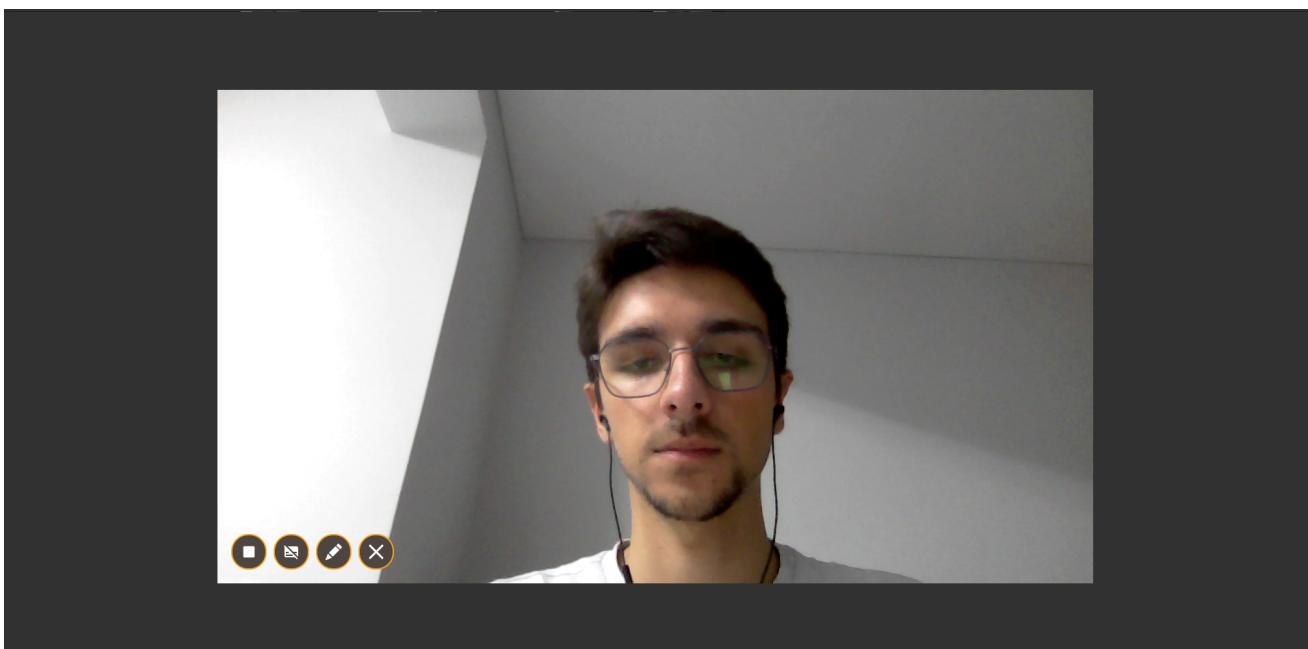


Capturing and recording

To capture a photo the user only has to click on the first button on the bottom left corner, the output is imediately shown to the user:



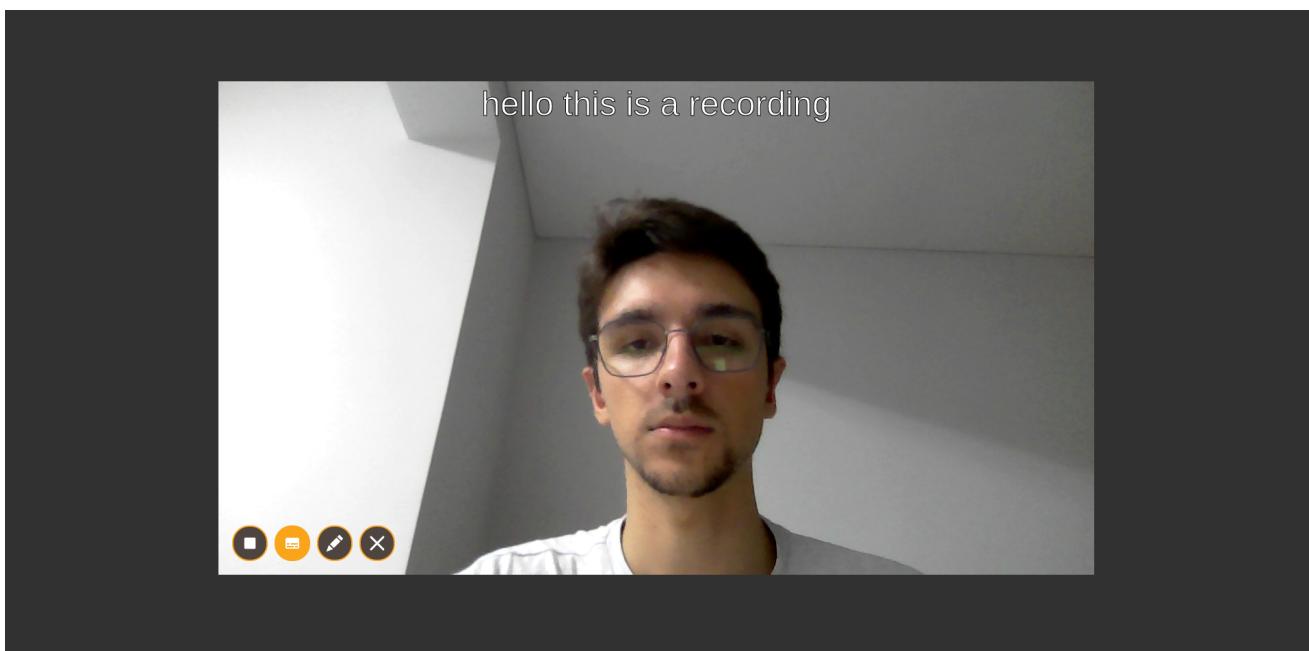
To record a video the user has to press the second button on the bottom left corner, this will start the recording:



As it is visible the user has now a different set of buttons available. The first one will stop the recording, redirecting the user to a page container a video player where the recordings can be watched.



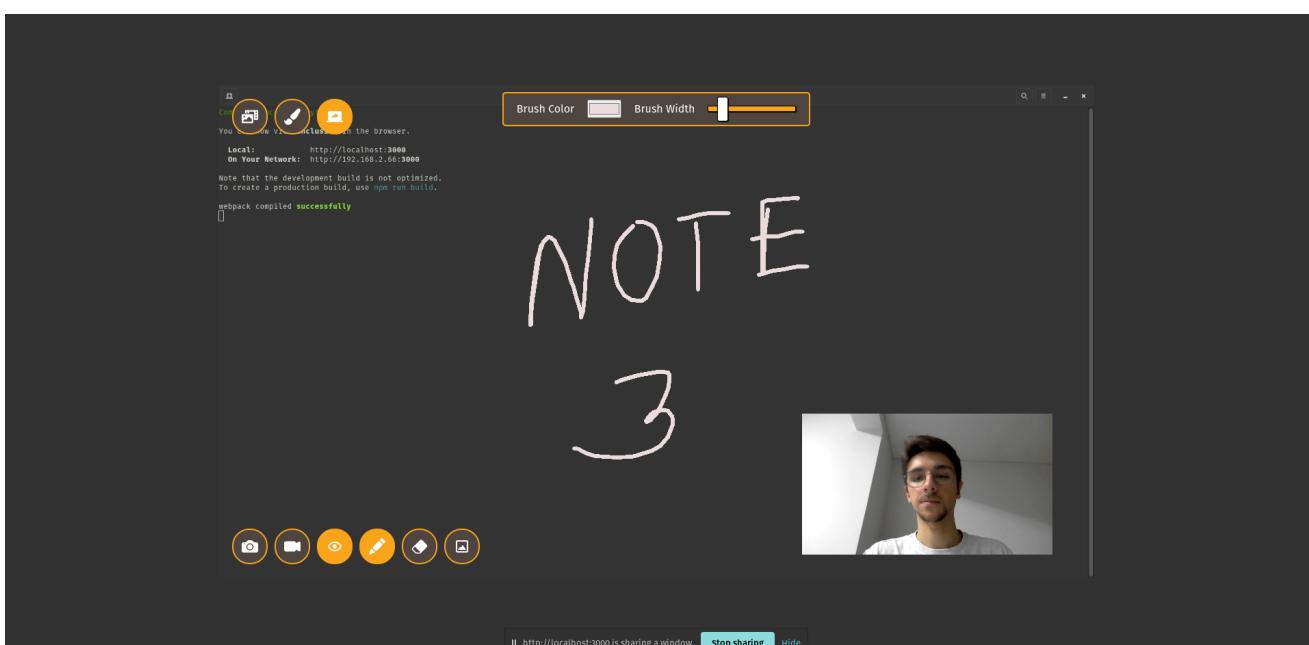
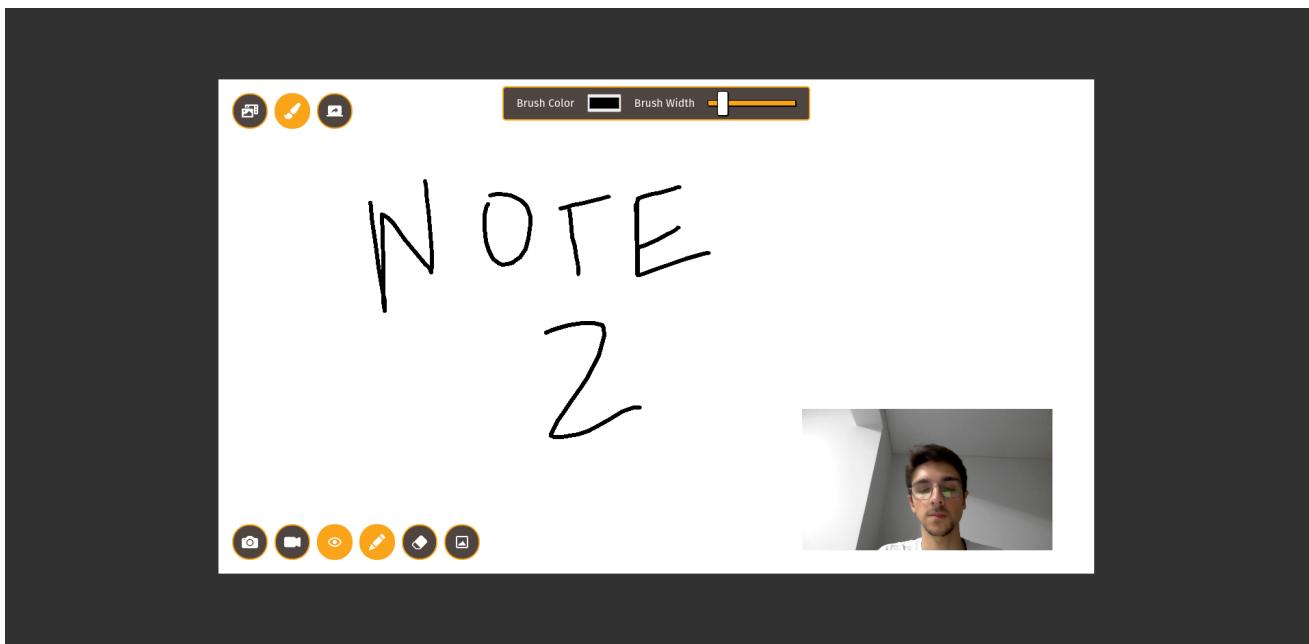
The second button will activate the generation of transcripts generated by the integration of a speech to text API that enables the user to see on the screen what he is saying at the moment:



The behaviour of the third button will be explained on the next section and the last one will stop the recording and redirect the user to the initial page.

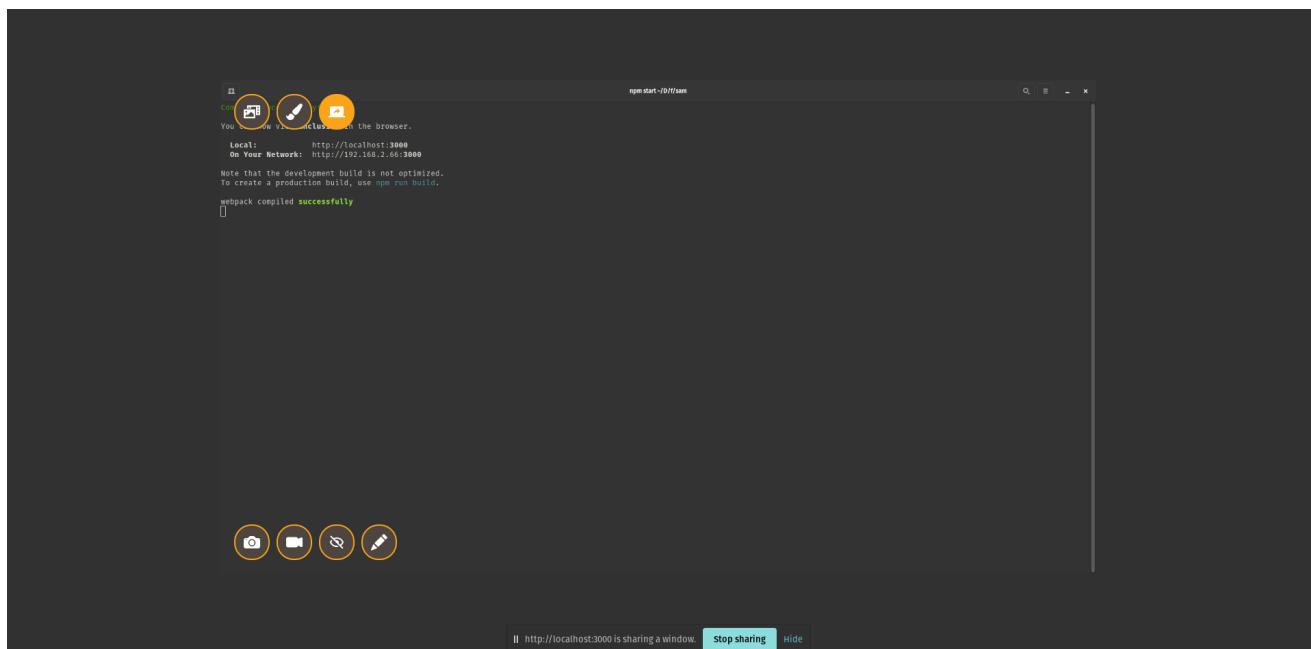
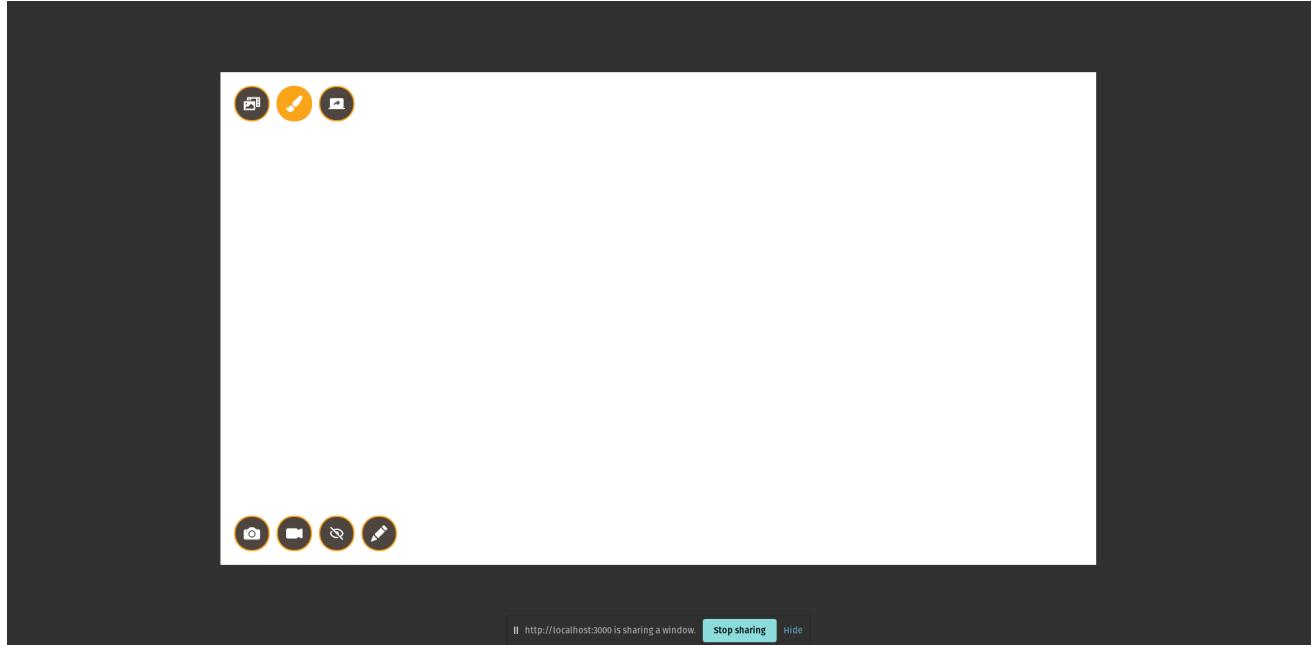
Annotations

The third icon on the bottom left corner (pencil) will enable the user to start drawing on the screen. The user is able to select the brush color and width. There is also an erasing button. Here are examples of notes taken in the three different modes:



Hide camera

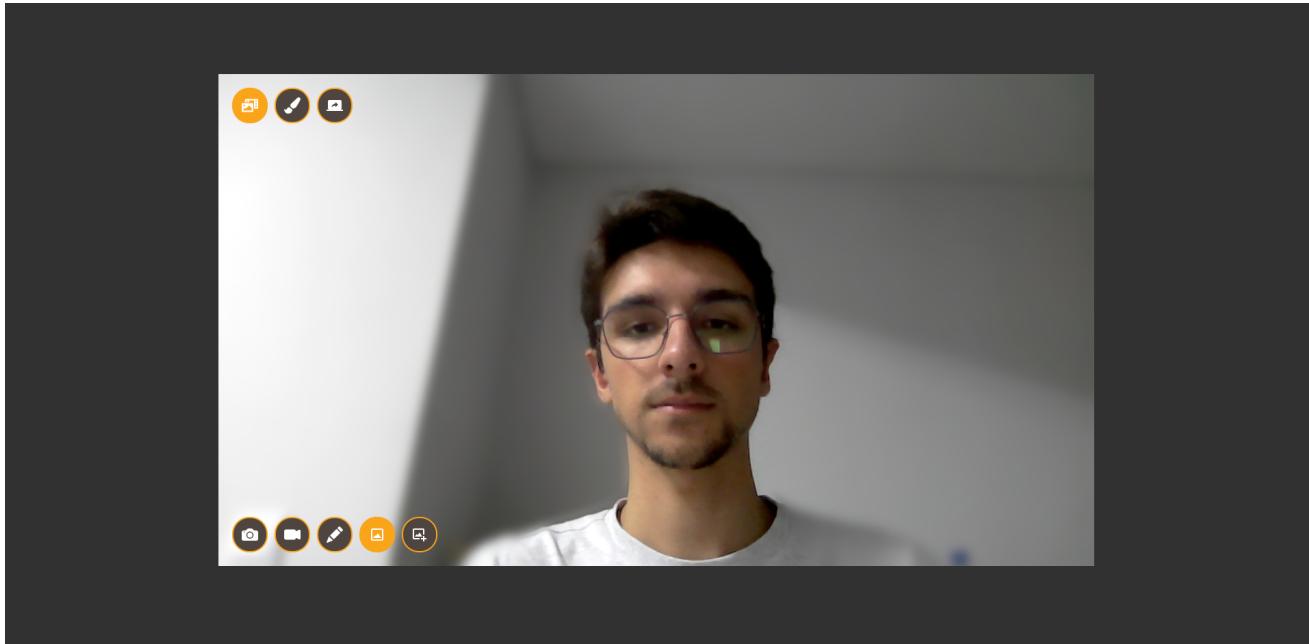
When on the white board or screen sharing modes the user may want to show or hide the camera for diverse reasons. A button with an icon with the shape of an eye was included so that the camera is hidden/shown when clicked. Here are the two modes without the camera:



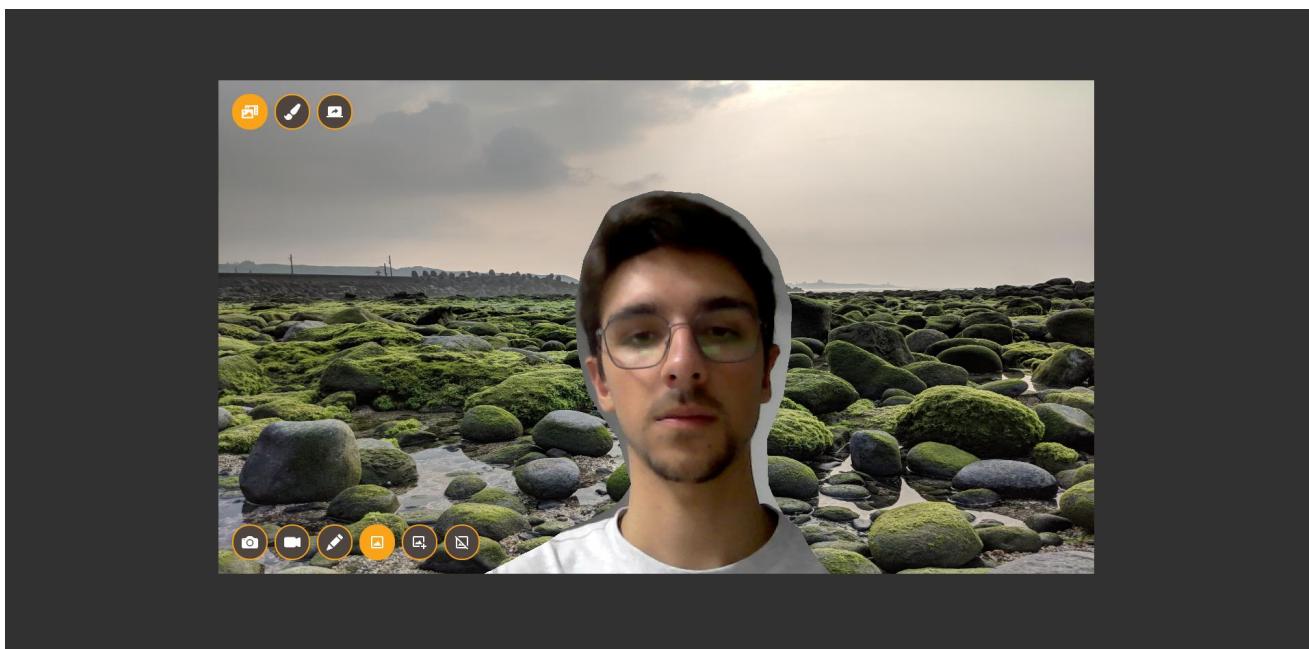
Virtual background

We wanted the user to have the ability to keep the privacy of the place he is recording at, so we implemented a virtual background functionality where the user can blur the background or insert an image to hide the background. This was achieved by integrating a computer vision model for human body parts segmentation.

By clicking on the last icon on the bottom left corner the user will activate this functionality:



And by clicking on the button that appears on the right side of the clicked one, the user is able to import a photo to display as background:



When a photo is uploaded another button will appear that when clicked restores the blurred background previously applied without the photo.

Problems faced and lessons learned

- Since we wanted to do a standalone web application with no server side processing some features of video editing that were idealised in the initial stages of development were not possible to implement.
- The computer vision model for human body parts detection may not be optimized to every use case scenario and it may slow down the application when running on less powerfull hardware.

- Browser compatibility was also an issue since we are using APIs provided by Google Chrome. This way, we recommend the use of our application in the browser so that there are no unexpected behaviours.
 - Overall we believe we have built a product with good quality that may prove to be useful. Also it has margin to grow, as it is easy to implement new features on top of the ones implemented.
-