Nonwimp2 Assignment Writeup — Won Kim, Krish Savla, Yihui Hu

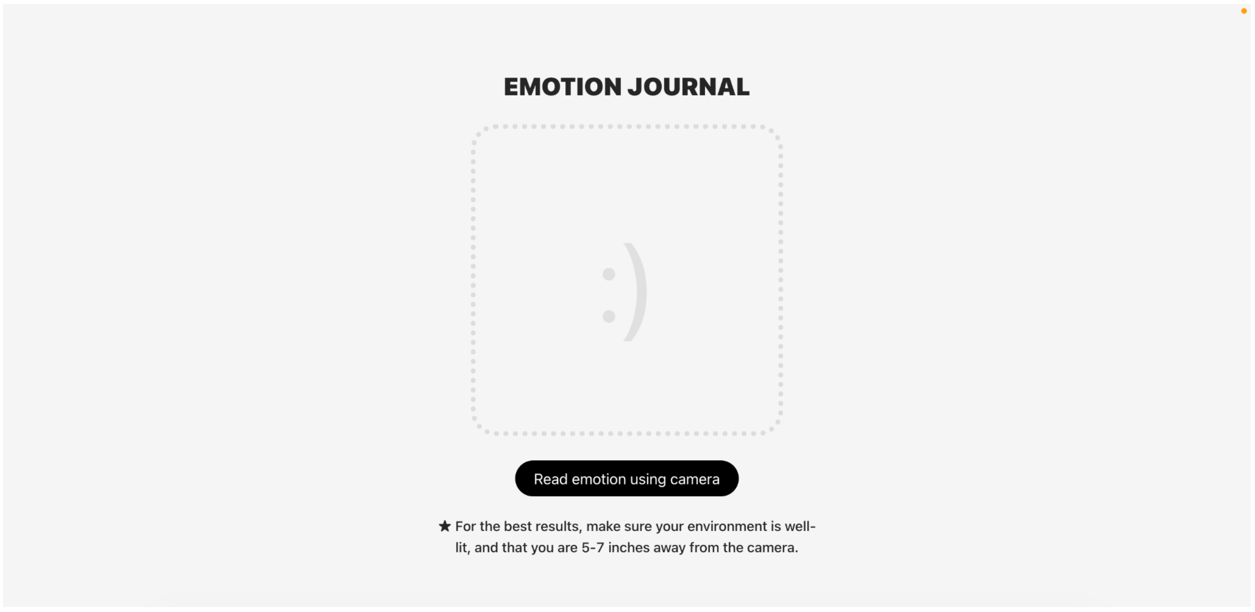
Documentation (15 pts)

- **How we arrived at the final design:**
    - From the start, we prioritized a minimal and fuss-free design that allows users to quickly record down their emotion(s) and thoughts.
    - Our site comprises two main sections: the top section utilizes Affectiva.js (fig1) and the bottom section utilizes the Vosk API for speech-to-text transcription (fig2). Users are free to begin with whichever section they wish.

- **How we implemented the main features of the program:**
    - We adapted and simplified the code from Prof. Robert Jacob's Affectiva demo, as well as the Vosk React Template, mostly through JavaScript. For frontend, we use React to optimize it for mobile/tablet view as well.

- **A short description of the good aspects of your UI design:**
    - We highlight key elements of the UI, such as various buttons to begin emotion reading or edit the transcripted speech,
    - We tap into real-life motifs, such as making the bottom section look like a sheet of paper to make the connection to analog writing.
    - There is ample feedback and cues in the UI to let users know at every step when a process is running (especially useful since the APIs we're using can be finicky/slow).

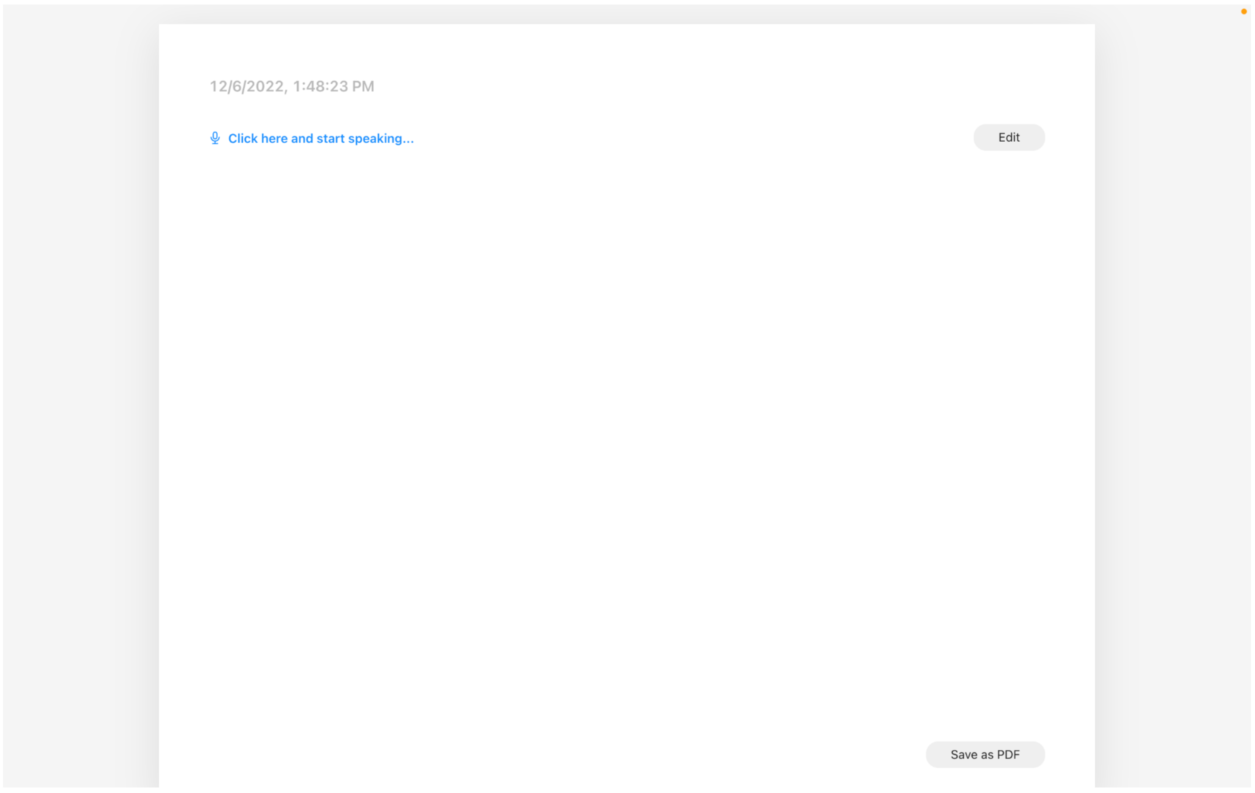
- **Interesting / unusual technical features or issues:**
    - Ability to edit then save the transcripted speech!
    - Ability to save the page as a PDF (with a record of the emotion captured as well as journaled thoughts)


EMOTION JOURNAL

Read emotion using camera

★ For the best results, make sure your environment is well-lit, and that you are 5-7 inches away from the camera.

(fig1)


12/6/2022, 1:48:23 PM

Click here and start speaking...

Edit

Save as PDF

(fig2)

- **Some interesting aspects of your UI design:**
    - We’ve included a timestamp of the current date such that when the user saves the page as a PDF, there is a clear record of when it was created.
    - The color of the page changes based on the captured emotion to reflect it clearly, and the emoji will also be displayed on the page.
    - As speech is being transcribed by the Vosk API, the words will appear one by one on screen as a way to let users know that it is tracking their speech as intended.

- **Any modifications we needed to make along the way:**
    - For the live transcription to work, the API generates each word as a <span> and appends it to a <div> element. When we implemented the ‘edit’ feature, we had to concatenate the text values in the <span(s)> so users can edit the entire ‘block’ of text.

- **Instructions for how to run your program:**
    - Users can visit https://emotion-journal.netlify.app to get started :)
        - Click on the ‘read emotion using camera button’ to record your emotion
        - Click on the speech button to start talking and click it again if you want to stop
        - Edit by clicking on the edit button, close edit by clicking it again
        - Saves as pdf by clicking on the ‘save as pdf’ button located on the bottom right of the document

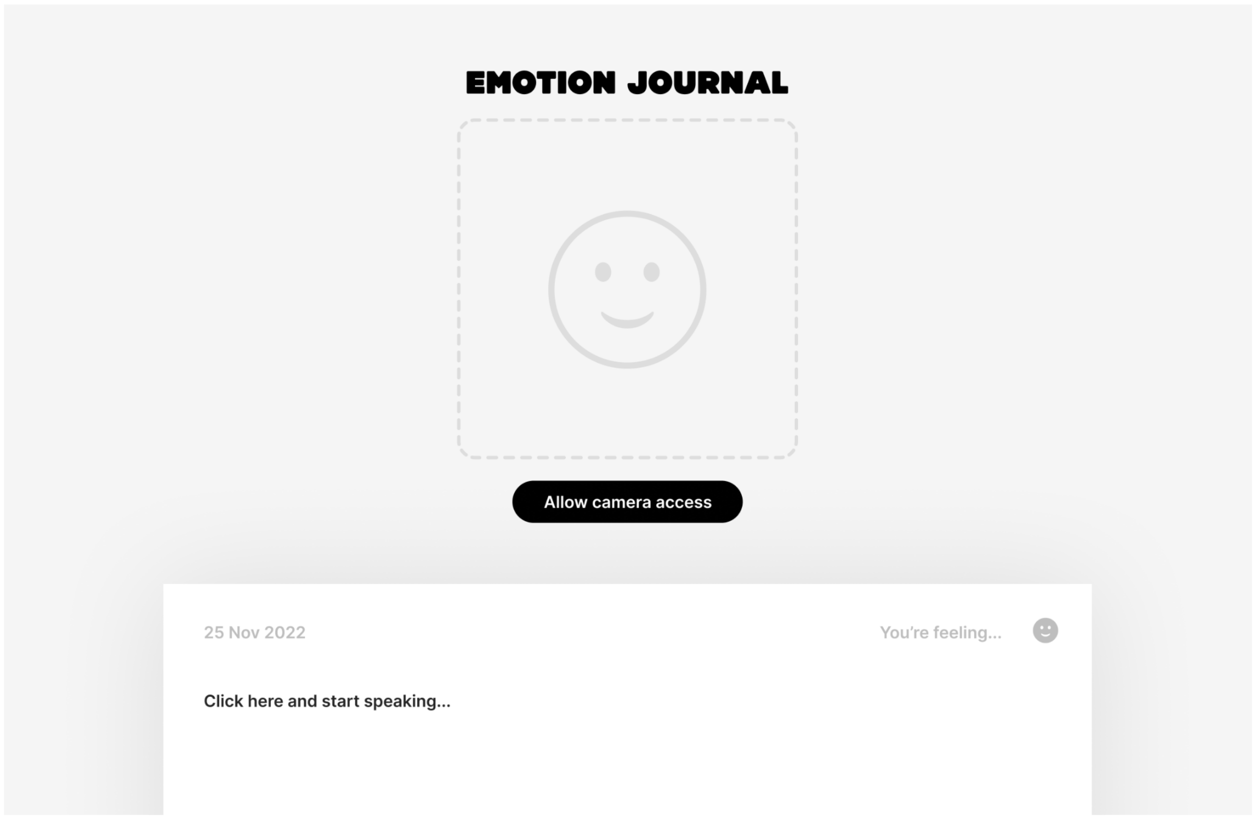
Nonwimp1 Assignment Writeup — Won Kim, Krish Savla, Yihui Hu

Project sketch (20 pts):

Describe your project in detail (the goal of your application / simulation and the initial design etc).

- Our application will be a simple journal-writing application. Using a webcam/front-facing camera, the app detects the user's current emotion and prompts them to speak/talk about their day. The app creates a PDF (which the user can download) containing the current date, an icon representing the user's emotion, and a transcription of their speech.
- The initial design is barebones, with only two 'input zones': the video/image of the user's face, as well as a 'sheet' of paper where the live transcribing is written as the user speaks.

Mockup on Figma:



Technology requirements/technical risks (20 pts):

List the technologies you are planning to use for your project (eg. APIs, a different language we have not covered in class, third party software, etc).

- We will mostly be using plain JavaScript for the backend, as well as HTML & CSS to display the app on the web. We might be using React as well for better UI management. For APIs, we are looking at Affectiva.js for the emotion detection, as well as Vosk for speech-to-text transcription. We might also have to use Express.js to set up a server to process data (?)
- Background color will change based on emotion.
- If we have time, we could make it mobile friendly as well.

List the potential risks that could come with the technology you plan to use. You should make some tests to make sure you can use them or find out its peculiarities.

- **Affectiva.js**: A user's emotions are sometimes difficult to track if they are too far from the camera or if the surrounding lighting is too dark or bright. We need to make it clear to the user that they need to be in a well-lit environment and close enough to the screen for the program to work as intended.
- **Vosk**: Speech recognition is not always perfect, so there will most likely be times when the journal will not be able to translate the user's voice to text perfectly. We can get around this by allowing the user to edit any text afterwards.

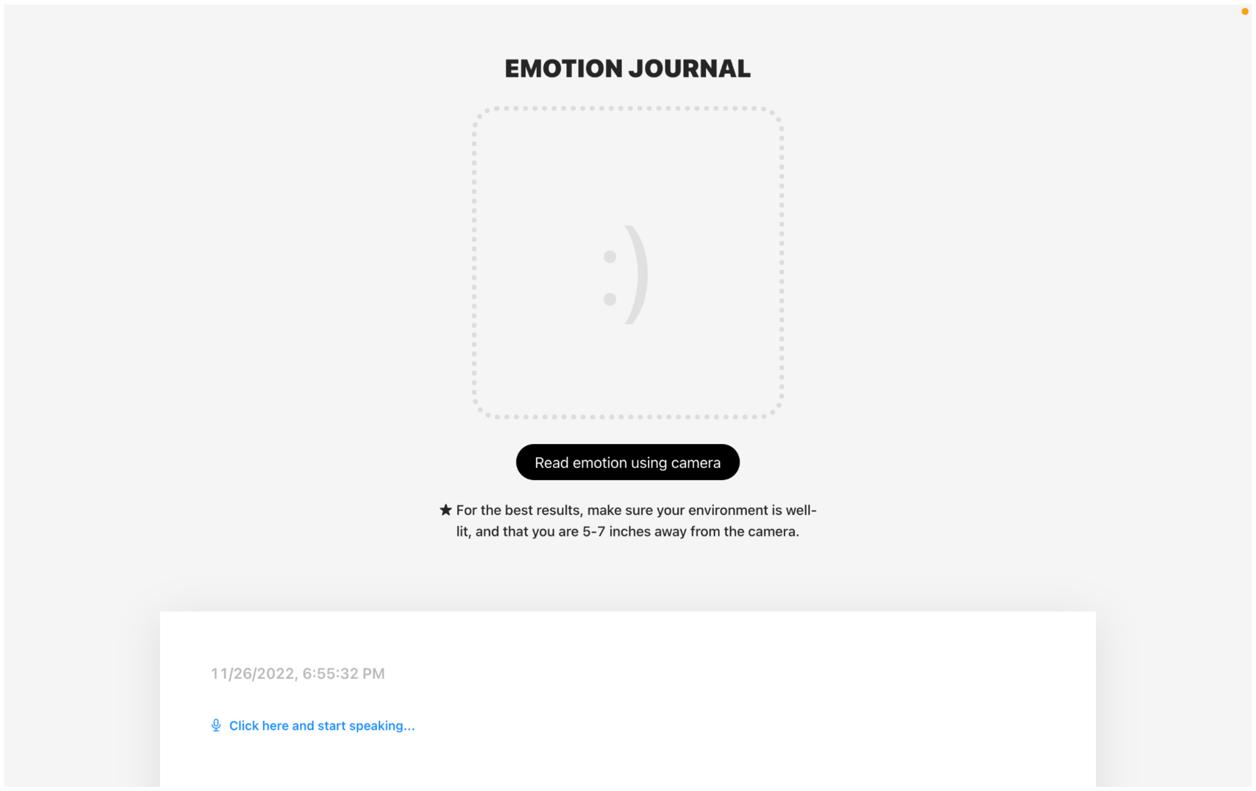
Technology feasibility test (20 pts):

Come up with a small feasibility test for the system or technology you want to use. An example would be if you are planning to use some API, trying out the demo code on your local system to make sure you can run it is a valid feasibility test.

- We have successfully gotten Vosk to work using a React demo that we forked from https://github.com/ccoreilly/vosk-browser. We have cleaned up all redundant code that we don't need, and formatted the code for our use.

- One challenge we face currently is the ability to edit the text of the live transcription. As they are generated live as discrete HTML elements (more specifically <span> elements),they cannot be edited as one whole chunk/paragraph.
- We have also begun experimenting with the Affectiva SDK/web demo on the course website, adapting it to suit our needs.
- As a side note, as this app does not require any keyboard input, it's great for people with mobility issues/those not skilled with typing. Or more generally, people who'd rather talk through their day to record it down rather than typing it (talking is faster than typing for a lot of people). Though, mouse input is still required to click buttons and initiate the emotion reading & transcription.

Screenshot of current progress (uses React for frontend):



★ For the best results, make sure your environment is well-lit, and that you are 5-7 inches away from the camera.

11/26/2022, 6:55:32 PM

In-Class Presentation (Mon Nov 28) - 20 Pts:

Each group should give a very brief presentation of your project plans in class (max of 2 minutes, 2 slides) for the group to discuss, critique, and provide feedback for completing the project.

- Presentation Link: https://docs.google.com/presentation/d/1y1w7GYaCsUaNs6TQCZpbgaTFgNa1fkwMEPRKXb8wh58/eCS86_Nonwimp_1_Presentationdit#slide=id.p