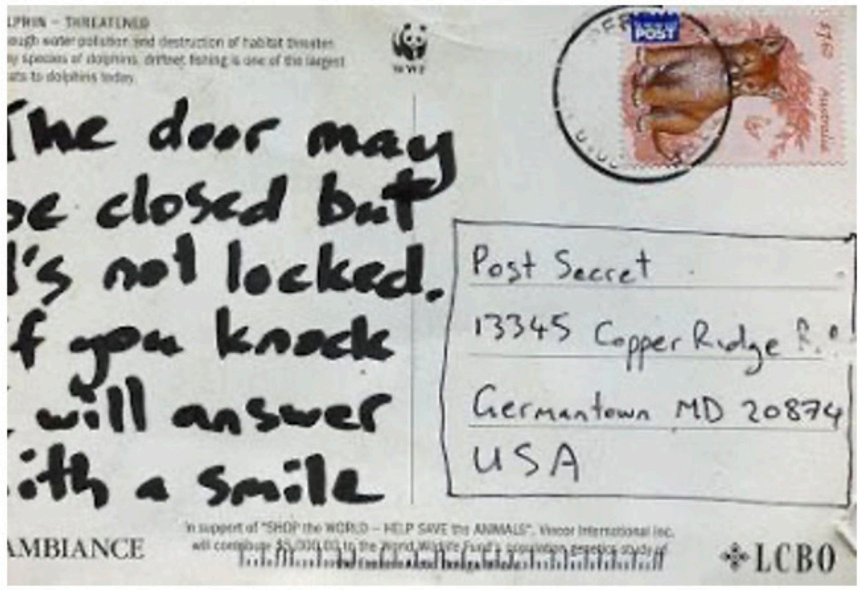
Portfolio

Portfolio

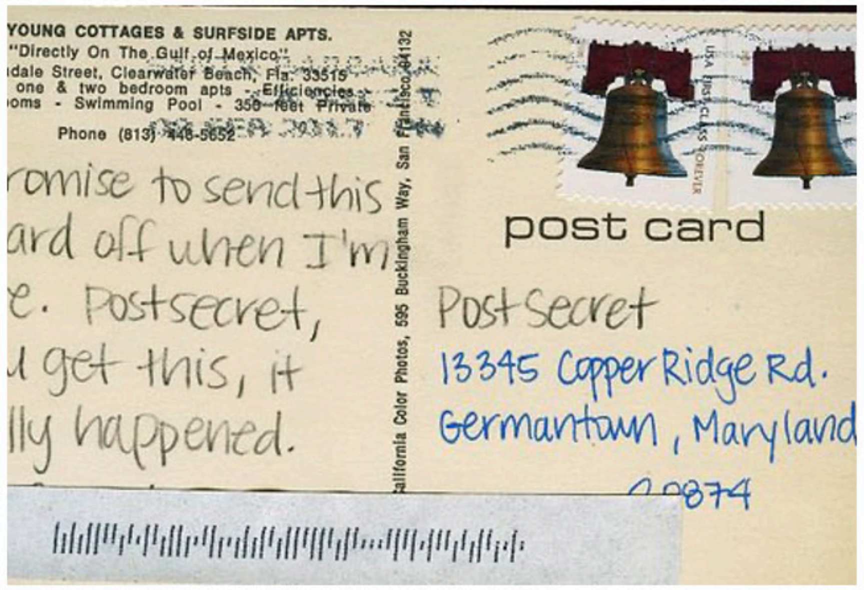
BIG DATA, THE SELF AND SOCIAL PLATFORMS

2022


The door may
e closed but
's not locked.
f you knock
will answer
ith a smile.

Post Secret
13345 Copper Ridge R.
Germantown MD 20874
USA

AMBIANCE
LCBO

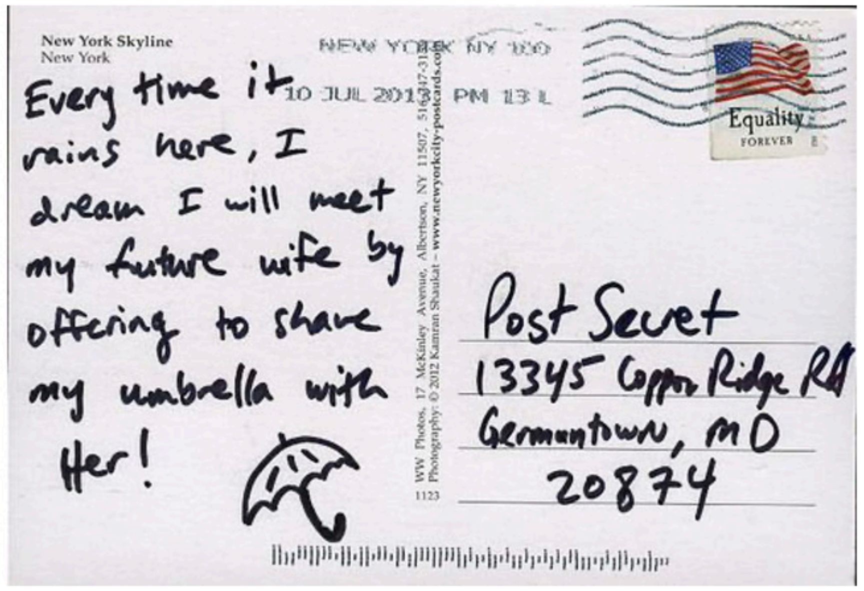

YOUNG COTTAGES & SURFSIDE APTS.
"Directly On The Gulf of Mexico"
dale Street, Clearwater Beach, Fla. 33515
one & two bedroom apts - Efficiencies -
oms - Swimming Pool - 350 feet Private
Phone (813) 446-5652

California Color Photos, 595 Buckingham Way, San Francisco 94132

romise to send this
ard off when I'm
e. Postsecret,
u get this, it
lly happened.

post card

PostSecret
13345 Copper Ridge Rd.
Germantown, Maryland
20874

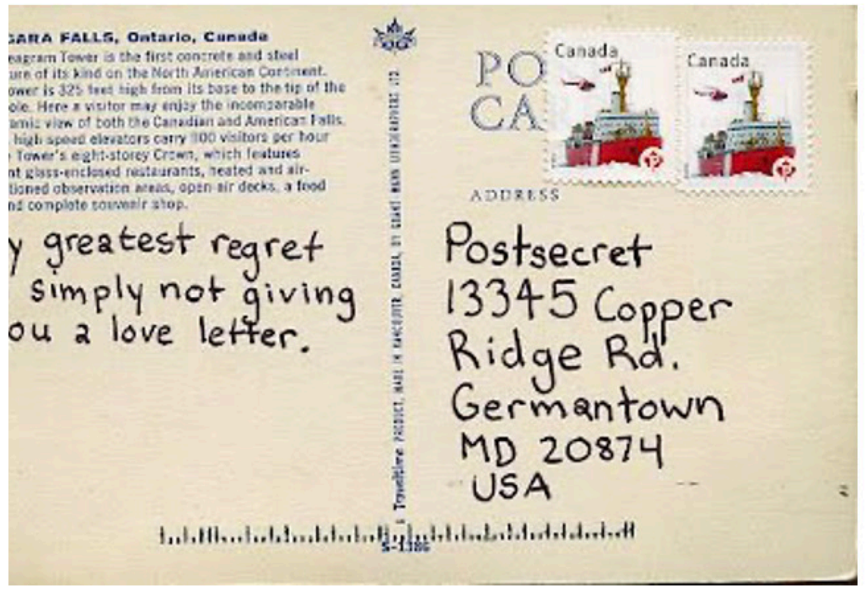

New York Skyline
New York

NEW YORK NY 100
10 JUL 2013 PM 13 L

Equality
FOREVER

Every time it
rains here, I
dream I will meet
my future wife by
offering to share
my umbrella with
Her!

Post Secret
13345 Copper Ridge Rd
Germantown, MD
20874


GARA FALLS, Ontario, Canada

Canada
Canada

ADDRESS

y greatest regret
simply not giving
ou a love letter.

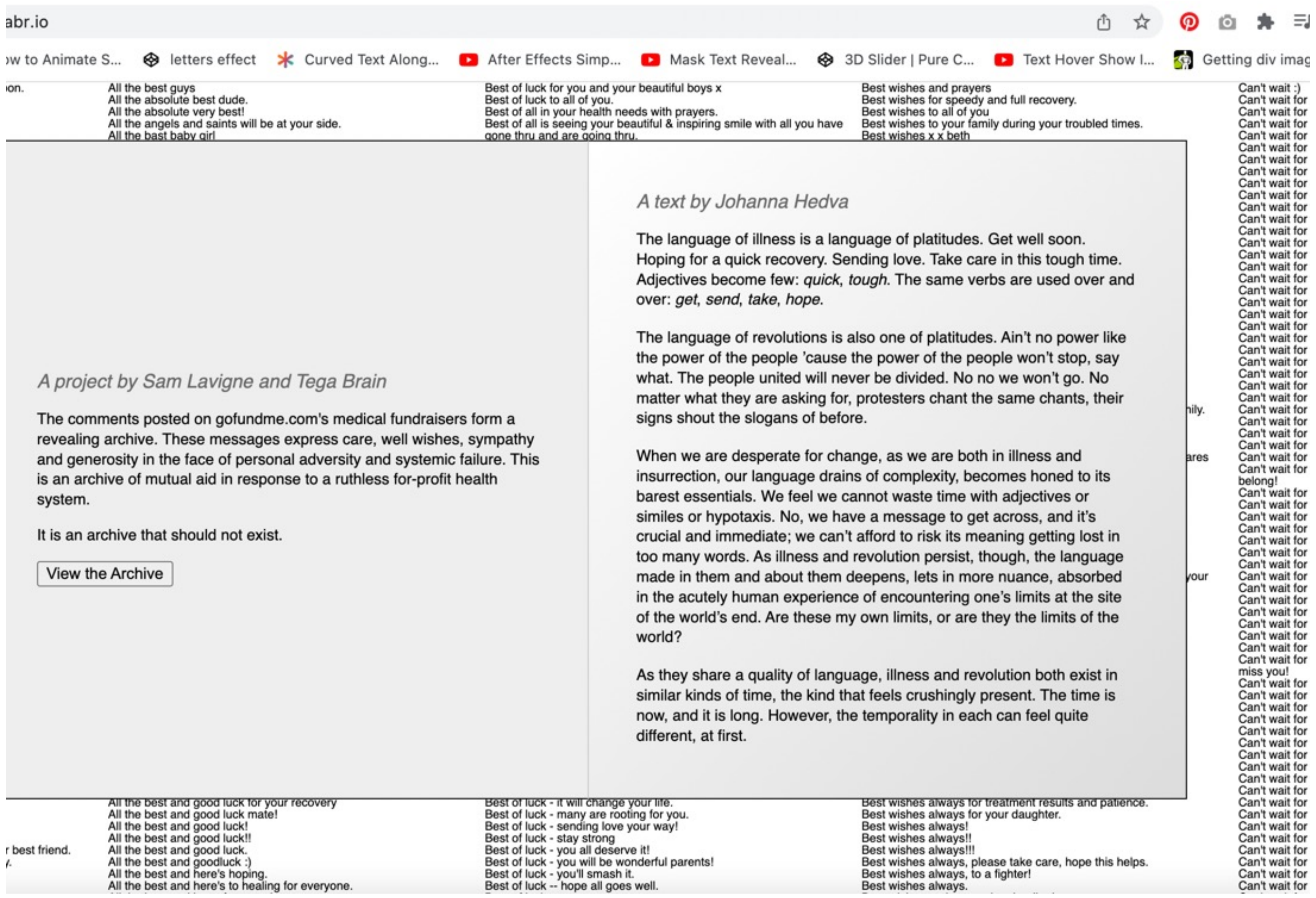
Postsecret
13345 Copper
Ridge Rd.
Germantown
MD 20874
USA

Abstract

The brief explains to explore an idea in depth, both technically and conceptually. It had to involve Big Data in some way, as well as include a web-based experiment.

The web based experiment chosen was a dynamic website, and the project itself is to visualise the handwritten messages on the back of lost postcards. Using OCR to extract the text and then convert it into text one can manipulate. How interesting it is to see a message written to another person out there; the emotions and thoughts revealed.

Dilemmas


abr.io

ow to Animate S... letters effect Curved Text Along... After Effects Simp... Mask Text Reveal... 3D Slider | Pure C... Text Hover Show I... Getting div imag

ion.

All the best guys
All the absolute best dude.
All the absolute very best!
All the angels and saints will be at your side.
All the bast baby girl

Best of luck for you and your beautiful boys x
Best of luck to all of you.
Best of all in your health needs with prayers.
Best of all is seeing your beautiful & inspiring smile with all you have gone thru and are going thru.

Best wishes and prayers
Best wishes for speedy and full recovery.
Best wishes to all of you
Best wishes to your family during your troubled times.
Best wishes x x beth

A project by Sam Lavigne and Tega Brain

The comments posted on gofundme.com's medical fundraisers form a revealing archive. These messages express care, well wishes, sympathy and generosity in the face of personal adversity and systemic failure. This is an archive of mutual aid in response to a ruthless for-profit health system.

It is an archive that should not exist.

View the Archive

A text by Johanna Hedva

The language of illness is a language of platitudes. Get well soon. Hoping for a quick recovery. Sending love. Take care in this tough time. Adjectives become few: *quick, tough.* The same verbs are used over and over: *get, send, take, hope.*

The language of revolutions is also one of platitudes. Ain't no power like the power of the people 'cause the power of the people won't stop, say what. The people united will never be divided. No no we won't go. No matter what they are asking for, protesters chant the same chants, their signs shout the slogans of before.

When we are desperate for change, as we are both in illness and insurrection, our language drains of complexity, becomes honed to its barest essentials. We feel we cannot waste time with adjectives or similes or hypotaxis. No, we have a message to get across, and it's crucial and immediate; we can't afford to risk its meaning getting lost in too many words. As illness and revolution persist, though, the language made in them and about them deepens, lets in more nuance, absorbed in the acutely human experience of encountering one's limits at the site of the world's end. Are these my own limits, or are they the limits of the world?

As they share a quality of language, illness and revolution both exist in similar kinds of time, the kind that feels crushingly present. The time is now, and it is long. However, the temporality in each can feel quite different, at first.

ily.

ares

your

r best friend.

All the best and good luck for your recovery
All the best and good luck mate!
All the best and good luck!
All the best and good luck!!
All the best and good luck.
All the best and goodluck :)
All the best and here's hoping.
All the best and here's to healing for everyone.

Best of luck - it will change your life.
Best of luck - many are rooting for you.
Best of luck - sending love your way!
Best of luck - stay strong
Best of luck - you all deserve it!
Best of luck - you will be wonderful parents!
Best of luck - you'll smash it.
Best of luck -- hope all goes well.

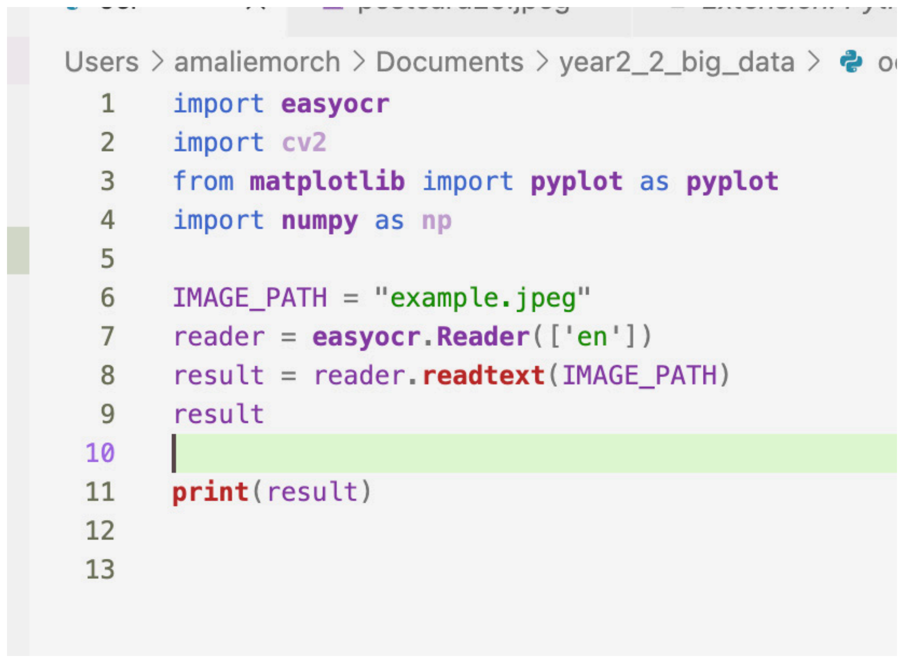
Best wishes always for treatment results and patience.
Best wishes always for your daughter.
Best wishes always!
Best wishes always!!
Best wishes always!!!
Best wishes always, please take care, hope this helps.
Best wishes always, to a fighter!
Best wishes always.

Can't wait :)
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
belong!
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
miss you!
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for
Can't wait for

The process started with the idea of a dynamic website where a new poem/ message is presented on every refresh. A possible addition would be to add a voice to read it out loud. Some dilemmas were:

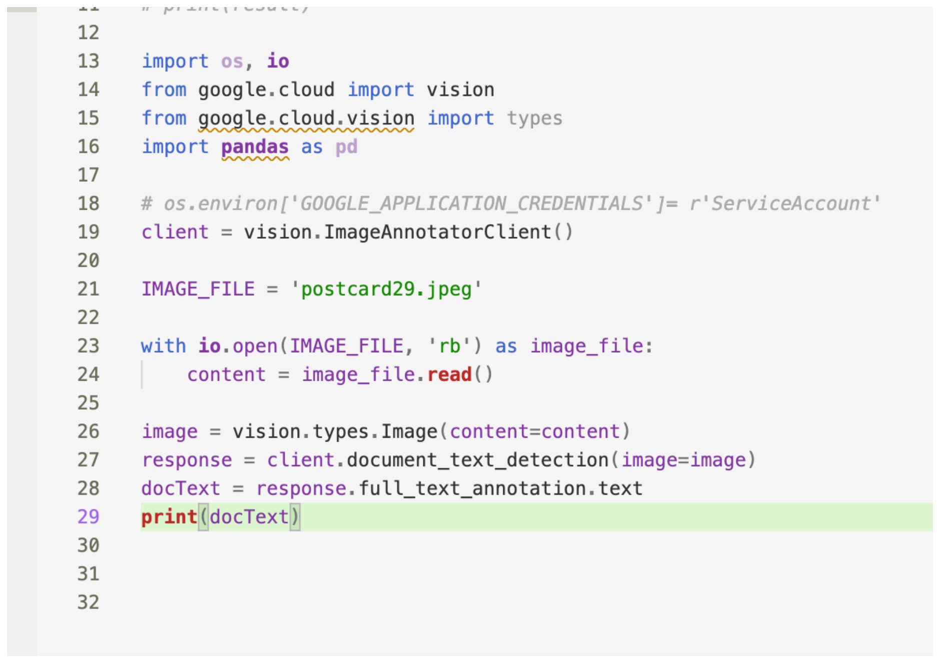
- To keep the handwriting or convert it? The presentation is more personal if the handwriting is kept.
- If the messages were to be read out loud, would it be a real person or automated; again, different impression.
- To include data as in how many letters sent and lost a year?
- Possible automated messages where one enters a phone number or email and a random message is sent.

Development

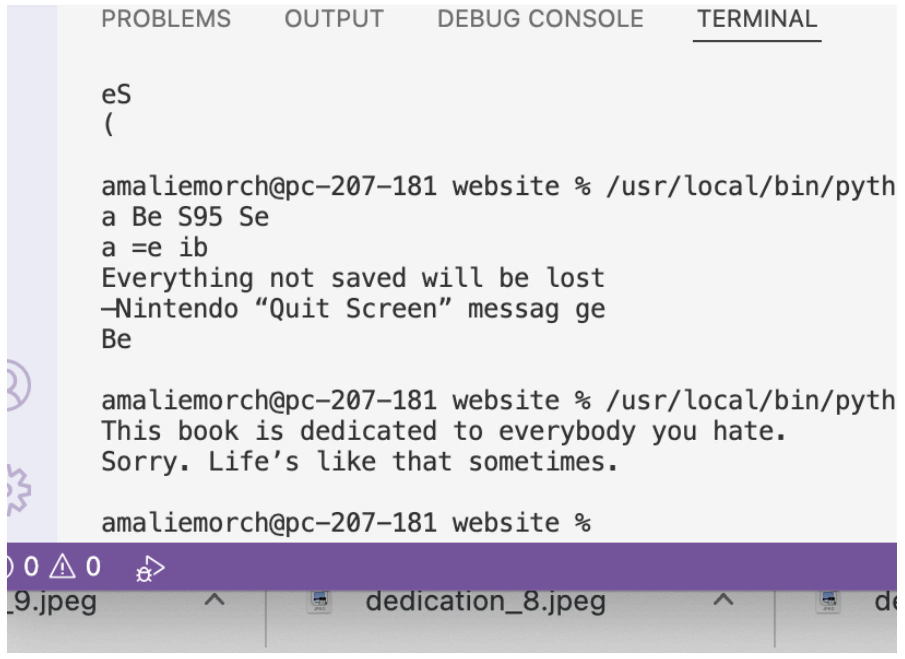


Users > amaliemorch > Documents > year2_2_big_data >

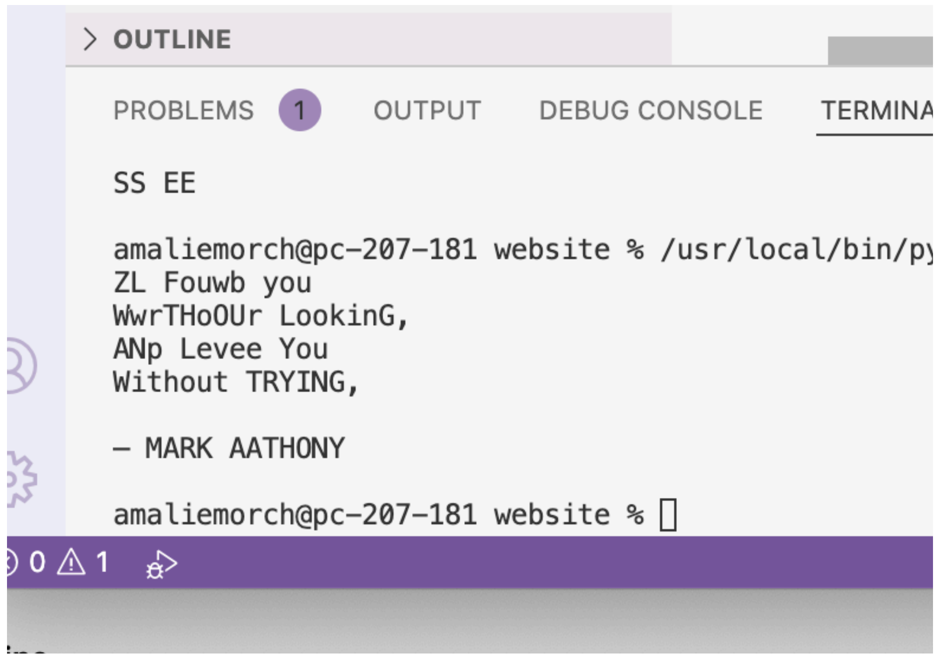
```python
import easyocr
import cv2
from matplotlib import pyplot as pyplot
import numpy as np

IMAGE_PATH = "example.jpeg"
reader = easyocr.Reader(['en'])
result = reader.readtext(IMAGE_PATH)
result

print(result)
```



```python
import os, io
from google.cloud import vision
from google.cloud.vision import types
import pandas as pd

# os.environ['GOOGLE_APPLICATION_CREDENTIALS']= r'ServiceAccount'
client = vision.ImageAnnotatorClient()

IMAGE_FILE = 'postcard29.jpeg'

with io.open(IMAGE_FILE, 'rb') as image_file:
    content = image_file.read()

image = vision.types.Image(content=content)
response = client.document_text_detection(image=image)
docText = response.full_text_annotation.text
print(docText)
```



```
eS
(

amaliemorch@pc-207-181 website % /usr/local/bin/pyth
a Be S95 Se
a =e ib
Everything not saved will be lost
–Nintendo “Quit Screen” messag ge
Be

amaliemorch@pc-207-181 website % /usr/local/bin/pyth
This book is dedicated to everybody you hate.
Sorry. Life’s like that sometimes.

amaliemorch@pc-207-181 website %
```



```
SS EE

amaliemorch@pc-207-181 website % /usr/local/bin/py
ZL Fouwb you
WwrTHoOUr LookinG,
ANp Levee You
Without TRYING,

— MARK AATHONY

amaliemorch@pc-207-181 website %
```

There were multiple options as to how the project could be coded. Each with problems:

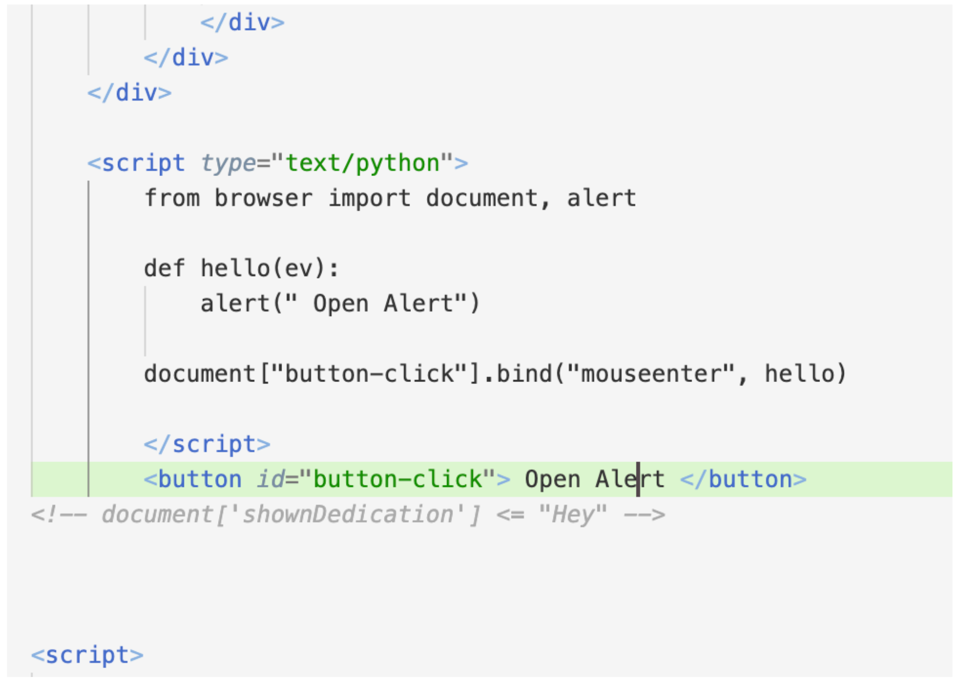
- Option 1 Write html in python
  - Problem adding css
- Option 2 Write python in html
  - Problem getting code to work
- Option 3 Write js in html
  - Problem linking code

The first attempts were done in Python. Different OCR’s were tried out, but pytesseract proved to work the best. However, as can be seen by the bottom pictures, the accuracy wasnt the best. It was at that point that the handwritten postcards were temporarily changed to the dedications from books as to more easily interpret the converted text.

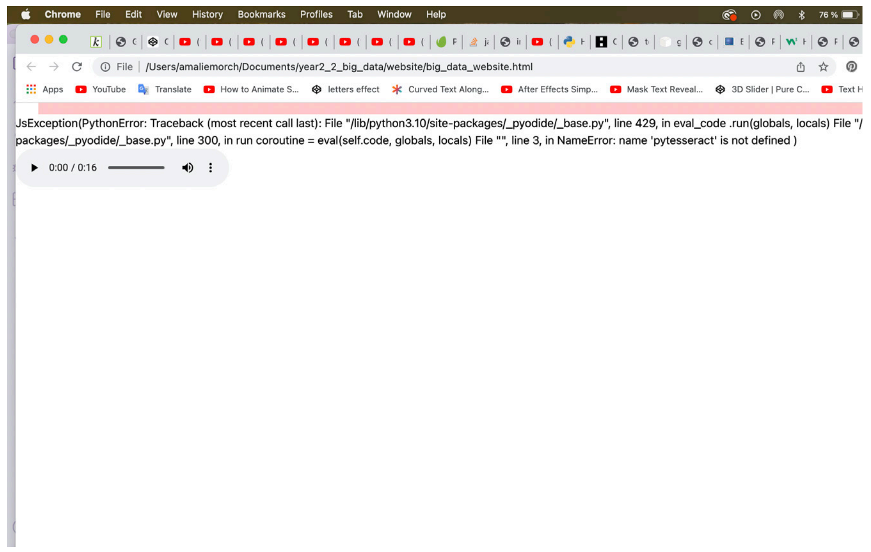
Experimentation

Python

The next step was to add text to speech to play in the background, which was done with gtts and playsound. The python code was then pretty much complete. That’s when there was an attempt to link the python code to the HTML file using pyscript, however errors either occured or the page would be stuck loading. It was later attempted to be done using brython as well, however that wasn’t very successful either.

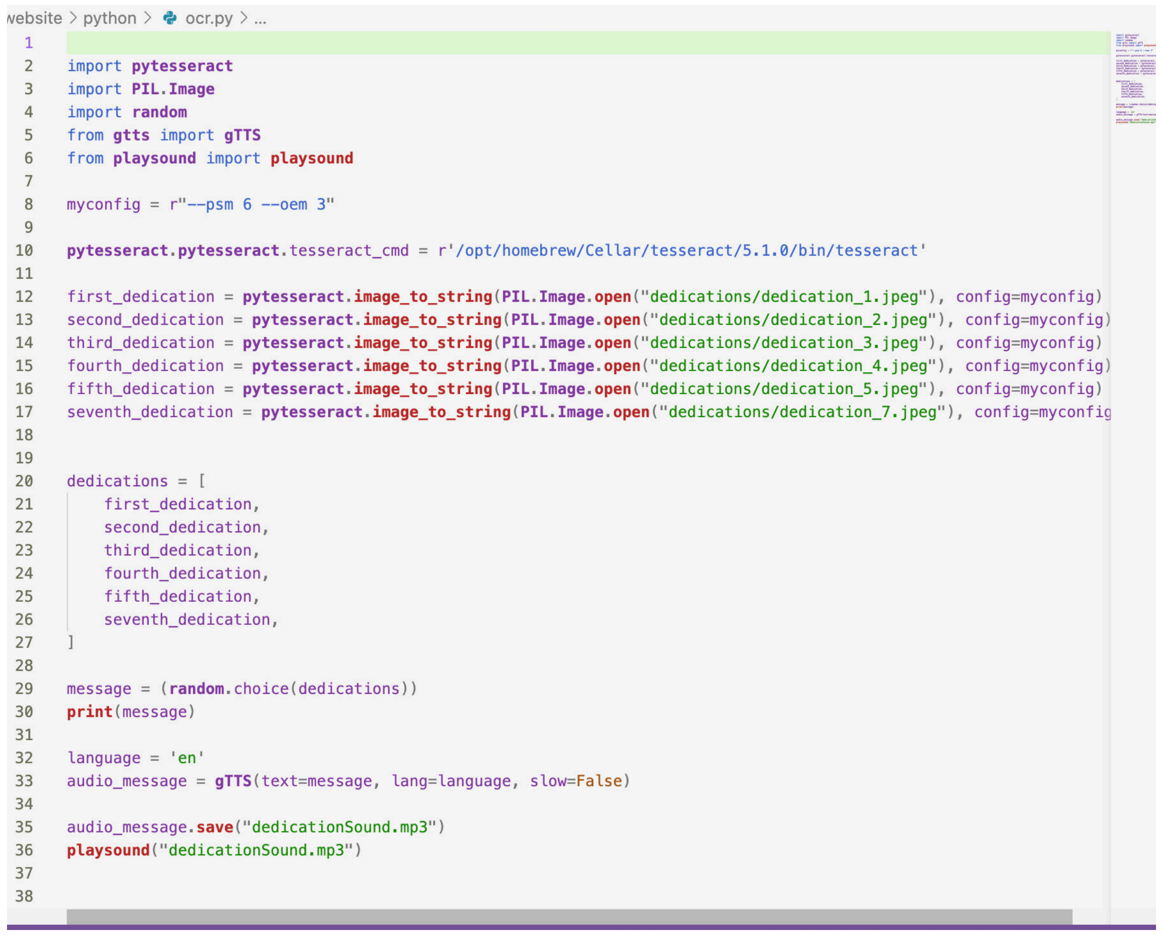


```
            </div>
        </div>
    </div>

    <script type="text/python">
        from browser import document, alert

        def hello(ev):
            alert(" Open Alert")

        document["button-click"].bind("mouseenter", hello)

        </script>
        <button id="button-click"> Open Alert </button>
<!-- document['shownDedication'] <= "Hey" -->


<script>
```

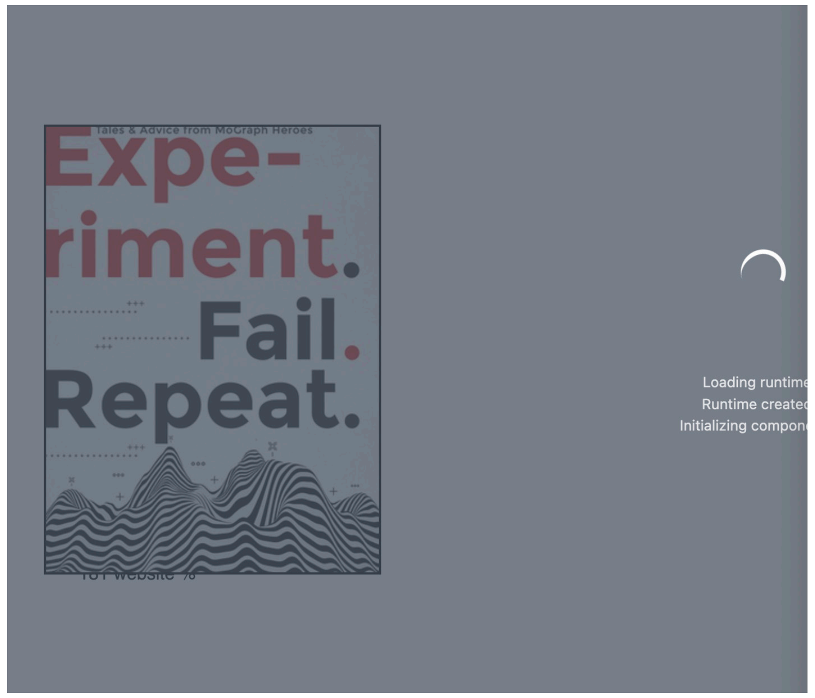


Chrome File Edit View History Bookmarks Profiles Tab Window Help
/Users/amaliemorch/Documents/year2_2_big_data/website/big_data_website.html
Apps YouTube Translate How to Animate S... letters effect Curved Text Along... After Effects Simp... Mask Text Reveal... 3D Slider | Pure C... Text H
0:00 / 0:16

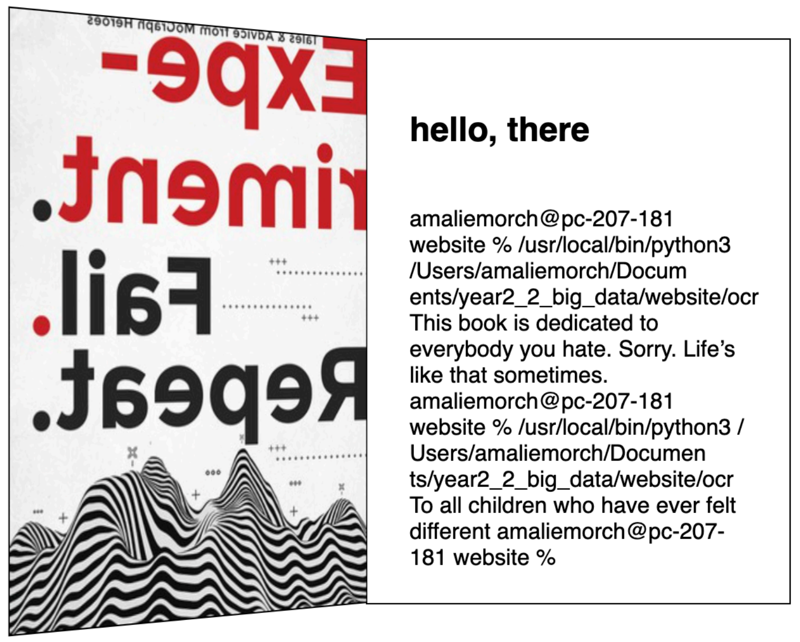
```
JsException(PythonError: Traceback (most recent call last): File "/lib/python3.10/site-packages/_pyodide/_base.py", line 429, in eval_code .run(globals, locals) File "/
packages/_pyodide/_base.py", line 300, in run coroutine = eval(self.code, globals, locals) File "", line 3, in NameError: name 'pytesseract' is not defined )
```



```python

import pytesseract
import PIL.Image
import random
from gtts import gTTS
from playsound import playsound

myconfig = r"--psm 6 --oem 3"

pytesseract.pytesseract.tesseract_cmd = r'/opt/homebrew/Cellar/tesseract/5.1.0/bin/tesseract'

first_dedication = pytesseract.image_to_string(PIL.Image.open("dedications/dedication_1.jpeg"), config=myconfig)
second_dedication = pytesseract.image_to_string(PIL.Image.open("dedications/dedication_2.jpeg"), config=myconfig)
third_dedication = pytesseract.image_to_string(PIL.Image.open("dedications/dedication_3.jpeg"), config=myconfig)
fourth_dedication = pytesseract.image_to_string(PIL.Image.open("dedications/dedication_4.jpeg"), config=myconfig)
fifth_dedication = pytesseract.image_to_string(PIL.Image.open("dedications/dedication_5.jpeg"), config=myconfig)
seventh_dedication = pytesseract.image_to_string(PIL.Image.open("dedications/dedication_7.jpeg"), config=myconfig


dedications = [
    first_dedication,
    second_dedication,
    third_dedication,
    fourth_dedication,
    fifth_dedication,
    seventh_dedication,
]

message = (random.choice(dedications))
print(message)

language = 'en'
audio_message = gTTS(text=message, lang=language, slow=False)

audio_message.save("dedicationSound.mp3")
playsound("dedicationSound.mp3")


```

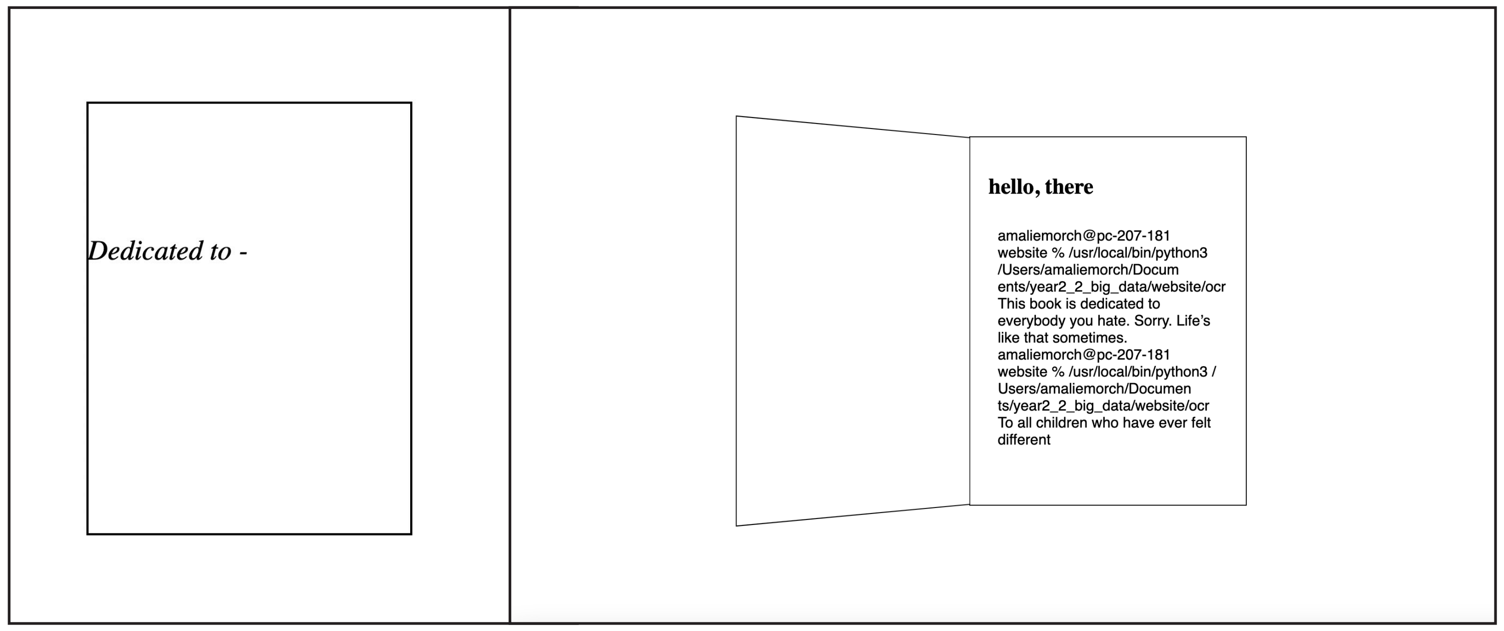


Tales & Advice from MoGraph Heroes
Expe-riment. Fail. Repeat.
Loading runtime
Runtime created
Initializing compon



hello, there

amaliemorch@pc-207-181
website % /usr/local/bin/python3
/Users/amaliemorch/Docum
ents/year2_2_big_data/website/ocr
This book is dedicated to
everybody you hate. Sorry. Life's
like that sometimes.
amaliemorch@pc-207-181
website % /usr/local/bin/python3 /
Users/amaliemorch/Documen
ts/year2_2_big_data/website/ocr
To all children who have ever felt
different amaliemorch@pc-207-
181 website %



Dedicated to -

hello, there

amaliemorch@pc-207-181
website % /usr/local/bin/python3
/Users/amaliemorch/Docum
ents/year2_2_big_data/website/ocr
This book is dedicated to
everybody you hate. Sorry. Life's
like that sometimes.
amaliemorch@pc-207-181
website % /usr/local/bin/python3 /
Users/amaliemorch/Documen
ts/year2_2_big_data/website/ocr
To all children who have ever felt
different

Experimentation

Visual HTML

The original idea was to have an envelope opening, a letter slide out with the converted text. However that idea changed to a book opening instead. The book started out as closed with the text printed at the front like a title. However, the dedications in books are inside so it was then changed to be presented as open. Furtehr, it was found a little boring, so an animation was created on hover so the book opens and closes.

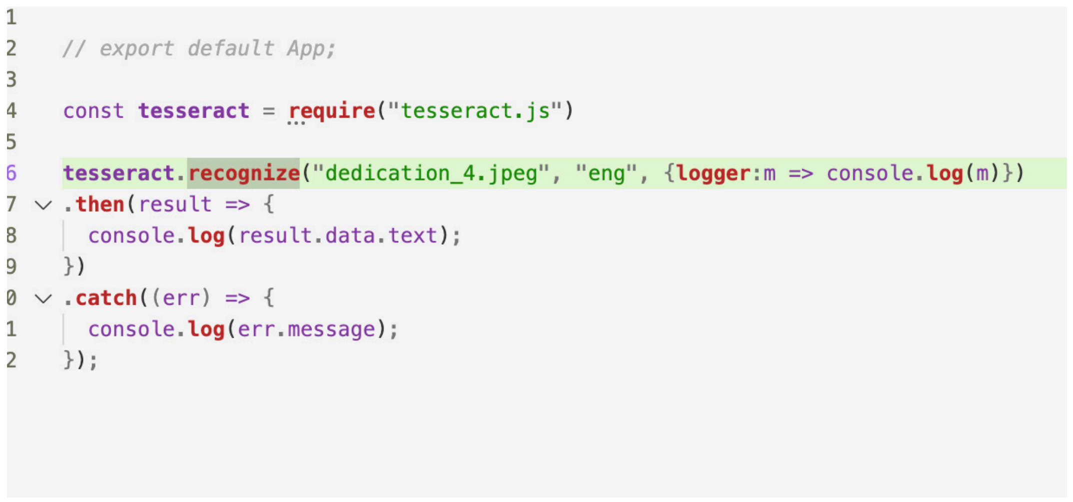
Experimentation

Javascript

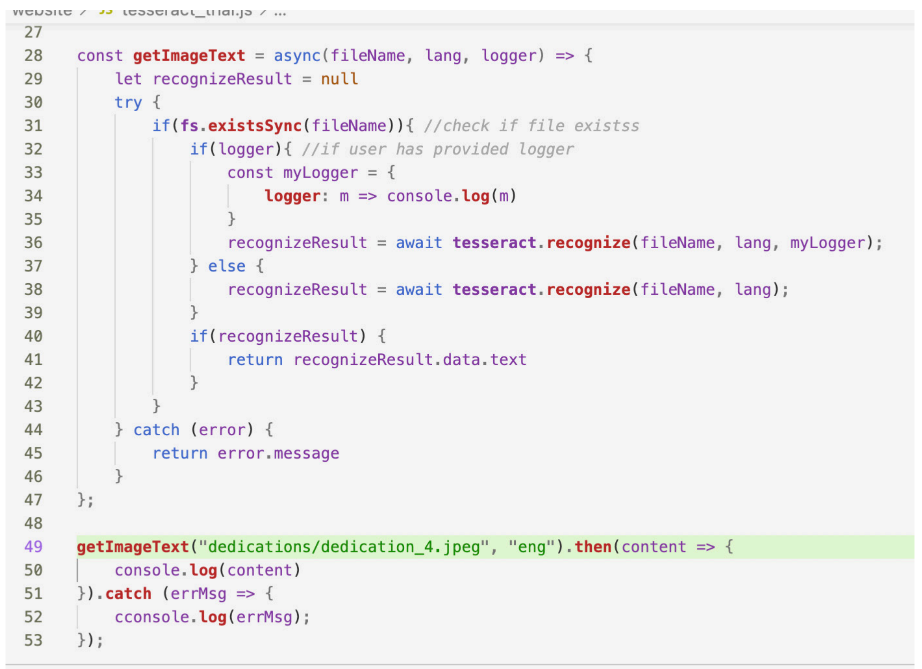
The next step was to try to implement the OCR in Javascript, instead of Python. This was a big challenge as there wasn't much experience with this. There were several attempt at implementing the OCR, as it simply wouldnt visualise in the HTML. It would however print fine in the console, so the code was indeed working.

Many possible solutions were implemented to try to solve the problem:

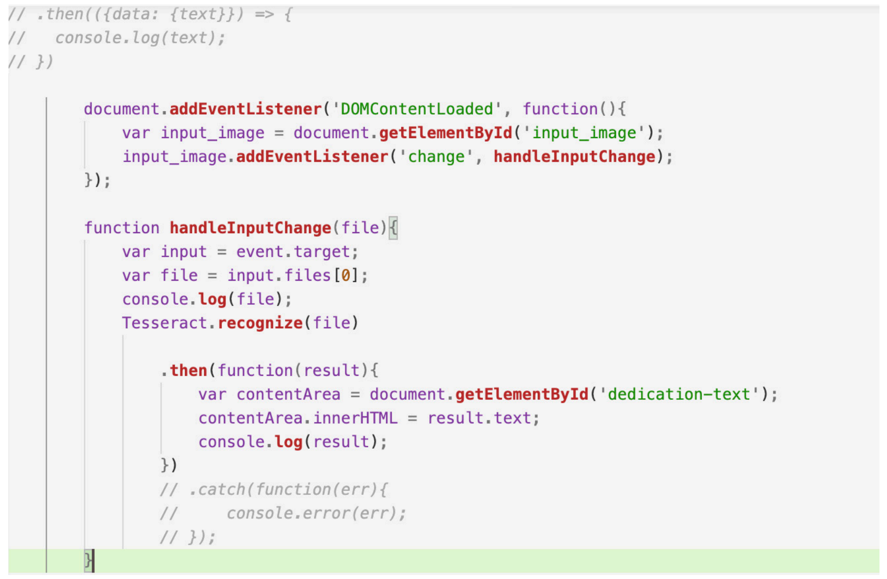
- To replace innerHTML with innerText.
- To move the id element before and after the script element (before the JS was moved to a seperate file).
- To change the linked HTML element.
- To replace require() with import, as well several other attempts to remove the error occuring in the browser as to items not being defined.
- Turned out the solution was a missing CDN link in the head.



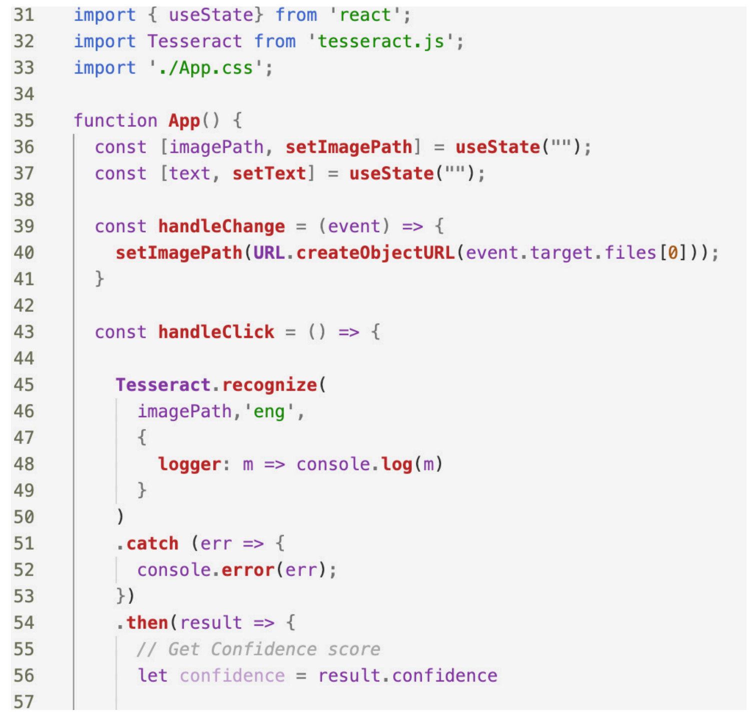
```javascript

// export default App;

const tesseract = require("tesseract.js")

tesseract.recognize("dedication_4.jpeg", "eng", {logger:m => console.log(m)})
.then(result => {
  console.log(result.data.text);
})
.catch((err) => {
  console.log(err.message);
});
```



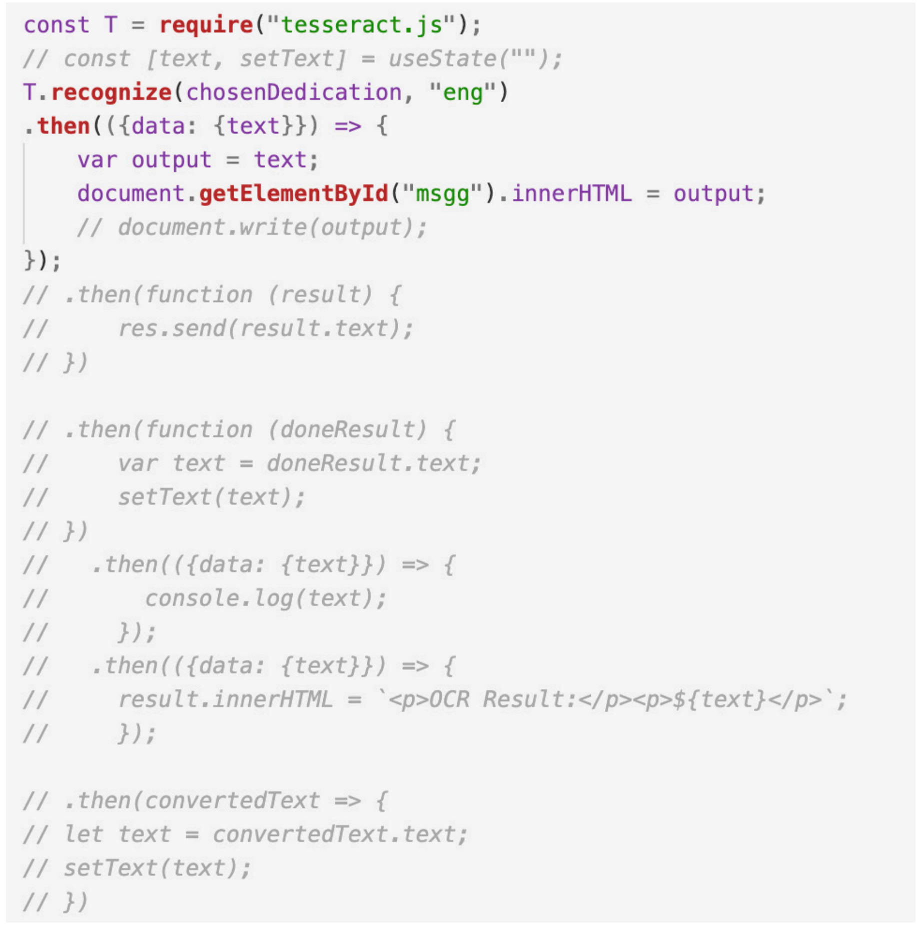
```javascript

const getImageText = async(fileName, lang, logger) => {
    let recognizeResult = null
    try {
        if(fs.existsSync(fileName)){ //check if file existss
            if(logger){ //if user has provided logger
                const myLogger = {
                    logger: m => console.log(m)
                }
                recognizeResult = await tesseract.recognize(fileName, lang, myLogger);
            } else {
                recognizeResult = await tesseract.recognize(fileName, lang);
            }
            if(recognizeResult) {
                return recognizeResult.data.text
            }
        }
    } catch (error) {
        return error.message
    }
};

getImageText("dedications/dedication_4.jpeg", "eng").then(content => {
    console.log(content)
}).catch (errMsg => {
    cconsole.log(errMsg);
});
```



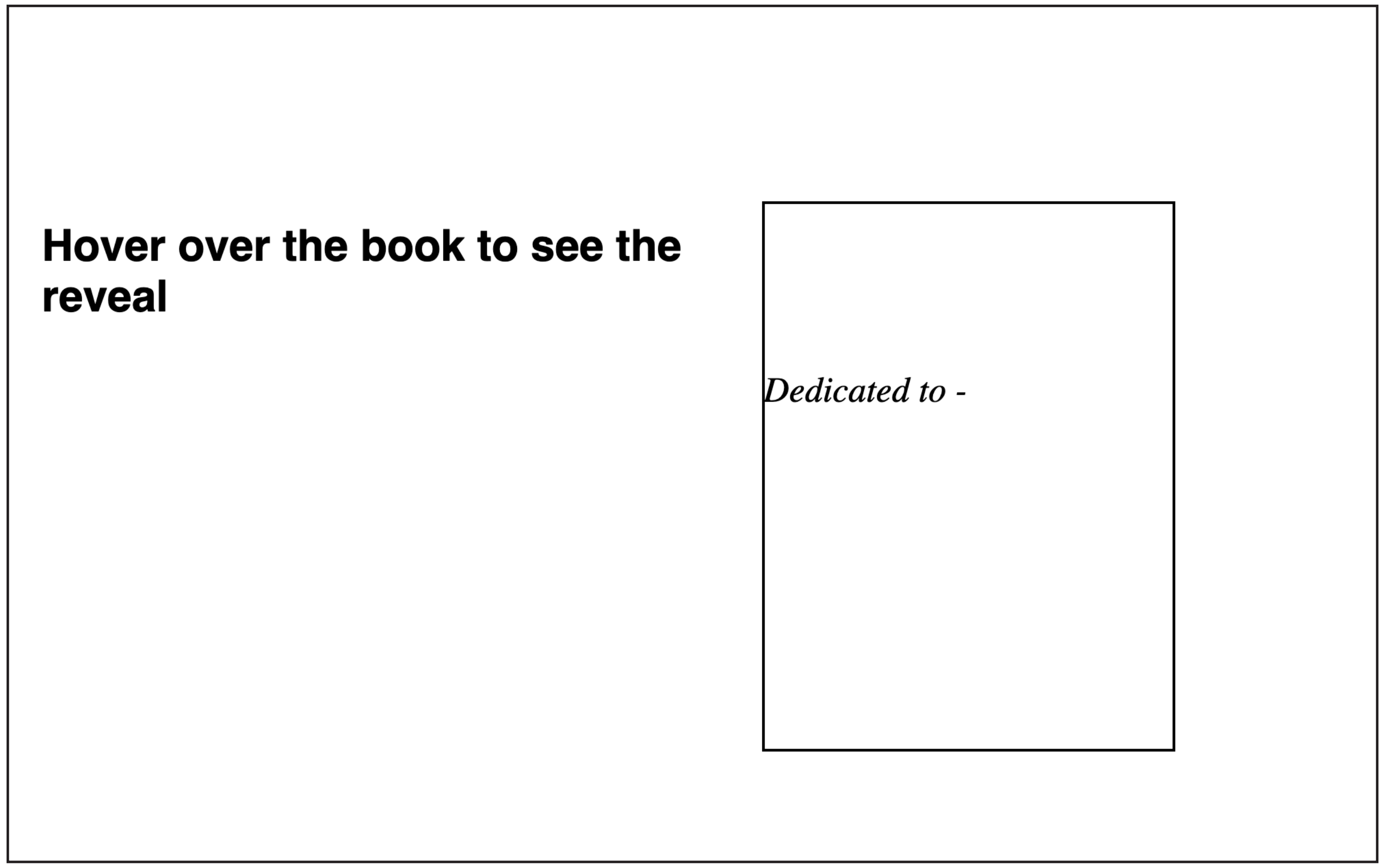
```javascript
// .then(({data: {text}}) => {
//   console.log(text);
// })

    document.addEventListener('DOMContentLoaded', function(){
        var input_image = document.getElementById('input_image');
        input_image.addEventListener('change', handleInputChange);
    });

    function handleInputChange(file){
        var input = event.target;
        var file = input.files[0];
        console.log(file);
        Tesseract.recognize(file)

            .then(function(result){
                var contentArea = document.getElementById('dedication-text');
                contentArea.innerHTML = result.text;
                console.log(result);
            })
            // .catch(function(err){
            //     console.error(err);
            // });
    }
```



```javascript
import { useState} from 'react';
import Tesseract from 'tesseract.js';
import './App.css';

function App() {
  const [imagePath, setImagePath] = useState("");
  const [text, setText] = useState("");

  const handleChange = (event) => {
    setImagePath(URL.createObjectURL(event.target.files[0]));
  }

  const handleClick = () => {

    Tesseract.recognize(
      imagePath,'eng',
      {
        logger: m => console.log(m)
      }
    )
    .catch (err => {
      console.error(err);
    })
    .then(result => {
      // Get Confidence score
      let confidence = result.confidence

```

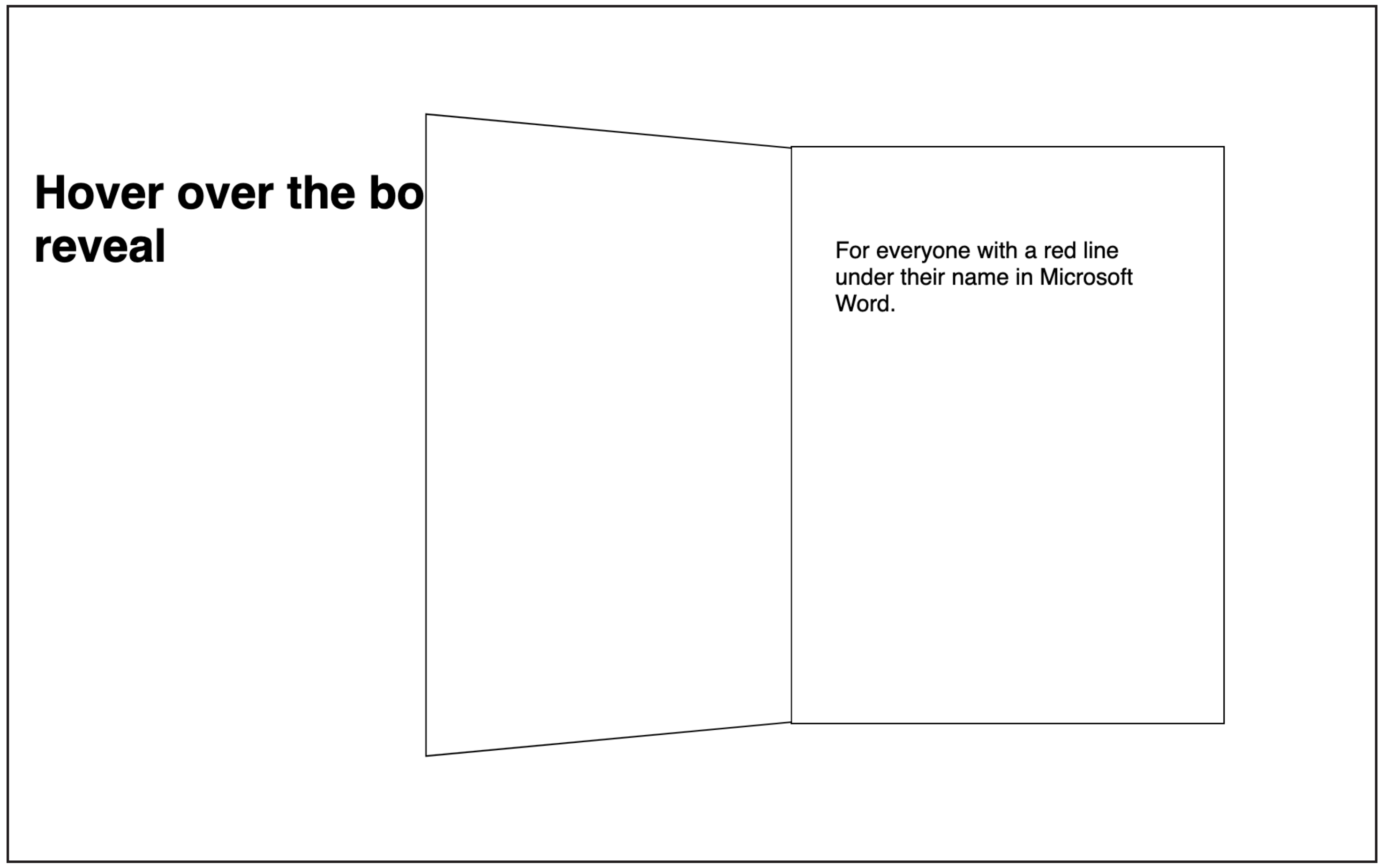


```javascript
const T = require("tesseract.js");
// const [text, setText] = useState("");
T.recognize(chosenDedication, "eng")
.then(({data: {text}}) => {
    var output = text;
    document.getElementById("msgg").innerHTML = output;
    // document.write(output);
});
// .then(function (result) {
//     res.send(result.text);
// })

// .then(function (doneResult) {
//     var text = doneResult.text;
//     setText(text);
// })
//   .then(({data: {text}}) => {
//       console.log(text);
//     });
//   .then(({data: {text}}) => {
//     result.innerHTML = `<p>OCR Result:</p><p>${text}</p>`;
//     });

// .then(convertedText => {
// let text = convertedText.text;
// setText(text);
// })
```



Hover over the book to see the reveal

Dedicated to -

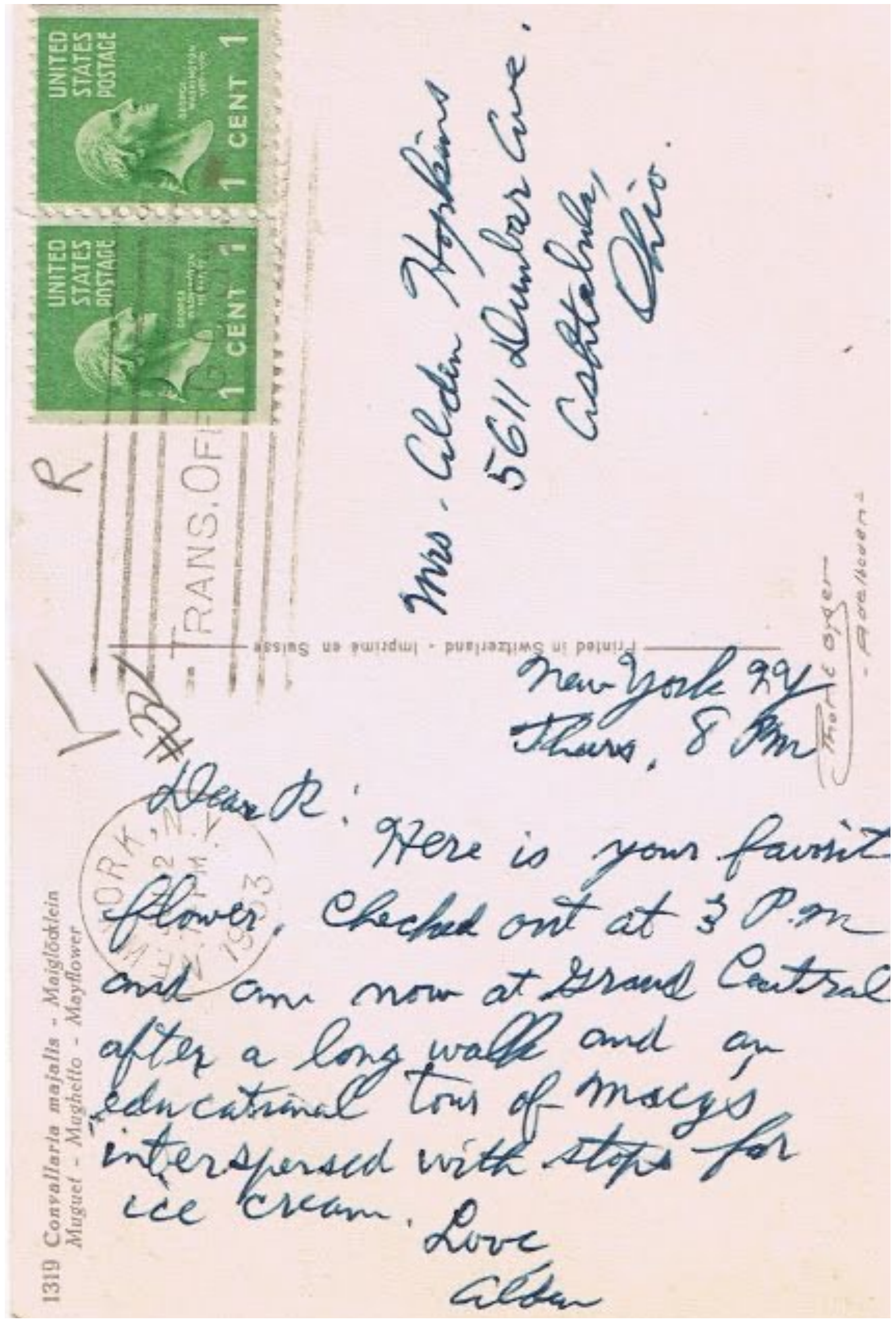


Hover over the bo
reveal

For everyone with a red line under their name in Microsoft Word.

Finished project

The finished project is a dynamic website which shows a book; on hover, the book opens and presents one random dedication out of an array. There is also a text to speech feature playing in the background.



UNITED STATES POSTAGE 1 CENT 1

UNITED STATES POSTAGE 1 CENT 1

TRANS. OF

NEW YORK, N.Y. 1953

Printed in Switzerland - Imprimé en Suisse

1319 Convallaria majalis - Maiglöcklein
Muguet - Mughetto - Mayflower

Mrs. Alden Hopkins
5611 Dunbar Ave.
Ashtabula,
Ohio.

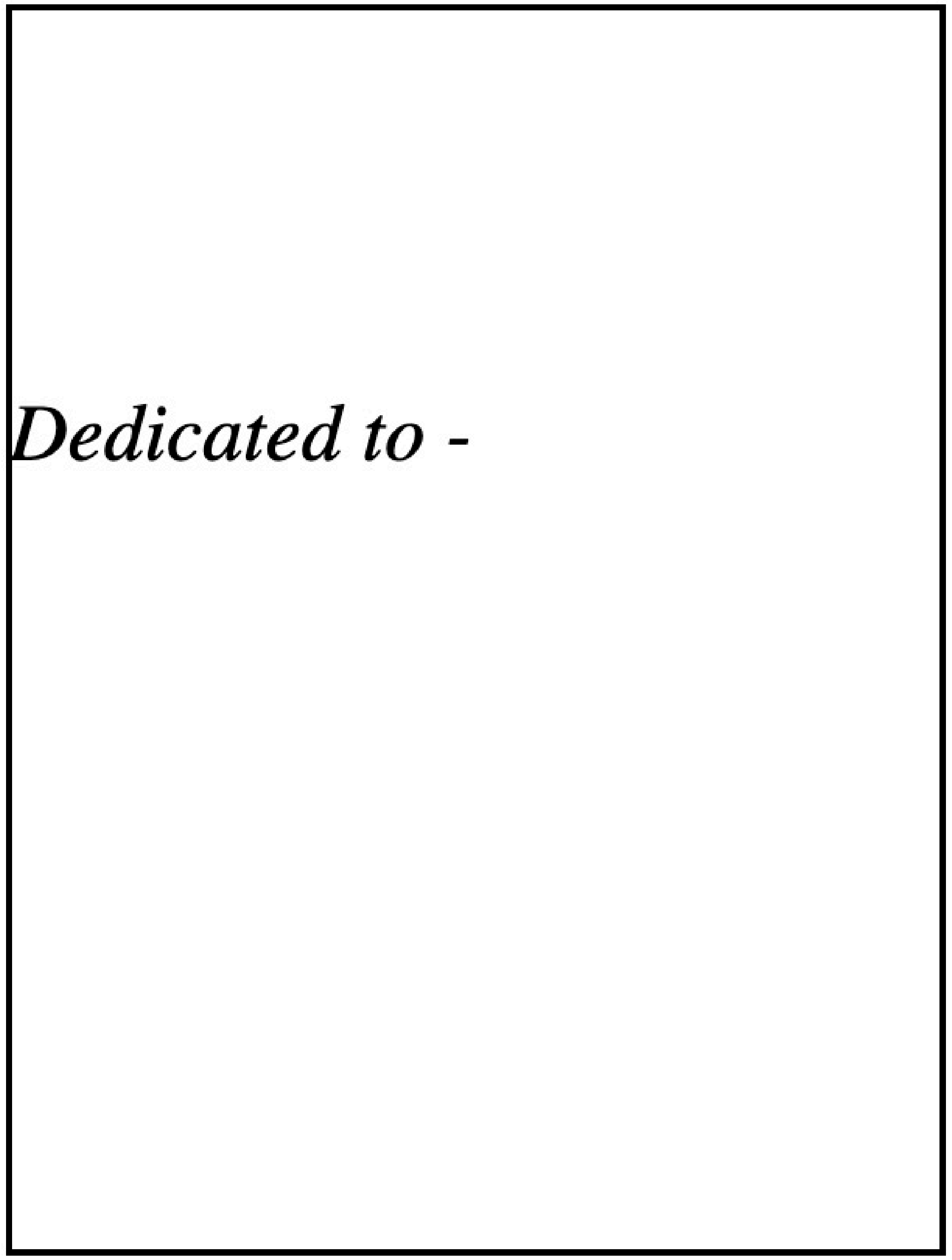
New York 22
Thurs. 8 PM

Dear R: Here is your favorite flower. Checked out at 3 P.M and am now at Grand Central after a long walk and an educational tour of Macy's interspersed with stops for ice cream.
Love
Alden

Possible improvements

Further development

- Accuracy: If further developed, there would be improvements made as to the accuracy of the OCR, as the one currently used includes errors.
- Handwriting: There would be an attempt to figure out the original idea of using postcards by making the conversion more accurate, as well as within specific area.
- Visualise a map (as to loctaions) or timeline (as to dates) postcards were posted; possible using D3.
- If continued with dedications, the option to choose a genre of book to see differences in dedications.


Dedicated to -

Biggest challenge

The biggest challenge was simply visualising the OCR. It was managable to code it in both Python and Javascript, but to actually show the output in HTML was extremely difficult for some reason. Python and HTML wasn't very compatible, and for some reason Javascript simply didn't want to link up to the HTML file. However, with much frustration, and a few tears, it was managable.