

main.py

```
# imports - only import what is needed for security
# while url_for is not directly used in this file it is imported so the HTML files
can use it
# so the HTML files can use it to connect to CSS and JS files
from flask import Flask, render_template, url_for, request, jsonify
from flask_cors import CORS
from yagmail import SMTP
from json import dumps, loads
from os import listdir

# local imports
from functions.openweather import OpenWeather
from functions.ambee import Ambee
from functions.defra import get_pollution
from functions.advice import get_advice

ow = OpenWeather()
am = Ambee()

# create the app
app = Flask(__name__)
CORS(app)

# return pages
@app.route('/')
def homepage():
    return render_template('home.html')

@app.route('/blog')
def blogpage():
    return render_template('blog.html')

@app.route('/weather')
def weatherpage():
    return render_template('weather.html')

@app.route('/contact')
def contactpage():
    return render_template('contact.html')

@app.route('/admin')
def adminpage():
    return render_template('admin.html')

@app.route('/blog/view/<id>')
def viewpage(id): # the argument is not used but is there so that <id> in the URL
can be anything
    return render_template('view.html')
```

```

# weather page content generation
@app.route('/weather/content', methods=['POST'])
def weather_data_generation():
    # get json body
    values = request.get_json()
    longitude = values['longitude']
    latitude = values['latitude']
    time = values['time']

    # openWeather interactions
    city, country = ow.reverse_geocoding(longitude,latitude)
    forecast = ow.forecast(longitude,latitude,time)

    # Ambee interactions
    pollen = am.get_pollen(longitude,latitude)

    # DEFRA map data
    pollution = get_pollution(longitude, latitude)

    # advice generation
    advice = get_advice(forecast,pollen)

    response = {
        'location': [city,country],
        'forecast': forecast,
        'pollen': pollen,
        'pollution': pollution,
        'advice': advice
    }

    return jsonify(response), 200

# send email
@app.route('/contact/send', methods=['POST'])
def send_email():
    # get json body
    data = request.get_json()

    # load the email and password from file of the sender
    with open('sender.txt', 'r') as f:
        contents = [line[:-1] for line in f.readlines()]
        email = contents[0]
        passwd = contents[1]

    # load the recipient from receiver.txt
    with open('receiver.txt', 'r') as f:
        receiver = f.read()

    # formulate email body
    subject = data['subject']

```

```
body = [f"This is {data['name']}. Send replies to  
{data['reply_email']}.\\n\\n\\n{data['body']}"]
```

```
SMTP(email,passwd).send(receiver, subject, body)
```

```
response = {  
    'message' : 'Email sent successfully.'  
}  
return jsonify(response), 201
```

```
# check admin password  
@app.route('/passcheck',methods=['POST'])  
def passcheck():  
    # get json body  
    data = request.get_json()  
    guess_hash = data['hash']  
  
    # load the correct hash  
    with open('correct_pass_hash.txt', 'r') as f:  
        correct_hash = f.read()  
  
    # compare  
    if guess_hash == correct_hash:  
        message = 'Correct password'  
        code = 200  
    else:  
        message = 'Wrong password'  
        code = 401  
  
    # send response  
    response = {  
        'message': message  
    }  
    return jsonify(response), code
```

```
# change admin password  
@app.route('/admin/password', methods=['POST'])  
def changepass():  
    # get json body  
    data = request.get_json()  
    hash = data['hash']  
  
    # write to the file  
    with open('correct_pass_hash.txt', 'w') as f:  
        f.write(hash)  
  
    # response  
    response = {
```

```
        'message': 'Password updated successfully'
    }
    return jsonify(response), 201
```

```
# change / get contact email
@app.route('/admin/email', methods=['GET', 'POST'])
def contact_email():
    # get current email
    if request.method == 'GET':
        with open('receiver.txt', 'r') as f:
            email = f.read()
            response = {
                'email': email
            }
            return jsonify(response), 200
    # update email
    else:
        data = request.get_json()
        with open('receiver.txt', 'w') as f:
            f.write(data['email'])
            response = {
                'message': 'Email updated successfully'
            }
            return jsonify(response), 200
```

```
# blog post creation
@app.route('/admin/create', methods=['POST'])
def create_post():
    blog_json = request.get_json()
    timestamp = blog_json['timestamp']
    # create blog file
    with open(f'static/blog/{timestamp}.json', 'w') as f:
        f.write(dumps(blog_json))
    # response
    response = {
        'message': 'Blog post created successfully.'
    }
    return jsonify(response), 201
```

```
# gets all blog posts in reverse release order
@app.route('/blog/all', methods=['GET'])
def load_blogs():
    # get a list of all files
    posts = listdir('static/blog')
    posts.sort()
    # display them in console for debug
```

```

print(f'Contents of static/blog: {posts}')

# for each one load it and store the contents in all_posts
# which is a list so they stay in the correct order when sent
all_posts = []
for post_name in posts:
    with open(f'static/blog/{post_name}', 'r') as f:
        post_str = f.read()

        post_json = loads(post_str)
        all_posts.append(post_json)

# newest first
all_posts.reverse()

# response
response = {
    'all_posts': all_posts
}
return jsonify(response), 200

# gets just the most recent blog post to save on transfer time
# this function is very similar to the above one but without the loop
@app.route('/lastblog', methods=['GET'])
def last_blog():
    # get a list of all files
    posts = listdir('static/blog')
    posts.sort()
    # display them in console for debug
    print(f'Contents of static/blog: {posts}')

    # get contents of most recent
    with open(f'static/blog/{posts[-1]}', 'r') as f:
        post_str = f.read()

    # response - directly send the blog json so less indexing problems at the other
end
response = loads(post_str)
return jsonify(response), 200

# gets a specific blog post
# very similar to the above but after a specific value instead of the last one
# so it doesn't need to list all of the options
@app.route('/getblog/<id>', methods=['GET'])
def specific_blog(id):
    # get contents of correct post
    with open(f'static/blog/{id}.json', 'r') as f:
        post_str = f.read()

```

```
# response - directly send the blog json so less indexing problems at the other
end
response = loads(post_str)
return jsonify(response), 200
```

```
## RUN ##
if __name__ == '__main__':
    app.run('127.0.0.1', 8080)
```

correct_pass_hash.txt

f8459a0433df302e1c87fd8e9729f36f80b040c5ee1bba2262795a82e490dc35

sender.txt

healthadvicegroup@gmail.com
eqryuepvrttkqjx
H341thc4r3

first line is email, second line is app password,
third is normal password for your access

DO NOT EDIT THIS FILE!!!

receiver.txt

jacob.vincent.331@accesstomusic.ac.uk

/functions/advice.py

```
"""This file will generate the advice for the weather
page. At the moment, it just generates sample advice
based on the other sample data."""
```

```
def get_advice(forecast, pollen):
    return [
        {
            'title': 'Cold temperatures soon',
            'description': 'Wrap up nice and warm! Additionally, maybe stay at home and
keep warm with a hot water bottle if you can.'
        },
        {
```

```

        'title': 'Raining soon',
        'description' : 'Make sure to wear waterproof clothing, or bring an umbrella!
Also, keep in mind that roads can be more slippery when wet.'
    },
    {
        'title': 'High pollen count',
        'description' : 'If you suffer from hay fever, be sure you know the risks and
learn to prevent them.'
    }
]

```

functions/ambee.py

```

"""This file contains functions that relate to
Ambee's API. Until a subscription is
acquired this will generate dummy data.

Subscriptions are not free, however after talking
to Ambee's support team they have agreed to give me
5000 free requests that last until August, which is
long enough for this project to be graded."""

```

```

class Ambee:
    # space for __init__ to load auth key
    def __init__(self):
        pass

    def get_pollen(self, long, lat):
        return {
            'grass' : 2,
            'trees' : 120,
            'weeds' : 90
        }

```

functions/defra.py

```

"""This file contains functions that relate to
DEFRA map data. This returns dummy data for now, but
will use actual data via Pandas once downloaded"""

def get_pollution(long, lat):
    return [
        {
            'year': 2021, # this is the newest publicly available data
            'value': 0.808
        },
        {
            'year': 2020,
            'value': 0.917
        }
    ]

```

```

    },
    {
        'year': 2019,
        'value': 0.902
    },
    {
        'year': 2018,
        'value': 1.024
    },
    {
        'year': 2017,
        'value': 1.056
    },
]

```

/functions/openweather.py

```

"""This file contains functions that relate to
OpenWeather's API. Until a free subscription is
acquired this will generate dummy data."""

```

```

class OpenWeather:
    # space for __init__ to load auth key
    def __init__(self):
        pass

    def reverse_geocoding(self, long, lat):
        return 'Norwich', 'UK'

    def forecast(self, long, lat, time):
        return {
            'Now': {
                'type': '01d', # the type refers to the icons openweather provide
                'temp': 16
            },
            '11am': {
                'type': '01d',
                'temp': 16
            },
            '12pm': {
                'type': '02d',
                'temp': 17
            },
            '1pm': {
                'type': '04d',
                'temp': 13
            },
            '2pm': {
                'type': '09d',
                'temp': 9
            }
        }

```



```
{
  '3pm': {
    'type': '09d',
    'temp': 9
  },
  '4pm': {
    'type': '09d',
    'temp': 10
  },
  '5pm': {
    'type': '09d',
    'temp': 10
  },
  '6pm': {
    'type': '04d',
    'temp': 9
  },
  '7pm': {
    'type': '09n',
    'temp': 8
  }
}
```

/static/blog/1679654311845.json

```
{
  "title": "Example Post 1",
  "image": "https://openweathermap.org/img/wn/10d@2x.png",
  "preview": "Here is the first blog post that has been created. The aim of this page is to give people more generic advice that isn't specific to only certain individuals.",
  "html": "<h1>Example Post 1 - The blog system</h1><br><p>These blogs exist to spread advice to the masses without needing to contact everyone individually. While the 'Contact us' page can be used for people to ask you for help, this system effectively lets you give widespread help to anyone who uses the site.<br><br>With the ability to add your own HTML to these posts, you could include:</p><ul><li>Bullet point lists</li><li>Images <img src='https://openweathermap.org/img/wn/10d@2x.png'></li><li><a href='/'>Links</a></li><li>Or basically anything else!</li></ul><p>So go wild with it, and experiment to see what is possible.</p>",
  "timestamp": 1679654311845
}
```

/static/blog/1679654688955.json

```
{
  "title": "Example Post 2",
  "image": "https://openweathermap.org/img/wn/09d@2x.png",
  "preview": "10 paragraphs of lorem ipsum!",
  "html": "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. A scelerisque purus semper eget duis. Egestas diam in arcu cursus euismod quis viverra. Eu mi bibendum neque egestas congue quisque egestas diam. Proin libero nunc consequat interdum varius sit amet mattis vulputate. Sed augue lacus viverra vitae congue eu consequat"
}
```

ac felis. Etiam erat velit scelerisque in dictum non consectetur. Aliquet sagittis id consectetur purus. Feugiat vivamus at augue eget arcu dictum varius duis. Cras semper auctor neque vitae. Ornare arcu odio ut sem nulla pharetra
diam.

Sodales ut etiam sit amet. Pretium nibh ipsum consequat nisl vel. Egestas pretium aenean pharetra magna ac placerat. Diam vulputate ut pharetra sit amet aliquam id diam. Tincidunt lobortis feugiat vivamus at augue eget. Sit amet dictum sit amet justo. Proin sagittis nisl rhoncus mattis rhoncus urna neque viverra. Fringilla urna porttitor rhoncus dolor purus. In hac habitasse platea dictumst vestibulum. Ullamcorper morbi tincidunt ornare massa eget egestas purus. Euismod quis viverra nibh cras pulvinar. Sit amet luctus venenatis lectus magna fringilla. Ut aliquam purus sit amet luctus venenatis lectus magna fringilla. Dui faucibus in ornare quam. Non tellus orci ac auctor augue mauris augue. Condimentum vitae sapien pellentesque habitant morbi tristique senectus et. Lacus vel facilisis volutpat est velit. Tincidunt tortor aliquam nulla facilisi cras. Adipiscing vitae proin sagittis nisl rhoncus mattis rhoncus urna. Pellentesque nec nam aliquam sem et tortor consequat id porta.

Et tortor consequat id porta nibh venenatis. Volutpat sed cras ornare arcu. Porta non pulvinar neque laoreet suspendisse interdum consectetur libero. Adipiscing diam donec adipiscing tristique risus nec feugiat in fermentum. Iaculis urna id volutpat lacus laoreet. Interdum varius sit amet mattis vulputate enim. At tempor commodo ullamcorper a lacus vestibulum sed arcu non. Mattis aliquam faucibus purus in massa tempor nec feugiat nisl. Nisi quis eleifend quam adipiscing vitae proin sagittis. Ornare massa eget egestas purus viverra. Dictum sit amet justo donec enim. Sapien pellentesque habitant morbi tristique senectus et netus et malesuada. Convallis aenean et tortor at risus viverra. Tellus at urna condimentum mattis pellentesque id nibh.

Nunc sed id semper risus in hendrerit gravida rutrum quisque. Varius morbi enim nunc faucibus a pellentesque. Habitasse platea dictumst quisque sagittis purus. Nunc mi ipsum faucibus vitae aliquet nec ullamcorper. Morbi tempus iaculis urna id volutpat lacus laoreet non curabitur. Viverra tellus in hac habitasse platea dictumst vestibulum rhoncus. Ornare massa eget egestas purus viverra. Lobortis feugiat vivamus at augue eget arcu dictum varius duis. Bibendum at varius vel pharetra vel turpis nunc eget lorem. Lectus nulla at volutpat diam ut venenatis tellus in metus. Sed cras ornare arcu dui vivamus arcu felis bibendum. Aenean et tortor at risus viverra. Sit amet justo donec enim diam vulputate ut pharetra sit. Sed arcu non odio euismod lacinia. Convallis aenean et tortor at risus viverra adipiscing. Sodales ut etiam sit amet nisl purus in mollis. Tincidunt tortor aliquam nulla facilisi. Feugiat nisl pretium fusce id velit ut tortor pretium viverra. Aliquam sem fringilla ut morbi tincidunt.

Cursus turpis massa tincidunt dui ut ornare lectus sit. Porttitor leo a diam sollicitudin tempor id eu nisl nunc. Nam libero justo laoreet sit. Mauris vitae ultricies leo integer malesuada nunc. Eu facilisis sed odio morbi quis commodo odio aenean sed. Egestas egestas fringilla phasellus faucibus scelerisque. Consequat id porta nibh venenatis. Phasellus faucibus scelerisque eleifend donec pretium vulputate sapien. Vivamus at augue eget arcu dictum varius duis. Vel orci porta non pulvinar neque. Eu facilisis sed odio morbi quis commodo. Tellus rutrum tellus pellentesque eu tincidunt. Et netus et malesuada fames ac turpis egestas sed. Nulla facilisi cras fermentum odio eu feugiat pretium. Eget nullam non nisi est sit. Dui sapien eget mi proin sed libero enim. Nunc consequat interdum varius sit amet mattis. Sed turpis tincidunt id aliquet risus feugiat in ante.

Faucibus ornare suspendisse sed nisi lacus. Lacus viverra vitae congue eu. Magna ac placerat vestibulum lectus

mauris ultrices. Risus commodo viverra maecenas accumsan lacus vel facilisis
volutpat est. Cursus in hac habitasse platea dictumst quisque. Nibh venenatis cras
sed felis eget velit aliquet sagittis. Etiam sit amet nisl purus in mollis.
Ullamcorper sit amet risus nullam eget felis eget. Erat nam at lectus urna duis
convallis. Elit eget gravida cum sociis natoque penatibus. Lacus laoreet non
curabitur gravida arcu ac. Habitant morbi tristique senectus et netus et malesuada
fames ac. Augue interdum velit euismod in. Quis lectus nulla at volutpat diam ut
venenatis tellus in. Viverra suspendisse potenti nullam ac tortor vitae. Felis
imperdiet proin fermentum leo vel orci porta non. Ridiculus mus mauris vitae
ultricies leo. Proin libero nunc consequat interdum varius sit amet mattis
vulputate.\n

\nEu turpis egestas pretium aenean pharetra magna ac. Facilisis
magna etiam tempor orci. Velit laoreet id donec ultrices tincidunt arcu. Risus in
hendrerit gravida rutrum quisque non tellus orci ac. Ac orci phasellus egestas
tellus rutrum tellus. Eu scelerisque felis imperdiet proin fermentum. Sagittis eu
volutpat odio facilisis mauris sit amet massa vitae. Id semper risus in hendrerit
gravida rutrum quisque non. Rhoncus aenean vel elit scelerisque mauris pellentesque
pulvinar. Amet aliquam id diam maecenas ultricies. Lacus sed turpis tincidunt id.
Enim ut sem viverra aliquet. Dictum fusce ut placerat orci nulla pellentesque
dignissim. Aliquam eleifend mi in nulla posuere sollicitudin aliquam. Lorem dolor
sed viverra ipsum nunc aliquet bibendum enim. Fames ac turpis egestas integer eget
aliquet nibh. At imperdiet dui accumsan sit amet nulla facilisi morbi tempus. Est
placerat in egestas erat imperdiet sed.\n

\nTortor posuere ac ut consequat
semper viverra. Ullamcorper eget nulla facilisi etiam dignissim diam. Sed vulputate
odio ut enim. Sit amet tellus cras adipiscing enim eu turpis egestas pretium.
Mattis pellentesque id nibh tortor id aliquet lectus. Tincidunt eget nullam non
nisi. Vitae semper quis lectus nulla at. Aenean pharetra magna ac placerat. At
lectus urna duis convallis convallis tellus id. Tempus egestas sed sed risus
pretium quam vulputate dignissim suspendisse. Vitae sapien pellentesque habitant
morbi tristique senectus et. Augue neque gravida in fermentum. Urna molestie at
elementum eu facilisis. Odio euismod lacinia at quis risus sed vulputate odio ut.
Aliquet risus feugiat in ante metus dictum at. Adipiscing bibendum est ultricies
integer quis auctor elit sed vulputate. Leo vel orci porta non. Nulla aliquet
porttitor lacus luctus accumsan tortor posuere. Integer quis auctor elit
sed.\n

\nAt tellus at urna condimentum mattis pellentesque id nibh tortor.
Velit dignissim sodales ut eu sem integer. Est pellentesque elit ullamcorper
dignissim cras tincidunt lobortis feugiat. Massa massa ultricies mi quis. Et
malesuada fames ac turpis egestas sed. Tincidunt lobortis feugiat vivamus at augue
eget arcu dictum. Vel risus commodo viverra maecenas accumsan lacus. Commodum odio
aenean sed adipiscing diam donec. Non blandit massa enim nec dui. Nunc vel risus
commodo viverra maecenas accumsan lacus. Massa id neque aliquam vestibulum. At
imperdiet dui accumsan sit. Mauris augue neque gravida in fermentum et sollicitudin
ac.\n

\nConvallis convallis tellus id interdum velit. Mollis nunc sed id
semper risus in. Est ullamcorper eget nulla facilisi etiam dignissim diam quis
enim. Id donec ultrices tincidunt arcu non sodales neque. Nisl suscipit adipiscing
bibendum est ultricies integer quis auctor. Sit amet venenatis urna cursus eget
nunc scelerisque. Elit at imperdiet dui accumsan sit amet nulla. Ultricies mi eget
mauris pharetra. Fermentum dui faucibus in ornare quam viverra. Eu ultrices vitae
auctor eu augue ut lectus arcu bibendum. Gravida quis blandit turpis cursus in hac
habitasse platea dictumst. Placerat in egestas erat imperdiet. Nunc pulvinar sapien
et ligula ullamcorper malesuada proin libero nunc. Purus in massa tempor nec

```
feugiat nisl. Imperdiet dui accumsan sit amet nulla facilisi morbi. Arcu bibendum  
at varius vel pharetra vel turpis. Volutpat sed cras ornare arcu.", "timestamp":  
1679654688955}
```

/static/blog/1679655156413.json

```
{"title": "Example post 3", "image": "/static/site/logo.png", "preview": "Look, the  
logo! I ran out of ideas when it comes to what to put in these example posts, but I  
need three to properly create and test the blog page. So, examiner, for your  
convenience, here is a link to the admin panel.", "html": "<a href='/admin'>Admin  
panel</a>", "timestamp": 1679655156413}
```

/static/site/go-arrow.png



Normally transparent background

/static/site/logo.png



/static/site/sun.png



/static/favicon.ico



/static/admin.css

```
body {  
  background-color: #ececec;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
/* Navbar */
.navbar {
  background-color: #7EF297;
  width: 100%;
  height: 100px;
  position: fixed;
  top: 0%;
  left: 0%;
  border-bottom-color: #253031;
  border-bottom-width: 3px;
  border-bottom-style: solid;
}

.navbar > a {
  color: black;
  top: 68px;
  height: 30px;
  position: fixed;
  font-family: Arial, Helvetica, sans-serif;
  width: 146px;
  text-align: center;
  text-decoration: none;
  border-width: 2px;
  border-color: #253031;
  border-style: solid;
  border-bottom: 0cm;
}

.contact {
  right: 10%;
  background-color: #00bf63;
}

.blog {
  right: calc(10% + 153px);
  background-color: #00bf63;
}

.weather {
  right: calc(10% + 306px);
  background-color: #00bf63;
}

.home {
  right: calc(10% + 459px);
  background-color: #00bf63;
}

/* Main body */
```

```
/* password lock */
.password {
  position: absolute;
  top: 50%;
  border-style: solid;
  left: 50%;
  transform: translateX(-50%);
}

.main {
  display: none;
}

/* password update */
.password-change {
  position: absolute;
  top: 100px;
  left: 10px;
  width: 375px;
  height: 130px;
  border-style: solid;
  border-width: 3px;
}

.password-change > div > input {
  position: absolute;
  right: 0%;
}

/* email update */
.email-change {
  position: absolute;
  top: 233px;
  left: 10px;
  border-style: solid;
  height: 130px;
  width: 375px;
}

.email {
  width: 75%;
  position: absolute;
  right: 0px;
}

/* blog editor */
.blog-editor {
  position: absolute;
  left: 388px;
  right: 10px;
  top: 100px;
  bottom: 40px;
  border-style: solid;
```

```
}
```

```
.title {  
  position: absolute;  
  width: 80%;  
  right: 20px;  
  top: 80px;  
}
```

```
.title-label {  
  position: absolute;  
  width: 20%;  
  text-align: center;  
  top: 80px;  
}
```

```
.image-url {  
  position: absolute;  
  width: 80%;  
  right: 20px;  
  top: 105px;  
}
```

```
.image-url-label {  
  position: absolute;  
  width: 20%;  
  text-align: center;  
  top: 105px;  
}
```

```
.preview {  
  position: absolute;  
  width: 40%;  
  bottom: 50px;  
  left: 5%;  
  resize: none;  
  height: calc(100% - 225px);  
}
```

```
.preview-label {  
  position: absolute;  
  left: 5%;  
  top: 150px;  
}
```

```
.post-body {  
  position: absolute;  
  width: 40%;  
  bottom: 50px;  
  right: 5%;  
  resize: none;
```

```
height: calc(100% - 225px);  
}
```

```
.post-body-label {  
position: absolute;  
right: 5%;  
width: calc(40% + 7px);  
top: 150px;  
}
```

```
.create-post {  
position: absolute;  
bottom: 5px;  
left: 5px;  
border-color: #00bf63;  
width: 100px;  
}
```

```
.clear-blog {  
position: absolute;  
bottom: 5px;  
left: 110px;  
border-color: crimson;  
}
```

```
.blog-editor-title {  
position: absolute;  
left: 10px;  
top: 0px;  
}
```

```
/* Update posts */  
.existing-posts {  
position: absolute;  
top: 366px;  
left: 10px;  
border-style: solid;  
width: 375px;  
bottom: 40px;  
overflow-y: scroll;  
}
```

```
.existing-posts > div {  
position: absolute;  
width: 100%;  
border-top-style: solid;  
border-bottom-style: solid;  
border-color: #aaaaaa;  
height: 175px;  
}
```



```
.existing-posts > div > h3 {  
  position: absolute;  
  text-align: center;  
  left: 50%;  
  transform: translateX(-50%);  
}
```

```
.existing-posts > div > img {  
  position: absolute;  
  height: 125px;  
  width: 60%;  
  top: 50px;  
}
```

```
.existing-posts > div > button {  
  position: absolute;  
  border-color: orange;  
  bottom: 60px;  
  transform: translateY(50%);  
  right: 20px;  
}
```

```
/* message updates */  
.response {  
  position: absolute;  
  bottom: 5px;  
  width: 99%;  
  text-align: center;  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 15px;  
  height: 20px;  
}
```

/static/blog.css

```
body {  
  background-color: #ececec;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
/* Navbar */  
.navbar {  
  background-color: #7EF297;  
  width: 100%;  
  height: 100px;  
  position: fixed;  
  top: 0%;  
  left: 0%;  
  z-index: 9999;
```

```
border-bottom-color: #253031;
border-bottom-width: 3px;
border-bottom-style: solid;
}
```

```
.navbar > a {
color: black;
top: 68px;
height: 30px;
position: fixed;
font-family: Arial, Helvetica, sans-serif;
width: 146px;
text-align: center;
text-decoration: none;
border-width: 2px;
border-color: #253031;
border-style: solid;
border-bottom: 0cm;
}
```

```
.contact {
right: 10%;
background-color: #00bf63;
}
```

```
.blog {
right: calc(10% + 153px);
background-color: #ececec;
}
```

```
.weather {
right: calc(10% + 306px);
background-color: #00bf63;
}
```

```
.home {
right: calc(10% + 459px);
background-color: #00bf63;
}
```

```
/* Main body */
```

```
.loading {
position: absolute;
left: 50%;
top: 200px;
transform: translateX(-50%);
}
```

```
/* Posts */
```

```
.posts > div {  
  position: absolute;  
  width: 80%;  
  left: 50%;  
  transform: translateX(-50%);  
  height: 325px;  
  background-color: #7EF297;  
  border-radius: 50px;  
  border-color: #253031;  
  border-width: 5px;  
  border-style: solid;  
}
```

```
.posts > div > h3 {  
  position: absolute;  
  width: calc(100% - 325px);  
  text-align: center;  
  right: 0%;  
  font-size: 30px;  
  top: 0px;  
}
```

```
.posts > div > img {  
  position: absolute;  
  width: 350px;  
  top: 50%;  
  transform: translateY(-50%);  
}
```

```
.posts > div > p {  
  font-size: 20px;  
  position: absolute;  
  top: 70px;  
  width: calc(100% - 375px);  
  right: 25px;  
  text-align: center;  
}
```

```
.posts > div > button {  
  position: absolute;  
  border-radius: 1000px;  
  width: 10%;  
  font-size: 25px;  
  color: white;  
  background-color: #00bf63;  
  right: 50px;  
  bottom: 25px;  
}
```

static/contact.css

```
body {
  background-color: #ececec;
}

/* Navbar */
.navbar {
  background-color: #7EF297;
  width: 100%;
  height: 100px;
  position: fixed;
  top: 0%;
  left: 0%;
  border-bottom-color: #253031;
  border-bottom-width: 3px;
  border-bottom-style: solid;
}

.navbar > a {
  color: black;
  top: 68px;
  height: 30px;
  position: fixed;
  font-family: Arial, Helvetica, sans-serif;
  width: 146px;
  text-align: center;
  text-decoration: none;
  border-width: 2px;
  border-color: #253031;
  border-style: solid;
  border-bottom: 0cm;
}

.contact {
  right: 10%;
  background-color: #ececec;
}

.blog {
  right: calc(10% + 153px);
  background-color: #00bf63;
}

.weather {
  right: calc(10% + 306px);
  background-color: #00bf63;
}

.home {
  right: calc(10% + 459px);
```

```
background-color: #00bf63;
}
```

```
/* Main body */
```

```
.title {
  position: absolute;
  top: 125px;
  font-size: 50px;
  font-family: Arial, Helvetica, sans-serif;
  width: 100%;
  text-align: center;
}
```

```
p {
  position: absolute;
  top: 185px;
  font-size: 25px;
  font-family: Arial, Helvetica, sans-serif;
  text-align: center;
  width: 30%;
  left: 35%;
}
```

```
.form {
  position: absolute;
  top: 350px;
  width: 80%;
  left: 10%;
  height: calc(95% - 350px);
  background-color: #7EF297;
  border-radius: 50px;
  border-color: #253031;
  border-width: 4px;
  border-style: solid;
}
```

```
.reply_email {
  position: absolute;
  top: 50px;
  left: 50px;
  height: 20px;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 18px;
  width: calc(60% - 50px);
  background-color: #ececec;
}
```

```
.name {
  position: absolute;
  top: 50px;
```

```
right: 50px;
height: 20px;
font-family: Arial, Helvetica, sans-serif;
font-size: 18px;
width: calc(40% - 80px);
background-color: #ececec;
}
```

```
.subject {
position: absolute;
top: 90px;
left: 50%;
transform: translateX(-50%);
height: 20px;
font-family: Arial, Helvetica, sans-serif;
font-size: 18px;
width: 50%;
background-color: #ececec;
}
```

```
.email_body {
position: absolute;
top: 130px;
left: 50%;
transform: translateX(-50%);
height: 50%;
font-family: Arial, Helvetica, sans-serif;
font-size: 18px;
width: 90%;
resize: none;
background-color: #ececec;
}
```

```
.Send {
position: absolute;
bottom: 50px;
right: 100px;
width: 130px;
height: 50px;
border-radius: 100px;
background-color: #00bf63;
color: white;
font-family: Arial, Helvetica, sans-serif;
font-size: 30px;
border-color: #253031;
}
```

```
.Message {
position: absolute;
left: 50px;
bottom: 50px;
```

```
width: 300px;
height: 20px;
transform: translateY(250px);
}
```

/static/home.css

```
body {
background-color: #ececec;
font-family: Arial, Helvetica, sans-serif;
}
```

```
/* Navbar */
.navbar {
background-color: #7EF297;
width: 100%;
height: 100px;
position: fixed;
top: 0%;
left: 0%;
border-bottom-color: #253031;
border-bottom-width: 3px;
border-bottom-style: solid;
}
```

```
.navbar > a {
color: black;
top: 68px;
height: 30px;
position: fixed;
font-family: Arial, Helvetica, sans-serif;
width: 146px;
text-align: center;
text-decoration: none;
border-width: 2px;
border-color: #253031;
border-style: solid;
border-bottom: 0cm;
}
```

```
.contact {
right: 10%;
background-color: #00bf63;
}
```

```
.blog {
right: calc(10% + 153px);
background-color: #00bf63;
}
```

```
.weather {  
  right: calc(10% + 306px);  
  background-color: #00bf63;  
}
```

```
.home {  
  right: calc(10% + 459px);  
  background-color: #ececec;  
}
```

```
/* Main body */  
.sungraphic {  
  position: absolute;  
  bottom: 0%;  
  width: max(300px,30%);  
  left: calc(25% - max(150px,15%));  
}
```

```
.about {  
  position: absolute;  
  top: 150px;  
  width:40%;  
  left: 5%;  
  text-align: center;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
.description {  
  top: 250px;  
  position: absolute;  
  width: 440px;  
  left: calc(25% - 220px);  
  text-align: center;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
.gobutton {  
  position: absolute;  
  top: 400px;  
  color: white;  
  background-color: #00bf63;  
  width: 150px;  
  border-radius: 1000px;  
  left: calc(25% - 75px);  
  height: 50px;  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 25px;  
}
```

```
/* Featured Blog */  
.featured-article-outer {
```



```
position: absolute;
width: 45%;
right: 25%;
transform: translateX(50%);
border-radius: 50px;
border-width: 3px;
border-color: #253031;
border-style: solid;
background-color: #00bf63;
height: calc(100% - 200px);
top: 150px;
}
```

```
.featured-article-outer > h2 {
position: absolute;
width: 100%;
top: 10px;
font-size: 30px;
text-align: center;
}
```

```
.featured-article-inner {
position: absolute;
width: calc(100% - 50px);
bottom: 25px;
border-radius: 25px;
border-width: 2px;
border-color: #253031;
border-style: solid;
background-color: #7EF297;
left: 25px;
top: 100px;
}
```

```
.title {
position: absolute;
width: 100%;
top: 0px;
font-size: 25px;
text-align: center;
}
```

```
.image {
position: absolute;
height: 25%;
top: 75px;
left: 50%;
transform: translateX(-50%);
}
```

```
.preview {
```

```
position: absolute;
top: calc(100px + 25%);
width: 80%;
text-align: center;
left: 50%;
transform: translateX(-50%);
font-size: 18px;
}
```

```
.blog-button {
position: absolute;
color: white;
background-color: #00bf63;
width: 150px;
border-radius: 1000px;
height: 50px;
font-family: Arial, Helvetica, sans-serif;
font-size: 25px;
bottom: 25px;
left: 50%;
transform: translateX(-50%);
}
```

/static/view.css

```
body {
background-color: #ececec;
font-family: Arial, Helvetica, sans-serif;
}
```

```
/* Navbar */
.navbar {
background-color: #7EF297;
width: 100%;
height: 100px;
position: fixed;
top: 0%;
left: 0%;
z-index: 9999;
border-bottom-color: #253031;
border-bottom-width: 3px;
border-bottom-style: solid;
}
```

```
.navbar > a {
color: black;
top: 68px;
height: 30px;
position: fixed;
font-family: Arial, Helvetica, sans-serif;
```

```
width: 146px;
text-align: center;
text-decoration: none;
border-width: 2px;
border-color: #253031;
border-style: solid;
border-bottom: 0cm;
}

.contact {
right: 10%;
background-color: #00bf63;
}

.blog {
right: calc(10% + 153px);
background-color: #ececec;
}

.weather {
right: calc(10% + 306px);
background-color: #00bf63;
}

.home {
right: calc(10% + 459px);
background-color: #00bf63;
}

/* Main body */

.Content {
position: absolute;
top: 125px;
}
```

/static/weather.css

```
body {
background-color: #ececec;
}

/* Navbar */
.navbar {
background-color: #7EF297;
width: 100%;
height: 100px;
position: fixed;
top: 0%;
left: 0%;
```

```
border-bottom-color: #253031;
border-bottom-width: 3px;
border-bottom-style: solid;
}
```

```
.navbar > a {
color: black;
top: 68px;
height: 30px;
position: fixed;
font-family: Arial, Helvetica, sans-serif;
width: 146px;
text-align: center;
text-decoration: none;
border-width: 2px;
border-color: #253031;
border-style: solid;
border-bottom: 0cm;
}
```

```
.contact {
right: 10%;
background-color: #00bf63;
}
```

```
.blog {
right: calc(10% + 153px);
background-color: #00bf63;
}
```

```
.weather {
right: calc(10% + 306px);
background-color: #ececec;
}
```

```
.home {
right: calc(10% + 459px);
background-color: #00bf63;
}
```

```
/* Main body */
```

```
/* Weather */
.Weather-title > h1 {
position: absolute;
top: 100px;
font-family: Arial, Helvetica, sans-serif;
font-size: 33px;
width: 300px;
text-align: center;
}
```

```
.Weather-title > h3 {  
  position: absolute;  
  top: 113px;  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 20px;  
  left: 300px;  
  width: 150px;  
  text-align: center;  
}
```

```
.Location {  
  position: absolute;  
  top: 113px;  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 20px;  
  right: 20px;  
}
```

```
.Weather-body {  
  position: absolute;  
  width: 100%;  
}
```

```
.Weather-body > div {  
  position: absolute;  
  top: 170px;  
  background-color: #7EF297;  
  border-color: #253031;  
  border-width: 2px;  
  border-top-width: 0px;  
  border-bottom-width: 0px;  
  border-style: solid;  
  width: max(8%,102px);  
}
```

```
.Weather-body > div > h6 {  
  font-family: Arial, Helvetica, sans-serif;  
  text-align: center;  
  font-size: 20px;  
  padding-top: 2px;  
  margin-top: 2px;  
  padding-bottom: 2px;  
  margin-bottom: 2px;  
}
```

```
.Weather-body > div > img {  
  display: block;  
  margin-left: auto;  
  margin-right: auto;  
  width: 100px;  
}
```

```

}

/* Individual forecast sections */
.Hour4 {
  right: 50%;
}
.Hour5 {
  left: 50%;
}
.Hour3 {
  right: calc(50% + max(8%,102px) + 4px)
}
.Hour6 {
  left: calc(50% + max(8%,102px) + 4px)
}
.Hour2 {
  right: calc(50% + (2 * max(8%,102px)) + 8px)
}
.Hour7 {
  left: calc(50% + (2 * max(8%,102px)) + 8px)
}
.Hour1 {
  right: calc(50% + (3 * max(8%,102px)) + 12px)
}
.Hour8 {
  left: calc(50% + (3 * max(8%,102px)) + 12px)
}
.Hour0 {
  right: calc(50% + (4 * max(8%,102px)) + 16px)
}
.Hour9 {
  left: calc(50% + (4 * max(8%,102px)) + 16px)
}

```

```

/* Pollen */

```

```

.Pollen-title {
  position: absolute;
  top: 40%;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 33px;
  width: 300px;
  text-align: center;
}

```

```

.Pollen > h3 {
  position: absolute;
  font-family: Arial, Helvetica, sans-serif;
  width: 50px;
}

```

```
height: 50px;
border-color: #00bf63;
border-style: solid;
border-width: 10px;
text-align: center;
border-radius: 100%;
padding: 50px;
}
```

```
.Pollen-Grass {
top: calc(50% + 30px);
left: 75px;
}
```

```
.Pollen-Trees {
top: calc(50% + 30px);
left: 275px;
}
```

```
.Pollen-Weeds {
top: calc(50% + 210px);
left: 175px;
}
```

```
/* Air quality */
```

```
.Air-Q-title > h1 {
position: absolute;
top: 40%;
font-family: Arial, Helvetica, sans-serif;
font-size: 33px;
width: 300px;
text-align: center;
right: calc(50% + 100px);
}
```

```
.Air-Q-title > h2 {
position: absolute;
top: calc(40% + 16px);
font-family: Arial, Helvetica, sans-serif;
font-size: 20px;
right: calc(50% - 160px);
width: 350px;
text-align: center;
}
```

```
.Air-Q > h3 {
position: absolute;
font-family: Arial, Helvetica, sans-serif;
text-align: center;
```

```
right: calc(50% + 90px);
transform: translateX(50%);
}
```

```
.Year0 {
  top: 50%;
  font-size: 30px;
}
```

```
.Year1 {
  top: calc(50% + 70px);
  font-size: 23px;
}
```

```
.Year2 {
  top: calc(50% + 123px);
  font-size: 23px;
}
```

```
.Year3 {
  top: calc(50% + 176px);
  font-size: 23px;
}
```

```
.Year4 {
  top: calc(50% + 229px);
  font-size: 23px;
}
```

```
.Copyright {
  top: calc(50% + 282px);
  font-size: 10px;
  width: 300px;
}
```

```
/* Advice */
```

```
.Advice > h1 {
  position: absolute;
  top: 40%;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 33px;
  width: 300px;
  text-align: center;
  left: calc(50% + 100px);
}
```

```
.Advice > div > h2 {
  position: absolute;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 25px;
  text-align: left;
  width: 400px;
  left: calc(50% + 200px);
}
```



```

}

.help0 > h2 {
  top: 50%;
}
.help1 > h2 {
  top: calc(50% + 130px);
}
.help2 > h2 {
  top: calc(50% + 260px);
}

.Advice > div > p {
  position: absolute;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 20px;
  text-align: left;
  width: 33%;
  left: calc(50% + 200px);
}

.help0 > p {
  top: calc(50% + 50px);
}
.help1 > p {
  top: calc(50% + 180px);
}
.help2 > p {
  top: calc(50% + 310px);
}

```

/static/admin.js

```

// get current contact email
function get_contact_email() {
  fetch('/admin/email').then(response => response.json()).then(data => {
    document.querySelector('.current-email').innerHTML = 'Currently ' +
data['email']
  })
}

// assemble a post object on the html
const post_space = document.querySelector('.existing-posts')
function create_post(json, index) {
  let post_div = document.createElement('div')
  let classname = 'post' + index
  post_div.setAttribute('class', classname)
  let distance = (index * 178) + 70
  post_div.style.top = distance + 'px'
  post_space.appendChild(post_div)
}

```

```

let title = document.createElement('h3')
classname = 'title' + index
title.setAttribute('class', classname)
title.innerHTML = json['title']
post_div.appendChild(title)

let image = document.createElement('img')
classname = 'img' + index
image.setAttribute('class', classname)
image.setAttribute('src', json['image'])
post_div.appendChild(image)

let button = document.createElement('button')
classname = 'button' + index
button.setAttribute('class', classname)
let command = 'UpdateMode(' + json['timestamp'] + ')'
button.setAttribute('onclick', command)
button.innerHTML = 'Update'
post_div.appendChild(button)
}

```

```

// get all blog posts so they are ready to be chosen for update
function get_all_posts() {
  fetch('/blog/all').then(response => response.json()).then(data => {
    // iterate through each blog post assembling the structure
    // first declare a bunch of needed variables
    let posts = data['all_posts']
    let post_json = {}
    for (i = 0; i < posts.length; i++) {
      post_json = posts[i]
      // create the post
      create_post(post_json, i)
    }
  })
}

```

```

// this function is directly from MDN Web Docs, and is free to use for sha-256
hashing strings
async function digestMessage(message) {
  const msgUint8 = new TextEncoder().encode(message); // encode as (utf-8)
  Uint8Array
  const hashBuffer = await crypto.subtle.digest("SHA-256", msgUint8); // hash the
  message
  const hashArray = Array.from(new Uint8Array(hashBuffer)); // convert buffer to
  byte array
  const hashHex = hashArray
    .map((b) => b.toString(16).padStart(2, "0"))

```

```

    .join(""); // convert bytes to hex string
    return hashHex;
}

```

```

// check password, if correct reveal everything else
function checkpass() {
    let password = document.querySelector('.pword_entry').value
    digestMessage(password).then(hash => {
        console.log(hash)

```

```

        let body = {
            hash: hash
        }

```

```

        // send to /passcheck
        fetch('/passcheck', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json;charset=utf-8'
            },
            body: JSON.stringify(body)
        })
        .then(response => response.json())
        // handle response
        .then(data => {
            if (data['message'] == 'Correct password') {
                console.log(data)
                // hide the password entry
                document.querySelector('.password').style.display = 'none'
                // show everything else
                document.querySelector('.main').style.display = 'inline'
            }
        })
    })
}

```

```

// update the password
function changepass() {
    document.querySelector('.response').innerHTML = 'Please wait...'
    let password1 = document.querySelector('.new_pword_entry1').value
    let password2 = document.querySelector('.new_pword_entry2').value
    if (password1 == password2) {
        digestMessage(password1).then(hash => {
            console.log(hash)

```

```

        let body = {
            hash: hash

```

```

    }

    // send to /admin/password
    fetch('/admin/password', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json;charset=utf-8'
      },
      body: JSON.stringify(body)
    })
    .then(response => response.json())

    // handle response
    .then(data => {
      document.querySelector('.response').innerHTML = data['message']
      document.querySelector('.new_pword_entry1').value = ''
      document.querySelector('.new_pword_entry2').value = ''
    })
  })
} else {
  document.querySelector('.response').innerHTML = 'Passwords do not match.'
}
}
}

```

```

// change destination email for contact
function changeemail() {
  document.querySelector('.response').innerHTML = 'Please wait...'
  let email = document.querySelector('.email').value
  // validation - '@' present, only one '@', at least one '.' after the '@'
  valid_email = (email.indexOf('@') > -1) && (email.split('@').length == 2) &&
(email.split('@').pop().indexOf('.') > -1)
  if (valid_email) {
    // send POST request
    let body = {
      email: email
    }

```

```

    // send to /admin/email
    fetch('/admin/email', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json;charset=utf-8'
      },
      body: JSON.stringify(body)
    })
    .then(response => response.json())

    // handle response
    .then(data => {
      document.querySelector('.response').innerHTML = data['message']

```

```

        document.querySelector('.email').value = ''
        get_contact_email()
    })
} else {
    document.querySelector('.response').innerHTML = 'Please enter a valid email'
}
}
}

```

```

// upload blog post to backend
function createpost(update) {
    document.querySelector('.response').innerHTML = 'Please wait...'
    // get values, cancel if empty
    let title = document.querySelector('.title').value
    if (title == '') {
        document.querySelector('.response').innerHTML = 'Please enter a title.'
        return
    }
    let image_url = document.querySelector('.image-url').value
    if (image_url == '') {
        document.querySelector('.response').innerHTML = 'Please enter an image URL.'
        return
    }
    let preview = document.querySelector('.preview').value
    if (preview == '') {
        document.querySelector('.response').innerHTML = 'Please enter some preview
text'
        return
    }
    let html = document.querySelector('.post-body').value
    if (html == '') {
        document.querySelector('.response').innerHTML = 'Please enter the blog
contents'
        return
    }
}

```

```

let timestamp = 0

// get the timestamp
if (update == false) {
    timestamp = Date.now()
} else {
    timestamp = update
}

```

```

// assemble into POST request body
body = {
    title: title,
    image: image_url,

```

```
    preview: preview,
    html: html,
    timestamp: timestamp
  }
}
```

```
// send request to /admin/create
fetch('/admin/create', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json;charset=utf-8'
  },
  body: JSON.stringify(body)
})
.then(response => response.json())
// handle response
.then(data => {
  document.querySelector('.response').innerHTML = data['message']
  get_all_posts()
})
}
```

```
// swap to update mode
function UpdateMode(timestamp) {
  console.log(timestamp) //debug
  // get the blog details
  fetch_url = '/getblog/' + timestamp
  fetch(fetch_url).then(response => response.json()).then(data => {
    document.querySelector('.title').value = data['title']
    document.querySelector('.image-url').value = data['image']
    document.querySelector('.preview').value = data['preview']
    document.querySelector('.post-body').value = data['html']
  })
}
```

```
// other changes
document.querySelector('.blog-editor-title').innerHTML = 'Update this post'
document.querySelector('.create-post').innerHTML = 'Update post'
let command = 'createpost(' + timestamp + ')'
document.querySelector('.create-post').setAttribute('onclick', command)
document.querySelector('.clear-blog').innerHTML = 'Back to creation'
}
```

```
// clear the contents (or go back to creation mode)
function clearblog() {
  document.querySelector('.title').value = ''
  document.querySelector('.image-url').value = ''
  document.querySelector('.preview').value = ''
  document.querySelector('.post-body').value = ''
}
```

```
document.querySelector('.blog-editor-title').innerHTML = 'Create a new blog post'
```

```
document.querySelector('.create-post').innerHTML = 'Create post'
document.querySelector('.create-post').setAttribute('onclick',
'createpost(false)')
document.querySelector('.clear-blog').innerHTML = 'Clear'
}
```

```
// run automatically
get_contact_email()
get_all_posts()
```

/static/blog.js

```
// redirect to the view page with the timestamp in the URL so it can be used on the
other side
function goToPage(timestamp) {
  let local_url = '/blog/view/' + timestamp
  window.location.href = local_url
}
```

```
// function to create a blog post preview object given its json and index
const posts_div = document.querySelector('.posts')
function create_post(json, index) {
  let post_div = document.createElement('div')
  let classname = 'post' + index
  post_div.setAttribute('class', classname)
  let distance = (index * 400) + 150
  post_div.style.top = distance + 'px'
  posts_div.appendChild(post_div)

  let title = document.createElement('h3')
  classname = 'title' + index
  title.setAttribute('class', classname)
  title.innerHTML = json['title']
  post_div.appendChild(title)

  let image = document.createElement('img')
  classname = 'img' + index
  image.setAttribute('class', classname)
  image.setAttribute('src', json['image'])
  post_div.appendChild(image)

  let preview = document.createElement('p')
  classname = 'prev' + index
  preview.setAttribute('class', classname)
  preview.innerHTML = json['preview']
  post_div.appendChild(preview)

  let button = document.createElement('button')
```

```

    classname = 'button' + index
    button.setAttribute('class', classname)
    let command = 'goToPage(' + json['timestamp'] + ')'
    button.setAttribute('onclick', command)
    button.innerHTML = 'View '
    post_div.appendChild(button)
  }

  // get all blog posts
  fetch('/blog/all').then(response => response.json()).then(data => {
    // debug - see test log for what was being tested
    console.log(data)

    // remove the loading text
    document.querySelector('.loading').style.display = 'none'

    // iterate through each blog post assembling the structure
    // first declare a bunch of needed variables
    let posts = data['all_posts']
    let post_json = {}
    for (i = 0; i < posts.length; i++) {
      post_json = posts[i]
      // create the post
      create_post(post_json, i)
    }
  })
})

```

/static/contact.js

```

// function for sending request to send email
function Submit() {
  document.querySelector('.Message').innerHTML = 'Please wait'

  // get values of input fields
  let reply_email = document.querySelector('.reply_email').value
  let name = document.querySelector('.name').value
  let subject = document.querySelector('.subject').value
  let body = document.querySelector('.email_body').value

  // piece together json
  let request_body = {
    reply_email: reply_email,
    name: name,
    subject: subject,
    body: body
  }

  // send to /contact/send
  fetch('/contact/send', {

```



```

    method: 'POST',
    headers: {
      'Content-Type': 'application/json;charset=utf-8'
    },
    body: JSON.stringify(request_body)
  })
  .then(response => response.json()).then(data => {

    document.querySelector('.Message').innerHTML = data['message']

  })
}

```

/static/home.js

```

function goNow() {
  window.location.href = '/weather'
}

// redirect to the view page with the timestamp in the URL so it can be used on the
// other side
function goToPage(timestamp) {
  let local_url = '/blog/view/' + timestamp
  window.location.href = local_url
}

// get the last blog post to show in a featured article
fetch('/lastblog').then(response => response.json()).then(data => {
  console.log(data)
  document.querySelector('.title').innerHTML = data['title']
  document.querySelector('.image').setAttribute('src', data['image'])
  document.querySelector('.preview').innerHTML = data['preview']
  let command = 'goToPage(' + data['timestamp'] + ')'
  document.querySelector('.blog-button').setAttribute('onclick',command)
})

```

/static/view.js

```

//get the timestamp from the URL
const timestamp = (window.location.pathname).split('/').pop()
console.log(timestamp)

// get the blog contents
fetch_url = '/getblog/' + timestamp
fetch(fetch_url).then(response => response.json()).then(data => {
  console.log(data)
  // show the HTML from the response
  document.querySelector('.Content').innerHTML = data['html']
})

```

/static/weather.js

```
// OPENWEATHER IMAGES FOR FORECAST ARE AT
// https://openweathermap.org/img/wn/<id>@2x.png

var latitude = 0
var longitude = 0

// get location - automatically asks permission
navigator.geolocation.getCurrentPosition(function(position) {

  latitude = position.coords.latitude;
  longitude = position.coords.longitude;

})

// build http request body
let body = {
  longitude: longitude,
  latitude: latitude,
  time: Date.now()
}

let data = {}
// send a request to main.py for the weather data
fetch('/weather/content', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json;charset=utf-8'
  },
  body: JSON.stringify(body)
})
.then(response => response.json())

// handle response
.then(data => {

  let full_location = (data.location[0] + ', ' + data.location[1])
  document.querySelector('.Location').innerHTML = full_location

  // times, temperatures, and images

  let times = Object.keys(data['forecast'])
  let time = ''

  // this is as Now is at the end of the json
  time = 'Now'
  img_code = data['forecast']['Now']['type']
```

```
temp = data['forecast']['Now']['temp']
object_name = '.Hour0'
document.querySelector(object_name).innerHTML = '<h6 class="Time0">Now</h6><h6
class="Temp0">'+temp+'°C</h6>'
```

```
// iterate the rest
for (i=0; i<9; i++) {
  time = times[i]
  img_code = data['forecast'][time]['type']
  temp = data['forecast'][time]['temp']
  object_name = '.Hour'+(i+1)
  document.querySelector(object_name).innerHTML = '<h6
class="Time0">'+time+'</h6><h6
class="Temp0">'+temp+'°C</h6>'
}
```

```
// pollen
const green = '#00c51b'
const yellow = '#ff9900'
const red = '#ff0000'
const purple = '#830b4a'
```

```
let grass = document.querySelector('.Pollen-Grass')
let trees = document.querySelector('.Pollen-Trees')
let weeds = document.querySelector('.Pollen-Weeds')
grass.innerHTML = 'Grass ' + data['pollen']['grass'] + 'ppm'
trees.innerHTML = 'Trees ' + data['pollen']['trees'] + 'ppm'
weeds.innerHTML = 'Weeds ' + data['pollen']['weeds'] + 'ppm'
```

```
// the numbers for the colour change come from the table on Kleenex Pollen Pal
// trees colour change
if (data['pollen']['trees'] < 96) {
  trees.style.borderColor = green
} else if (data['pollen']['trees'] < 208) {
  trees.style.borderColor = yellow
} else if (data['pollen']['trees'] < 704) {
  trees.style.borderColor = red
} else {
  trees.style.borderColor = purple
}
```

```
// grass colour change
if (data['pollen']['grass'] < 30) {
  grass.style.borderColor = green
} else if (data['pollen']['grass'] < 61) {
  grass.style.borderColor = yellow
}
```

```

} else if (data['pollen']['grass'] < 342) {
  grass.style.borderColor = red
} else {
  grass.style.borderColor = purple
}

// weeds colour change
if (data['pollen']['weeds'] < 21) {
  weeds.style.borderColor = green
} else if (data['pollen']['weeds'] < 78) {
  weeds.style.borderColor = yellow
} else if (data['pollen']['weeds'] < 267) {
  weeds.style.borderColor = red
} else {
  weeds.style.borderColor = purple
}

// pollution
let target = ''
for (i=0;i<5;i++) {
  target = '.Year' + i
  document.querySelector(target).innerHTML = data['pollution'][i]['year'] +
  '&nbsp;&nbsp;&nbsp;' + data['pollution'][i]['value'] + ' µg/m<sup>3</sup>'
}

// advice
for (i=0;i<3;i++) {
  target = '.help' + i
  document.querySelector(target).innerHTML = '<h2>'+data['advice'][i]['title']
+ '</h2><p>'+data['advice'][i]['description']+'</p>'
}

});

```

/templates/admin.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Health Advice Group | Admin</title>
  <link rel="stylesheet" href="{{url_for('static',filename='admin.css')}}">
  <link rel="shortcut icon" href="{{url_for('static', filename='favicon.ico')}}">
/>
</head>
<body>
  <div class="navbar">

```

```

    
    <a class="home" href="/">Home</a>
    <a class="blog" href="/blog">Blog</a>
    <a class="weather" href="/weather">Weather</a>
    <a class="contact" href="/contact">Contact</a>
  </div>
  <!--Password check-->
  <div class="password">
    <p>Password:</p>
    <input class="pword_entry" type="password">
    <button onclick="checkpass()">Submit</button>
  </div>

  <!--Main tools-->
  <div class="main">
    <!--Change password-->
    <div class="password-change">
      <h2>Update Password</h2>
      <div>
        <label for="new_pword_entry1">New password:</label>
        <input class="new_pword_entry1" type="password" id="new_pword_entry1"
minlength="1" placeholder="Password">
      </div>
      <div>
        <label for="new_pword_entry2">Confirm:</label>
        <input class="new_pword_entry2" type="password" id="new_pword_entry2"
placeholder="Password">
      </div>
      <button onclick="changepass()">Change</button>
    </div>

    <div class="email-change">
      <h2>Contact Email</h2>
      <div>
        <label class="current-email">Currently</label>
      </div>
      <div>
        <label for="email">New email:</label>
        <input class="email" type="email" id="email" inputmode="email"
placeholder="Email">
      </div>
      <button onclick="changeemail()">Change</button>
    </div>

    <!--Blog editing-->
    <div class="blog-list">
      <div>
      </div>
    </div>
    <div class="blog-editor">
      <h1 class="blog-editor-title">Create a new blog post</h1>
      <label for="title" class="title-label">Title:</label>

```

```

        <input type="text" class="title" id="title" placeholder="Top 10 Clever Blog
Post Titles">
        <label for='image-url' class="image-url-label">Image URL:</label>
        <input type="url" class="image-url" id="image-url"
placeholder="http://www.example.com/image.png">
        <label for="preview" class="preview-label">Preview text:</label>
        <textarea class="preview" id="preview" placeholder="What the user sees before
clicking on the post"></textarea>
        <label for="post-body" class="post-body-label">Body (HTML):</label>
        <textarea class="post-body" id="post-body" placeholder="Only include the
inside of the <body> tag."></textarea>
        <button onclick="createpost(false)" class="create-post">Create post</button>
        <button onclick="clearblog()" class="clear-blog">Clear</button>
    </div>
    <!--Area for available posts to update-->
    <div class="existing-posts">
        <h2>Update a post</h2>
    </div>

    <!--Message popup-->
    <p class="response">Welcome</p>
</div>

<script src="{{url_for('static',filename='admin.js')}}"></script>
</body>
</html>

```

/templates/blog.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Health Advice Group - Blog</title>
    <link rel="stylesheet" href="{{url_for('static',filename='blog.css')}}">
    <link rel="shortcut icon" href="{{url_for('static', filename='favicon.ico')}}"
/>
</head>
<body>
    <div class="navbar">
        
        <a class="home" href="/">Home</a>
        <a class="blog" href="/blog">Blog</a>
        <a class="weather" href="/weather">Weather</a>
        <a class="contact" href="/contact">Contact</a>
    </div>
    <h1 class="loading">Loading blog, please wait...</h1>

```

```

<div class="posts">
  <!--The blog posts are created in here by blog.js-->
</div>
<script src="{{url_for('static',filename='blog.js')}}"></script>
</body>
</html>

```

/templates/contact.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Health Advice Group | Contact</title>
  <link rel="stylesheet" href="{{url_for('static',filename='contact.css')}}">
  <link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}"
/>
</head>
<body>
  <div class="navbar">
    
    <a class="home" href="/">Home</a>
    <a class="blog" href="/blog">Blog</a>
    <a class="weather" href="/weather">Weather</a>
    <a class="contact" href="/contact">Contact</a>
  </div>
  <h1 class="title">Contact Us</h1>
  <p>Have a question about what we do? Need some more specific advice but can't
find it anywhere?<br>Send us an email!</p>

  <!-- form -->
  <div class="form">
    <input class="reply_email" value="Email" type="text">
    <input class="name" value="Name" type="text">
    <input class="subject" value="Subject" type="text">
    <textarea class="email_body">Body</textarea>
    <p class="Message"></p>
    <button class="Send" onclick="Submit()">Send </button>
  </div>
  <script src="{{url_for('static',filename='contact.js')}}"></script>
</body>

```

/templates/home.html

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Health Advice Group | Home</title>
  <link rel="stylesheet" href="{{url_for('static',filename='home.css')}}">
  <link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}"
/>
</head>
<body>
  <div class="navbar">
    
    <a class="home" href="/">Home</a>
    <a class="blog" href="/blog">Blog</a>
    <a class="weather" href="/weather">Weather</a>
    <a class="contact" href="/contact">Contact</a>
  </div>
  <h1 class="about">Find weather and pollen where you are.</h1>
  <p class="description">Our simple weather tracking tool allows you to see the
weather forecasts, pollen counts, and air quality data for where you are. All it
takes is a couple clicks!</p>
  
  <button type="button" onclick="goNow()" class="gobutton">Go Now </button>
  <!--Featured Article-->
  <div class="featured-article-outer">
    <h2>Featured Blog</h2>
    <div class="featured-article-inner">
      <h3 class="title">Loading</h3>
      <img class="image" src="" alt="Please wait">
      <p class="preview">Please wait</p>
      <button class="blog-button" onclick="goToPage()">View </button>
    </div>
  </div>
  <script src="{{url_for('static',filename='home.js')}}"></script>
</body>
</html>

```

/templates/view.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Health Advice Group | Blog</title>
  <link rel="stylesheet" href="{{url_for('static',filename='view.css')}}">
  <link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}"

```



```

/>
</head>
<body>
  <div class="navbar">
    
    <a class="home" href="/">Home</a>
    <a class="blog" href="/blog">Blog</a>
    <a class="weather" href="/weather">Weather</a>
    <a class="contact" href="/contact">Contact</a>
  </div>
  <div class="Content">
    <!--The custom stuff sits in here so the CSS can be changed slightly and it
looks nice-->
  </div>
  <script src="{{url_for('static',filename='view.js')}}"></script>
</body>

```

/templates/weather.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Health Advice Group | Weather</title>
  <link rel="stylesheet" href="{{url_for('static',filename='weather.css')}}">
  <link rel="shortcut icon" href="{{url_for('static', filename='favicon.ico')}}">
/>
</head>
<body>
  <div class="navbar">
    
    <a class="home" href="/">Home</a>
    <a class="blog" href="/blog">Blog</a>
    <a class="weather" href="/weather">Weather</a>
    <a class="contact" href="/contact">Contact</a>
  </div>
  <div class="Weather-title">
    <h1>Weather Forecast</h1>
    <h3>Next 10 hours</h3>
  </div>
  <h2 class="Location">City, COUNTRY</h2>

  <!--Weather forecast-->
  <div class="Weather-body">
    <div class="Hour0">
      <h6 class="Time0">Now</h6>
      
      <h6 class="Temp0">Now</h6>

```

```

</div>
<div class="Hour1">
  <h6 class="Time1">Now</h6>
  
  <h6 class="Temp1">Now</h6>
</div>
<div class="Hour2">
  <h6 class="Time2">Now</h6>
  
  <h6 class="Temp2">Now</h6>
</div>
<div class="Hour3">
  <h6 class="Time3">Now</h6>
  
  <h6 class="Temp3">Now</h6>
</div>
<div class="Hour4">
  <h6 class="Time4">Now</h6>
  
  <h6 class="Temp4">Now</h6>
</div>
<div class="Hour5">
  <h6 class="Time5">Now</h6>
  
  <h6 class="Temp5">Now</h6>
</div>
<div class="Hour6">
  <h6 class="Time6">Now</h6>
  
  <h6 class="Temp6">Now</h6>
</div>
<div class="Hour7">
  <h6 class="Time7">Now</h6>
  
  <h6 class="Temp7">Now</h6>
</div>
<div class="Hour8">
  <h6 class="Time8">Now</h6>
  
  <h6 class="Temp8">Now</h6>
</div>
<div class="Hour9">
  <h6 class="Time9">Now</h6>
  
  <h6 class="Temp9">Now</h6>
</div>
</div>
<script src="{url_for('static',filename='weather.js')}"></script>

```

```

<!-- Pollen counts -->

```

