

# Weather Crisis

CMPSC 463 (section 1)

[GitHub](#)

Avik Bhuiyan & Allen Chea

---

## Project Description

---

Weather web application that receives data and info from the publicly and freely available government web API for weather. Displays received information to the user and provides a warning label based on the severity of the weather.

## Significance

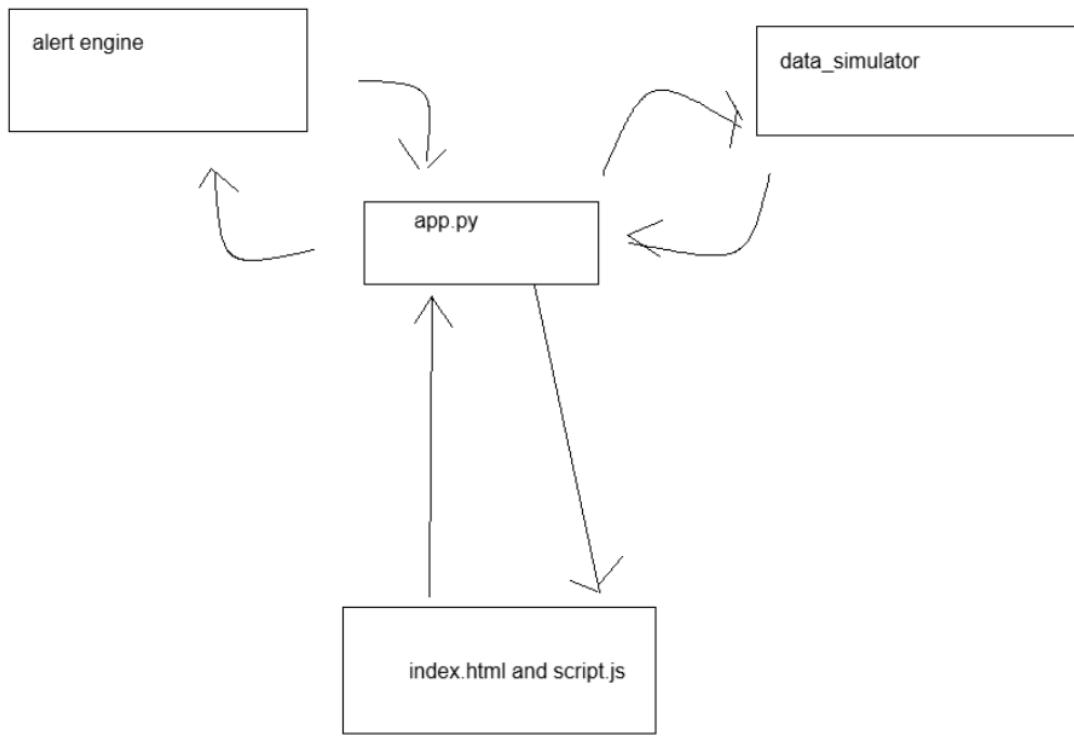
---

Could be helpful to show weather warnings as soon as the government catches wind of it, but unfortunately may fall short of local weather stations and forecasts. Unfortunately, we don't have the resources to be able to predict the weather or the datasets to train a model to. Nor access to a model to train. So this is more or less a best attempt to create a springboard for these ideas.

## Code Structure

---

The website runs a java script that receives json from app.py. The app.py program receives data from the data\_simulator python program that receives data and parses it from the public weather API. Currently, the weather API is configured for use in Pennsylvania. Then it judges the data using the alert\_engine.py program to determine the severity. Then the javascript on the website displays said information for the user.



## Description of Algorithms

---

Data\_simulator uses python libraries in order to receive and parse json. Alert Engine uses comparative algorithms in order to create appropriate warnings based off a judged severity of the situation and numbers given to it for the weather situation that may be encountered.

## Verification of Algorithms

---

See the demonstration video, the data is properly received and given to the user. We believe the data is accurate because it comes from the government.

## Functions

---

All functions are used to grab and send and communicate data. These were explained in the code structure.

## Results and Analysis

---

## Weather Crisis Early Warning System

The screenshot shows a web-based application for a weather crisis early warning system. At the top, a green header bar displays the text "ALERT LEVEL: GREEN" and "Stable conditions". Below this, a section titled "Current Weather Data" provides real-time information: Temperature (21 °C), Rainfall (1 mm), Wind Speed (0 km/h), and Humidity. A "Refresh Data" button is located at the bottom of this section.

Parameter	Value
Temperature	21 °C
Rainfall	1 mm
Wind Speed	0 km/h
Humidity	(not explicitly shown)

It works as best as it can

### Conclusion

---

Lots of issues in regards to getting this working and having to make a front end and a back end. Issues with actually running and locally hosting for the web interface. Applications of course learning we implemented basic python and used libraries. It was difficult trying to find exactly what to do for this project with such a vague description and no resources given.