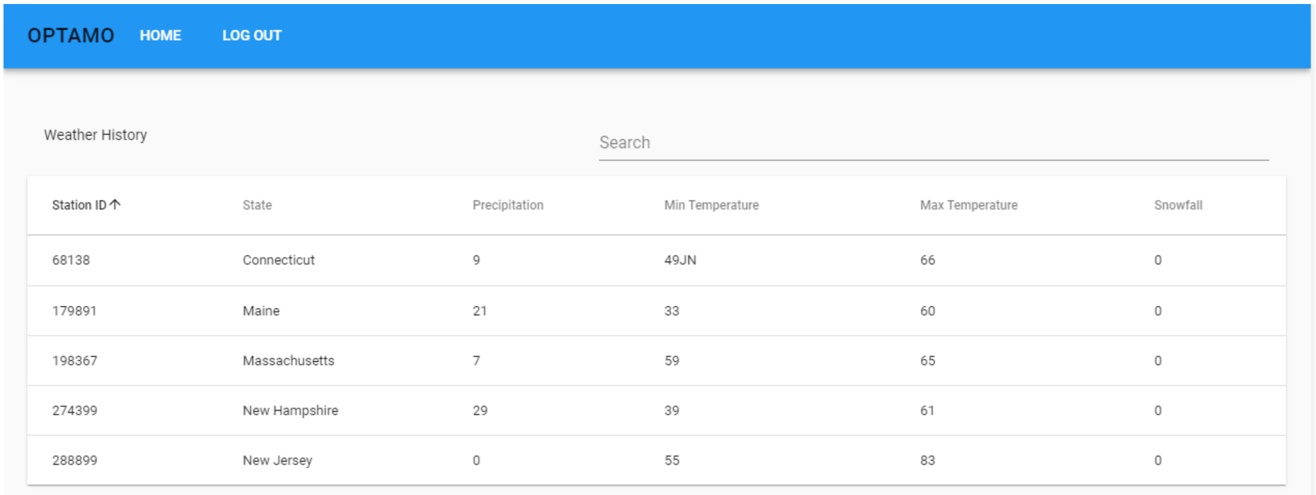
OPTAMO Weather Application



Weather History

Station ID ↑	State	Precipitation	Min Temperature	Max Temperature	Snowfall
68138	Connecticut	9	49JN	66	0
179891	Maine	21	33	60	0
198367	Massachusetts	7	59	65	0
274399	New Hampshire	29	39	61	0
288899	New Jersey	0	55	83	0

https://optamo-frontend.herokuapp.com/
username: optamo
password: optamo1234

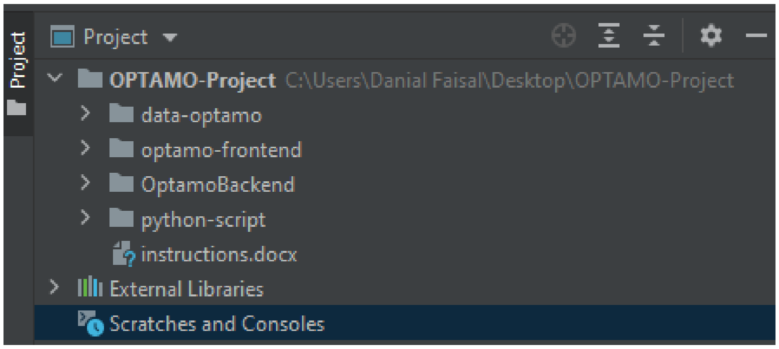
Contents

Technologies used

1. **Python Script** for parsing out data from directory and Posting API data on the backend platform.
2. **Python Django framework** for backend. The backend hosts the API data for the front-end application. Utilizes **PostgreSQL** for production environment and **SQLite** for development environment.
3. **Vue js** for frontend. The material design framework **Vuetify** is used for the data table.

Development Environment Set up

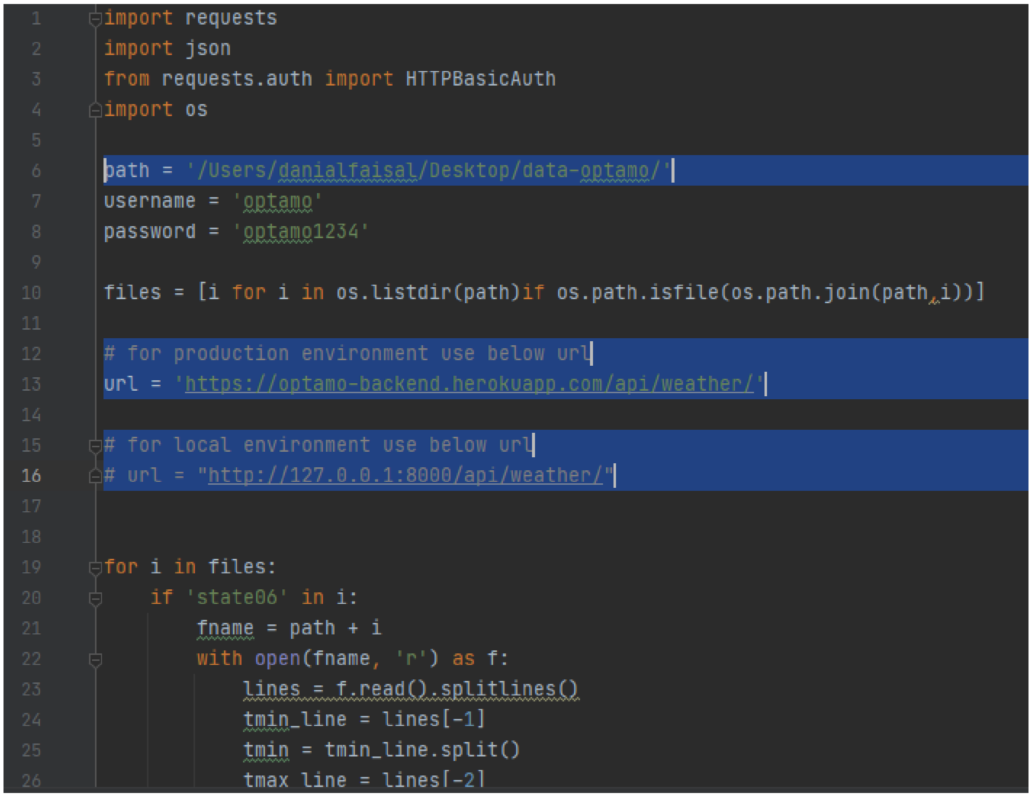
Below is how your project directory is structured.



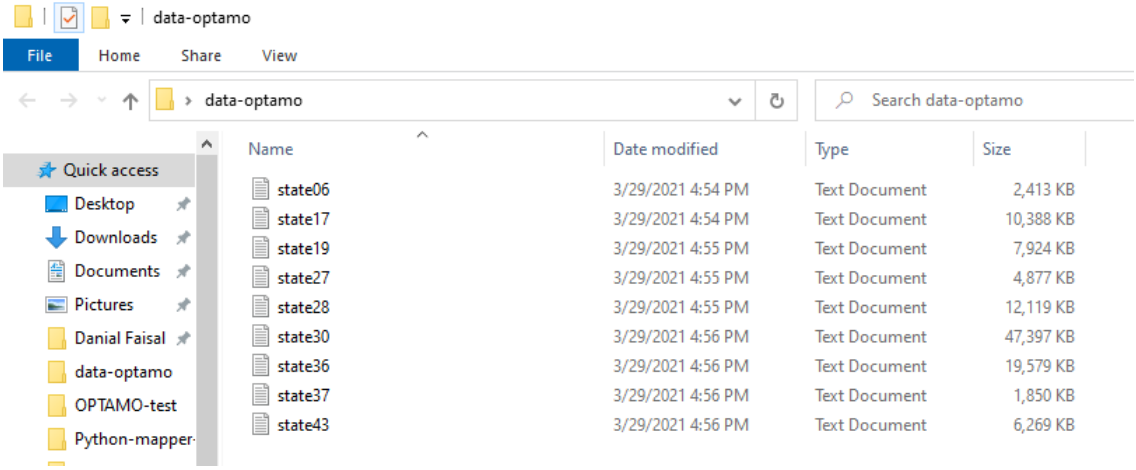
```
Project
OPTAMO-Project C:\Users\Danial Faisal\Desktop\OPTAMO-Project
    data-optamo
    optamo-frontend
    OptamoBackend
    python-script
    instructions.docx
External Libraries
Scratches and Consoles
```

Python Script

The Python script parses data from the folders in a specific directory and transmits API data, in JSON format to the specified back-end URL. The production environment for the backend has the URL [***https://optamo-backend.herokuapp.com/api/weather/***](https://optamo-backend.herokuapp.com/api/weather/) however if you are running the application on the local environment use [***http://127.0.0.1:8000/api/weather/***](http://127.0.0.1:8000/api/weather/)***.*** Make sure the Path entered is the directory where the data is stored. In my environment data is stored in a folder named ***data-optamo***



```python
import requests
import json
from requests.auth import HTTPBasicAuth
import os

path = '/Users/danialfaisal/Desktop/data-optamo/'
username = 'optamo'
password = 'optamo1234'

files = [i for i in os.listdir(path)if os.path.isfile(os.path.join(path,i))]

# for production environment use below url
url = 'https://optamo-backend.herokuapp.com/api/weather/'

# for local environment use below url
# url = "http://127.0.0.1:8000/api/weather/"


for i in files:
    if 'state06' in i:
        fname = path + i
        with open(fname, 'r') as f:
            lines = f.read().splitlines()
            tmin_line = lines[-1]
            tmin = tmin_line.split()
            tmax_line = lines[-2]
```

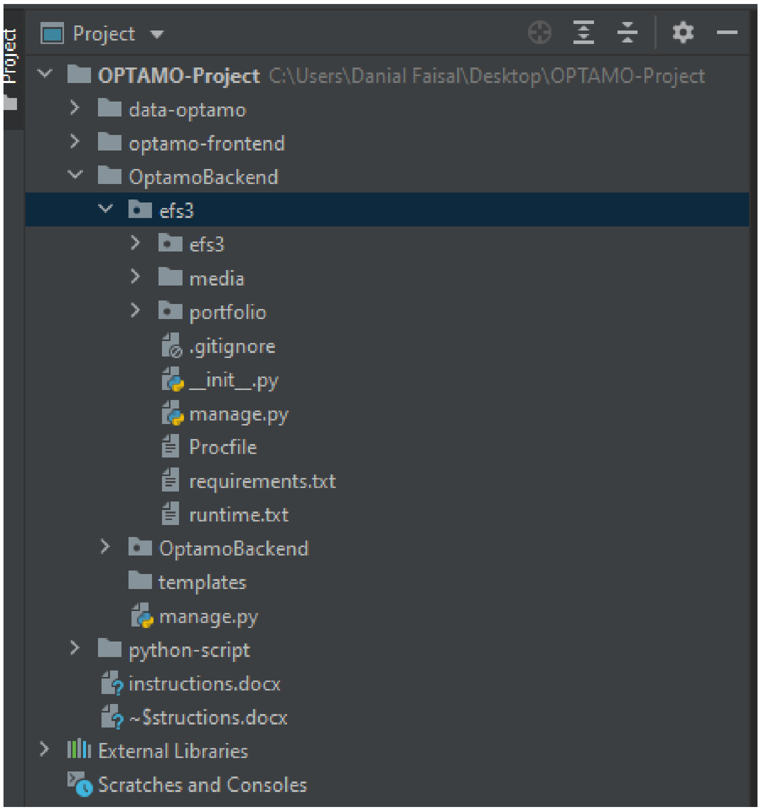


Name	Date modified	Type	Size
state06	3/29/2021 4:54 PM	Text Document	2,413 KB
state17	3/29/2021 4:54 PM	Text Document	10,388 KB
state19	3/29/2021 4:55 PM	Text Document	7,924 KB
state27	3/29/2021 4:55 PM	Text Document	4,877 KB
state28	3/29/2021 4:55 PM	Text Document	12,119 KB
state30	3/29/2021 4:56 PM	Text Document	47,397 KB
state36	3/29/2021 4:56 PM	Text Document	19,579 KB
state37	3/29/2021 4:56 PM	Text Document	1,850 KB
state43	3/29/2021 4:56 PM	Text Document	6,269 KB

Python Django Back-end

a) First you need to be in the ef3 directory. (note: There are 2 ef3 directories, make sure you are in the root level one) For me the directory was:

C:\Users\Danial Faisal\Desktop\OPTAMO-Project\OptamoBackend\efs3>

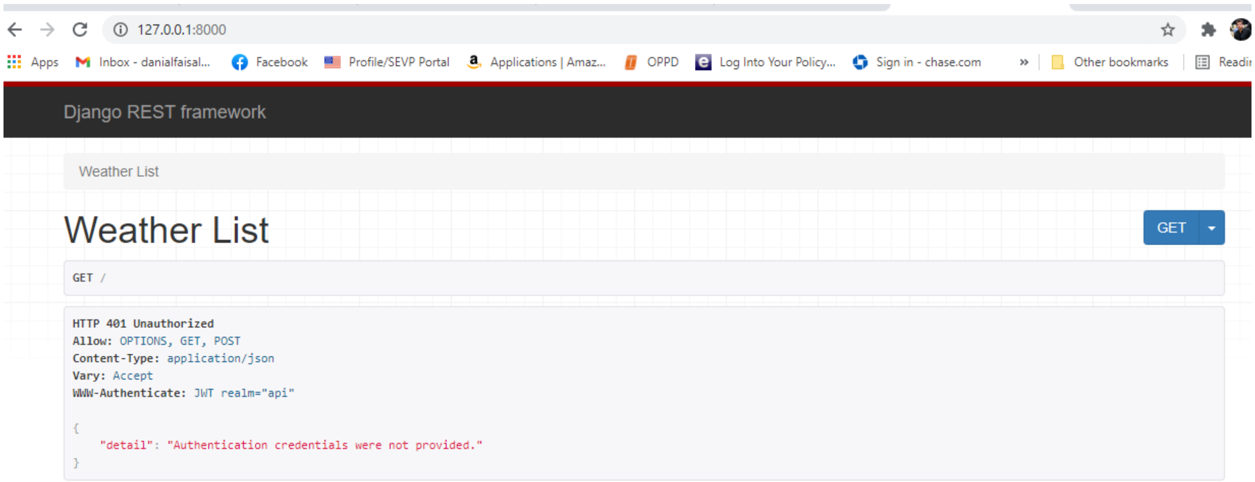


```
OPTAMO-Project C:\Users\Danial Faisal\Desktop\OPTAMO-Project
    data-optamo
    optamo-frontend
    OptamoBackend
        efs3
            efs3
            media
            portfolio
            .gitignore
            __init__.py
            manage.py
            Procfile
            requirements.txt
            runtime.txt
        OptamoBackend
        templates
        manage.py
    python-script
    instructions.docx
    ~$structions.docx
External Libraries
Scratches and Consoles
```

b) Once you are in the directory, run the following commands:

- ***pip install -r requirements.txt***
- ***python manage.py runserver***

c) Start development server ***http://127.0.0.1:8000/*** You will see an authentication error.

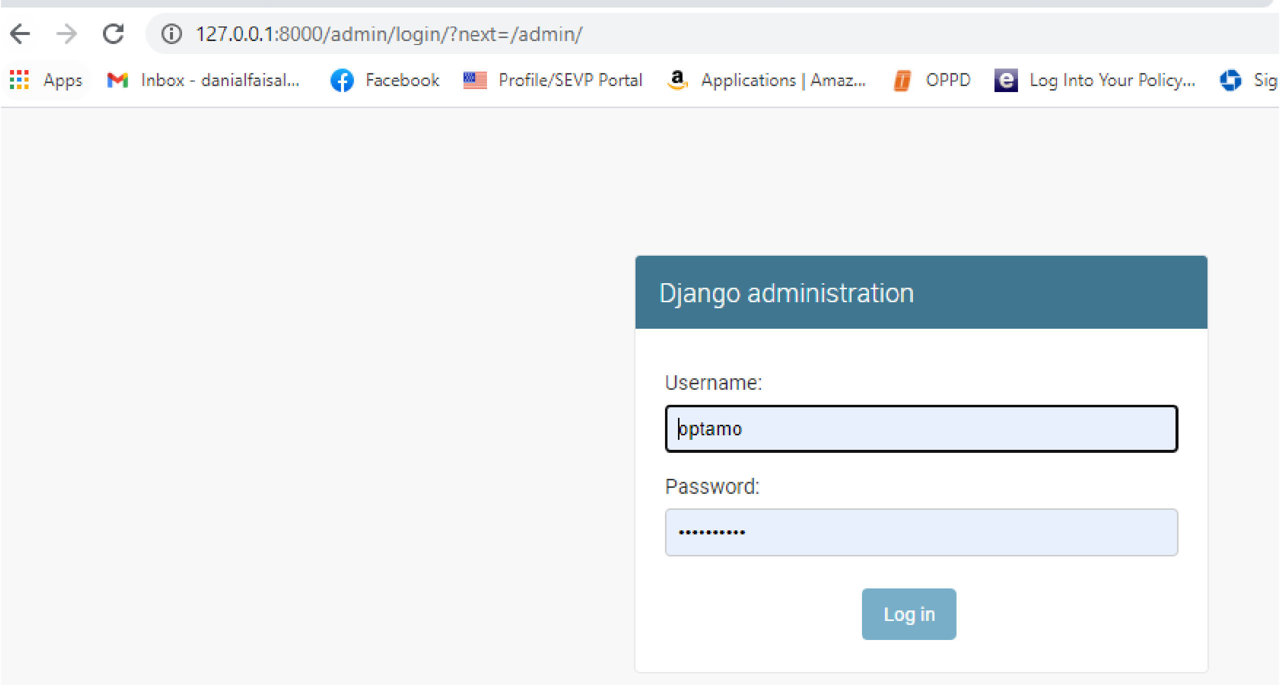


Weather List

```
GET /
```

```
HTTP 401 Unauthorized
Allow: OPTIONS, GET, POST
Content-Type: application/json
Vary: Accept
WWW-Authenticate: JWT realm="api"

{
    "detail": "Authentication credentials were not provided."
}
```

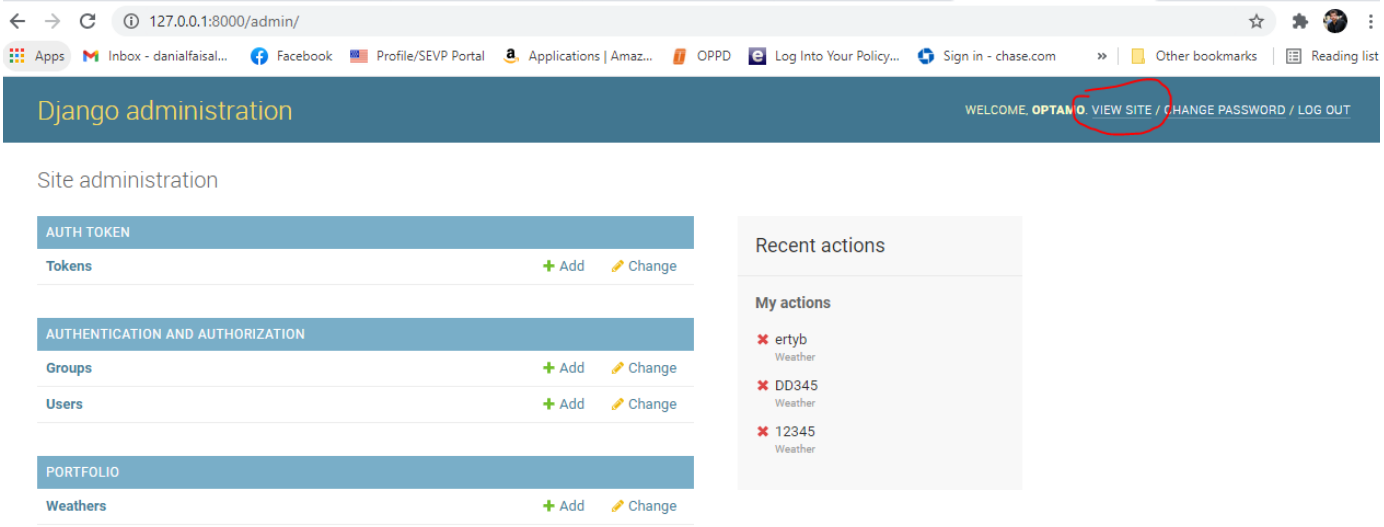
d) Go to the admin of the site ***http://127.0.0.1:8000/admin*** and enter your credentials, username: **optamo** password: **optamo1234**



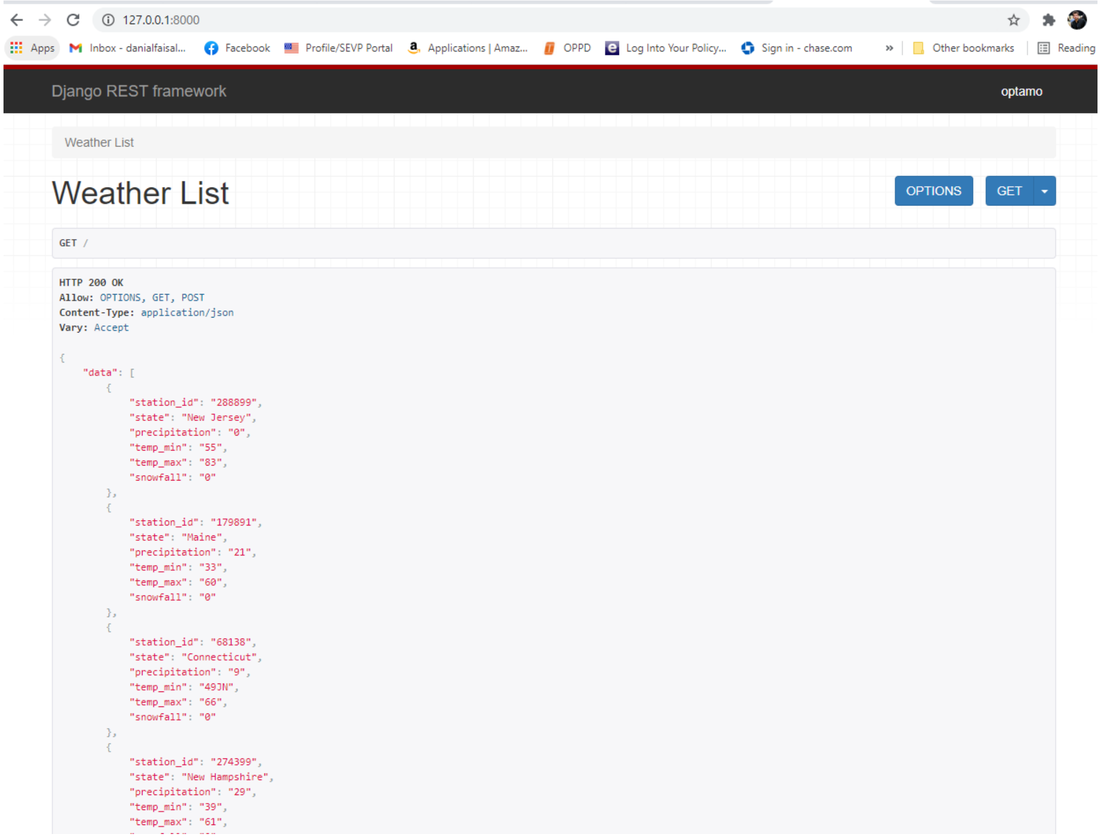
Username: optamo

Password: ••••••••••

e) On the admin site you can manage users and user groups and you can also perform CRUD functions on the data. Click “***VIEW SITE***” on the top right corner.



f) After you have logged in, you should be able to see all the data.



```
GET /
```

```
HTTP 200 OK
Allow: OPTIONS, GET, POST
Content-Type: application/json
Vary: Accept

{
    "data": [
        {
            "station_id": "288899",
            "state": "New Jersey",
            "precipitation": "0",
            "temp_min": "55",
            "temp_max": "83",
            "snowfall": "0"
        },
        {
            "station_id": "179891",
            "state": "Maine",
            "precipitation": "21",
            "temp_min": "33",
            "temp_max": "60",
            "snowfall": "0"
        },
        {
            "station_id": "68138",
            "state": "Connecticut",
            "precipitation": "9",
            "temp_min": "49JN",
            "temp_max": "66",
            "snowfall": "0"
        },
        {
            "station_id": "274399",
            "state": "New Hampshire",
            "precipitation": "29",
            "temp_min": "39",
            "temp_max": "61",
```

VueJS Front-end

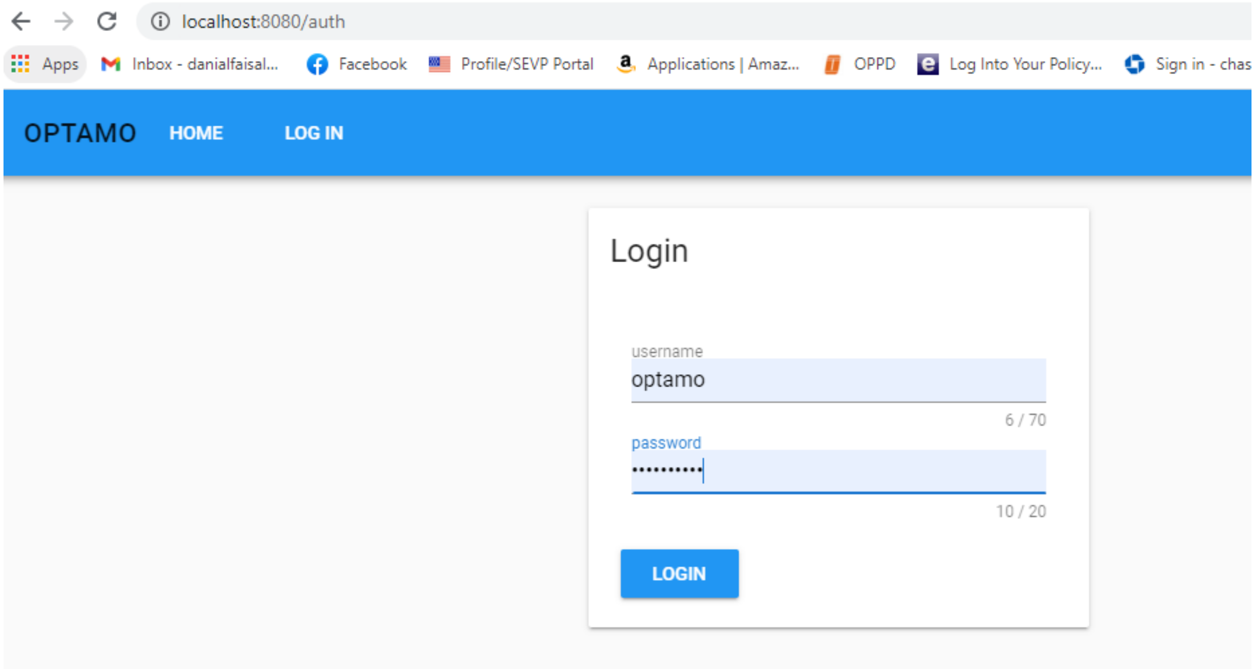
a) First you need to be in the optamo-frontend directory. For me the directory was:

C:\Users\Danial Faisal\Desktop\OPTAMO-Project\optamo-frontend>

b) Once you are in the directory, run the following commands:

- ***npm install***
- ***npm run dev***

c) Enter your credentials, username: **optamo** password: **optamo1234**

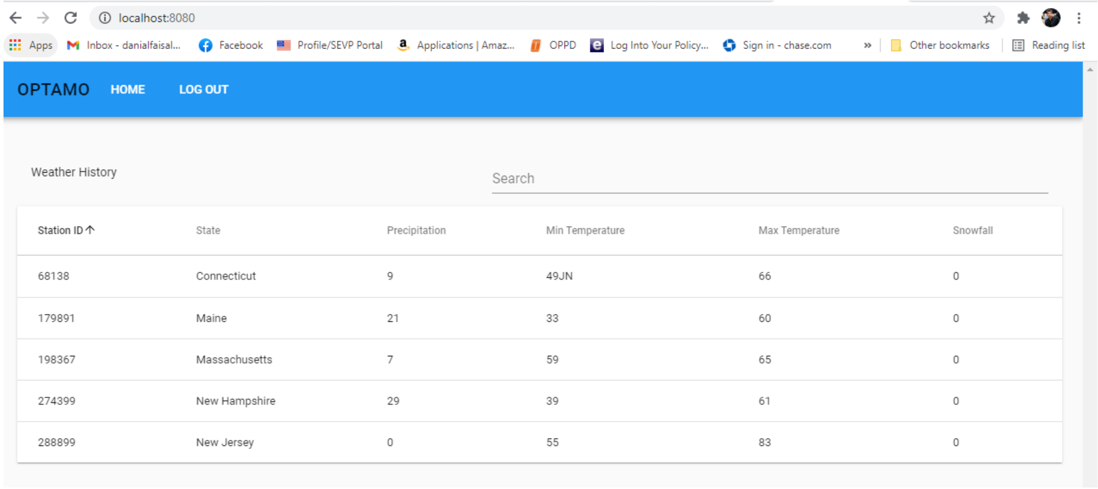


Login

username
optamo
6 / 70

password
••••••••••
10 / 20

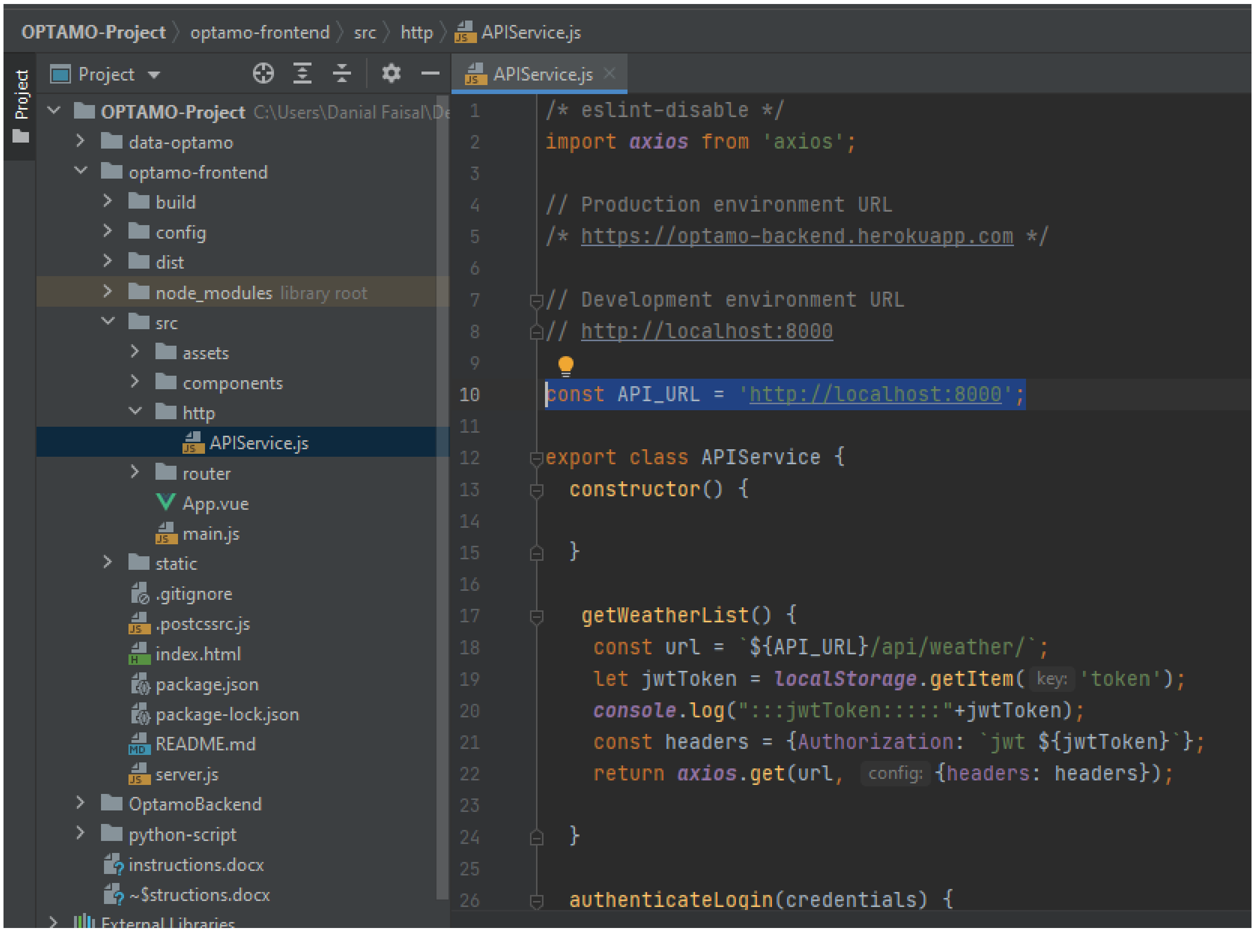
d) You should be able to see the home page.



Weather History

Station ID ↑	State	Precipitation	Min Temperature	Max Temperature	Snowfall
68138	Connecticut	9	49JN	66	0
179891	Maine	21	33	60	0
198367	Massachusetts	7	59	65	0
274399	New Hampshire	29	39	61	0
288899	New Jersey	0	55	83	0

e) Note: in the APIService.js file the API_URL determines which URL the API data is pulled from. For development environment backend we use ***http://localhost:8000*** but in case you are not able to run the back-end, you can use the production environment Back-end as well. ***https://optamo-backend.herokuapp.com***



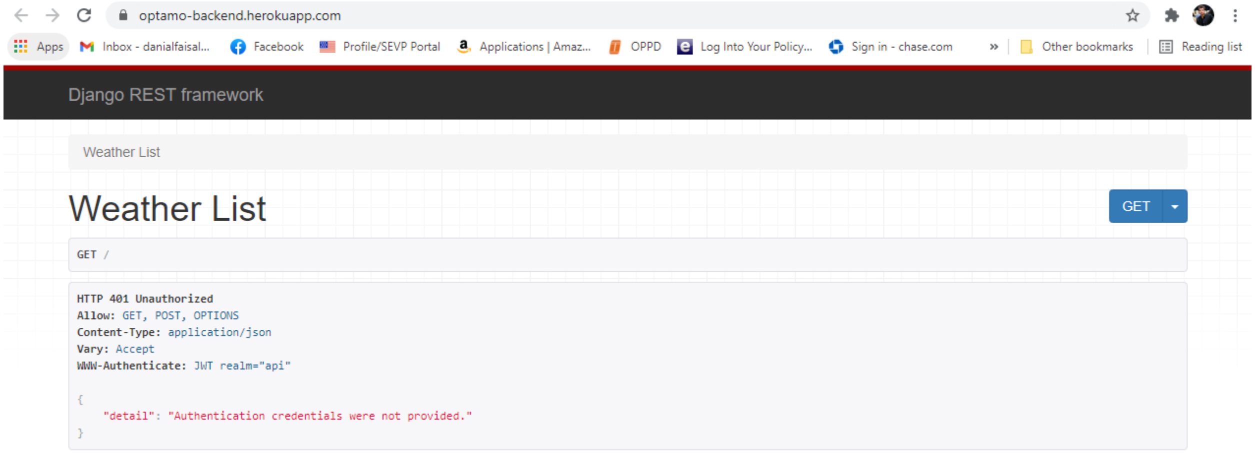
```
/* eslint-disable */
import axios from 'axios';

// Production environment URL
/* https://optamo-backend.herokuapp.com */

// Development environment URL
// http://localhost:8000

const API_URL = 'http://localhost:8000';

export class APIService {
  constructor() {

  }

   getWeatherList() {
    const url = `${API_URL}/api/weather/`;
    let jwtToken = localStorage.getItem( key: 'token');
    console.log(":::jwtToken:::::"+jwtToken);
    const headers = {Authorization: `jwt ${jwtToken}`};
    return axios.get(url,  config: {headers: headers});

  }

  authenticateLogin(credentials) {
```

Production Environment

Python Django Backend

Backend is deployed on the Heroku platform. The URL for the backend user interface is ***https://efs-backend.herokuapp.com/*** In order to access the records you need to go to the **admin** of the site, ***https://efs-backend.herokuapp.com/admin***, enter your credentials and click on **view site.**

- When you enter the URL it will show an authentication error



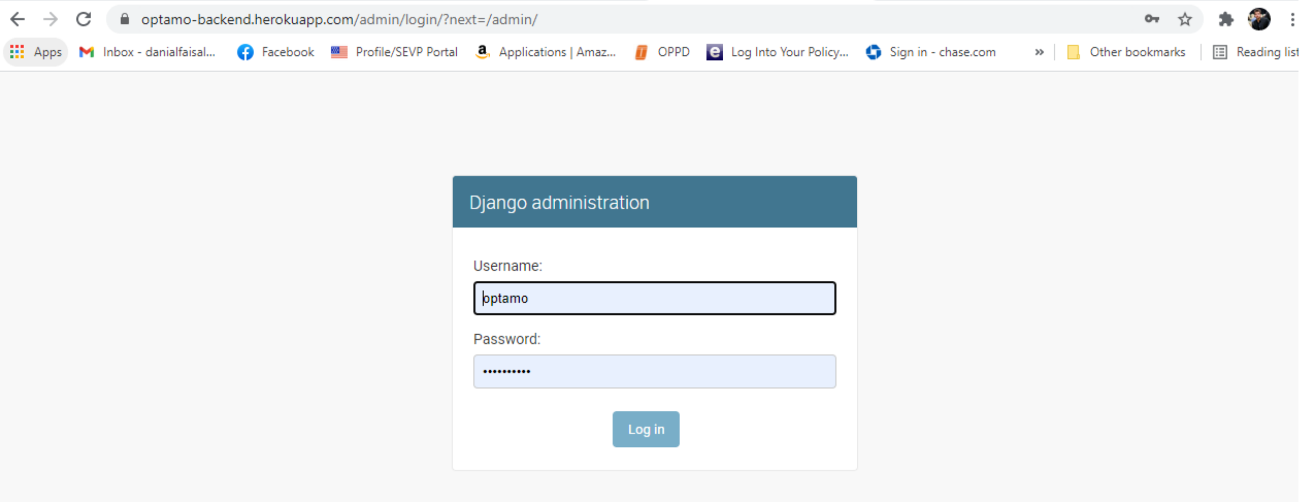
Weather List

Weather List

```
GET /
```

```
HTTP 401 Unauthorized
Allow: GET, POST, OPTIONS
Content-Type: application/json
Vary: Accept
WWW-Authenticate: JWT realm="api"

{
    "detail": "Authentication credentials were not provided."
}
```

- Add admin at the end of the URL and enter your credentials, username: **optamo** password: **optamo1234**

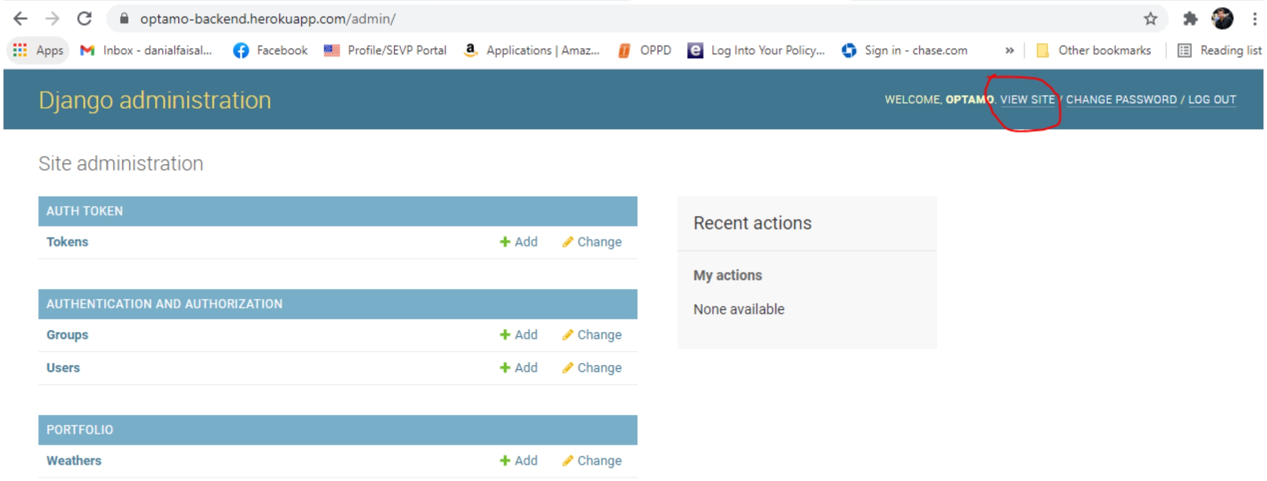


Django administration

Username: optamo

Password: ••••••••••

- In the admin site, you can perform User management and perform CRUD actions on the data. Click on **View Site** on the top right corner.



Django administration

Site administration

AUTH TOKEN		
Tokens	+ Add	Change

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

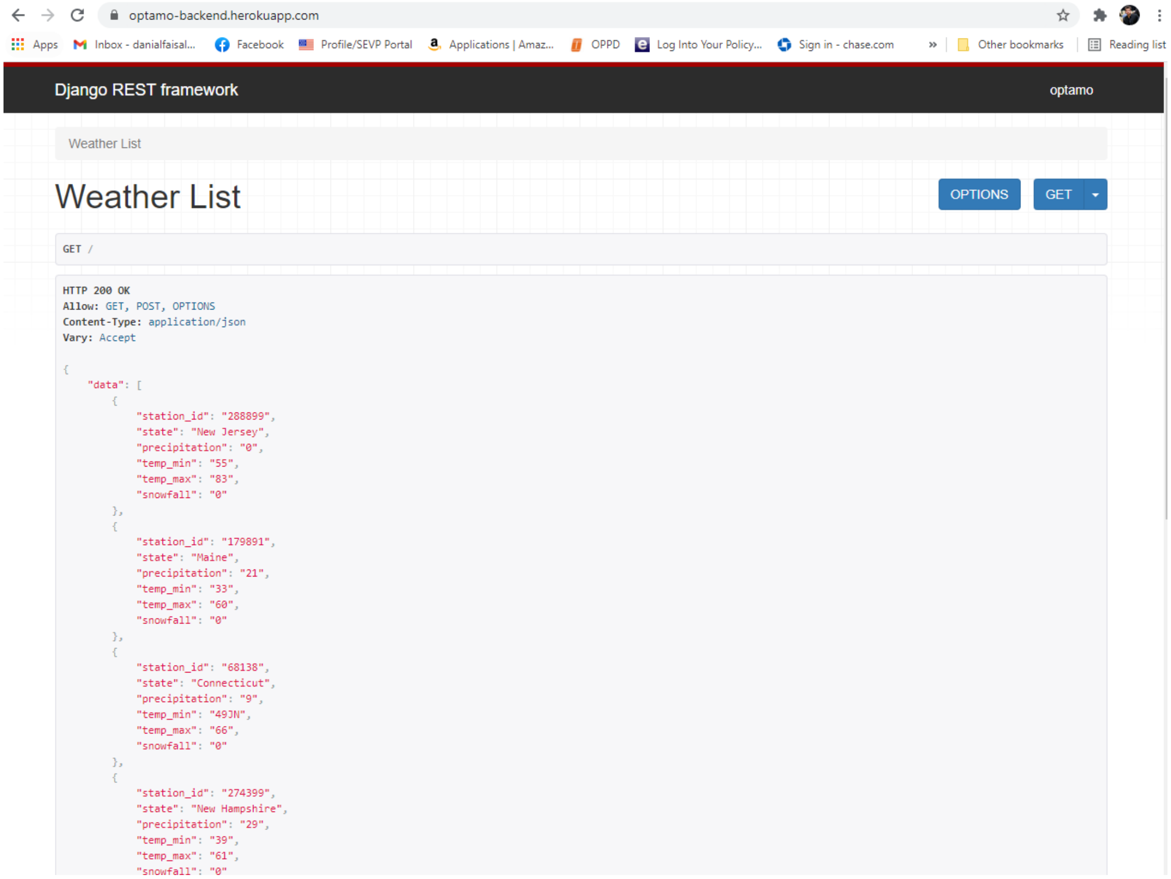
PORTFOLIO		
Weathers	+ Add	Change

Recent actions

My actions

None available

- Once you have logged in, you will be able to see all the data posted on the platform. The URL where the data is posted is https://optamo-backend.herokuapp.com/api/weather/ but the home page also displays the same thing.



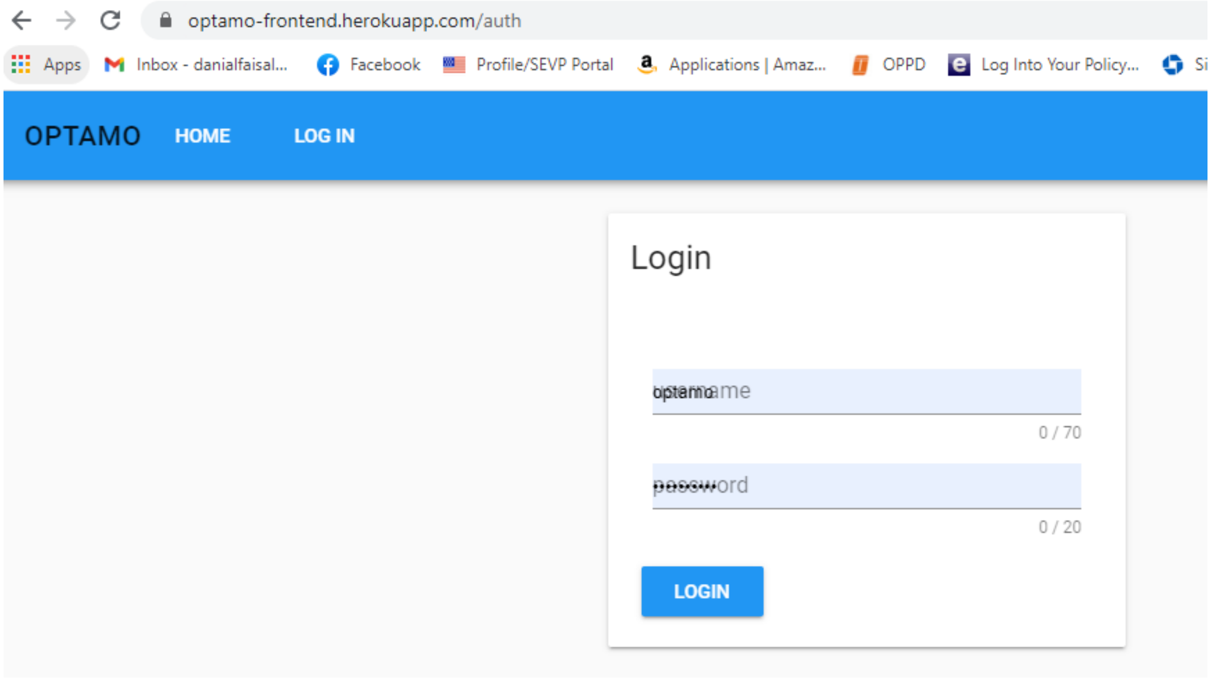
Weather List

```
GET /
```

```
HTTP 200 OK
Allow: GET, POST, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "data": [
        {
            "station_id": "288899",
            "state": "New Jersey",
            "precipitation": "0",
            "temp_min": "55",
            "temp_max": "83",
            "snowfall": "0"
        },
        {
            "station_id": "179891",
            "state": "Maine",
            "precipitation": "21",
            "temp_min": "33",
            "temp_max": "60",
            "snowfall": "0"
        },
        {
            "station_id": "68138",
            "state": "Connecticut",
            "precipitation": "9",
            "temp_min": "49JN",
            "temp_max": "66",
            "snowfall": "0"
        },
        {
            "station_id": "274399",
            "state": "New Hampshire",
            "precipitation": "29",
            "temp_min": "39",
            "temp_max": "61",
            "snowfall": "0"
```

Vue JS front end

Frontend is also deployed on the Heroku platform. The URL for the frontend is *https://optamo-frontend.herokuapp.com/* You will need to enter your credentials to access the page username: **optamo** password: **optamo1234**



Login

optamo username

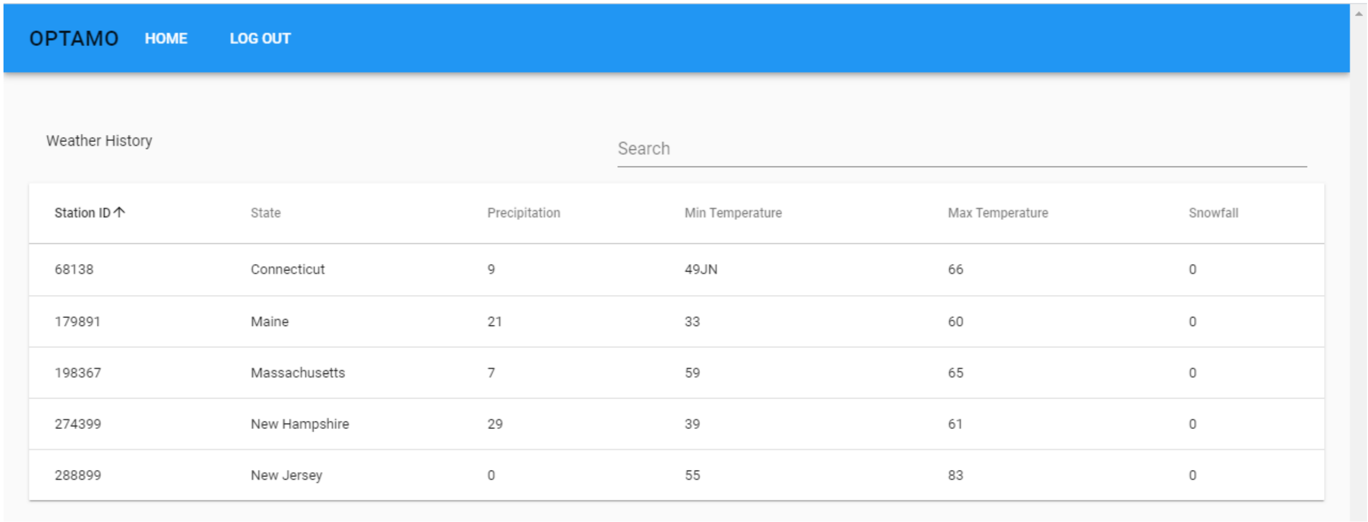
0 / 70

password

0 / 20

LOGIN

- Once you have logged in, you will see the data that was initially posted by the Python script.



Weather History

Search

Station ID ↑	State	Precipitation	Min Temperature	Max Temperature	Snowfall
68138	Connecticut	9	49JN	66	0
179891	Maine	21	33	60	0
198367	Massachusetts	7	59	65	0
274399	New Hampshire	29	39	61	0
288899	New Jersey	0	55	83	0

Improvements that could have been made.

1. Firstly, I used the Django framework instead of Flask because Django is a better framework for large scale applications. Also, I wanted to show my work deployed in a production environment and I have experience deploying applications on Django.

2. I used Vue JS instead of Angular, because again, my strongest front-end language is Vue. Both languages are based on the JavaScript framework, so they are very similar. I have worked a little on Angular as well and could have easily built this on Angular, but I would have needed more time to understand how to make the API connection and deploy the application.

3. For this test, I manually unzipped the tar file and converted the files inside it into the txt format. I believe that I could have automated that step as well if I had a little more time. Basically, we just to add a few more lines in the Python script to unzip a folder first, convert the files into the txt format and then perform the functions.