Task 3

Group Members: Lance Fenicle, Joshua White

Introduction:

The goal is to create a web page that will ask the user for a form to take in information about their name and location. It will then use that information to access an API and display the weather in that location. There are two parts to this project: including the Django web page, which acts as the I/O, and the script that accesses the API to get the weather data.

Table of Functions:

| Function Prototype | Description |
| --- | --- |
| Ger\_user\_info(request) | Handles the form submissions for the user location input. If the form is valid, the data of the session is redirected to the weather display page. |
| Show\_weather(request) | Retrieves the user’s location from the session and fetches the weather data using OpenWeatherMap API, processes it, and passes it to the display html template. |
| Create\_link\_geo(city,state,country,api) | Constructs the API request URL for converting the input location into longitude and latitude. |
| Create\_link\_weather(lat, lon, api) | Constructs the API request URL for getting the current weather data using latitude and longitude. |

Screenshots:

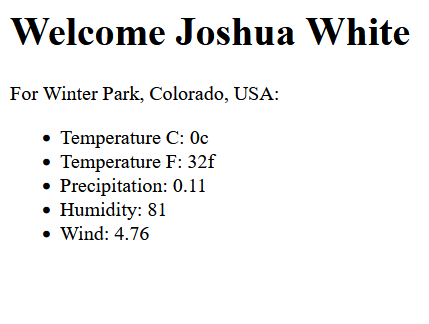
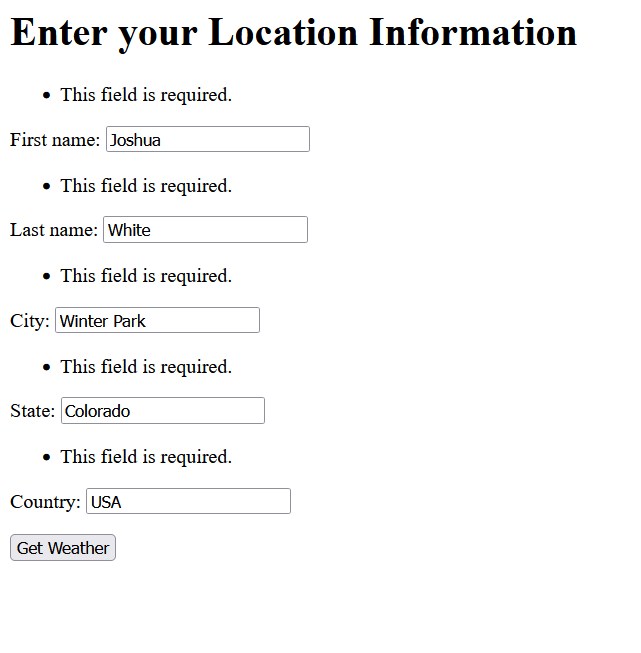
Colorado Springs, Colorado, USA

A screenshot of a computer

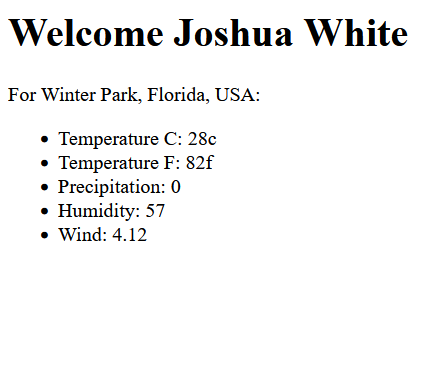
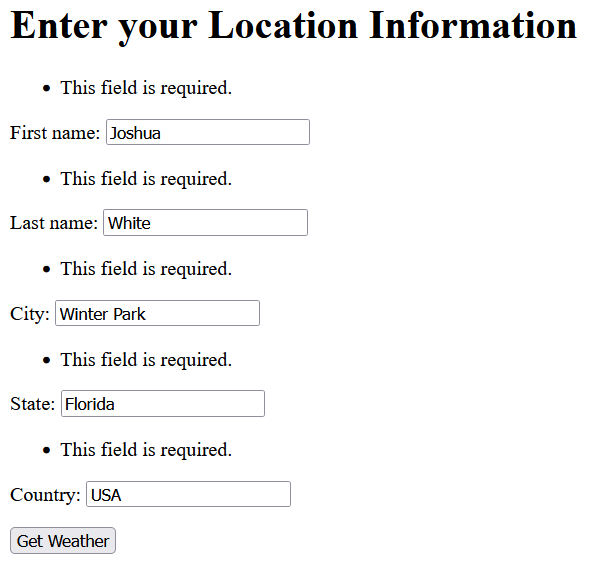
AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

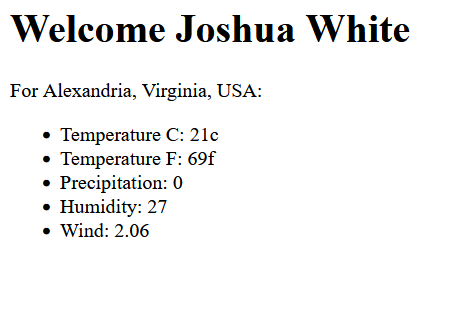
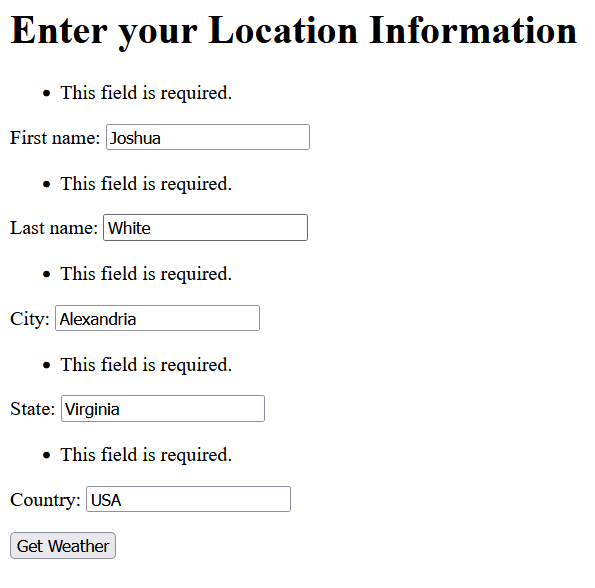
Winter Park, Colorado, USA



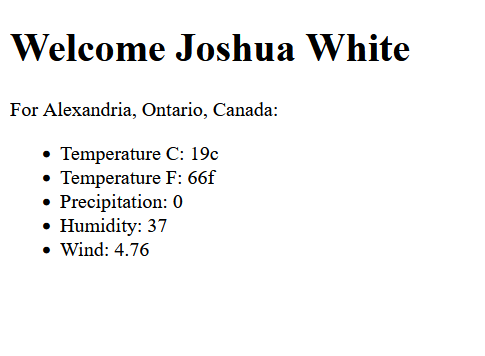
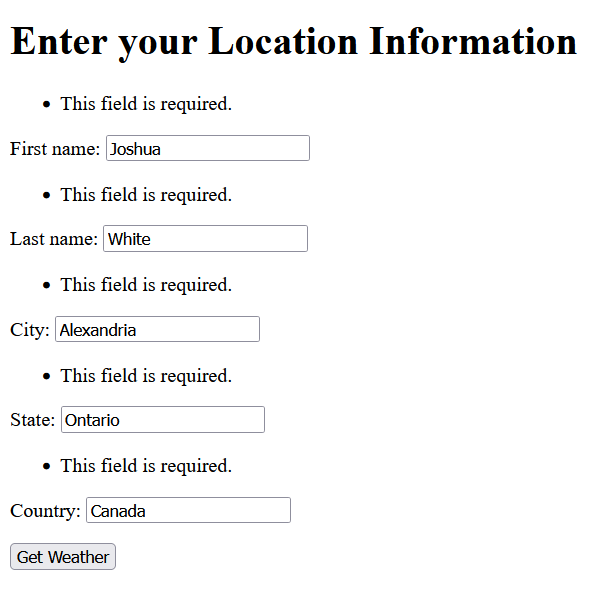
Winter Park, Florida, USA



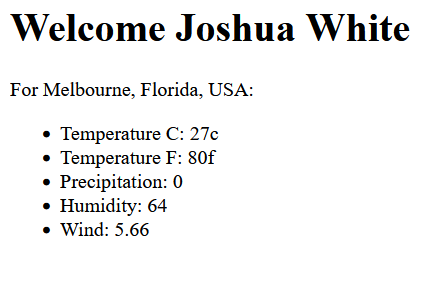
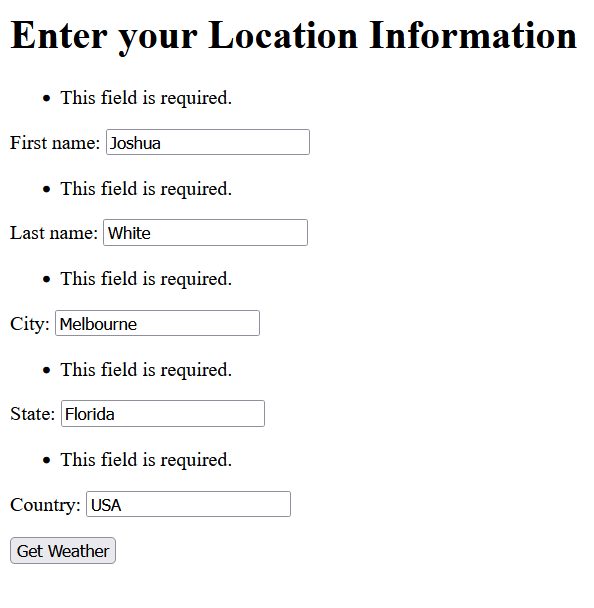
Alexandria, Virginia, USA



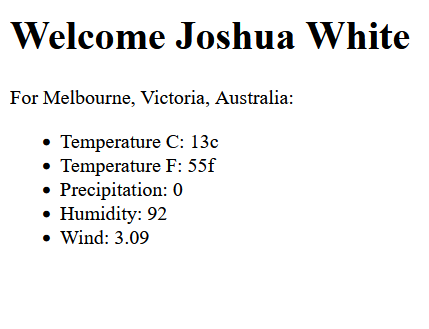
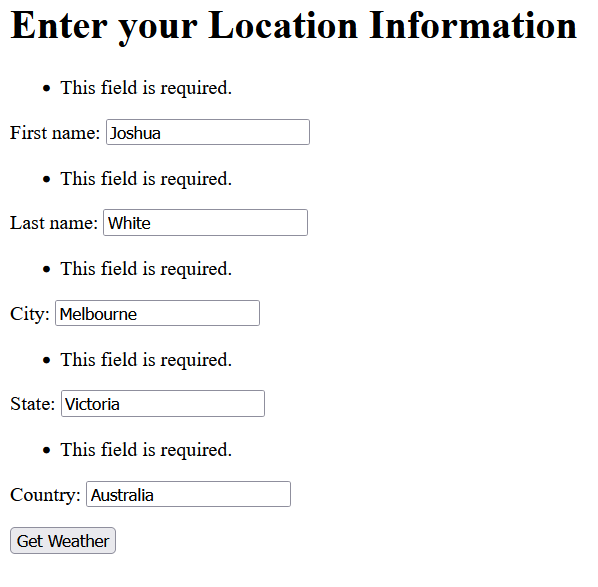
Alexandria, Ontario, Canada



Melbourne, Florida, USA



Melbourne, Victoria, Australia



Steps to run the task:

1. Open Command Prompt
2. Create a Virtual Environment by using py -m venv [name of virual environment]. Python must be installed, if it’s not, then install it
3. Activate the virtual environment my using [name of virtual environment]\Scripts\activate.bat
4. If Django not already installed, install with py -m pip install Django
5. Using the file explorer, move the weatherapp folder inside the virtual environment folder.
6. Navigate to the weatherapp folder (if you have not moved from the moment of making the virtual environment it should be cd [name of virtual environment]\weatherapp
7. Use py manage.py runserver
8. Paste <http://127.0.0.1:8000/> into your browsers search bar
9. Fill out the information, making sure to have correct spelling for city state and country

Conclusion:

This task was more previous than the previous 2, but I’m still proud of how it turned out. One of the more difficult parts was getting the website to work, and then the utilization of WeatherAPI caused some issues. Using WeatherAPI was fun to learn though, I had to learn how to interact with JSON files before we used did in class because it’s needed to use WeatherAPI, then slowly learning how WeatherAPI organizes it’s API. First requiring a call to get city information to grab the latitude and longitude, then a call using those pieces of data to get weather information. The precipitation was a tricky thing to do, because when there hasn’t been any rain or snow for the past hour, WeatherAPI will not show any data for it, meaning that I had to check to see if rain and snow were actually present in the JSON, and had to test with other cities that were getting rain and snow to find out how they organize the data in there because Colorado Springs was not getting any at the time.

The only thing I think I would change at this point is to create another file to hide the API key and grab it from that instead. Because while it’s fine to do for a school project, I understand that having my API key in plain text in the program isn’t best practice.