**Weather Forecast Project**

**Overview**

The Weather Forecast Project is a Python application that interacts with the WeatherAPI to fetch weather data for specified zip codes. It retrieves both daily and hourly weather forecasts, saves the data in CSV files, and logs the process in a structured manner. The project consists of the following core components:

retrieveWeatherForecast.py: Handles the main logic for fetching weather data, saving it to CSV, and logging.

constants.py: Contains constant configurations such as API keys, file paths, and headers.

testcases.py: Provides unit tests for verifying the functionality of the various functions in retrieveWeatherForecast.py.

logs/: Directory to store log files created during execution.

**Project Structure**

weatherforecast/

src/

retrieveWeatherForecast.py Main logic for retrieving, processing, and saving weather data  
 constants.py Configuration constants for API, file paths, headers, etc.

test/

testcases.py Unit tests for weather forecast functions

logs/   
 Directory where log files will be saved

weather\_forecast\_data/   
 Folder where CSV files containing weather forecasts are saved. These files are named based on the forecast type (daily\_forecast\_YYYYMMDD\_HHMM.csv or hourly\_forecast\_YYYYMMDD\_HHMM.csv).

**Installation of Prerequisites**

Make sure you have Python 3.x and the following libraries installed:

requests for API requests

unittest for testing

logging for logging functionality

We can install the necessary libraries using pip:

bash pip install requests

**Running the Application**

You can run the project by executing the retrieveWeatherForecast.py script:

bash python src/retrieveWeatherForecast.py

The script will:

* Set up the logging directory.
* Create folders for saving the forecast data if they don't already exist.
* Fetch the weather data for the zip codes listed in constants.py.
* Save the data into CSV files under weather\_forecast\_data/ (one for daily data, and one for hourly data).
* Log all activities (both successes and errors) in the logs/ directory.

**Functions**

setup\_log\_directory()

* Creates a log directory if it doesn't exist and sets up logging.
* Logs are saved with filenames that include a timestamp for easy tracking.

setup\_forecast\_data\_location()

* Ensures the weather\_forecast\_data folder exists to store CSV files containing forecast data.

initialize\_data\_files(forecastType, header)

* Initializes the CSV files for storing daily or hourly weather forecast data.
* The function returns the filename of the created CSV file.

get\_weather\_data(zip\_code)

* Fetches the weather data for a given zip code using the WeatherAPI.
* Logs the success or failure of the request.

save\_daily\_forecast(daily\_file\_name, data, zip\_code)

* Extracts and saves the daily weather forecast data into the specified CSV file.

save\_hourly\_forecast(hourly\_file\_name, data, zip\_code)

* Extracts and saves the hourly weather forecast data into the specified CSV file.

main()

* The main function that coordinates the entire process of fetching and saving forecast data for the zip codes specified in constants.py.

**Unit Tests**

The project includes unit tests for each function using the unittest framework. The tests are in test/testcases.py.

To run the tests, simply run the following command:

bash

python m unittest test/testcases.py

Key Tests

* test\_initialize\_data\_files(): Verifies that the data files are correctly initialized with the specified headers.
* test\_get\_weather\_data\_success(): Mocks a successful weather data retrieval and verifies the response.
* test\_save\_daily\_forecast(): Verifies that the daily weather data is correctly written to the CSV file.
* test\_save\_hourly\_forecast(): Verifies that the hourly weather data is correctly written to the CSV file.
* test\_main(): Tests the entire flow of fetching and saving forecast data.

(.venv) m-vgwp473vcq:$ python -m unittest testcases.py

/Users/m0m08kn/PycharmProjects/weather\_forecast/.venv/lib/python3.9/site-packages/urllib3/\_\_init\_\_.py:35: NotOpenSSLWarning: urllib3 v2 only supports OpenSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. See: https://github.com/urllib3/urllib3/issues/3020

warnings.warn(

.....

----------------------------------------------------------------------

Ran 5 tests in 0.006s

OK

**View the Logs**

Logs are created in the logs/ directory with filenames formatted as weather\_forecast\_<timestamp>.log. These logs contain detailed information about the following:

* Initialization of data files.
* Successes and failures in fetching data from the API.
* Data saving events for both daily and hourly forecasts.
* Any exceptions that occur during the process.

Example Output

1. Log Example (saved in logs/ directory):

weather\_forecast\_20241110194047.log

2024-11-10 19:40:47,072 - INFO - daily forecast data will be saved on ./weather\_forecast\_data/daily\_forecast\_20241110\_1940.csv

2024-11-10 19:40:47,073 - INFO - Data written to ./weather\_forecast\_data/daily\_forecast\_20241110\_1940.csv

2024-11-10 19:40:47,073 - INFO - hourly forecast data will be saved on ./weather\_forecast\_data/hourly\_forecast\_20241110\_1940.csv

2024-11-10 19:40:47,075 - INFO - Data written to ./weather\_forecast\_data/hourly\_forecast\_20241110\_1940.csv

2024-11-10 19:40:47,075 - INFO - Fetching weather data for zip code: 07030

2024-11-10 19:40:47,222 - INFO - Successfully fetched data for zip code 07030

2024-11-10 19:40:47,236 - INFO - Daily forecast data saved for zip code 07030

2024-11-10 19:40:47,257 - INFO - Hourly forecast data saved for zip code 07030

2024-11-10 19:40:47,257 - INFO - Data successfully processed for zip code: 07030

2024-11-10 19:40:47,257 - INFO - Fetching weather data for zip code: 07086

2024-11-10 19:40:47,332 - INFO - Successfully fetched data for zip code 07086

2024-11-10 19:40:47,348 - INFO - Daily forecast data saved for zip code 07086

2024-11-10 19:40:47,365 - INFO - Hourly forecast data saved for zip code 07086

2024-11-10 19:40:47,365 - INFO - Data successfully processed for zip code: 07086

2. CSV File Example (saved in weather\_forecast\_data/ directory):

daily\_forecast\_20241110\_1940.csv

zipcode,date,maxtemp\_f,mintemp\_f,avgtemp\_f,daily\_chance\_of\_rain,daily\_chance\_of\_snow,condition.text,condition.icon,timestamp

07030,2024-11-10,57.4,39.4,49.4,0,0,Overcast ,//cdn.weatherapi.com/weather/64x64/day/122.png,2024-11-10 19:40:47

07030,2024-11-11,66.4,54.1,57.9,64,0,Patchy rain nearby,//cdn.weatherapi.com/weather/64x64/day/176.png,2024-11-10 19:40:47

07030,2024-11-12,54.3,38.8,49.7,0,0,Sunny,//cdn.weatherapi.com/weather/64x64/day/113.png,2024-11-10 19:40:47

07086,2024-11-10,57.2,39.6,49.4,0,0,Overcast ,//cdn.weatherapi.com/weather/64x64/day/122.png,2024-11-10 19:40:47

07086,2024-11-11,62.4,54.7,57.7,82,0,Patchy rain nearby,//cdn.weatherapi.com/weather/64x64/day/176.png,2024-11-10 19:40:47

07086,2024-11-12,53.8,38.9,49.9,0,0,Sunny,//cdn.weatherapi.com/weather/64x64/day/113.png,2024-11-10 19:40:47

hourly\_forecast\_20241110\_1940.csv

zipcode,date,time,temp\_f,feelslike\_f,heatindex\_f,windchill\_f,humidity,cloud,chance\_of\_rain,chance\_of\_snow,condition.text,condition.icon,timestamp

07030,2024-11-10,2024-11-10 00:00,45.3,45.3,45.3,45.3,42,0,0,0,Clear ,//cdn.weatherapi.com/weather/64x64/night/113.png,2024-11-10 19:40:47

07030,2024-11-10,2024-11-10 01:00,44.5,44.5,44.5,44.5,44,0,0,0,Clear ,//cdn.weatherapi.com/weather/64x64/night/113.png,2024-11-10 19:40:47

07030,2024-11-10,2024-11-10 02:00,43.8,43.8,43.8,43.8,46,0,0,0,Clear ,//cdn.weatherapi.com/weather/64x64/night/113.png,2024-11-10 19:40:47

07030,2024-11-10,2024-11-10 03:00,43.2,43.2,43.2,43.2,47,43,0,0,Partly Cloudy ,//cdn.weatherapi.com/weather/64x64/night/116.png,2024-11-10 19:40:47

07030,2024-11-10,2024-11-10 04:00,41.5,40.1,41.5,40.1,48,34,0,0,Partly Cloudy ,//cdn.weatherapi.com/weather/64x64/night/116.png,2024-11-10 19:40:47

07030,2024-11-10,2024-11-10 05:00,40.4,38.4,40.4,38.4,57,70,0,0,Cloudy ,//cdn.weatherapi.com/weather/64x64/night/119.png,2024-11-10 19:40:47

07030,2024-11-10,2024-11-10 06:00,39.9,37.8,39.9,37.8,58,43,0,0,Partly Cloudy ,//cdn.weatherapi.com/weather/64x64/night/116.png,2024-11-10 19:40:47

07030,2024-11-10,2024-11-10 07:00,39.7,38.2,39.7,38.2,59,64,0,0,Cloudy ,//cdn.weatherapi.com/weather/64x64/day/119.png,2024-11-10 19:40:47

07030,2024-11-10,2024-11-10 08:00,40.7,39.9,40.7,39.9,60,94,0,0,Overcast ,//cdn.weatherapi.com/weather/64x64/day/122.png,2024-11-10 19:40:47

07030,2024-11-10,2024-11-10 09:00,43.3,43.0,43.3,43.0,58,100,0,0,Overcast ,//cdn.weatherapi.com/weather/64x64/day/122.png,2024-11-10 19:40:47

07030,2024-11-10,2024-11-10 10:00,47.2,46.7,47.2,46.7,54,37,0,0,Partly Cloudy ,//cdn.weatherapi.com/weather/64x64/day/116.png,2024-11-10 19:40:47

07030,2024-11-10,2024-11-10 11:00,50.4,49.3,50.4,49.3,44,46,0,0,Partly Cloudy ,//cdn.weatherapi.com/weather/64x64/day/116.png,2024-11-10 19:40:47