### **Assignment 1: Basic Flask Setup**

* Install Flask (pip install flask).
  + Open terminal, type: pip install flask
* Create a simple Flask app (app.py) with one route (/) that returns "Hello, Weather App!".
* Learn to run Flask locally (python app.py) and access it in the browser.

**Assignment 2: Fetch Weather Data (without Flask)**

* Introduce the [OpenWeatherMap](https://openweathermap.org/) API.
* Write a standalone Python script (no Flask yet) that:
  + Uses requests to call the OpenWeatherMap API.
  + Prints out temperature, humidity, wind speed, and description in the terminal.
  + Convert temperature from Kelvin to Celsius and Fahrenheit with a helper function.

*Note: You can use* [*this link*](https://temp-mail.org/en/) *to generate a throwaway email if you don't want to sign up using your main email*

### **Assignment 3: Integrate API into Flask**

* Move the weather-fetching code into the Flask app.
* Use Flask’s render\_template with a simple index.html that just prints:
  + "Weather in Helsinki: XX°C"

### **Assignment 4: Improve Template & Display More Data**

* Expand the template (index.html) to show:
  + Temperature (°C/°F)
  + Feels like
  + Humidity
  + Wind speed
  + Description
* Add CSS styling for readability.

### **Assignment 5: Add Sunrise, Sunset, and Styling**

* Parse sunrise and sunset times from the API (convert from Unix timestamp to readable format).
* Add them to the template.
* Enhance the UI with:
  + A card layout
  + Background gradient
  + Emojis/icons

## **Final Project: Complete Weather App**

* Combine everything into a polished **Weather App**:
  + Flask backend + OpenWeatherMap API integration.
  + Temperature in Celsius/Fahrenheit.
  + Feels like, humidity, wind speed, condition.
  + Sunrise & sunset times (adjusted to local timezone).
  + Styled weather card UI with icons/emojis.

### Optional Additions

* Let users search for a city instead of hardcoding “Helsinki”
* Handle API errors gracefully (invalid city, no internet)
* Deploy to a free service (like Render, Railway, or PythonAnywhere