IoT Weather Station with Raspberry Pi

**Project Description:** Create an Internet of Things (IoT) weather station using a Raspberry Pi that collects weather data from various sensors, displays the information locally, and uploads it to a cloud platform for remote monitoring.

**Skills/Objectives:**

1. **Computer Hardware:** 
   * Set up and configure Raspberry Pi with the required peripherals.
2. **Computer Interfacing:** 
   * Connect and interface various sensors (e.g., temperature, humidity, and pressure sensors) with the Raspberry Pi GPIO pins.
3. **Internet of Things:** 
   * Use Wi-Fi connectivity on the Raspberry Pi to send collected sensor data to a cloud platform.
4. **Troubleshooting:** 
   * Implement error handling mechanisms and troubleshoot issues that may arise during the setup or data transmission.

**Project Phases:**

1. **Planning:**
   * Define project goals and objectives, emphasizing the integration of computer hardware, interfacing, IoT, and troubleshooting skills.
   * Identify the necessary hardware components, including sensors and peripherals.
   * Determine the visualizing the collected data.
   * Plan the overall architecture of the system, considering data flow and communication protocols.
2. **Design:**
   * Conduct initial tests to ensure the Raspberry Pi setup is functioning correctly.
   * Test sensor connections and data readings to verify accurate sensor interfacing.
   * Implement a testing phase for the Python script, checking local data display
   * Collect feedback from initial testing, making adjustments to the design as needed.
3. **Testing and Feedback:**
   * Bug reports and issue tracking records.
   * User feedback and improvement suggestions.
   * Revised code and user interface based on testing and feedback.
4. **Documentation:**
   * Develop comprehensive documentation that includes:
     1. Setup instructions for the Raspberry Pi, including OS installation and Wi-Fi configuration.
     2. Schematics for connecting sensors to GPIO pins.
     3. Detailed explanations of the Python script, including code comments.
     4. Troubleshooting guidelines and solutions for common issues.
     5. Compile a component list with links for easy procurement.

**Python Libraries:**

You can use Python libraries like tkinter, Matplotlib, Requests, Adafruit CircuitPython DHT and random for creating a graphical user interface (GUI) and generating displays from sensors.

**Project Structure:**

* sensors.py: The main Python script containing the calculator logic.
* app.py: The script for creating the GUI and handling user interactions.
* README.md: Documentation for using the calculator app.
* requirements.txt: List of required libraries and dependencies.