**Hardware and Software requirements:**

**Hardware:**

1. Arduino Uno (or any compatible Arduino board)
   * Acts as the microcontroller to read temperature data and control the cooling fan.
2. Temperature Sensor (e.g., LM35, DHT11, or DS18B20)
   * Measures the CPU temperature in real-time and sends data to the Arduino for processing.
3. Cooling Fan (DC fan or CPU fan)
   * The cooling fan regulates the temperature by turning on or adjusting its speed based on the readings from the temperature sensor.
4. Relay Module (e.g., 5V relay module)
   * Used to control the cooling fan by switching it on or off based on the temperature readings from the Arduino.
5. Jumper Wires
   * For connecting the components (Arduino, sensor, relay, fan) together.
6. Power Supply
   * For powering the Arduino and the cooling fan (can use a USB cable for Arduino and an external power source for the fan if needed).
7. Breadboard (optional)
   * Useful for easy prototyping and connecting components without soldering.
8. Resistors (optional, depending on sensor type)
   * Required for some sensors to ensure proper functioning (e.g., a 10kΩ resistor might be used with the DHT11 sensor).
9. Cooling Fan Mounting Hardware (optional)
   * If needed, to properly secure the fan in place and direct airflow efficiently**.**

**Software:**

1. **Arduino IDE**
   * The main software used for writing, compiling, and uploading code to the Arduino board. It supports multiple programming languages like C and C++.
2. **Arduino Libraries**
   * Depending on the temperature sensor you're using, you may need specific libraries. Some common libraries include:
     + **DHT.h**: For DHT11 or DHT22 temperature sensors.
     + **OneWire.h** and **DallasTemperature.h**: For DS18B20 temperature sensors.
     + **Adafruit\_Sensor.h**: A general library for sensor data.
3. **Serial Monitor** (within Arduino IDE)
   * Used for debugging and displaying temperature readings and other system information from the Arduino during development.
4. **Basic Text Editor** (optional)
   * If you prefer to write the code outside the Arduino IDE and later upload it to the board.