How I solved it :

Understanding the Input:

Alright so the message came in this thing called JSON It's big in the IoT world—basically how gadgets talk to each other. The message had all the important bits like temperature and where the device was hanging out

Creating the Function:

I whipped up a function to handle this JSON message. But here's the kicker sometimes the message comes in as a string and other times it's already in an object form.So I had to make sure my function was cool with both. If it's a string, I had to flip it into an object so the program wouldn't freak out.

Processing the Temperature: Once the message was all set, I checked if the temperature was there. If it was, I did a quick check:

- If the temp was over 30°C I slapped a "High" status on the message.
- If it was 30°C or less, I gave it a "Normal" status. Simple as that.

Error Handling:

Now, if the temperature wasn't in the message, we'd be in trouble. You can't give a status without a temp, right? So, I made the function throw an error if that happens and let the user know, "Hey, something's missing."

Returning the Updated Message:

Finally, after everything was sorted out, I sent back the updated message, now with the status all set.

Problems I faced:

Handling the Input: One thing I ran into was making sure the function didn't care if the message came in as a string or an object. JSON can show up in different ways, so I had to make sure it could handle both without breaking a sweat.

Missing Data: Another issue was what happens if the temperature isn't there. Without that, there's no way to figure out the status. So, I made sure the function handled that smoothly by throwing an error and telling the user what's up.