**Individual Risk Reductions:**

Zahid:

A screenshot of a computer

Description automatically generated

Left: Transmitter | Right: Receiver

*Transmitter:* Sending temperature and humidity as comma separated values, read from DHT22

Using Arduino Unified Library for sensor reading

*Receiver:*

Using strtok to separate text at comma

Using strtof to parse floating point

Text

Description automatically generated

*Upload to Firebase Cloud Firestore:*

Using open source Firebase ESP Client library to interface between receiver ESP32 and database

Text

Description automatically generated

Text

Description automatically generated

Each data entry in database gets a timestamp, temperature, and humidity value. The timestamp is obtained by syncing with a Network Time Protocol (NTP) server, using the NTPClient library.

Alvin:

*Vercel:*

Vercel is a platform that allows you to host and deploy web apps created with the Next.js React framework, just by linking to a GitHub repository containing a Next.js app.

React is a JavaScript library that lets you build user interfaces based on components and interface with other libraries more seamlessly. This makes it much easier to develop a web page as opposed to writing pure HTML, CSS, and JS.

The Next.js framework extends Reacts capabilities, making things even more modular. For example, it handles page routing to other URLs on your site as long as you add a corresponding JavaScript file in the pages folder of the project.

*Firebase interfacing:*

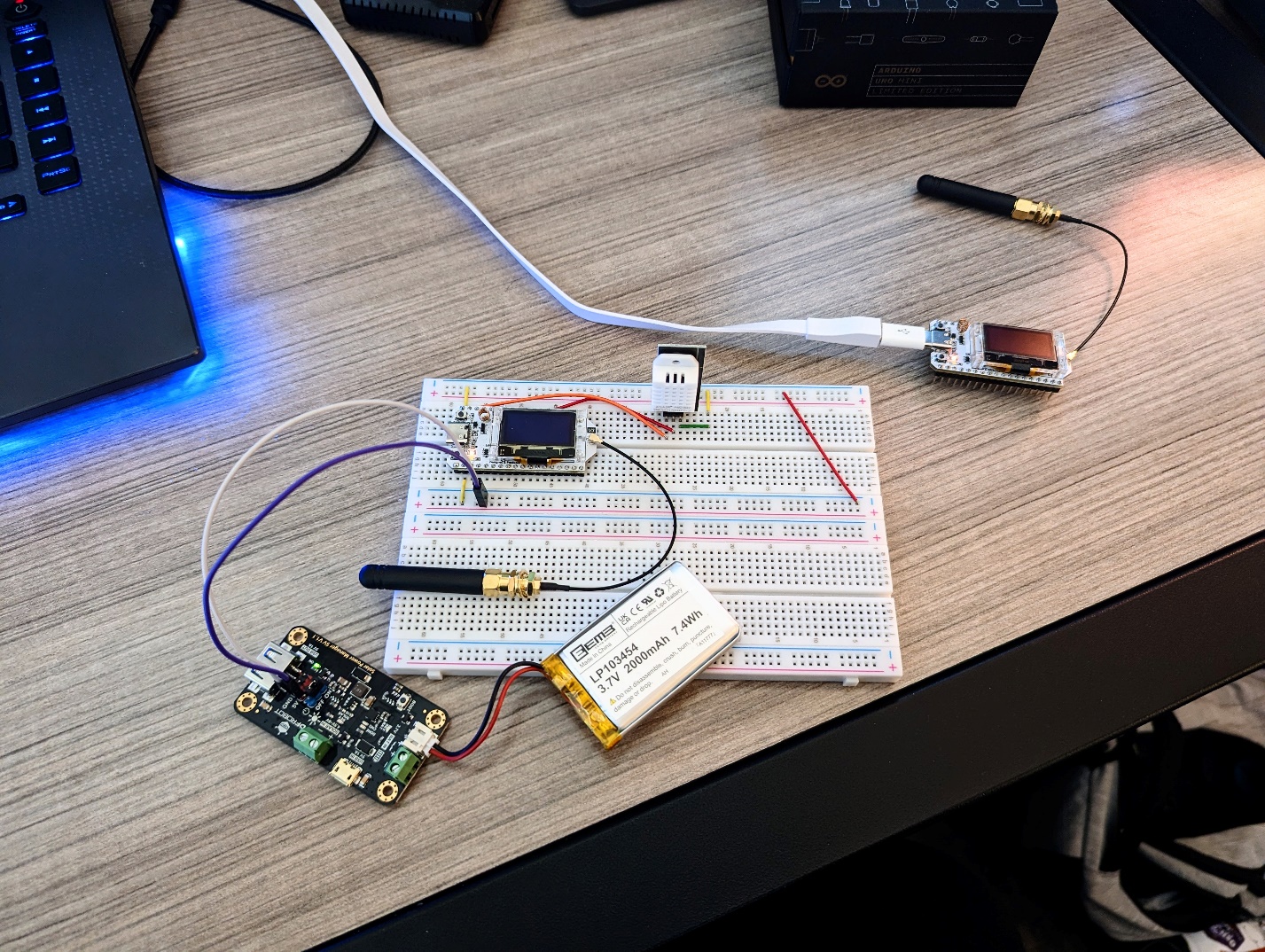
Google provides a Firebase JavaScript API that allows users to easily access their database with simple function calls:

Text

Description automatically generated

Here are the few lines we used to access our sensor data from the Cloud Firestore database.

**Team Risk Reduction:**

****

*On breadboard:*

LiPo connected to solar charge manager (panel not connected because it isn’t necessary for demonstration)

DHT22 temp+humidity sensor connected to ESP32 transmitter

*In corner:*

ESP32 receiver connected receiving LoRa packets, uploading to database over WiFi

*Database:*

A screenshot of a computer

Description automatically generated

In the serial monitor log, you can see the most recent uploaded packet matching with an entry in our Firebase database in the browser.

*Web app:*

Graphical user interface, text, application

Description automatically generated

A very crude homepage, clicking the button take you to this next page:

Chart

Description automatically generated with medium confidence

This page contains actual temperature data that our receiver uploaded after receiving from the transmitter. Tried to get humidity to show on the same chart, but we were struggling to get it working.