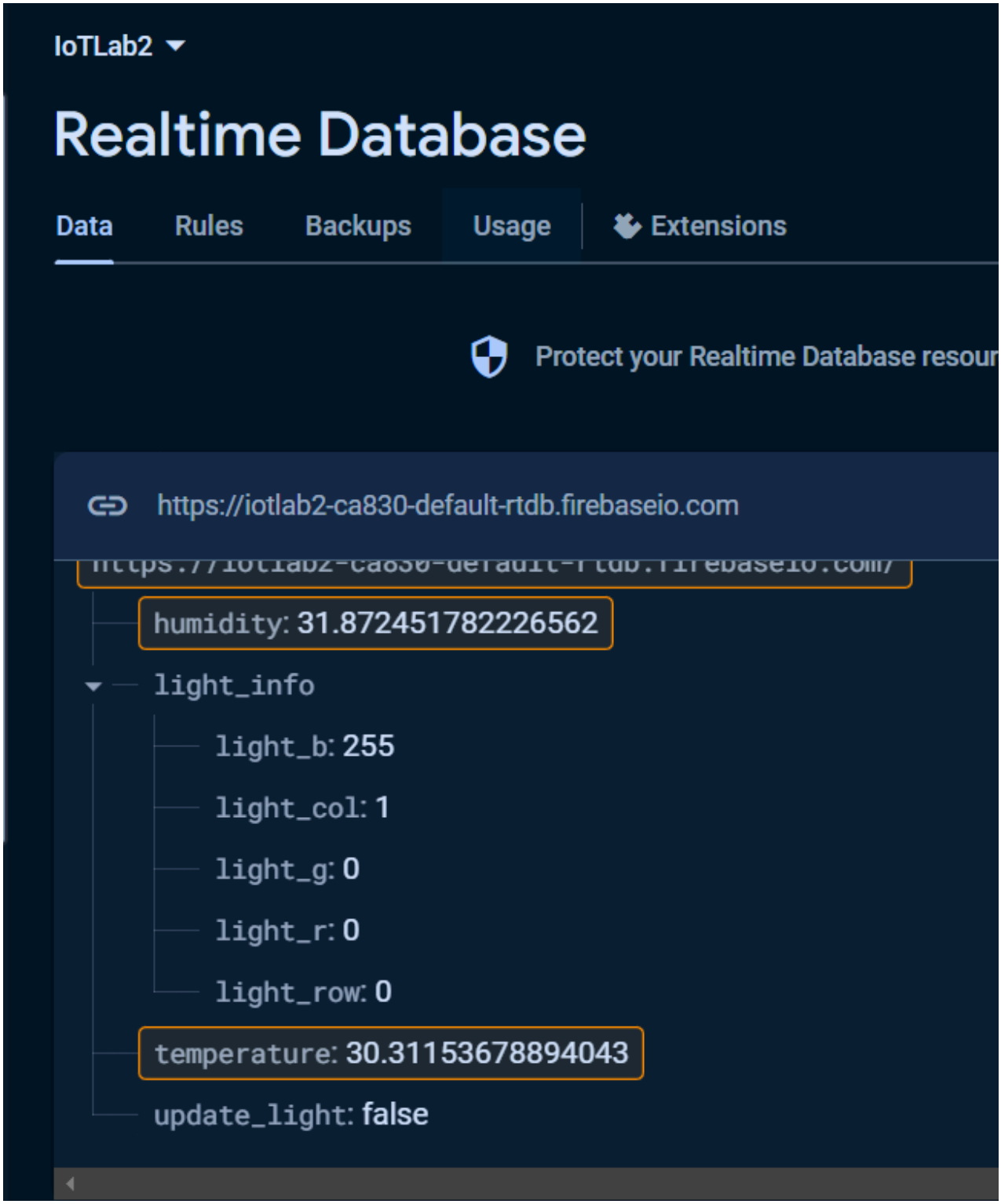
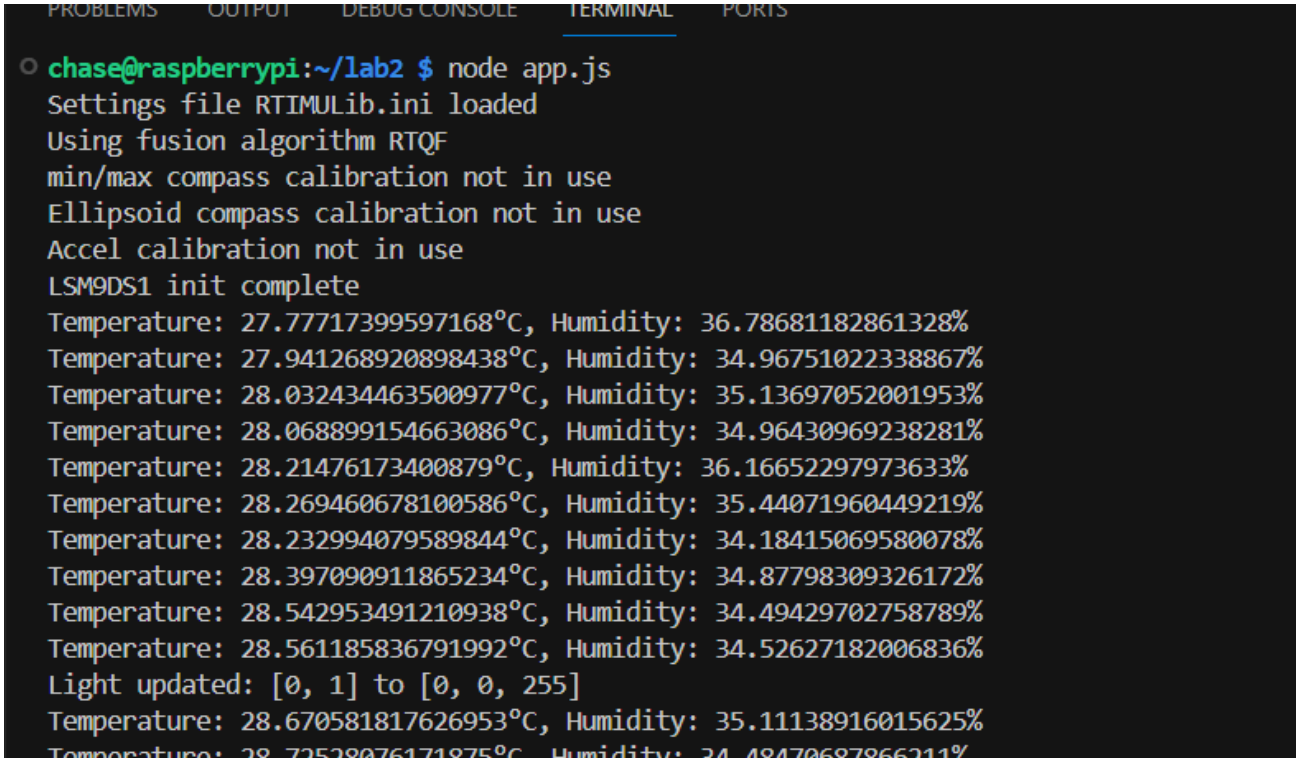

IoTLab2
Realtime Database
Data
Rules
Backups
Usage
Extensions
Protect your Realtime Database resour
https://iotlab2-ca830-default-rtdb.firebaseio.com
humidity: 31.872451782226562
light_info
light_b: 255
light_col: 1
light_g: 0
light_r: 0
light_row: 0
temperature: 30.31153678894043
update_light: false



Code:

```
var firebase = require('firebase/app');
require('firebase/analytics');
const { getDatabase, ref, onValue, set, update, get } =
require('firebase/database');
var nodeimu = require( '@trbll/nodeimu' );
var IMU = new nodeimu.IMU( );
var sense = require( '@trbll/sense-hat-led' );

// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyC2JWGJOylB21xTbAfTgzpA0gC2ceIG8fY",
  authDomain: "iotlab2-ca830.firebaseapp.com",
  projectId: "iotlab2-ca830",
  storageBucket: "iotlab2-ca830.appspot.com",
  messagingSenderId: "868635210220",
  appId: "1:868635210220:web:11b13892607ff5ebfb50c9",
```

```
  measurementId: "G-NB06Q0QTH2"
};

// Initialize Firebase
const app = firebase.initializeApp(firebaseConfig);
const database = getDatabase();

// Function to initialize default values in the database
function initializeDefaultValues() {
  const defaultValues = {
    temperature: 0,
    humidity: 0,
    update_light: false,
    light_info: {
      light_r: 0,
      light_g: 0,
      light_b: 0,
      light_row: 0,
      light_col: 0
    }
  };

  get(ref(database)).then((snapshot) => {
    if (!snapshot.exists()) {
      set(ref(database), defaultValues).then(() => {
        console.log("Default values set in Firebase database.");
      }).catch((error) => {
        console.error("Error setting default values in Firebase
database:", error);
      });
    }
  });
}

// Call initializeDefaultValues at the start of your application
initializeDefaultValues();

// Function to update temperature and humidity
function updateSensorData() {
  IMU.getValue((error, data) => {
```

```
    if (error) {
      console.log(error);
      return;
    }

    const { temperature, humidity } = data;
    console.log(`Temperature: ${temperature}°C, Humidity: ${humidity}%`);

    const updates = {};
    updates['/temperature'] = temperature;
    updates['/humidity'] = humidity;

    update(ref(database), updates);
  });
}

// Listen for updates
const lightUpdateRef = ref(database, 'update_light');
onValue(lightUpdateRef, (snapshot) => {
  const updateLight = snapshot.val();
  if (updateLight) {
    get(ref(database, 'light_info')).then((snapshot) => {
      if (snapshot.exists()) {
        const { light_r, light_g, light_b, light_row, light_col } =
snapshot.val();
        sense.setPixel(parseInt(light_row), parseInt(light_col),
parseInt(light_r), parseInt(light_g), parseInt(light_b));
        console.log(`Light updated: [${light_row}, ${light_col}] to
[${light_r}, ${light_g}, ${light_b}]`);

        set(ref(database, 'update_light'), false);
      }
    });
  }
});

// Periodically update sensor data
setInterval(updateSensorData, 5000); // Every 5 seconds
```