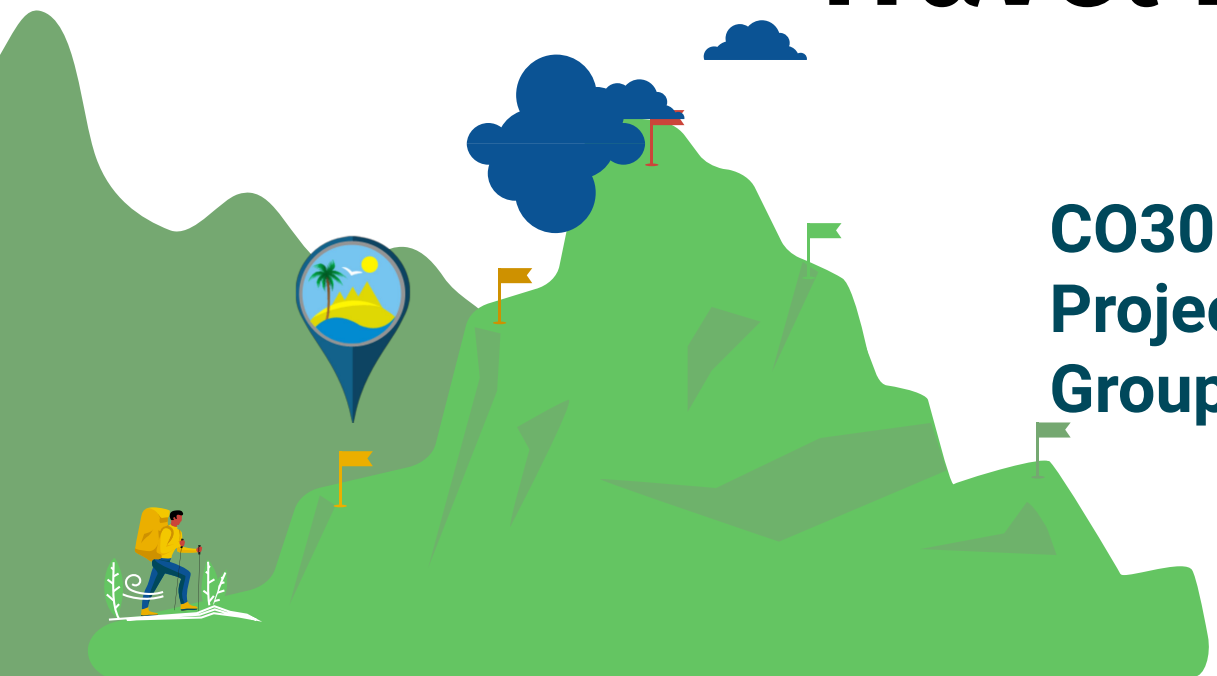
Weather Analytics and Travel Path Guider



CO300 : Third Year Project
Project Milestone 3
Group 07

Team Members

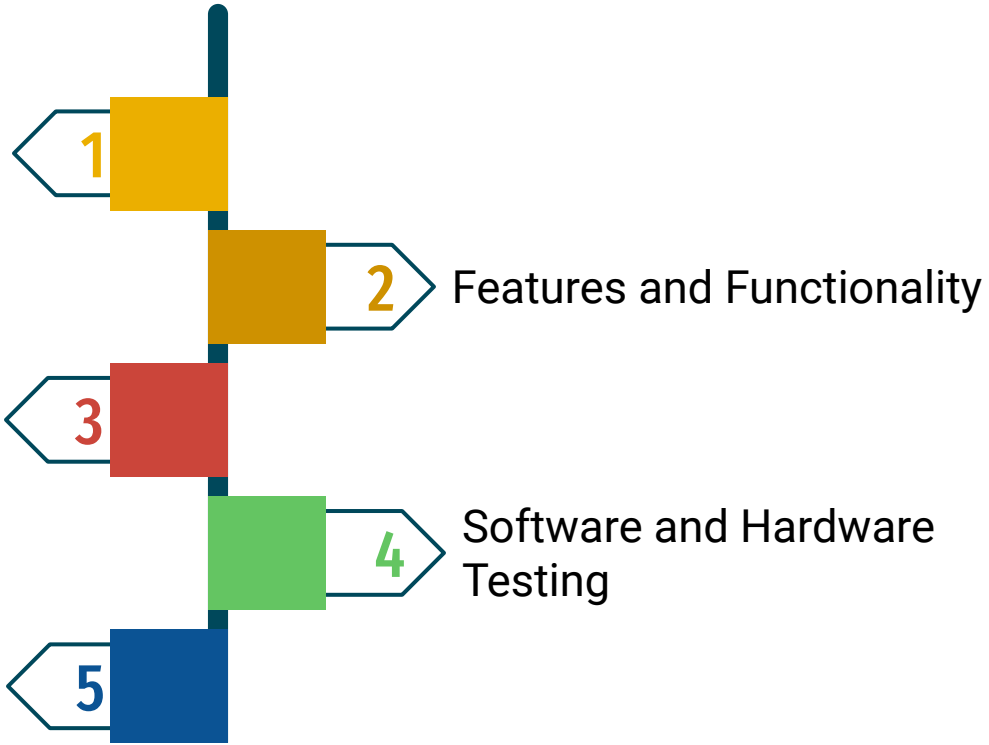
Ishan Fernando
E/18/098

Adeepa Fernando
E/18/100

Ridma Jayasundara
E/18/155

Outline

Project Overview

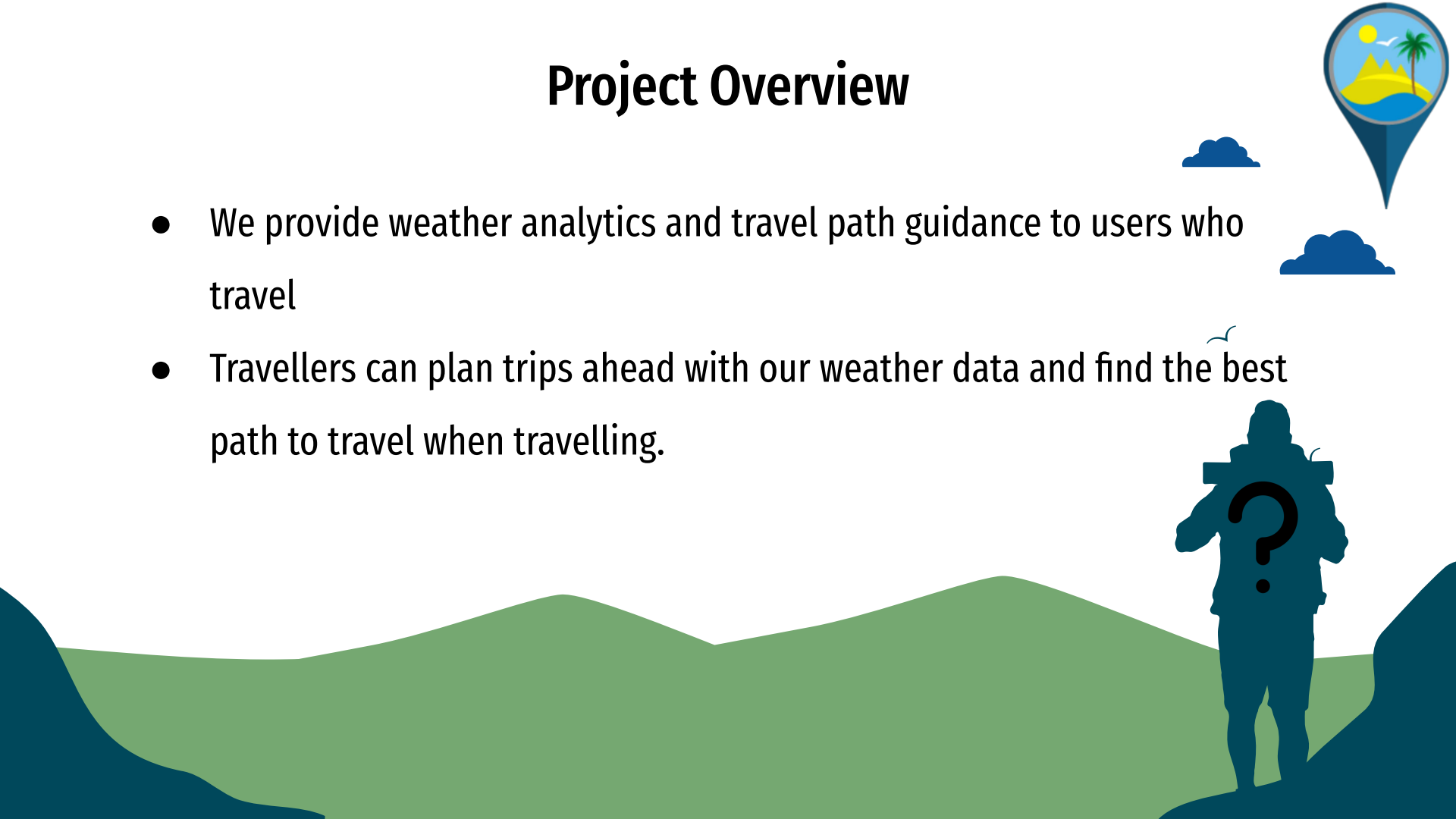


Implementation

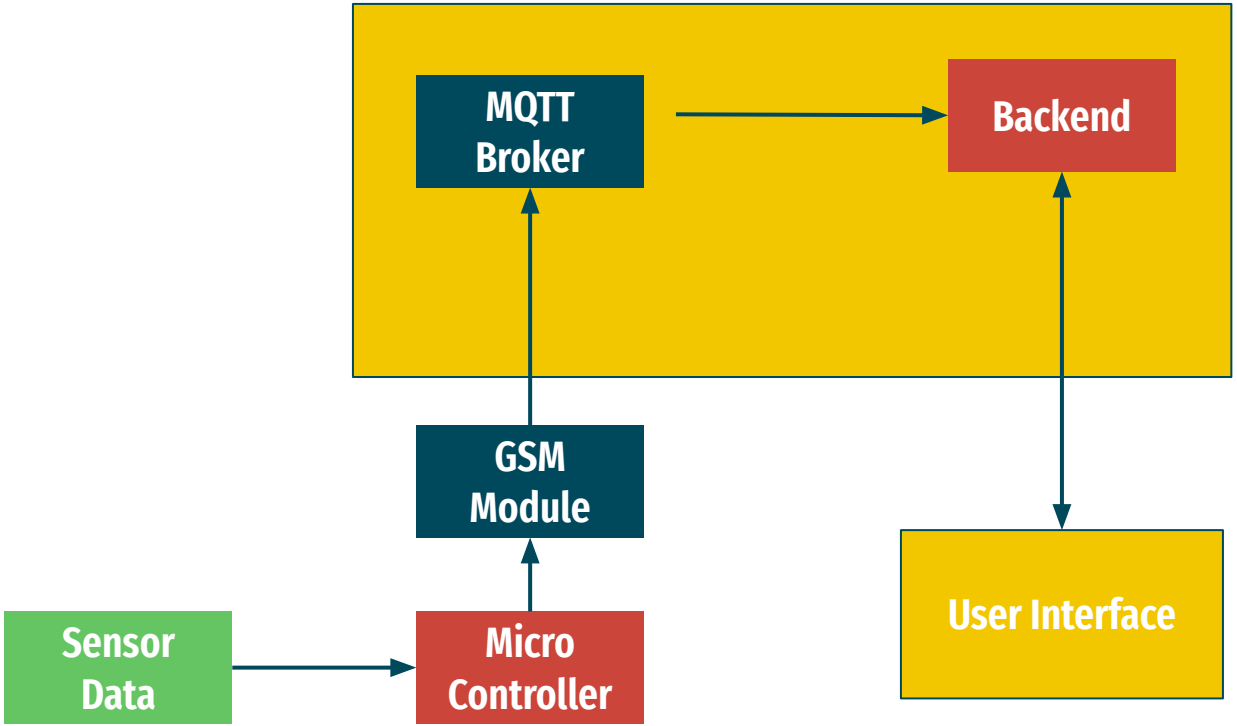
QnA

Project Overview

- We provide weather analytics and travel path guidance to users who travel
- Travellers can plan trips ahead with our weather data and find the best path to travel when travelling.



High Level System Overview

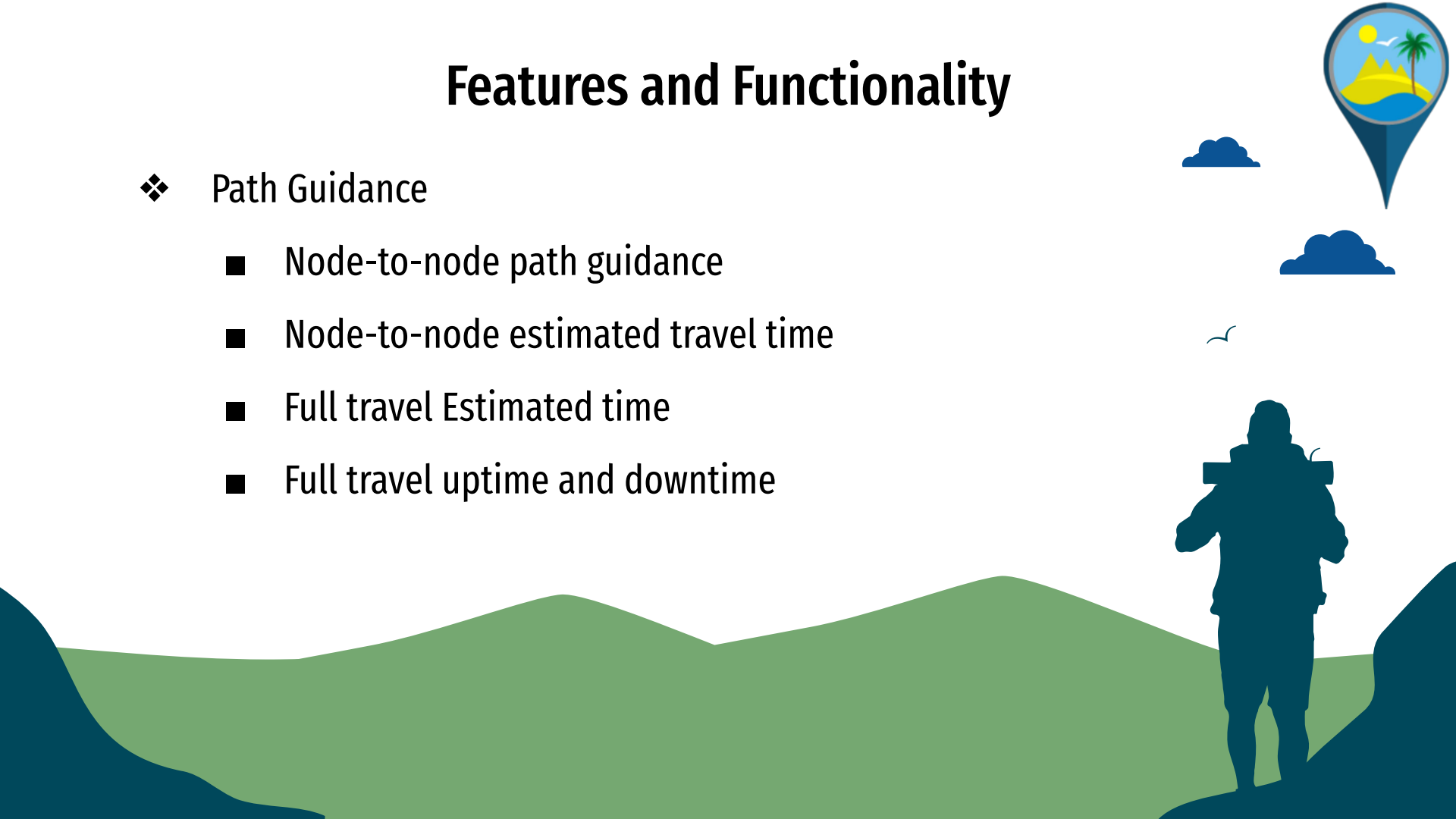


Features and Functionality

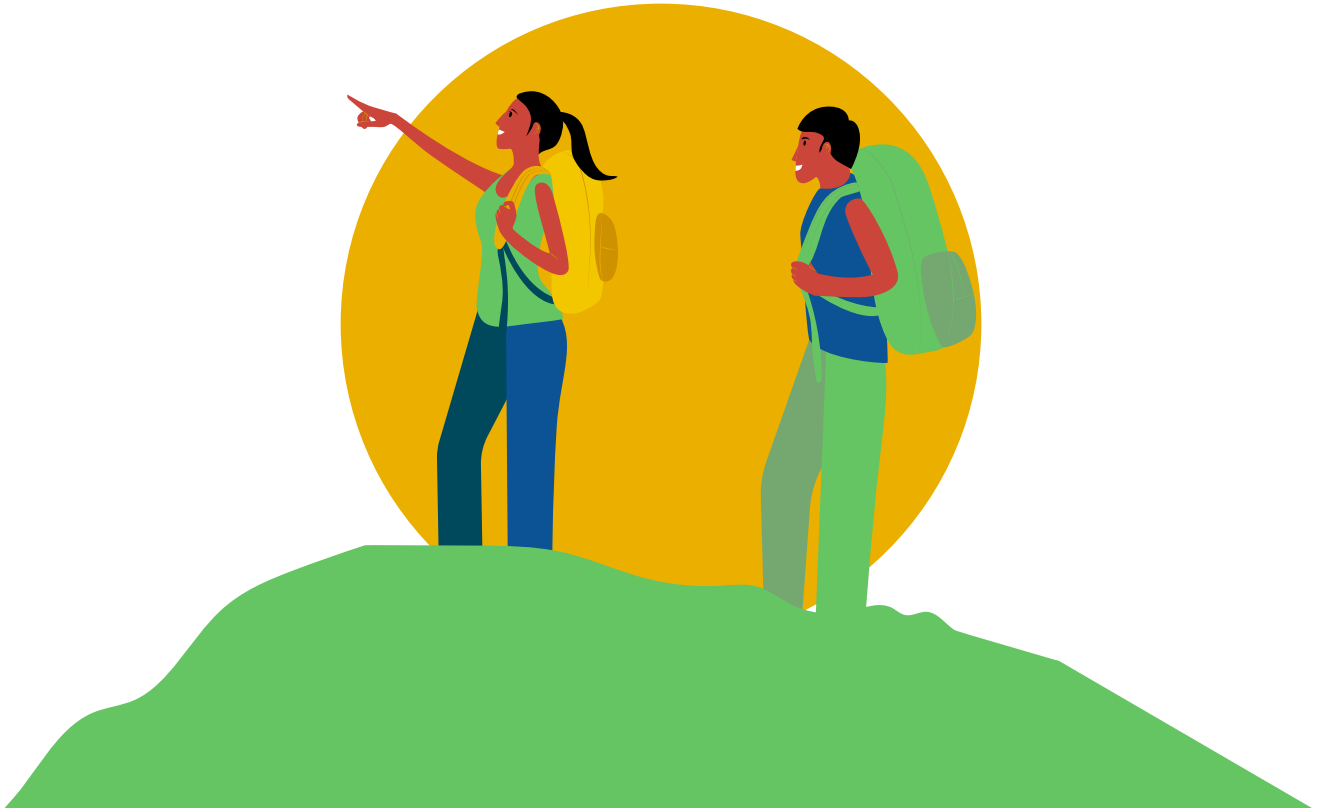
- Real time Weather Analytics data
  - Temperature Level
  - Humidity Level
  - Light Intensity Level
  - Rain Level in 4 categories : Very High, High, Medium, Low
  - Air Quality
- All this weather data for a year-back can be checked.

Features and Functionality

- Path Guidance
  - Node-to-node path guidance
  - Node-to-node estimated travel time
  - Full travel Estimated time
  - Full travel uptime and downtime

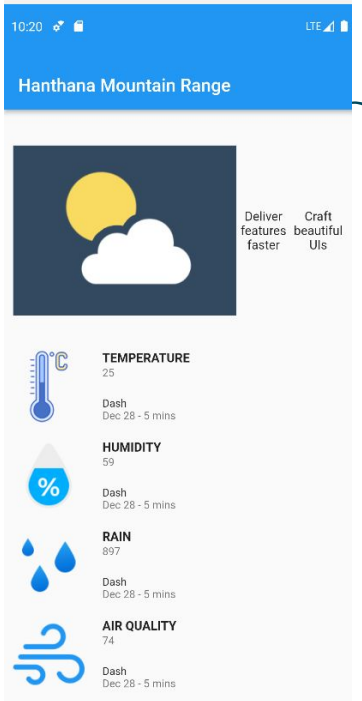


Implementation



Implementation - Frontend

- ✓ We are developing a mobile application for users to get information about weather and path guidance.
- ✓ User sign-in
- ✓ User sign-up
- ✓ Location Selection
- ✓ Real time Weather information View

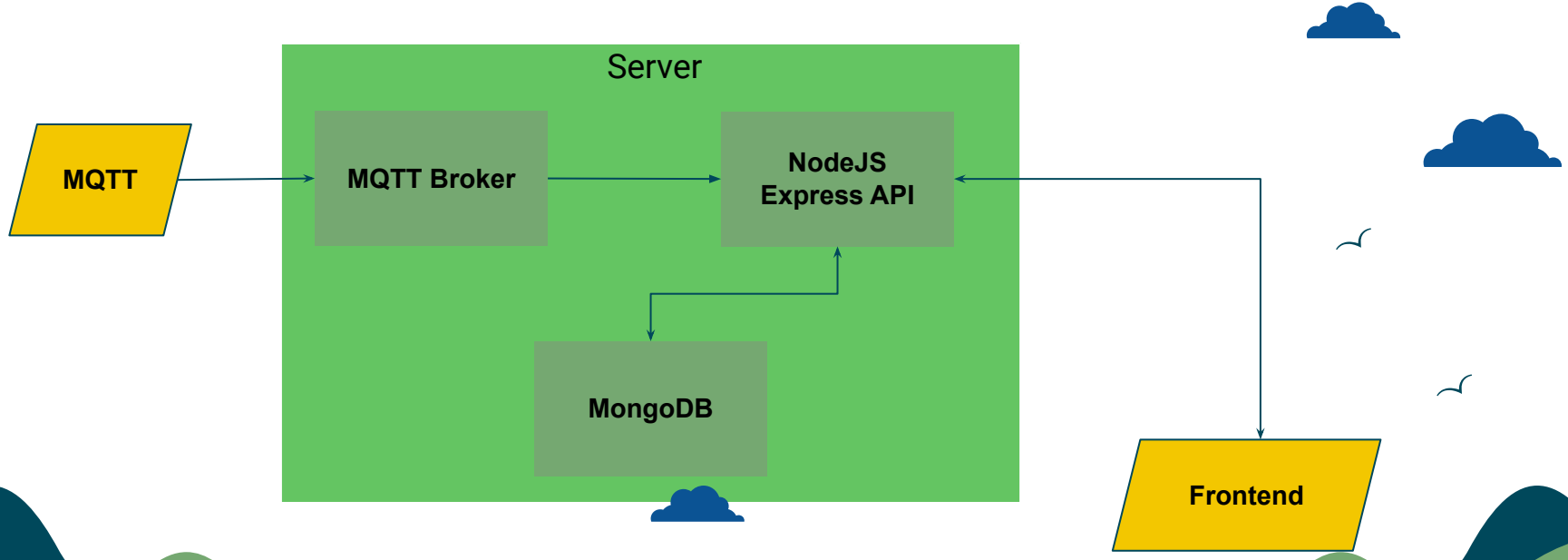


Implementation - Frontend

Implementation - Backend

- ✓ Deployed on Azure server
- ✓ Fully functional database
- ✓ Fully functional data transmission from the Hardware node
- ✓ Fully functional data transmission to the frontend
- ✓ All the required API's for the frontend

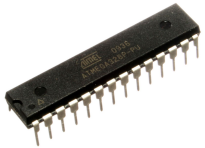
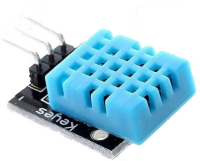
Backend Implementation Overview

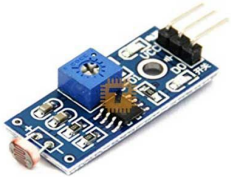


Implementation - Hardware

✓ Assembled the sensors and the microcontroller.

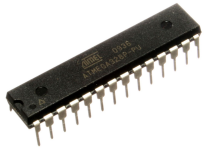
- DHT11 Temperature and Humidity Sensor
- YL-83 FC-37 Rain Sensor
- LDR Light Sensor
- Air Quality Sensor



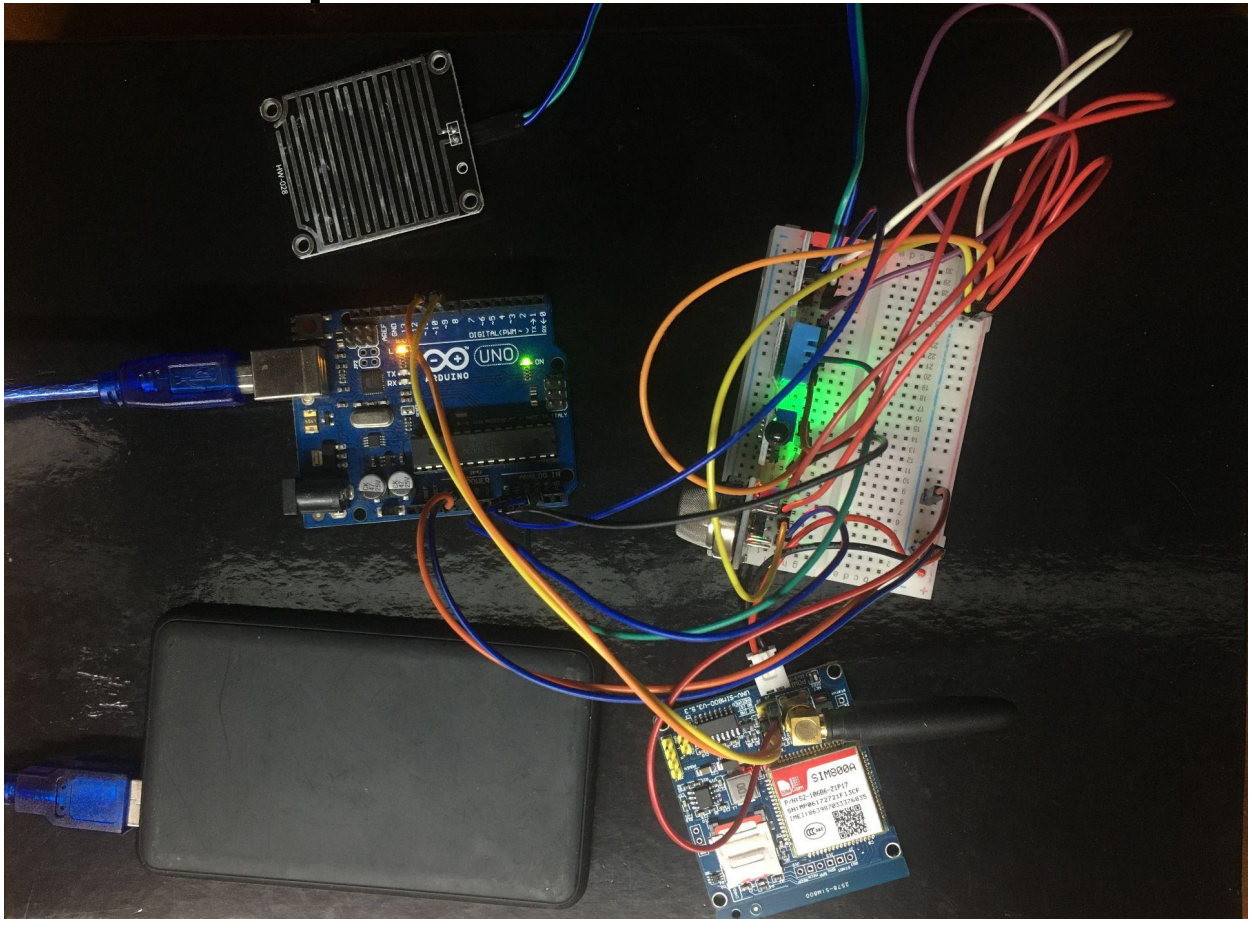

Implementation - Hardware

- ✓ Program the microcontroller to get the readings from the data
- ✓ Program the GSM module to connect to the Internet
- ✓ Program the microcontroller to connect to the server using MQTT and send data to the server

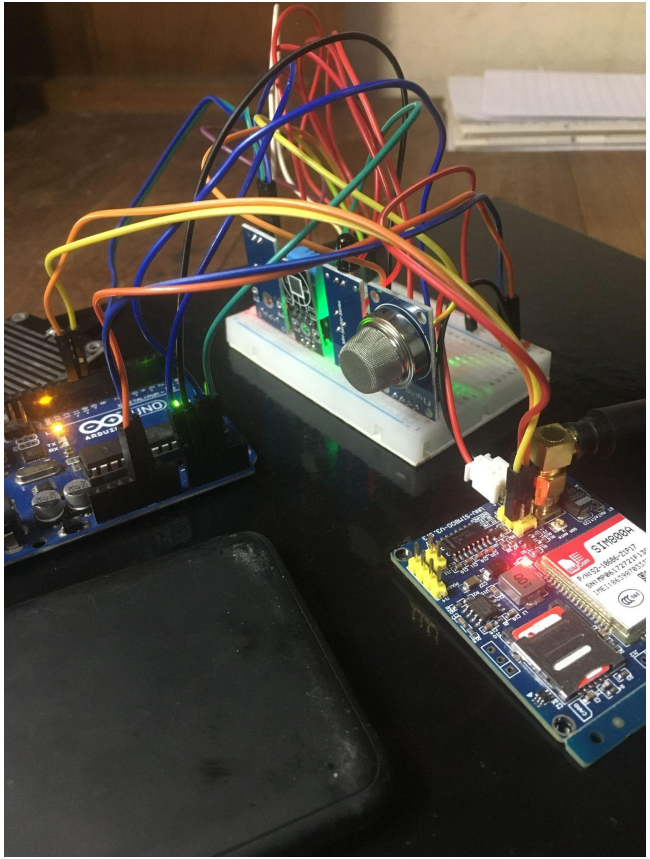


MQTT

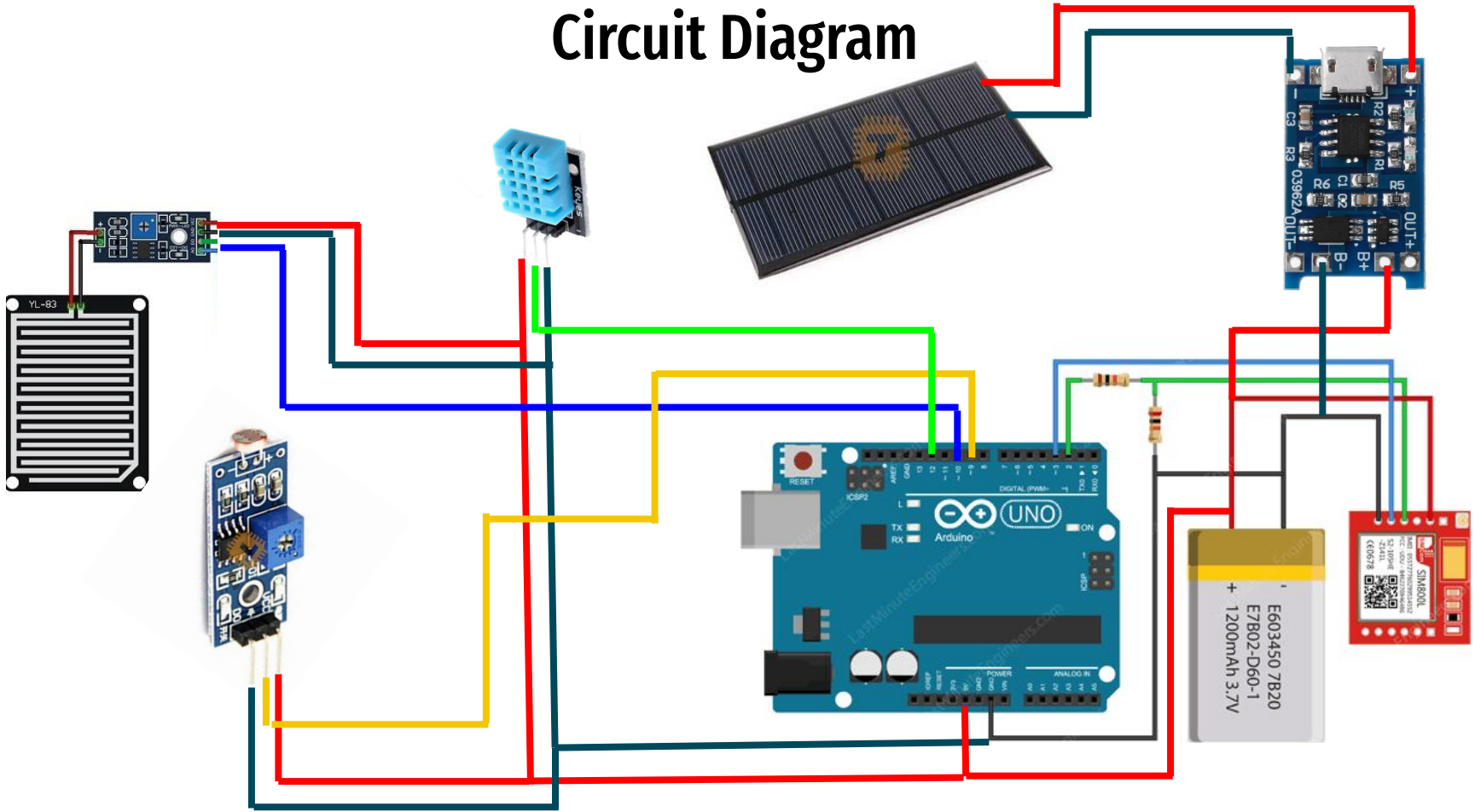
Implementation - Hardware



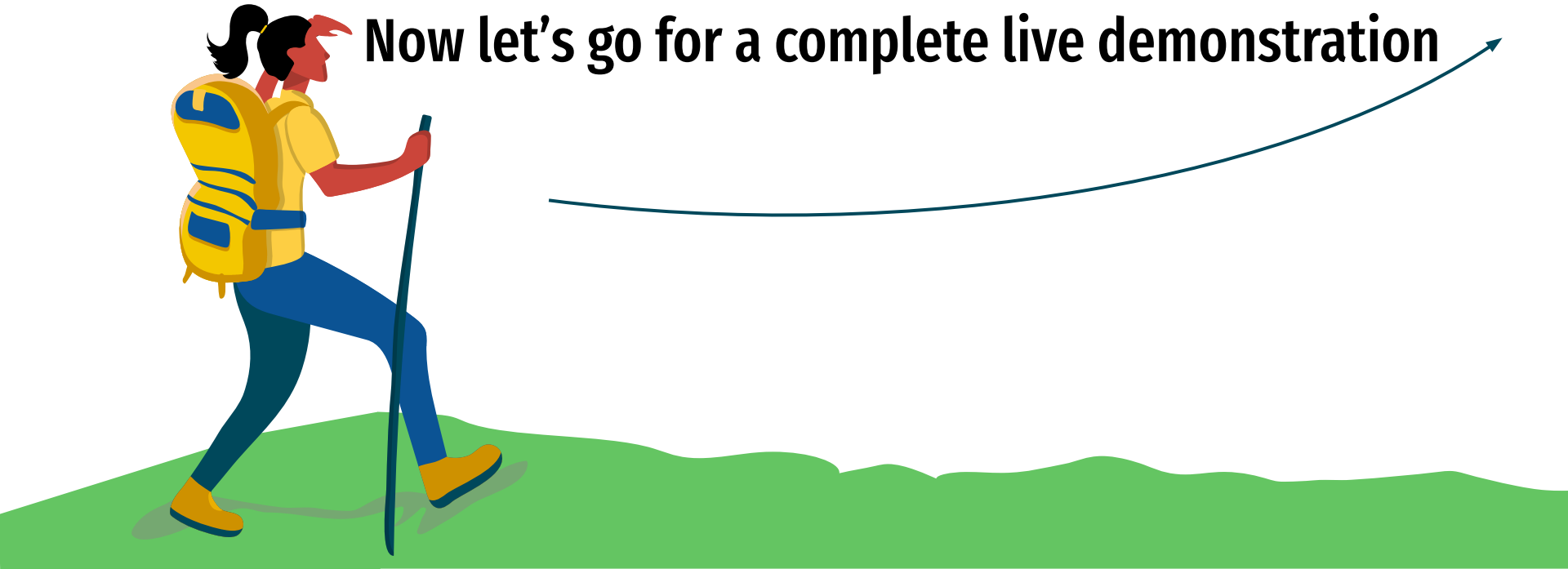
Implementation - Hardware

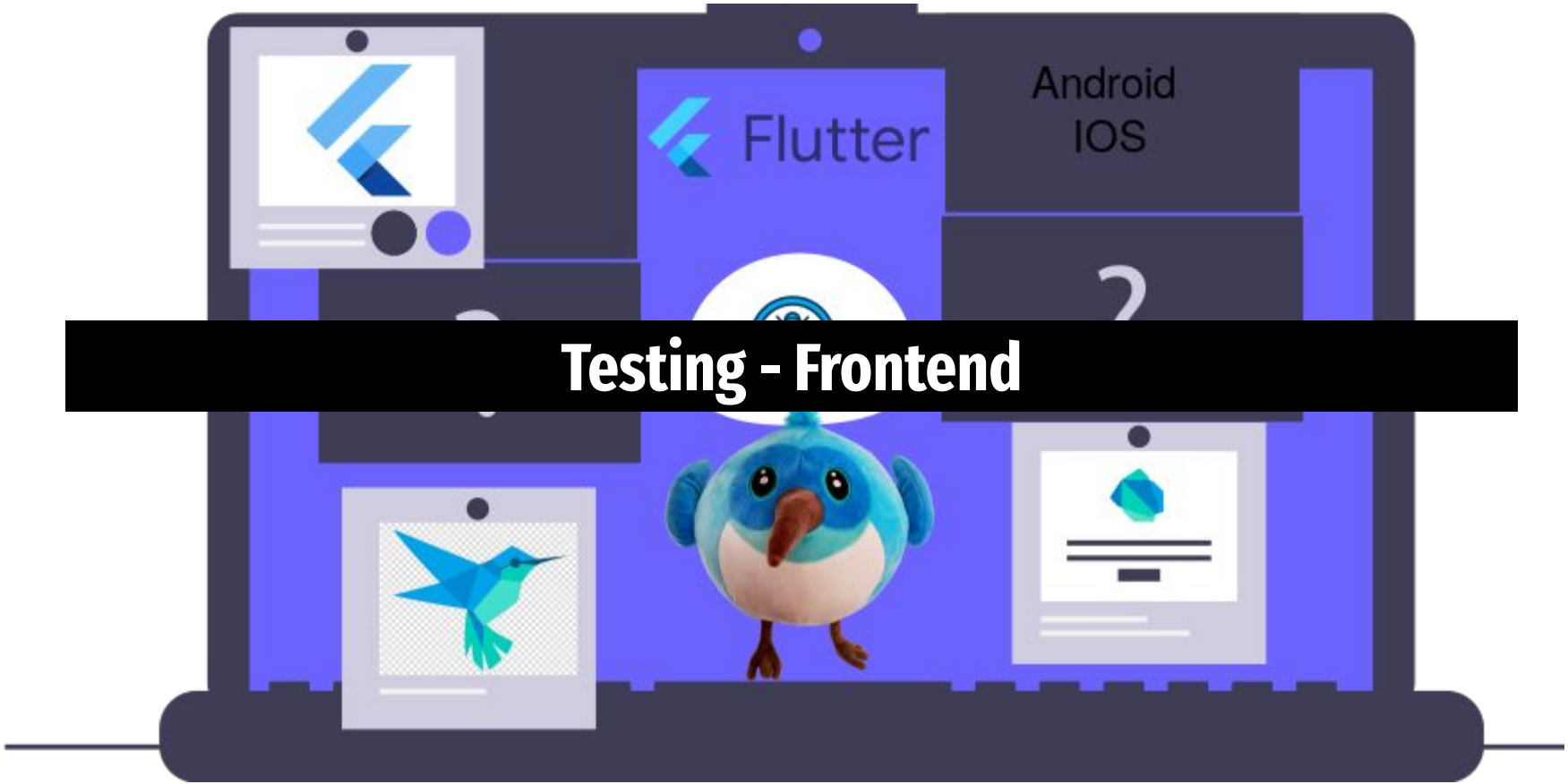


Circuit Diagram



Now let’s go for a complete live demonstration





Testing - Frontend

Testing - Frontend

- What is tested ?

  - all the input fields are tested

  - all the correct error messages are thrown

  - app is working for all the screen sizes

- Why Testing ?

  - to make sure all the components on the screen are working

  - to make sure correct errors are shown to the user

- How to test ?

  - using Flutter in-built testing library

Testing - Frontend

- Results and Findings

  - For some screen sizes the layout was not correct and it was fixed

  - Inputs fields were not checked for errors locally in the phone and it was fixed

Testing - Backend

home.test.js

code > backend > test > home.test.js > path

```
IshanFernando, 3 weeks ago | 1 author (IshanFernando)
1  const path = require('path');
2  require('dotenv').config({
3      path: path.resolve(__dirname, '../.env')      IshanFernando, 3 weeks ago • Small changes …
4  });
5  const request = require('supertest');
6  const db = require('../db');
7
8  const { SensorData } = require('../models/SensorData');
9
10 beforeAll(async () => {
11     await db.connectDB();
12 })
13 afterAll(async () => {
14     await db.disconnectDB();
15 })
16
17 describe('Root route', () => {
18     it('Server returns HTTP 200 OK + MongoDB Connected', async () => {
19         const res = await request('http://localhost:' + process.env.PORT).get('/')
20         expect(res.statusCode).toEqual(200)
21     })
22
23     it('Clearing data in db', async () => {
24         await request('http://localhost:' + process.env.PORT).get('/clear').expect(200);
25         const allData = await SensorData.findOne()
26         expect(allData).toBeNull();
```

PROBLEMS 169 OUTPUT DEBUG CONSOLE TERMINAL GITLENS

node - test

```
ridmajayasundara@My-MacBook-Pro test % npm run test

> weather-analytics-and-travel-path-guider@1.0.0 test
> jest --runInBand

RUNS test/seed.test.js
```

main* 5↓ 0↑ 1 1 167 Live Share AVR Ln 3, Col 45 Spaces: 4 UTF-8 LF JavaScript macOS (darwin) Prettier

Testing - Backend

- What is tested ?

  - all the new API endpoints are tested

- Why Testing ?

  - To make sure old features are working when a new one is implemented

  - Check load Handling capacity and limits

- How to test ?

  - JEST (Library in java to do testing)

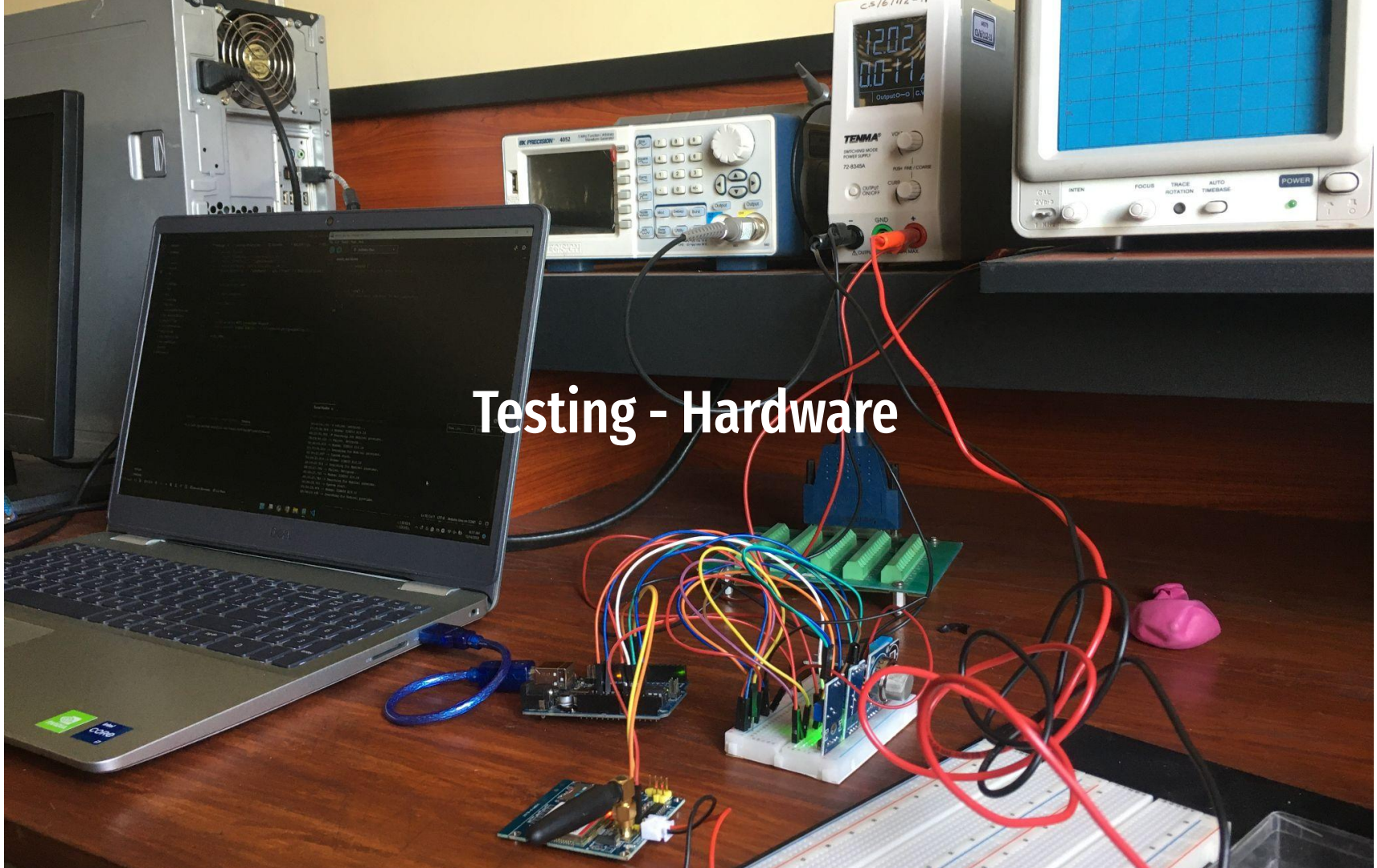
  - Artillery (Library in Node.js for load testing)

Testing - Backend

- Results and Findings

  - All the API endpoints are working as they are suppose to

  - A previously working route was not not working when a new route was added and it was fixed.


Testing - Hardware

Testing - Hardware

- What is tested ?

  - Connections to all the modules

  - Internet Connection to SIM800

  - Water proof, dustproof testing : Yet to be Done

  - Power Management

- How to test ?

  - Unit testing library in PlatformIO

  - Physical Testing on the Casing : Yet to be Done

Testing - Hardware

- Results and Findings

  - Lack of memory in the stack, had to move strings to the heap

  - can sustain continuous uptime

  - Power through the microcontroller is enough to power the GSM module

Q & A

Thank You !