Sri Lanka Institute of Information Technology

IT1040 - Fundamentals of Computing

Year 1, Semester 1- 2024

BodySQL - Real-Time Health Monitoring System with Daily Insights and Alerts

Progress Report

Group ID – P22

IT Number	Name
IT24103885	Senarathna Y.M.C.S
IT24104140	Bandara P.M.A.N
IT24102160	Gangamini A.H.A
IT24103862	Dias A.A.L
IT24102625	Fathima Minu R
IT24104019	Sajeebarath S

BodySQL is a distance health care management tool we are building for this assignment, and it helps individuals in rural areas by providing them with insights into their body temperature and heart rate.

Project Breakdown and Task Assignment

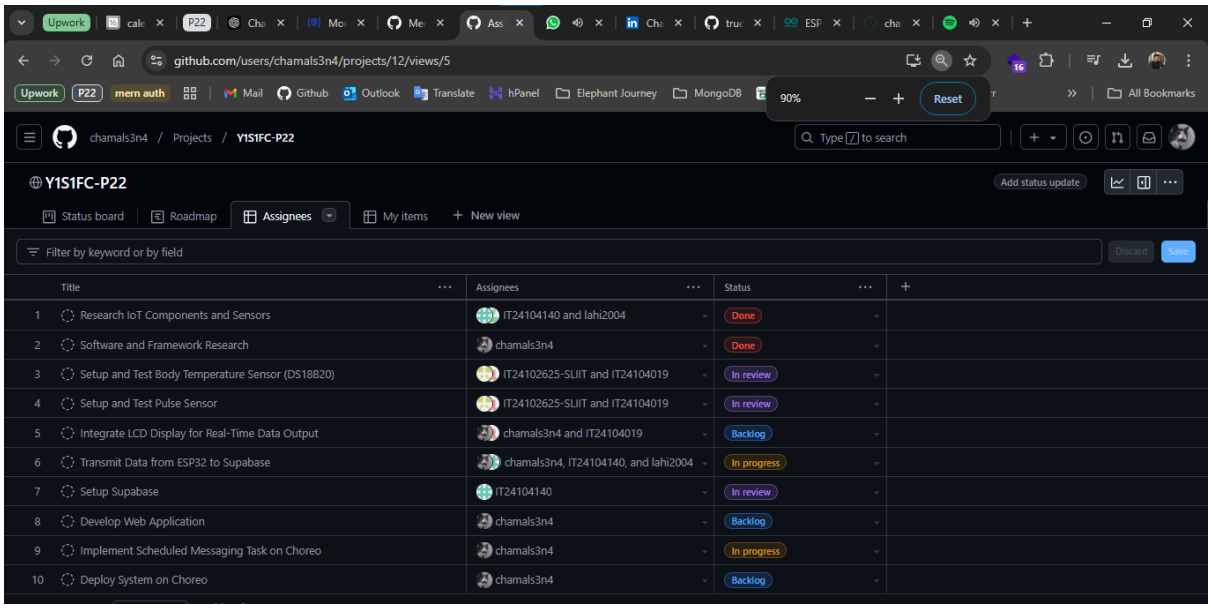
After proposing the project, the first thing we did was divide this project into smaller, manageable subparts. Then we assigned each task to our team member(s) based on their expertise and interest.

To make seamless collaboration and progress tracking, we:

1. Create a GitHub Repo to host the project
2. Setup a GitHub Project Board using Kanban Method to organize task into columns such as **TODO**, **IN PROGRESS**, **IN REVIEW** and **COMPLE**
3. Create issue templates for each subtask, documenting clear instructions and goals.
4. Assigned team members to each issue, linking their GitHub accounts.

As an open-source initiative 90% of the technologies used in BodySQL are open source, and the project is fully accessible on GitHub for public review and contribution. (YES, THIS IS OPEN SOURCE)

https://github.com/chamals3n4/Y1S1FC-P22



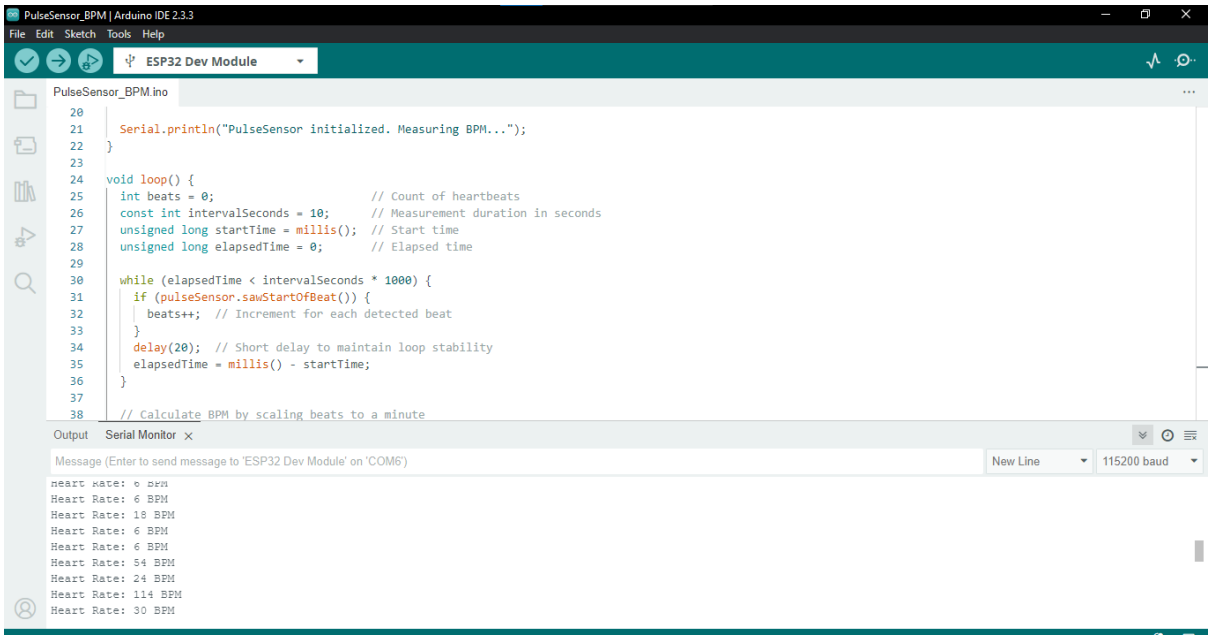
Project Progress

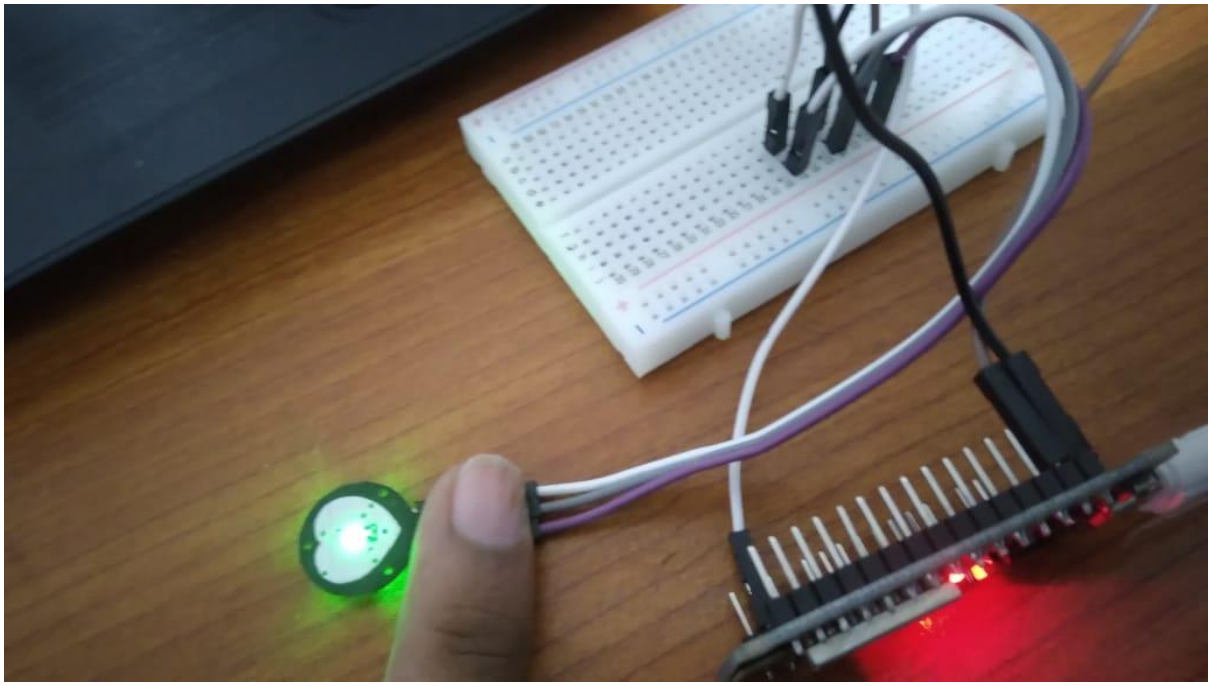
1. Hardware Research

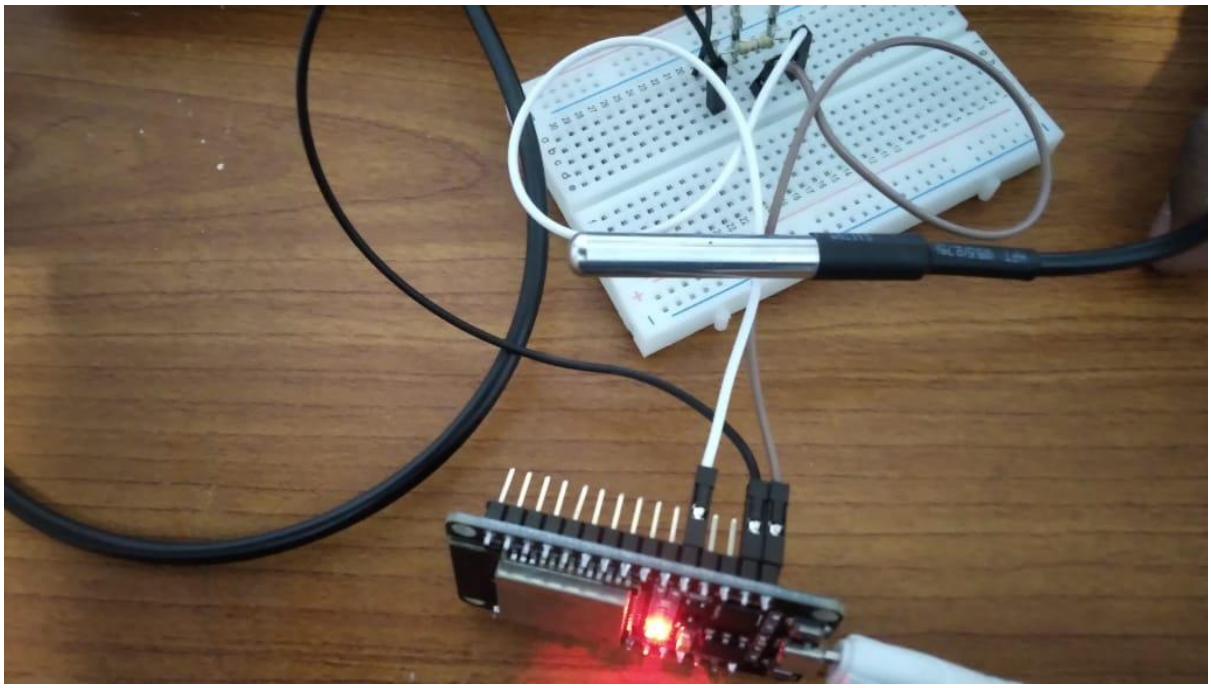
The first step was researching suitable hardware components for the project. We did research on the internet and using tools and forums like Perplexity and Arduino Forum to identify the most recommended and accurate sensors for body temperature and pulse rate measurements. Based on our findings we selected the DS18B20 and Pulse Sensor for each measurement.

2. Sensor Setup and Testing

We Connected both sensors individually and tested it accuracy under different conditions and, they provided stable reading to the serial monitor







3. Data Transmission to Supabase

As mentioned earlier, we successfully tested the body temperature and pulse sensors **individually** to verify their functionality. However, we have not yet integrated both sensors with the **ESP32** for real-time data transmission to Supabase.

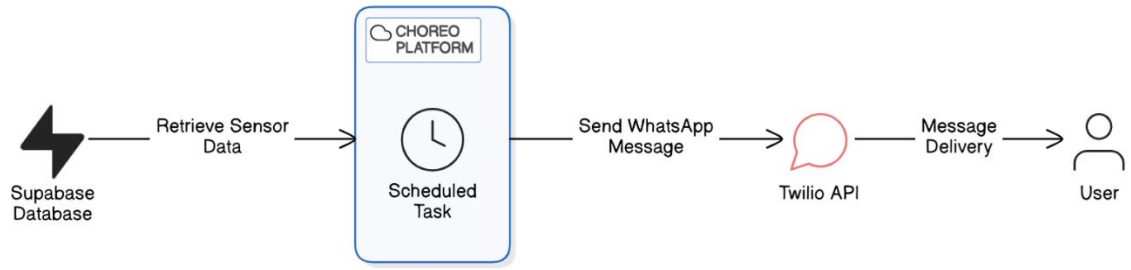
To verify the data transmission flow works, we added some mock data to Arduino IDE and then those data was transmitted to the Supabase database using the **ESPSupabase** Library through a REST API. After doing this test, we verified that the data transmission works well.

ESPSupabase C++

```cpp
// Import WiFi and ESPSupabase Library
#include <WiFi.h>
#include <ESPSupabase.h>

// Add you Wi-Fi credentials
const char* ssid = "YOUR SSID";
const char* password = "YOUR PASSWORD";

// Supabase credentials
const char* supabaseUrl = "YOUR SUPABASE URL";
const char* supabaseKey = "SUPABASE ANON KEY";

Supabase supabase;

void setup() {
  Serial.begin(115200);

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to Wi-Fi...");
  }
  Serial.println("Wi-Fi connected!");

  // Init Supabase
  supabase.begin(supabaseUrl, supabaseKey);

  // Add the table name here
  String tableName = "healthdata";
  // change the correct columns names you create in your table
  String jsonData = "{\"heartrate\": \"70\", \"bodytemp\": \"37\"}";

  // sending data to supabase
  int response = supabase.insert(tableName, jsonData, false);
  if (response == 200) {
    Serial.println("Data inserted successfully!");
  } else {
    Serial.print("Failed to insert data. HTTP response: ");
    Serial.println(response);
  }
}

void loop() {
}
```

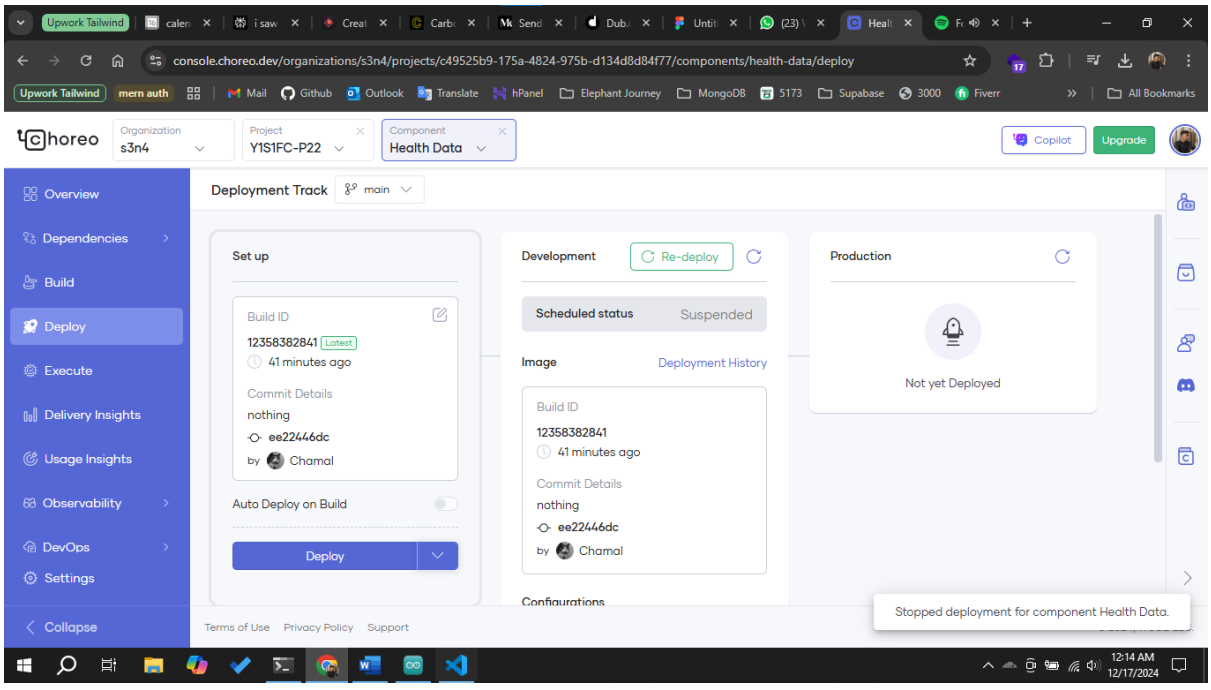
Scheduled Task and WhatsApp Messaging

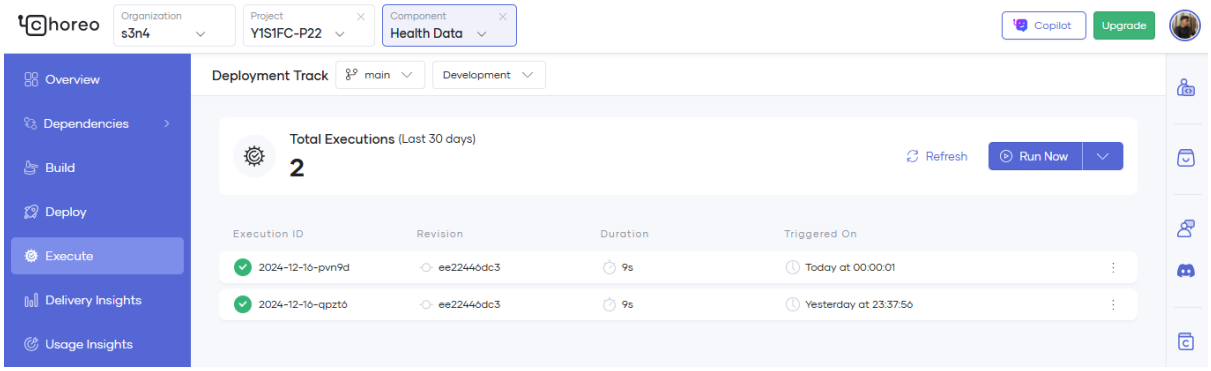
Another feature we tested and confirmed was Schedule Message. In this user can get the data he tested out using sensor through a WhatsApp message.



**In the proposal we said that we are going to send this data through an email service, but after doing some UX stuff, we decided to go with WhatsApp because most people don't check emails regularly.*

For this also we used a similar approach we did in the above task, that is we fetch the mockup data, that we stored on our database, and using that data, we try to send WhatsApp messages with Twilio Message API. After that we deployed this into **WSO2 Choreo** as a **Schedule Task Component**, and it Successfully Sent the data to Whatsapp.





These are the stuff we completed until this stage, and now here are the things we need complete to go for the **FINAL STAGE**

- Connecting **16x2 LCD Display** to ESP32 Microcontroller and Show Real Time Health Data.
- Connecting Both the sensors to the Microcontroller and transmit the actual data and send schedule messages.
- Building the Web Application (We completed the designing of the UI for this).
- Integrating LLM /w Schedule Message. For this we are going to user either Grok (**Elon's OPEN-SOURCE MODEL**) or GPT (**Sam's CLOSED MODEL**)

Here are some articles, and code lines wrote along with this project by S3n4

- How to Send Data from ESP32 to Supabase – https://chamalsena.medium.com/send-data-from-esp32-to-supabase-bb296bdb8e68

- Implementing Choreo Schedule Task - https://github.com/chamals3n4/choreo-scheduled-whatsapp