Department of Electronic & Telecommunication Engineering

University of Moratuwa

EN3250 : Internet of Things

Cal Track

Personal Calorie Tracker

Project Report

Name	Index number
1. Dhinesh Suntharalingam	170133B
2. Jathurshan Pradeepkumar	170248G
3. Mithunjha Anandakumar	170389M
4. Vinith Kugathasan	170654X

This report is submitted in partial fulfillment of the requirements

For the module EN3250: Internet of Things.

20th June 2021

Table of Contents

Introduction

Project Overview

A healthy diet is essential for good health and nutrition. It protects you against many chronic noncommunicable diseases, such as heart disease, diabetes and cancer. Eating varieties of foods and regulating the consumption of calories is essential for a healthy diet. Measuring the calorie intake would help to lose weight by giving an overview of what's eaten each day. This can help to identify eating patterns and to modify it by keeping us on track to reach health goals. But measuring the total calorie intake through each meal is a tedious task. Implementation of an Iot based solution to track the daily food/meal intake and monitor the total calorie consumed based on the food consumption would be an ideal solution to track the daily calorie intake.. In our application, the users can simply add all the contents of their meal from time to time and check their calorie intake for the day whenever they need it. Furthermore, sudden inquiry on the calorie of a specific food can also be checked. This IoT based solution, **Cal-Track** will be useful in tracking personal calorie intake in real time in a user-friendly manner. Cal-Track can be used to add the data on every meal intake, check the calorie intake of each meal, total calorie consumption per day and the overall progress of the tracking period. This IoT platform acts as an all in all solution for tracking calories in order to lead a healthy life.

Objectives

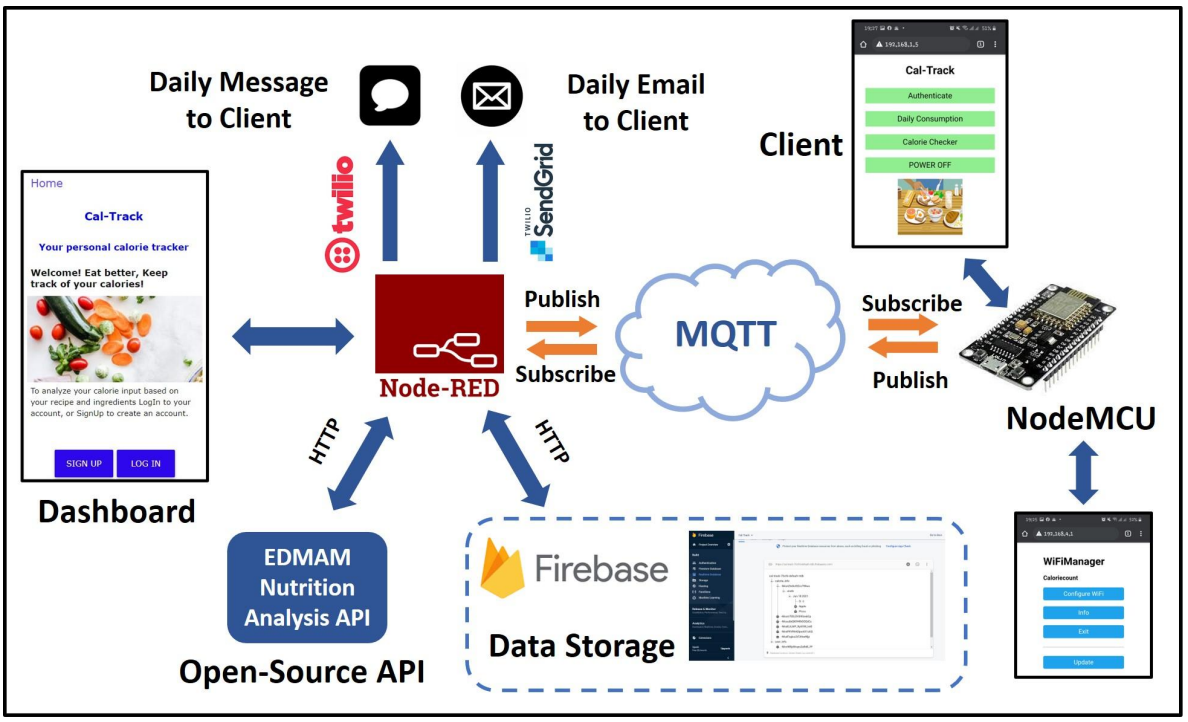
- Implementing a user friendly dashboard in Node-RED to create and log in to their personal profile.
- Presenting the calorie intake value for the given ingredients, using the Dashboard.
- Store all the data acquired in the database for identifying the trends and also for future purposes.
- Two step verification - authenticating the Node-RED account via the local webpage hosted by Node MCU in addition to providing login credentials.
- Accessing the personal profile through the local web pages hosted by NodeMCU.
- Sending the daily summary on the calorie intake as an email to the user, at the end of each day.

Scope

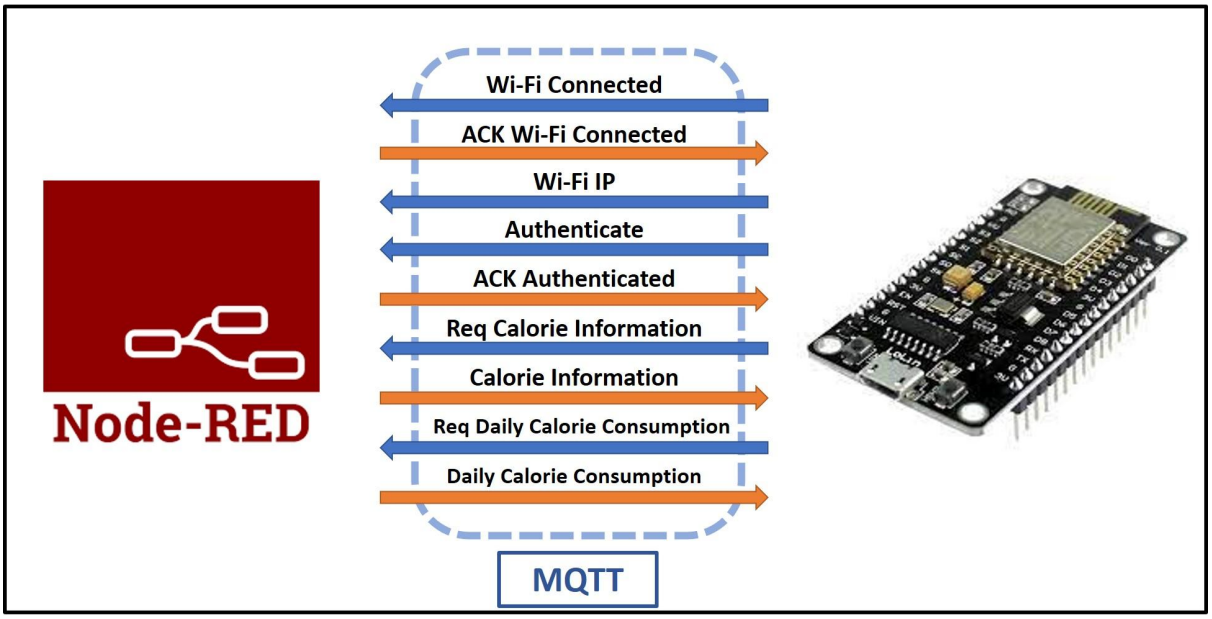
The user has to create a personal profile using the dashboard in Node-RED which can be used to add the food intake details and to track the calories consumed. The personal profile has details on calories of each food, daily consumption, total consumption (in a bar chart), and other details. At the initial stage of logging in, the account created on the Node-RED has to be authenticated using the device ID provided with the Node MCU. Later the individual meal intake information can be updated using the Node MCE local web page. Apart from that, calories of individual food items can be checked using the local web page. An open source API 'EDAMAM Nutrition Analysis API' is used to acquire all the information needed to calculate the calories of each type of food item. All the communication between the Node-RED and the Node MCU happens through the test.mosquitto.org (MQTT) server. The information collected is stored in the database 'Firebase'. It is stored user-wise with time stamps to carry out all the required processes.

Project Architecture

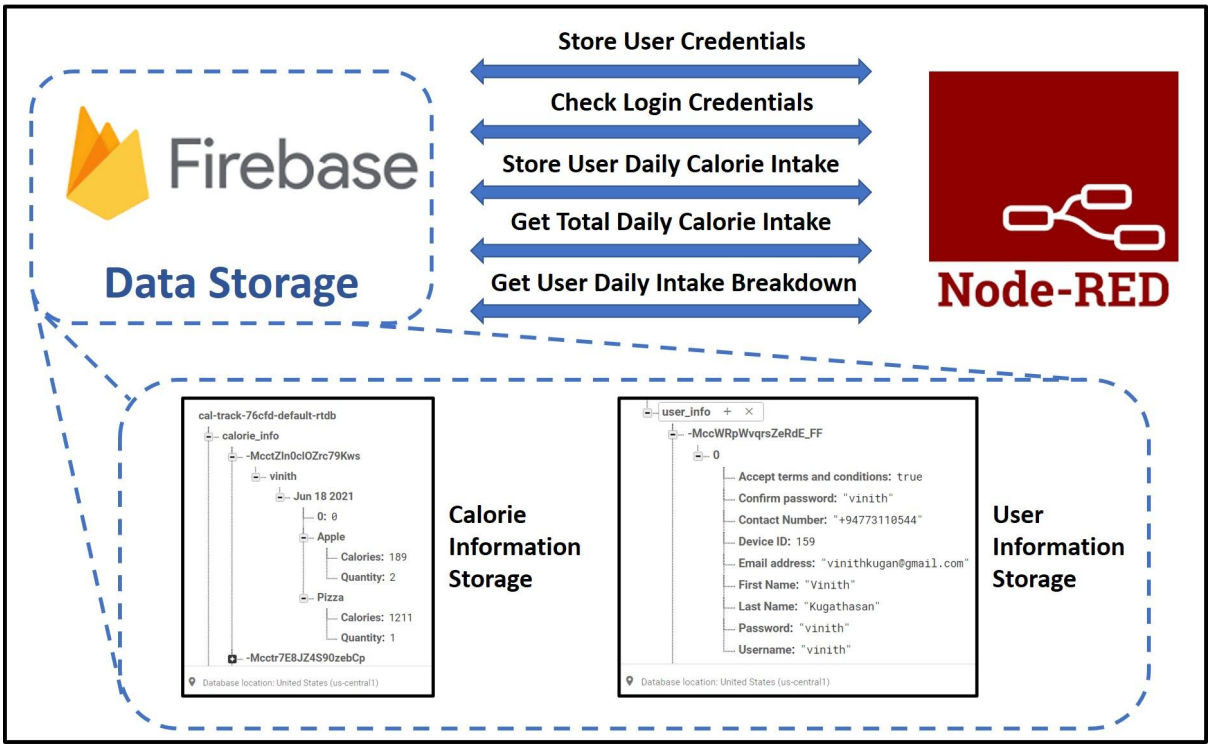
Overall Architecture



Node RED and NodeMCU



Node RED and Firebase Architecture



Methodology

Node red Dashboard and Firebase

○ SignUp & LogIn

A Dashboard was designed using Node-RED to visualize and manage the user's calories intake. In the Home page a short description about the dashboard is given and the users were given options to either directly login if they have an account already or sign up. A new user has to sign up before checking their calorie count using our "Cal-Track" dashboard. During the sign up the user has to provide his/ her full name, username(unique for the user), device ID (unique for the NodeMCU node), email address, contact number and password. These details are stored in the firebase database in realtime. The username and the passwords provided by the user in the login page are cross checked with the firebase database before allowing them to use the Cal-track dashboard.

○ Device ID

Device ID - The Device ID is specific to the NodeMCU device. The Device ID details are by the user during the signup itself. The details are stored at firebase and continuously used during the communication between NodeMCU and Node-RED dashboard. The messages published and subscribed carry the device ID as a header to the message. Since many users may access the dashboard using different NodeMCUs the device ID is used to identify and classify the messages published by different NodeMCUs. Similarly the messages published by the Node-RED also carry the device ID as the header, so that the NodeMCU devices can recognize the messages which are specific for them. This enables multiple users to access and use the dashboard at once without mixing the messages.

○ Authenticating using the NodeMCU

NodeMCU requires connection to the local/home wifi network to allow transmission and retrieval of information. If the device gets connected to the local/home wifi network, the NodeMCU publishes a message through the MQTT (<Device ID> WiFi is connected). The user is directed to the authentication page if the subscribed message is received. Instructions tab and authentication tab are designed to guide the users to connect to the WiFi and authenticate using the NodeMCU terminal respectively.

Two step verification was used in the dashboard to ensure the authorized accesses. Any device that is connected to the user's home network by the above mentioned method can be used for the authentication. Instructions for authentication are provided in the dashboard. The router address is published by the NodeMCU through MQTT server and the dashboard dynamically updates when the message(<Device ID> <IP address>) is received through MQTT server.

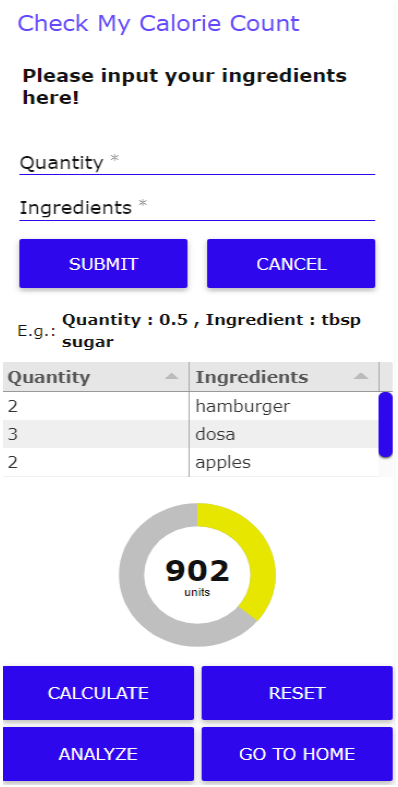
1. Go to the Router address (e.g.: 192.168.1.5)
2. Press Authenticate (button on the local webpage)

When the authenticated button is pressed a message will be published by the NodeMCU through MQTT, where the message will contain the device ID and the message (e.g: <Device

ID> Authenticated), and Node-RED will also publish a message (e.g : <Device ID> Device is authenticated). By this means the user and the device will be verified and then the user will be allowed to use the dashboard as a calorie tracker. The user will be directed to the 'check my calorie count' tab only if the subscribed message is received.

- ## Calorie tracker

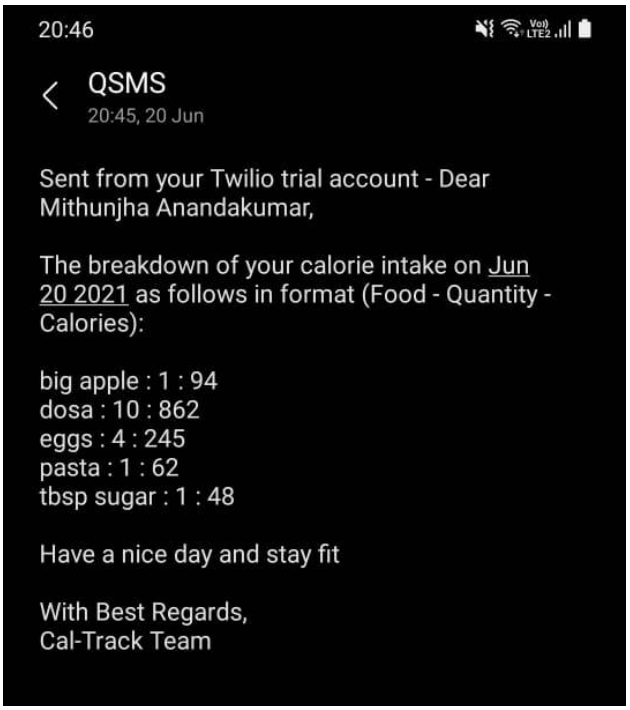
The nutrition analysis API of EDAMAM is used in our implementations. This API takes the food ingredients as input and gives the calorie of each ingredient/meal. In addition total weight, breakdown of nutrients and cautions are also provided by this specific API. In our implementations the calorie count is only considered. In the 'check my calorie count' tab the user can input his/her meal with quantity (e.g: 1 pizza, 1 hamburger, 0.5 tbsp sugar, 5 carrots, etc). When the submit button is pressed, the user's inputs are displayed in a table, and an HTTP request is made to the API to obtain the calorie count of the input of the user. The cancel button clears the Quantity and ingredient fields of the form. If the user inputs an incorrect ingredient/ food then the table will not update. After entering the ingredients the user can press the "calculate" button to calculate the total calories count of his ingredients on the table. The "Reset" button is used to clear the table and calorie count displayed. The inputs of the user are synced to the firebase database in realtime.

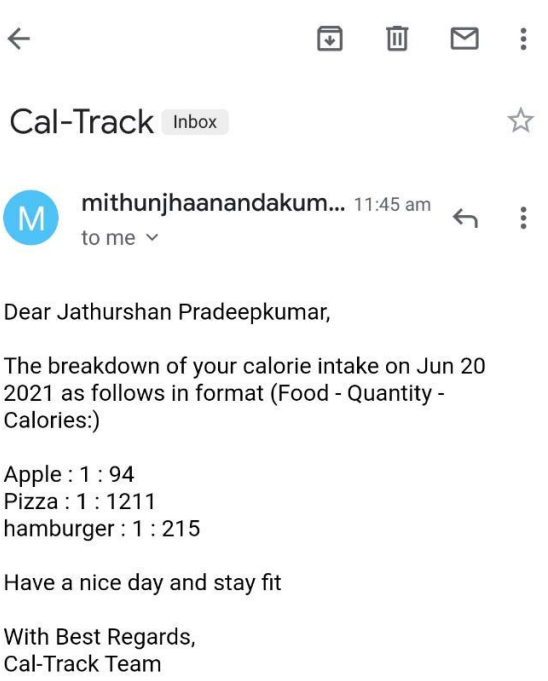


- ## Daily Calorie Intake

With the use of firebase database the daily calorie intakes can be logged and displayed to the user whenever requested (pressing analyze button). The analyze button in the 'check my calorie count' tab direct to the next tab where the chart of daily calorie intakes is displayed.

The email address and Contact number of the user is collected during the signup and used to send a detailed report of the user's daily calorie checks/inputs as an email and SMS during the end of the day (daily 11.55pm).





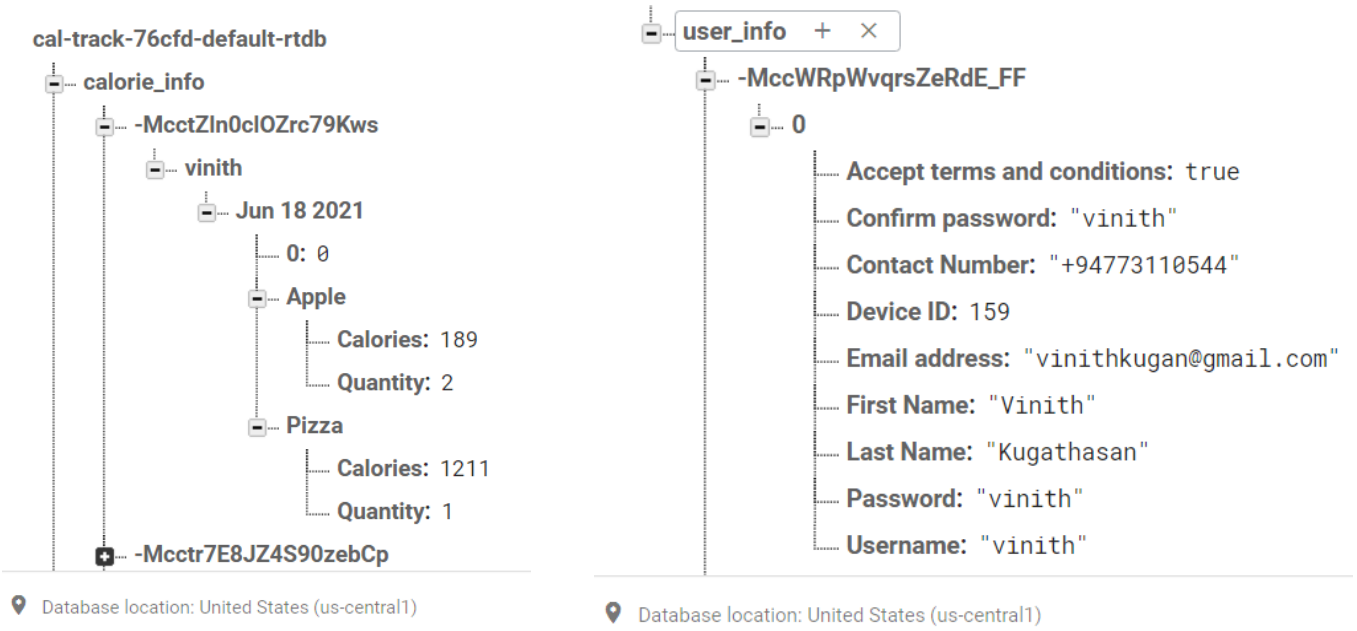
- **Calorie Checker : Services to NodeMCU**

Unlike the calorie tracker in the Node-RED dashboard, the NodeMCU hosts a local web page to display the calories of a given ingredient. Once the Node-RED receives the quantity and name of the ingredient published (<Device ID> <quantity> <name of the ingredient>)by NodeMCU through the MQTT server, the HTTP request is made to obtain the calorie count of the ingredient. Then the calorie count is published (<Device ID> <calorie count>) via MQTT. Similarly if the NodeMCU requests for daily calorie count, then the Node-RED publishes the date and total calories for the day obtained from firebase, as a message (<Device ID> <total calories for the specific day>) and publishes to NodeMCU.

- **Firebase - Database (in and out)**

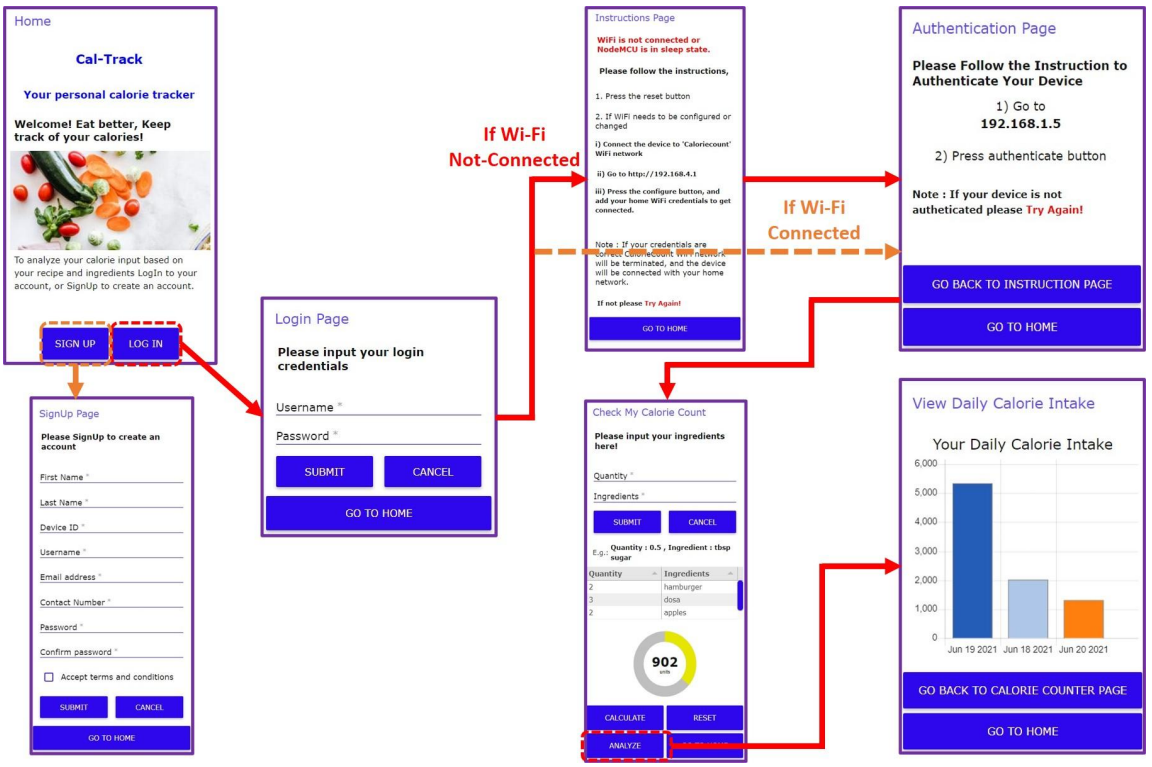
The Firebase-real time database was utilized in our application as storage for calorie tracking. The database consists of two sub-databases, which are as follows:

- User Information Database:
  - Information collected from the user while signup, and stored in this database.
  - The data from this database is utilized for user authentication.
- Calorie Information Database:
  - Stores the information about the food consumed by the user, its quantity and calories for each day separately.



The communication between the Firebase database and Node-RED was done through HTTPS. The information stored in the user information database was utilized for user authentications and to retrieve information about the user to send an email. The calorie information database was utilized to store the information given by the customer each day. This database gets updated real time, when the customer enters his/her food intake. This database is used to construct an email everyday containing a breakdown of the user's calorie intake for that day. It provides daily calorie consumption information to the users, when requested and supports visualization of the calorie intake patterns.

- **Flow of Node-RED Dashboard**



NodeMCU

○ WiFi SSID and Password credentials

NodeMCU requires connection to the local/home wifi network to allow transmission and retrieval of information. To facilitate the user to configure or change the wifi network, initially NodeMCU provides soft AP and acts as a wifi server of the name “Caloriecount”. User has to connect to that network to provide the credentials of the local wifi network. If the provided credentials are correct, NodeMCU gets connected to the corresponding network and this process was carried out using the WiFiManager library.

○ User authentication

In the Node-red dashboard, users will be asked to provide username and password credentials to login. Once provided, the dashboard requires the user to authenticate using the hosted authentication webpage (local) as a part of two step verification. Once the user completes the authentication by clicking the “Authenticate” button, NodeMCU publishes this to the Node-red dashboard through MQTT server. Also, NodeMCU receives information on whether the authentication has been received by the dashboard or not, through MQTT subscriptions, and makes corresponding changes to the authentication status.

○ Hosting local web pages

NodeMCU was also used to provide the user with a set of local web pages that can be accessed through the local ip address . These web pages were hosted by NodeMCU to all the devices connected in the local network via http transmission. CSS was used to provide styles while HTML was used to add contents accessible to the user, during the web page development. The assigned capabilities of the web pages include user authentication, calorie checker, daily consumption and power on/off features.

○ Calorie checker

Unlike the calorie tracker in the Node-red dashboard - which keeps track of the users’ calories, NodeMCU hosts a local web page to display the calories of a given ingredient. Once the user provides the quantity and name of the ingredient, it will be published to the dashboard via MQTT server. In the dashboard, the amount of calories will be obtained from the nutrition analysis API of EDAMAM and NodeMCU receives the information through MQTT server and displays it in the web page.

○ Daily Consumption

SImilar to the calorie calculator, users can also obtain the amount of calories consumed in that particular day. For this process, the Node-red dashboard retrieves the information from Firebase and transmits to the local webpage through NodeMCU.

- ## Power Conservation

As an IoT application, power conservation is vital in the implementation. To reduce the ON time of NodeMCU, deepsleep mode of indefinite time interval is enabled during the following three instances,

1. When the device is not connected to a WiFi network for 5 minutes.
2. When the user is inactive for 5 minutes.
3. When the user powers off the device through the corresponding web page.

The user can exit the deepsleep mode by pressing the reset button. The same approach is employed during WiFi disconnections to reconnect to the network.

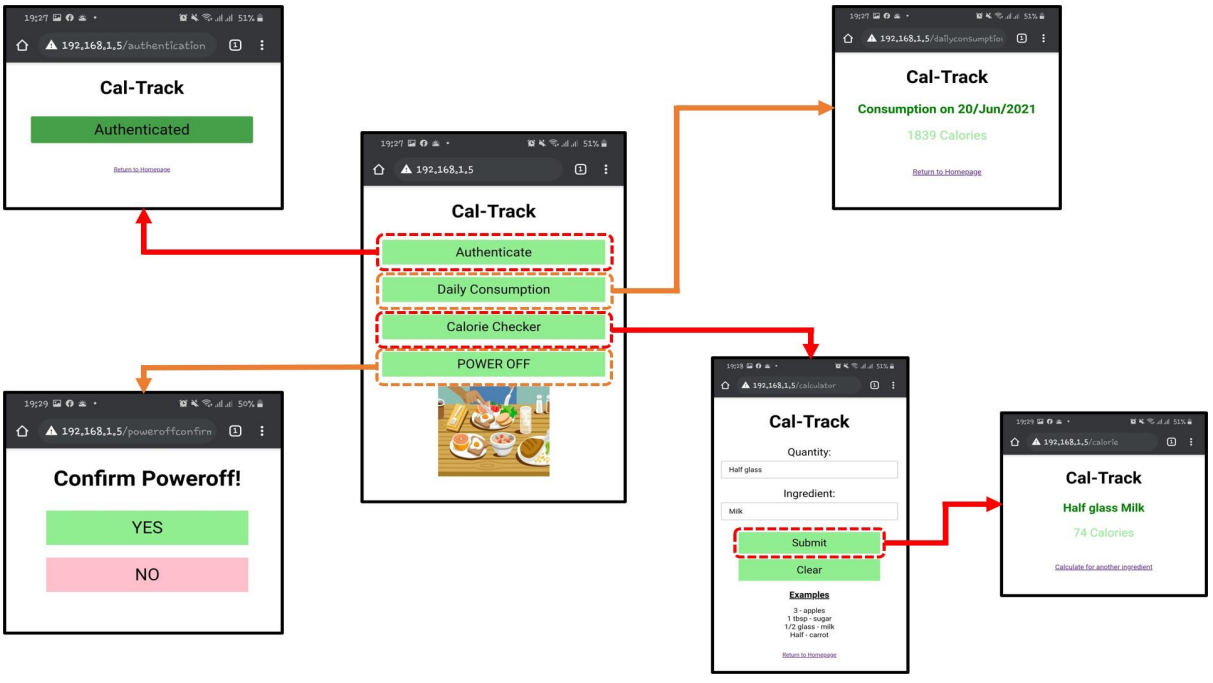
- ## Device ID

Each NodeMCU will have a unique device id, which will be provided to the user. The respective values of device ids will be hardcoded in the NodeMCU as well. When signing up in the Node-red dashboard, the user will be asked to provide the device ids, through which their details will be linked to the device ids. In all the transmissions from or to the Node-red via MQTT server, device id will be appended to the messages, to sort out the messages relevant to the user.

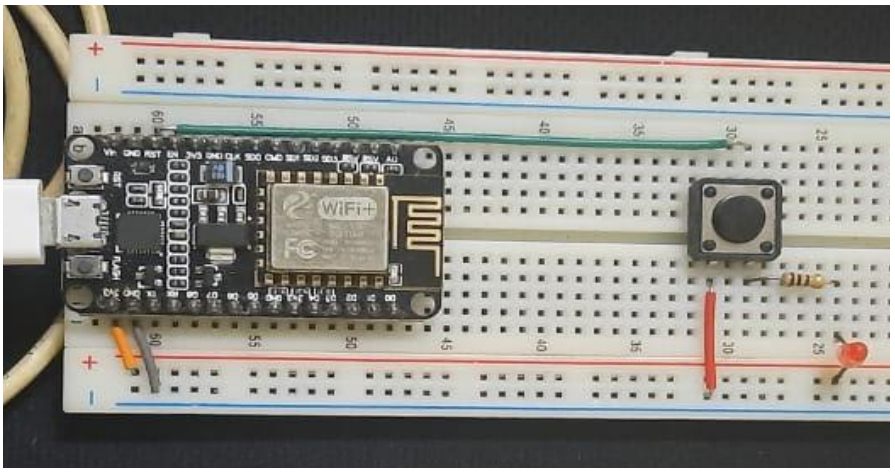
- ## Services required by NodeMCU

NodeMCU requires the service from Firebase for retrieval of daily calorie consumption information. Also it requires the external API of EDAMAM to check the calorie values of the given ingredients.

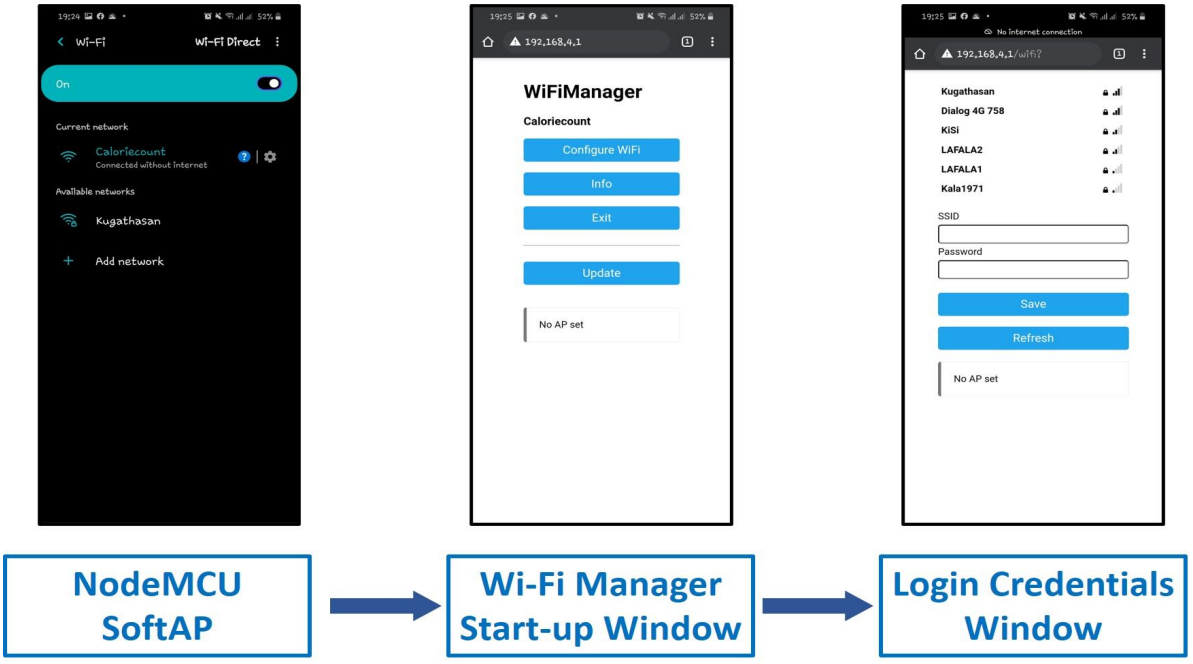
- ## Flow of the web pages hosted by NodeMCU



- **Deep sleep external wakeup configuration**



- **Flow of automatic acquisition of WiFi credentials**



Conclusion

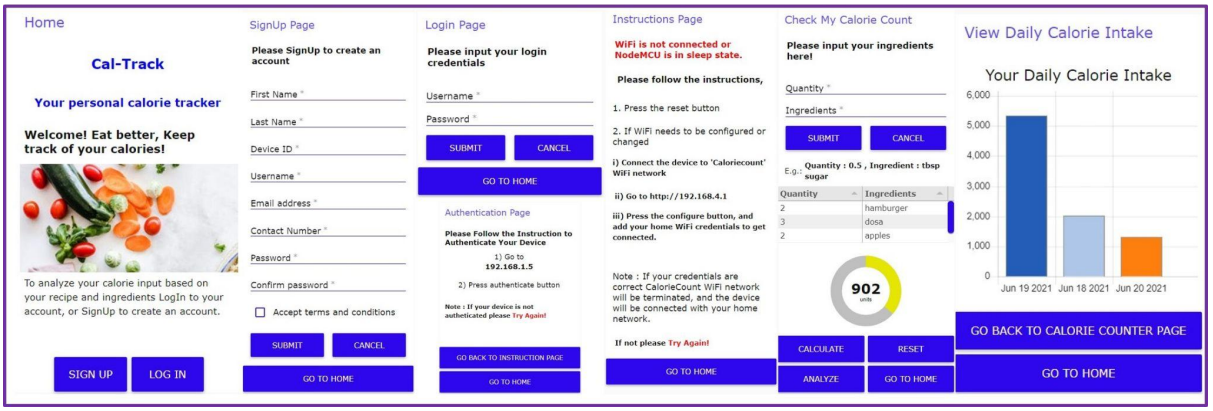
We have successfully implemented an IoT based solution to track personalized calorie intake, using Node-RED dashboard, NodeMCU and Firebase. Cal-Track can be used to add data on every meal intake, check the calorie intake of each meal, total calorie consumption per day and the overall progress within the tracking period. We have tested the performance of our implementation through 2 users with two different network locations, trying to login to the dashboard and access features simultaneously and it was flawless.

We tried other free APIs to increase the features of the system, such as Pushbullet and Twilio. Among these, Pushbullet requires the user to install another application while Ringcentral. We didn't implement the feature of sending an SMS on the daily calorie intake to the users since the Twilio API free version has limited the number of receivers to one.

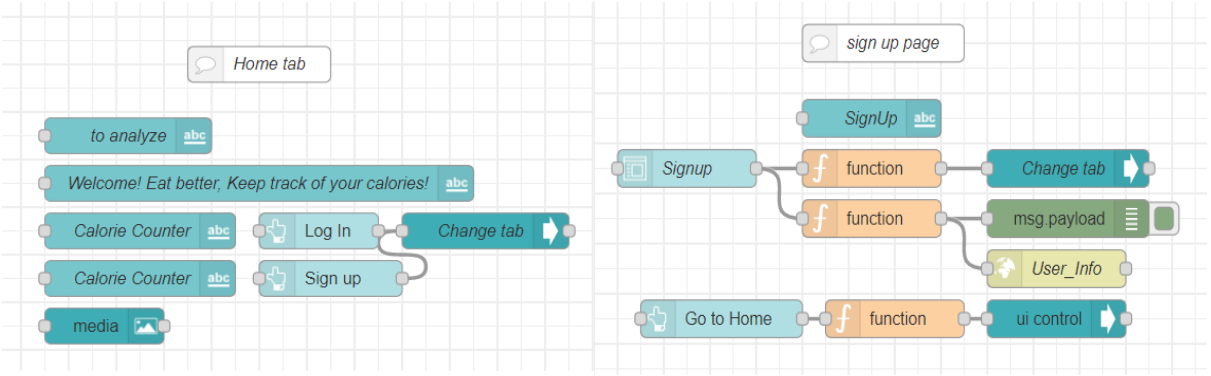
The device ID provided with the NodeMCU is used as the unique key to differentiate each user. HTML pages and forms are hosted locally in the Node MCU for the user to navigate through the authentication phase and applications. A deep-sleep mode is implemented on the NodeMCU as a power conservation mechanism from which the device can be woken up using an external switch attached to the NodeMCU. Our device implementation focuses on single user per device. i.e. With one device, only one user can keep track of his/her calorie intake. Improvements to our implementation include features such as multi user per device, hosting public web pages and utilizing messaging APIs.

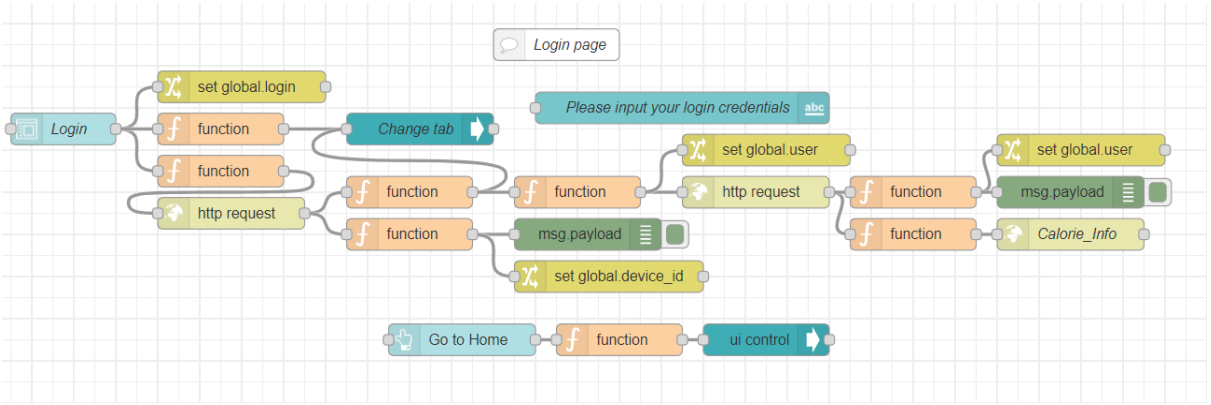
Annexes

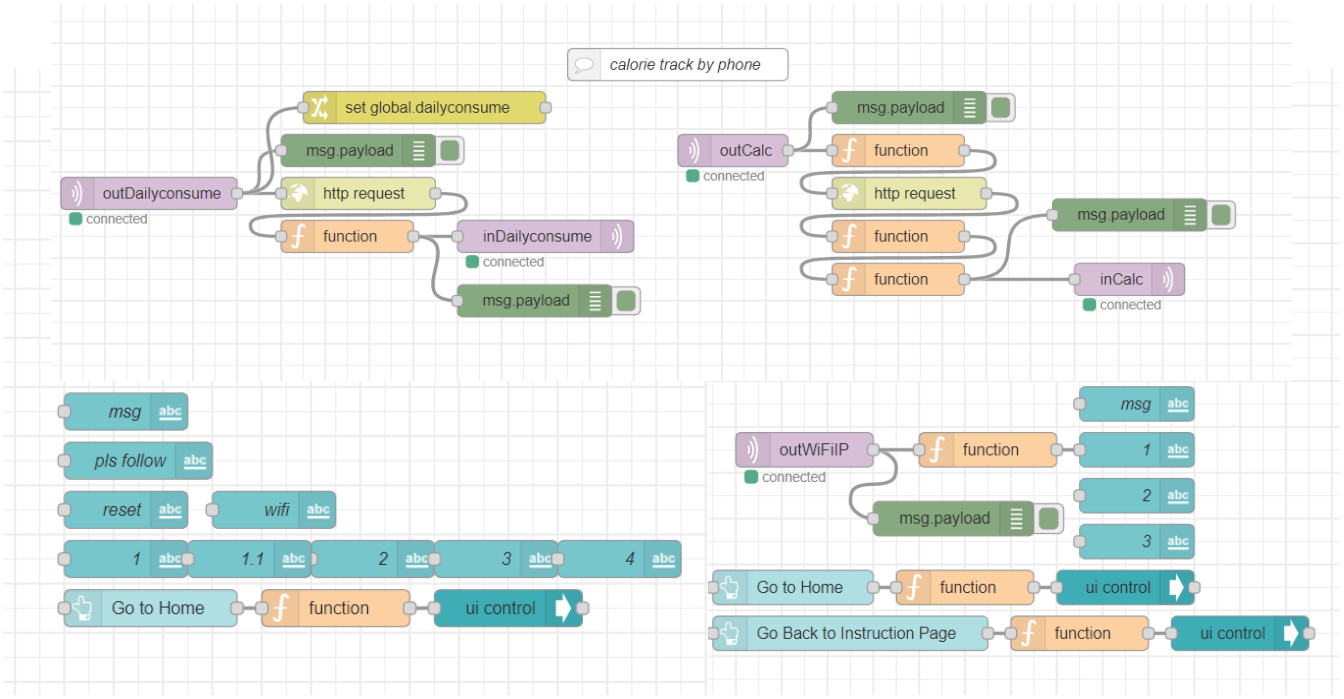
- ## The Dashboard

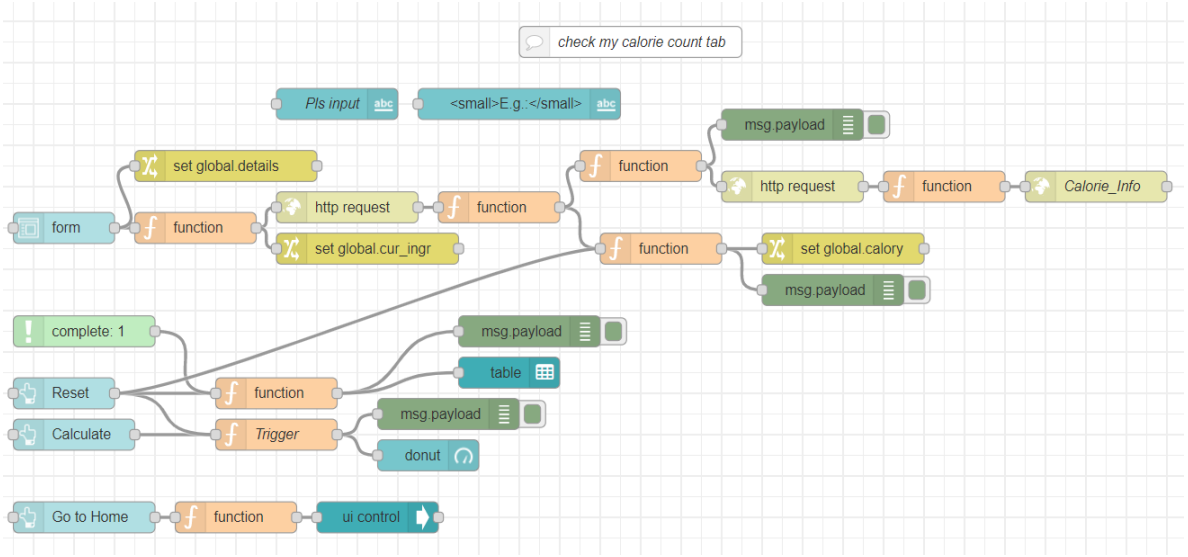


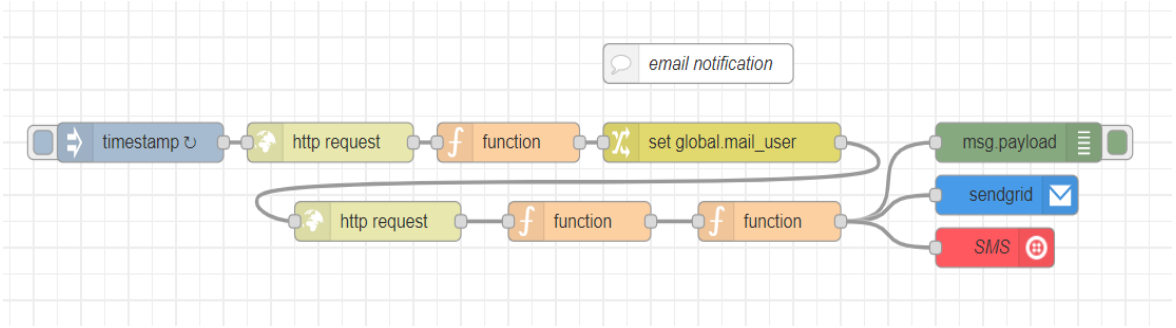
- ## The NodeRed Flow






calorie track by phone
set global.dailyconsume
msg.payload
outDailyconsume
connected
http request
function
inDailyconsume
connected
msg.payload
outCalc
connected
msg.payload
function
http request
function
function
msg.payload
inCalc
connected
msg
pls follow
reset
wifi
1
1.1
2
3
4
Go to Home
function
ui control
msg
outWiFiIP
connected
function
1
msg.payload
2
3
Go to Home
function
ui control
Go Back to Instruction Page
function
ui control


check my calorie count tab
Pls input
<small>E.g.:</small>
set global.details
form
function
http request
set global.cur_ingr
function
function
msg.payload
http request
function
Calorie_Info
function
set global.calory
msg.payload
complete: 1
Reset
Calculate
function
Trigger
msg.payload
table
msg.payload
donut
Go to Home
function
ui control


email notification
timestamp
http request
function
set global.mail_user
msg.payload
sendgrid
SMS
http request
function
function