**ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ**

Diagram

Description automatically generatedΑΝΑΛΥΤΙΚΑ:

Στο Hive MQ υπάρχουν 4 topics πάνω στα οποία μεταφέρονται τα δεδομένα από αισθητήρες -> βάση δεδομένων -> LEDs. 2 topics για να μεταφέρουμε περιοδικά τις θερμοκρασίες από τους αισθητήρες στη βάση και 2 topics για να μεταφέρουμε τις εντολές on/off από το backend στα LEDs.

Συγκεκριμενα διαβαζουμε τις τιμες από τους 2 αισθητηρες ανα δυο δευτερολεπτα και στις 5 τιμες αποθηκευουμε τον μεσο ορο στο database αποφευγωντας πιθανα spikes ή και λανθασμενες μετρησεις.

Diagram

Description automatically generatedΠαρακάτω φαίνονται τα API endpoints και σε ποια controls ανταποκρίνονται και πως.

Συνοπτικά στο frontend υπάρχουν 2 επίπεδα ελέγχων. Το πιο low level κάθε δωμάτιου ξεχωριστά (2 εδώ) και ένα high level για όλο το σπίτι συνολικά.

Τα controls είναι On/Off για το σπίτι και κάθε δωμάτιο χωριστά, o χρονοδιακόπτης για το σπίτι και κάθε δωμάτιο χωριστά. Η ρύθμιση θερμοκρασίας για κάθε δωμάτιο χωριστά και για το σπίτι – όταν ρυθμίζεται για το σπίτι τότε σε όλα τα δωμάτια εφαρμόζεται αυτή η θερμοκρασία. Τέλος η προβολή της Live θερμοκρασίας σε κάθε δωμάτιο από τους σένσορες, αυτή η λειτουργία προφανώς δεν είναι διαθέσιμη για όλο το σπίτι.

Η βάση δεδομένων υλοποιήθηκε με αρχεία txt και σε αυτά αποθηκεύονται οι τιμές των θερμοκρασιών.

Εδώ να σημειώσουμε ότι το Live Temp endpoint στέλνει περιοδικά τις τιμές θερμοκρασίας στο frontend και ότι τα Set Temp endpoints όταν είναι ενεργοποιημένος (state = on) ο διακόπτης τότε και αυτά στέλνουν περιοδικά τις τιμές των ζητούμενων θερμοκρασιών για να γίνει ο έλεγχος για αυτόματο on/off του διακόπτη. Τα υπόλοιπα controls υλοποιούνται όταν υπάρξει το αντίστοιχο input από το χρήστη.

**Οδηγίες Εγκατάστασης**

**Web App:**

Πρέπει να έχει εγκατασταθεί η **python** μαζί με τα απαραίτητα πακέτα: **flask**, **flask-cors, paho-mqtt**.

Για κάθε βιβλιοθήκη η εγκατάσταση γίνεται με την εντολή: pip install π.χ.: **pip install flask**

**ESP32**:

Η πλακέτα έχει προγραμματιστεί έτσι ώστε να συνδέεται σε ένα ήδη υπάρχον Wi-Fi από hotspot ενός laptop χωρίς κάποια extra εργασία.

Για να τρέξει κανονικά σε άλλες συνθήκες (χωρίς το laptop) θα πρέπει να αλλαχτεί ο κώδικας **esp.ino** και στις μεταβλητές **ssid** και **password** να μπουν αυτές του δικτύου Wi-Fi που θα συνδεθεί η πλακέτα.

Για να γίνει η αλλαγή και το compile του κώδικα πρέπει να εγκατασταθεί το Arduino IDE μαζί με κάποιες βιβλιοθήκες.

Μετά την εγκατάσταση ακολουθώντας το [link](https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/) θα εγκαταστήσετε υποστήριξη για το ESP32s.

Μετά θα επιλέξετε στα tools -> Board/Board Manager -> ESP32 -> **NodeMCU-32S**

Τέλος θα πρέπει να εγκαταστήσετε 2 βιβλιοθήκες από το Library Manager:

* PubSubClient (by Nick O’Leary Version 2.8)
* NTPClient (by Fabrice Weinberg Version 3.2)

Μετα ο κώδικας θα πρέπει να κάνει compile κανονικά με τα νέα Wi-Fi credentials.

Για να τρέξει η εφαρμογή συνδέουμε τη πλακέτα σε παροχή ρεύματος, μετά θα πρέπει να ανοίξουμε το terminal στον φάκελο: **iot-home/backend** και να τρέξουμε την εντολή: **python app.py**

Μετα στον φάκελο **iot-home/frontend** ανοίγουμε το αρχείο **index.html** και θα πρέπει να λειτουργούν όλα κανονικά.

**Χρήση Εφαρμογής**

Για την χρήση της εφαρμογής τώρα αρκεί να επιλέξουμε το command που θέλουμε και θα δούμε τα αποτελέσματα του στα LED της πλακέτας.