



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

ΠΑΝΟΣ ΧΡΗΣΤΟΣ
A.M. 18390085

Εισηγητής: Χρήστος Τρούσσας, Επ. Καθηγητής

Αθήνα, Μάρτιος 2024

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

**ΠΑΝΟΣ ΧΡΗΣΤΟΣ
Α.Μ. 18390085**

Εισηγητής:

Χρήστος Τρούσσας, Επ. Καθηγητής

Εξεταστική Επιτροπή:

**Παναγιώτα Τσελέντη, Μέλος ΕΔΙΠ
Ακριβή Κρούσκα, Μέλος ΕΔΙΠ**

Ημερομηνία εξέτασης Μάρτιος 2024

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Χρήστος Πάνος του Δημητρίου, με αριθμό μητρώου 18390085 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Βεβαιώνω ότι είμαι συγγραφέας αυτής της Διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

«Βεβαιώνω ότι είμαι συγγραφέας της παρούσας διπλωματικής εργασίας και ότι έχω αναφέρει ή παραπέμψει σε αυτή, ρητά και συγκεκριμένα, όλες τις πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, προτάσεων ή λέξεων, είτε αυτές μεταφέρονται επακριβώς (στο πρωτότυπο ή μεταφρασμένες) είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για την συγκεκριμένη διπλωματική εργασία»

Ο/Η Δηλών/ούσα



Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό των τεχνολογιών IoT. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής Χρήστος Τρούσσας, τον οποίο θα ήθελα να ευχαριστήσω. Ακόμα θα ήθελα να ευχαριστήσω την οικογένειά μου για τη συμπαράσταση κατά τη διάρκεια των σπουδών μου.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία ασχολείται με την κατασκευή μιας full-stack εφαρμογής τύπου Smart Home που διαχειρίζεται δεδομένα και παίρνει αποφάσεις, με βάση το υπάρχων σύστημα κανόνων, με τελικό σκοπό τον έλεγχο διάφορων οικιακών συσκευών, μέσω της χρήσης αισθητήρων. Στη σημερινή εποχή, όπου η τεχνολογία και οι υπηρεσίες του Διαδικτύου διαμορφώνουν έντονα την καθημερινότητά μας, επιδιώκουμε να αξιοποιήσουμε αυτές τις εξελίξεις για να δημιουργήσουμε μία καινοτόμα εφαρμογή που θα ανταποκρίνεται στις σύγχρονες ανάγκες και απαιτήσεις. Αυτή η εφαρμογή ενσωματώνει τεχνολογίες τόσο στο μέρος του χρήστη (Frontend) όσο και στο πίσω μέρος (Backend), δημιουργώντας έτσι μία ολοκληρωμένη και ενιαία εμπειρία χρήσης. Αποτελείται από μία κεντρική σελίδα όπου ο χρήστης, μέσω ενός εικονικού μενού επιλογών, αποκτά έλεγχο και μπορεί και αποφασίζει για τις συσκευές που είναι συνδεδεμένες στο οικιακό δίκτυο καθιστώντας την εφαρμογή αρκετά φιλική προς τον χρήστη. Συμπέρασμα της διπλωματικής είναι η ευκολία διαχείρισης συσκευών του Διαδικτύου των Πραγμάτων μέσω της χρήσης διαδικτυακών εφαρμογών, παρέχοντας μία βελτιωμένη εμπειρία χρήσης, εστιάζοντας στην άνεση και την αποτελεσματικότητα.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Σύστημα Κανόνων, Διαδικτυακές Εφαρμογές, Διαδίκτυο Αντικειμένων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Αισθητήρες, Fullstack, Smart Home, IoT

ABSTRACT

This thesis deals with the construction of a full-stack Smart Home application that manages data and makes decisions, based on the existing rules system, with the ultimate goal of controlling various home appliances through the use of sensors. In today's era, where technology and Internet services are strongly shaping our daily lives, we seek to leverage these developments to create an innovative application that meets modern needs and requirements. This application integrates technologies on both the user side (frontend) and the backend (backend), thus creating an integrated and unified user experience. It consists of a main page where the user, through a virtual menu of options, gains control

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

and can decide about the devices connected to the home network, making the application user-friendly. The conclusion of the thesis is the ease of managing IoT devices through the use of web applications, providing an improved user experience, focusing on convenience and efficiency.

SCIENTIFIC AREA: Rule System, Web Applications, Internet of Things (IoT)

KEYWORDS: Sensors, Fullstack, Smart Home, IoT

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	8
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	12
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ	13
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	14
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ	14
ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ	15
ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	18
2.1 ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ.....	18
2.1.1 ΙΣΤΟΡΙΑ ΚΑΙ ΕΞΕΛΙΞΗ.....	18
2.1.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ ΤΩΝ ΠΡΑΓΜΑΤΩΝ.....	19
2.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΕΛΑΤΗ – ΕΞΥΠΗΡΕΤΗΤΗ (CLIENT-SERVER)	22
2.2.1 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΠΕΛΑΤΗ – ΕΞΥΠΗΡΕΤΗΤΗ .	24
2.3 SOCKETS.....	26
2.4 THREADS	27
2.5 ΑΙΣΘΗΤΗΡΕΣ.....	28
2.6 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ.....	28
ΚΕΦΑΛΑΙΟ 3: ΜΕΘΟΔΟΛΟΓΙΑ	31
3.1 ΣΤΟΧΟΣ ΚΑΙ ΥΠΟΠΡΟΒΛΗΜΑΤΑ	31
3.2 ΠΡΩΙΜΑ ΣΤΑΔΙΑ.....	33
3.3 ΕΜΠΟΔΙΑ ΣΤΗΝ ΥΛΟΠΟΙΗΣΗ	34
ΚΕΦΑΛΑΙΟ 4: ΤΟ ΣΥΣΤΗΜΑ	35
4.1 ΔΟΜΗ ΑΡΧΕΙΩΝ ΕΦΑΡΜΟΓΗΣ.....	35
4.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΦΑΡΜΟΓΗΣ	36
4.2.1 ΒΙΒΛΙΟΘΗΚΗ FLASK.....	38
4.2.2 ΒΙΒΛΙΟΘΗΚΗ FLASK – CORS	38

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

4.2.3 ΒΙΒΛΙΟΘΗΚΗ WEBSOCKETS	39
4.2.4 ΒΙΒΛΙΟΘΗΚΕΣ THREADING ΚΑΙ ASYNCIO	39
4.3 ΕΜΠΡΟΣΘΙΟ ΜΕΡΟΣ (FRONTEND).....	40
4.3.1 ΔΕΔΟΜΕΝΑ ΘΕΡΜΟΚΡΑΣΙΑΣ	41
4.3.2 ΔΕΔΟΜΕΝΑ ΥΓΡΑΣΙΑΣ	46
4.3.3 ΔΕΔΟΜΕΝΑ ΦΩΤΕΙΝΟΤΗΤΑΣ.....	49
4.3.4 ΠΛΑΤΦΟΡΜΑ ΟΠΤΙΚΟΠΟΙΗΣΗΣ ΔΕΔΟΜΕΝΩΝ (THINGSPEAK)	55
4.4 ΠΙΣΩ ΜΕΡΟΣ (BACKEND).....	55
ΚΕΦΑΛΑΙΟ 5: ΑΞΙΟΛΟΓΗΣΗ	64
5.1 ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΗΣ.....	64
5.2 ΑΞΙΟΛΟΓΗΣΗ ΑΠΟ ΧΡΗΣΤΕΣ.....	67
5.2.1 ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ	68
5.2.2 ΑΞΙΟΠΙΣΤΙΑ.....	69
5.2.3 ΑΙΣΘΗΤΙΚΗ	70
ΚΕΦΑΛΑΙΟ 6: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ	72
6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	72
6.2 ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ	73
ΒΙΒΛΙΟΓΡΑΦΙΑ	74

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2.3: Αρχιτεκτονική client-server και χρήση πρωτοκόλλου HTTP	22
Εικόνα 4.3.1: Interface εφαρμογής (Dashboard)	41
Εικόνα 4.3.2: Διαφορετικές περιπτώσεις ενεργειών αυτόματου ελέγχου κλιματιστικού (Auto Control)	43
Εικόνα 4.3.3: Μήνυμα που λαμβάνει ο χρήστης κατά τον χειροκίνητο έλεγχο θερμοκρασίας (Manual Control)	44
Εικόνα 4.3.4: Διαφορετικές περιπτώσεις ενεργειών αυτόματου ελέγχου αφυγραντήρα (Auto Control)	47
Εικόνα 4.3.5: Διαφορετικές περιπτώσεις ενεργειών αυτόματου ελέγχου φωτιστικού (Auto Control)	50
Εικόνα 4.3.6: Τα βήματα που ακολουθεί ο χρήστης για την επιλογή επιπέδου φωτισμού (Manual Control)	52
Εικόνα 4.3.7: Διαγράμματα (graphs) που απεικονίζουν τις τιμές από κάθε sensor (Interface)	54
Εικόνα 5.1.1: Πρώτη περίπτωση χρήσης εφαρμογής	63
Εικόνα 5.1.2: Δεύτερη περίπτωση χρήσης εφαρμογής	64
Εικόνα 5.1.3: Τρίτη περίπτωση χρήσης εφαρμογής	65
Εικόνα 5.1.4: Τέταρτη περίπτωση χρήσης εφαρμογής	66

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 4.2: Διάγραμμα αρχιτεκτονικής εφαρμογής.....	37
Σχήμα 4.3: Διάγραμμα εφαρμογής Frontend κομματιού	53
Σχήμα 5.2.1: Στατιστικά αξιολόγησης λειτουργικότητας	68
Σχήμα 5.2.2: Στατιστικά αξιολόγησης αξιοπιστίας.....	69
Σχήμα 5.2.3: Στατιστικά αξιολόγησης αισθητικής	70

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 4.1: Δομή αρχείων εφαρμογής.....	36
Πίνακας 4.3: Αναφορικός πίνακας διευθύνσεων (paths) αιτημάτων στο Backend από το Frontend	62

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

IoT Internet of Things

WWW World Wide Web

HTTP HyperText Transfer Protocol

UI User Interface

AJAX Asynchronous JavaScript and XML

A/C Air conditioning

RH Relative Humidity

API Application Programming Interface

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

Η συνεχής εξέλιξη των τεχνολογιών συνδεδεμένων συσκευών και η άνοδος του Διαδικτύου των Πραγμάτων (IoT) έχουν επιφέρει μια επανάσταση στον τρόπο που συλλέγονται, διαχειρίζονται και αξιοποιούνται τα δεδομένα. Σε αυτό το πλαίσιο, η παρούσα διπλωματική εργασία επικεντρώνεται στη διαχείριση δεδομένων μέσω αισθητήρων και τη λήψη αποφάσεων, ενσωματώνοντας διάφορες τεχνικές και μεθοδολογίες για τη βελτιστοποίηση της διαδικασίας αυτής.

Η ανάγκη για αποτελεσματική διαχείριση δεδομένων και αυτοματοποιημένη λήψη αποφάσεων έχει επιτευχθεί με την ραγδαία αύξηση του όγκου και της ποικιλίας των δεδομένων που παράγονται από αισθητήρες. Αυτό είναι επίκαιρο και κρίσιμο σε πολλούς τομείς, όπως η βιομηχανία, η υγεία, η γεωργία και η περιβαλλοντική διαχείριση. Ένα ιδανικό παράδειγμα είναι αυτό του έξυπνου σπιτιού (Smart Home). Το Smart Home αντιπροσωπεύει ένα εξελιγμένο περιβάλλον διαβίωσης, όπου η τεχνολογία συγκλίνει με την καθημερινότητά μας για να προσφέρει προηγμένες λύσεις και εξυπηρετήσεις. Σε αυτό το συμπεριλαμβανόμενο οικιακό περιβάλλον, η διαχείριση δεδομένων και οι αλγόριθμοι λήψης αποφάσεων παίζουν καθοριστικό ρόλο στη διαμόρφωση της εμπειρίας των κατοίκων.

Η διαχείριση δεδομένων σε ένα Smart Home αφορά τη συλλογή, αποθήκευση και ανάλυση πληροφοριών που παράγονται από τα διάφορα έξυπνα συστήματα. Αισθητήρες, συσκευές IoT (Συσκευές του Διαδικτύου των Πραγμάτων), κάμερες και άλλες συνδεδεμένες συσκευές συγκεντρώνουν στοιχεία σχετικά με την κατανάλωση ενέργειας, την ασφάλεια, τη θερμοκρασία και πολλά άλλα.

Οι αλγόριθμοι λήψης αποφάσεων είναι υπεύθυνοι για την ανάλυση αυτών των δεδομένων και την εξαγωγή έξυπνων αποφάσεων για την αυτοματοποίηση διαφόρων λειτουργιών του σπιτιού. Ένα παράδειγμα είναι οι αλγόριθμοι που διαχειρίζονται τον έλεγχο της θέρμανσης ή του κλιματισμού, προβλέποντας και προσαρμόζοντας τις ρυθμίσεις με βάση τις προτιμήσεις και τις συνθήκες.

Με την εφαρμογή προηγμένων τεχνολογιών, όπως της μηχανικής μάθησης και των αλγορίθμων λήψης αποφάσεων, η διαχείριση αυτών των δεδομένων μπορεί να είναι πιο αποτελεσματική και προσαρμοστική. Ο αλγόριθμος λήψης αποφάσεων αντιπροσωπεύει ένα σύνολο διαδικασιών που επιτρέπουν σε ένα σύστημα ή σε μια εφαρμογή να λάβει αποφάσεις με βάση τα δεδομένα που διαθέτει. Ουσιαστικά, αντλεί πληροφορίες από πολύπλοκες συνθήκες, αναλύει τα δεδομένα αυτά, και παράγει αποφάσεις που οδηγούν στην επίλυση ενός συγκεκριμένου προβλήματος ή στην επίτευξη ενός στόχου. Ένας τέτοιος αλγόριθμος είναι χρήσιμος σε διάφορους τομείς, όπως η

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

αναγνώριση προτύπων, η διαχείριση συστημάτων, και η αυτοματοποιημένη λήψη αποφάσεων σε προηγμένα συστήματα IoT. Βασίζονται σε μαθηματικές αρχές και εξελίσσονται συνεχώς με την εισαγωγή νέων τεχνικών και προσεγγίσεων που επιτρέπουν την αντιμετώπιση ακόμη πιο σύνθετων προβλημάτων.

Το Διαδίκτυο των Πραγμάτων (IoT) αναφέρεται στο δίκτυο συσκευών, αισθητήρων και άλλων αντικειμένων που είναι συνδεδεμένα στο διαδίκτυο και ανταλλάσσουν δεδομένα μεταξύ τους. Η ορολογία IoT περιλαμβάνει την υποδομή, τις συσκευές και τα πρωτόκολλα που επιτρέπουν αυτήν τη σύνδεση και αλληλεπίδραση. Είναι γνωστό πως υπάρχουν διάφορες αρχιτεκτονικές IoT που εξυπηρετούν διάφορες ανάγκες και σενάρια εφαρμογής. Στην προκειμένη περίπτωση θα χρησιμοποιηθεί η αρχιτεκτονική πελάτη-εξυπηρετητή (Client-Server Architecture) όπου οι συσκευές IoT λειτουργούν ως πελάτες και αλληλεπιδρούν με εξυπηρετητές που είναι υπεύθυνοι για τη διαχείριση και τον έλεγχο του συστήματος.

Στην αρχιτεκτονική αυτή, οι συσκευές IoT δρουν ως πελάτες που αποστέλλουν δεδομένα σε έναν κεντρικό εξυπηρετητή (server). Ο εξυπηρετητής λαμβάνει, αποθηκεύει και επεξεργάζεται αυτά τα δεδομένα. Η σελίδα frontend προσφέρει μια διεπαφή μέσω της οποίας ο χρήστης μπορεί να επικοινωνεί με το σύστημα, να δίνει εντολές και να παρακολουθεί την κατάσταση των συσκευών.

Ο αλγόριθμος λήψης αποφάσεων στον εξυπηρετητή παίζει σημαντικό ρόλο στη διαχείριση των δεδομένων. Ελέγχει αν οι τιμές είναι εντός ορίων, εκτελεί ενέργειες διόρθωσης, και ενημερώνει την frontend για τυχόν αλλαγές. Η συνεργασία μεταξύ πελάτη και εξυπηρετητή επιτρέπει την αποτελεσματική διαχείριση και έλεγχο των συσκευών IoT, ενθαρρύνοντας την αυξημένη αξιοποίηση και αυτοματισμό στο πλαίσιο του έξυπνου περιβάλλοντος.

Η εργασία αυτή παρουσιάζει την χρήση ενός τέτοιου αλγορίθμου, αποσκοπώντας στην δημιουργία μιας εφαρμογής ελέγχου διάφορων οικιακών συσκευών μέσω της διαχείρισης δεδομένων βασισμένη σε αισθητήρες. Η σελίδα (Frontend page) αποτελεί ένα εικονικό πίνακα λειτουργιών, δίνοντας την δυνατότητα στον χρήστη να ελέγχει τις διάφορες συσκευές που βρίσκονται συνδεδεμένες στο τοπικό δίκτυο (πχ. κλιματιστικό, αφυγραντήρας, φωτιστικό) βασισμένη πάντα στον αλγόριθμο λήψης αποφάσεων και στα δεδομένα που συλλέγει από τους διάφορους αισθητήρες στο χώρο.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

2.1 ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ

Το Διαδίκτυο των Πραγμάτων, ή αλλιώς IoT, είναι ένα δίκτυο αλληλένδετων συσκευών που συνδέουν και ανταλλάσσουν δεδομένα με άλλες συσκευές IoT και το Cloud χωρίς να είναι αναγκαία η ανθρώπινη παρέμβαση. Οι συσκευές IoT είναι συνήθως ενσωματωμένες σε τεχνολογία όπως αισθητήρες (sensors) και λογισμικό (software) και μπορούν να περιλαμβάνουν μηχανικές και ψηφιακές μηχανές καθώς και αντικείμενα καταναλωτών [2]. Η χρήση του IoT εμφανίζεται όλο και περισσότερο στην καθημερινότητά μας, σε επιχειρήσεις, το σπίτι μας. Μία επιχείρηση, χρησιμοποιώντας συσκευές IoT, θα καταφέρει να αυξήσει την αποτελεσματικότητά της σε αρκετούς τομείς, θα υπάρχει καλύτερη εξυπηρέτηση όσον αφορά τους πελάτες καθώς και γρηγορότερη και πιο σωστή λήψη αποφάσεων, μεγαλώνοντας την αξία της κατα πολύ. Αντίστοιχα, σε μία οικία, ο άνθρωπος θα μπορέσει να έχει καλύτερο έλεγχο στο περιβάλλον του, κάνοντας ελάχιστες ενέργειες.

2.1.1 ΙΣΤΟΡΙΑ ΚΑΙ ΕΞΕΛΙΞΗ

Η έννοια του IoT ξεκίνησε να παίρνει σάρκα και οστά στις δεκαετίες του '80 και του '90 όπου την ανέφερε ο Βρετανός τεχνολόγος Kevin Ashton για να περιγράψει ένα μέλλον όπου τα καθημερινά φυσικά αντικείμενα θα συνδέονται στο διαδίκτυο και θα μπορούν να αναγνωρίζονται σε άλλες συσκευές [3]. Από τότε, επιστήμονες και μηχανικοί αρχίσανε να εξετάζουν τον τρόπο σύνδεσης φυσικών αντικειμένων στο διαδίκτυο. Οι πρώτες συσκευές που συμμετείχαν σε αυτήν τη σύνδεση ήταν συνήθως βιομηχανικού τύπου, όπως αισθητήρες και κινητήρες.

Οι αισθητήρες είναι φυσικές συσκευές που μετατρέπουν φυσικά φαινόμενα, όπως θερμοκρασία, κίνηση, ή φως, σε ηλεκτρονικά σήματα. Στο πλαίσιο του Internet of Things (IoT), οι αισθητήρες συνδέονται σε δίκτυα, επικοινωνούν μεταξύ τους και με άλλες συσκευές μέσω του Διαδικτύου.

Με την πάροδο του χρόνου, οι εξελίξεις στην τεχνολογία επέτρεψαν τη δημιουργία μικρότερων, πιο ισχυρών και οικονομικών αισθητήρων και συσκευών, κάνοντας το IoT πιο προσιτό και εφικτό για ευρύτερη χρήση [2]. Η ανάπτυξη του Διαδικτύου ήταν καθοριστική για την εξέλιξη του IoT. Κατά τη δεκαετία του '90, το Διαδίκτυο άρχισε να γίνεται προσβάσιμο στο ευρύ κοινό,

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

επιτρέποντας συνδέσεις μεταξύ υπολογιστών και συσκευών. Οι πρώτες εφαρμογές του IoT εμφανίστηκαν κυρίως στο βιομηχανικό περιβάλλον.

Η δεκαετία του 2000 είδε την εμφάνιση του έξυπνου σπιτιού (Smart Home), όπου έξυπνες συσκευές συνδέονταν στο διαδίκτυο για να παρέχουν αυτοματισμό και απομακρυσμένο έλεγχο. Έξυπνος θερμοστάτης, φώτα, και κάμερες ήταν μερικά από τα πρώτα στοιχεία του έξυπνου σπιτιού. Η σταδιακή αύξηση της ταχύτητας του Διαδικτύου και η διάδοση των τεχνολογιών ασύρματης επικοινωνίας δημιούργησαν τις ιδανικές συνθήκες για τη σύνδεση όλο και περισσότερων συσκευών. Ταυτόχρονα, η χρήση σύγχρονων αισθητήρων παράγει μεγάλες ποσότητες δεδομένων, ενώ η τεχνολογία Big Data επιτρέπει την αποτελεσματική αξιοποίηση και ανάλυση αυτών των πληροφοριών [3]. Οι συσκευές IoT εξελίχθηκαν από απλά ανταλλακτικά δεδομένα σε έξυπνα συστήματα που χρησιμοποιούν τεχνητή νοημοσύνη και μηχανική μάθηση για να κατανοούν, προβλέπουν και λαμβάνουν αυτόνομες αποφάσεις.

Σήμερα, το IoT εντασσόμενο σε ποικίλους τομείς όπως η υγεία, η γεωργία, η βιομηχανία και οι έξυπνες πόλεις, επιδρά θεαματικά στον τρόπο ζωής και την εργασία μας. Με τη συνεχή εξέλιξη των τεχνολογιών, το IoT υπόσχεται να συνεχίσει να μεταμορφώνει τον κόσμο μας και να αλλάζει τον τρόπο ζωής μας, προσφέροντας καινοτόμες λύσεις και δυνατότητες που πριν ήταν απλά αδιανόητες.

2.1.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ ΤΩΝ ΠΡΑΓΜΑΤΩΝ

Το Internet of Things (IoT) παρέχει αρκετά πλεονεκτήματα, αλλά και αντιμετωπίζει ορισμένα μειονεκτήματα. Ένα βασικό πλεονέκτημα είναι πως επιτρέπει την πρόσβαση σε πληροφορίες από οπουδήποτε και ανά πάσα στιγμή μέσω οποιασδήποτε συσκευής [2]. Αυτό σημαίνει ότι δεν είμαστε πλέον περιορισμένοι σε συγκεκριμένους φυσικούς τόπους ή συγκεκριμένες συσκευές για να αποκτήσουμε πρόσβαση σε δεδομένα. Αντίθετα, μπορούμε να συνδεόμαστε σε δίκτυα και να λαμβάνουμε πληροφορίες από παντού, χρησιμοποιώντας το IoT, με αποτέλεσμα να μας επιτρέπει την άμεση πρόσβαση σε δεδομένα, εφαρμογές και υπηρεσίες μέσω συνδεδεμένων συσκευών όπως τα smartphones, τα έξυπνα ρολόγια, οι αισθητήρες και άλλες έξυπνες συσκευές. Αυτή η συνεχής πρόσβαση επιτρέπει στους χρήστες να διαχειρίζονται, να παρακολουθούν και να αλληλεπιδρούν με το περιβάλλον τους από οπουδήποτε, προσφέροντας άνεση, ευελιξία και εξειδικευμένες υπηρεσίες.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Ένα άλλο πλεονέκτημα είναι πως επιτρέπει τη μεταφορά πακέτων δεδομένων μέσω ενός συνδεδεμένου δικτύου, το οποίο μπορεί να εξοικονομήσει χρόνο και χρήμα [2]. Η μεταφορά πακέτων δεδομένων μεταξύ των συνδεδεμένων συσκευών επιτρέπει τη συλλογή, ανάλυση και χρήση πληροφοριών με τρόπο πιο αποτελεσματικό. Αυτό συμβάλλει στην αυξημένη αυτοματοποίηση και ευελιξία σε διάφορους τομείς, όπως η βιομηχανία, η υγεία, η ενέργεια και η καθημερινή ζωή. Εκτός από την εξοικονόμηση χρόνου, η μεταφορά δεδομένων μέσω του συνδεδεμένου δικτύου οδηγεί σε οικονομία χρημάτων, καθώς διαδικασίες που προηγουμένως απαιτούσαν ανθρώπινη επέμβαση τώρα μπορούν να εκτελούνται αυτόματα. Αυτό ενισχύει την αποτελεσματικότητα, μειώνει τα λάθη και δημιουργεί ένα περιβάλλον όπου η τεχνολογία εξυπηρετεί τις καθημερινές ανάγκες με βιώσιμο και οικονομικά αποτελεσματικό τρόπο.

Κάτι άλλο που προσφέρει το IoT είναι η ικανότητα να συλλέγει μεγάλο όγκο δεδομένων από πολλές συσκευές (αισθητήρες, συσκευές κινητικότητας κλπ.), βοηθώντας τόσο τους χρήστες όσο και τους κατασκευαστές [2]. Επιπλέον η διαδικασία ανάλυσης δεδομένων στην άκρη (edge computing) αποτελεί κρίσιμο στοιχείο του IoT. Κατά τη διάρκεια αυτής της φάσης, η επεξεργασία δεδομένων διεξάγεται κοντά στην πηγή τους, δηλαδή στη συσκευή ή τον αισθητήρα που τα δημιούργησε, αντί να στέλνονται όλα τα δεδομένα στο Cloud για επεξεργασία. Αυτό έχει σημαντικά πλεονεκτήματα, όπως η μείωση του όγκου των δεδομένων που μεταφέρονται στο Cloud, η μείωση της καθυστέρησης στην ανταπόκριση, και η εξοικονόμηση εύρους ζώνης δικτύου. Συνολικά, αυτή η προσέγγιση επιτρέπει την αποτελεσματική επεξεργασία των δεδομένων ακόμα και σε περιβάλλοντα με περιορισμένη συνδεσιμότητα, βοηθώντας στην εξαγωγή σημαντικών πληροφοριών από το IoT οικοσύστημα.

Αξίζει να αναφερθεί και το πλεονέκτημα της βελτιωμένης επικοινωνίας μεταξύ συνδεδεμένων ηλεκτρονικών συσκευών [2]. Αυτή η δυνατότητα επιτρέπει στις συσκευές να ανταλλάσσουν δεδομένα και πληροφορίες μεταξύ τους, βελτιώνοντας τη συνολική απόδοση και αποτελεσματικότητα του συστήματος. Μέσω της ενισχυμένης επικοινωνίας, οι συνδεδεμένες συσκευές μπορούν να συνεργάζονται συντονισμένα για την υλοποίηση εργασιών, ανταλλάσσοντας πληροφορίες που είναι καίριες για τη λήψη αποφάσεων ή την εκτέλεση ειδικών λειτουργιών. Αυτή η αποτελεσματική επικοινωνία μεταξύ των συσκευών συμβάλλει στη δημιουργία έξυπνων συνδυασμών λειτουργιών, προσφέροντας ταυτόχρονα βελτιωμένη χρηστικότητα και εμπειρία στον χρήστη. Ως εκ τούτου, η βελτιωμένη αυτή επικοινωνία αποτελεί ουσιαστικό στοιχείο για την αποδοτική λειτουργία του IoT, επιτρέποντας την ομαλή συνεργασία και αλληλεπίδραση μεταξύ των έξυπνων συσκευών.

Ωστόσο, σε ένα τέτοιο δίκτυο μπορούμε να διακρίνουμε και ορισμένα μειονεκτήματα. Ένα από αυτά είναι η αύξηση του αριθμού των συνδεδεμένων συσκευών στο δίκτυο, που δημιουργεί μια

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

αυξανόμενη επιφάνεια επίθεσης και αποτελεί μια προκλητική πτυχή της τεχνολογικής εξέλιξης [2]. Καθώς οι συσκευές αυτές μοιράζονται περισσότερες πληροφορίες μεταξύ τους για την αποτελεσματικότερη λειτουργία, αυξάνεται σημαντικά η πιθανότητα κλοπής εμπιστευτικών πληροφοριών από επιθετικούς χάκερς. Η επέκταση της επιφάνειας επίθεσης προκύπτει από τον μεγάλο αριθμό σημείων εισόδου που παρέχουν οι πολλές συνδεδεμένες συσκευές. Αυτό δίνει στους κακόβουλους χρήστες περισσότερες ευκαιρίες να εκμεταλλευτούν ελλείψεις ασφάλειας ή αδυναμίες σε μια συγκεκριμένη συσκευή ή δίκτυο. Καθώς οι επιθέσεις μπορεί να επικεντρώνονται σε πολλαπλές συσκευές, η προστασία των εμπιστευτικών πληροφοριών απαιτεί εξειδικευμένες προσεγγίσεις ασφάλειας και συνεχή ενημέρωση των μέτρων προστασίας. Είναι, επομένως, σημαντικό να αναπτύσσονται συστήματα που ενισχύουν την ασφάλεια των συνδεδεμένων συσκευών, προκειμένου να αντιμετωπίζονται αποτελεσματικά πιθανές απειλές και να διασφαλίζεται η προστασία των προσωπικών δεδομένων.

Ένα άλλο μειονέκτημα είναι πώς η αύξηση των συσκευών IoT δημιουργεί προκλήσεις στη διαχείριση τους για οργανισμούς, καθώς η πολυπλοκότητα και ο όγκος των συσκευών αυξάνονται [2]. Αυτή η κατάσταση μπορεί να καταστήσει τη διαχείριση των συσκευών προβληματική, καθώς οι οργανισμοί θα χρειαστεί να αντιμετωπίσουν έναν χαοτικό αριθμό συνδεδεμένων συσκευών IoT. Η δυσκολία στη διαχείριση προκύπτει από την ανάγκη για συλλογή, ανάλυση και ερμηνεία των δεδομένων που παράγονται από χιλιάδες ή ακόμα και εκατομμύρια συσκευές. Οι οργανισμοί πρέπει να αντιμετωπίσουν την πρόκληση της αποθήκευσης, επεξεργασίας και διαχείρισης αυτών των δεδομένων με τρόπο που να είναι αποτελεσματικός, ασφαλής και συμμορφούμενος με τους κανονισμούς περί προστασίας δεδομένων. Οι προηγμένες λύσεις διαχείρισης συσκευών IoT είναι ουσιώδεις για την αποφυγή χαοτικών καταστάσεων, προσφέροντας αυτοματοποιημένες διαδικασίες, ευέλικτες πλατφόρμες διαχείρισης και ισχυρές λειτουργίες ασφαλείας. Με αυτόν τον τρόπο, οι οργανισμοί μπορούν να απολαύσουν τα οφέλη του IoT χωρίς να θέτουν σε κίνδυνο την αποτελεσματική διαχείριση και ασφάλεια των συνδεδεμένων συσκευών.

Ένα από τα μειονεκτήματα του IoT είναι επίσης η πιθανότητα καταστροφής άλλων συνδεδεμένων συσκευών σε περίπτωση σφάλματος στο σύστημα [2]. Αυτό συμβαίνει όταν μια συσκευή IoT, λόγω κάποιου τύπου επιθέσεων ή ασφαλείας, δεν λειτουργεί σωστά και ενδέχεται να προκαλέσει ζημιά σε άλλες συσκευές στο ίδιο δίκτυο. Αυτό το φαινόμενο γίνεται περισσότερο πιθανό όταν οι συσκευές δεν είναι προστατευμένες από κατάλληλα μέτρα ασφαλείας. Σε περιπτώσεις επίθεσης, όπως η εισβολή κακόβουλου λογισμικού, οικονομικών επιθέσεων ή ακόμη και φυσικής καταστροφής, μια ευάλωτη συσκευή μπορεί να αντιδράσει με τρόπο που επηρεάζει τις λειτουργίες

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

των γύρω συσκευών. Για να αντιμετωπιστεί αυτό το πρόβλημα, είναι ζωτικής σημασίας η εφαρμογή στρατηγικών ασφαλείας, όπως η χρήση αξιόπιστων πρωτοκόλλων ασφαλείας, η τακτική ενημέρωση του λογισμικού και η υιοθέτηση βέλτιστων πρακτικών ασφάλειας από τους κατασκευαστές και τους χρήστες των συσκευών IoT.

Αξίζει να αναφέρουμε ένα ακόμη μειονέκτημα, το οποίο είναι η αύξηση των προβλημάτων συμβατότητας μεταξύ συσκευών, καθώς δεν υπάρχει διεθνές πρότυπο συμβατότητας για το IoT [2]. Καθώς οι συσκευές IoT προέρχονται από διάφορους κατασκευαστές και λειτουργούν με διάφορα πρωτόκολλα επικοινωνίας, η απουσία ενός κοινού προτύπου δυσκολεύει τις συσκευές να αλληλεπιδρούν αμοιβαίως μεταξύ τους. Αυτό δημιουργεί προκλήσεις όσον αφορά τη συμβατότητα και τον συντονισμό μεταξύ διαφορετικών συσκευών, περιορίζοντας την ομαλή ενσωμάτωση και αλληλεπίδραση. Οι χρήστες μπορεί να αντιμετωπίσουν δυσκολίες όταν προσπαθούν να συνδέσουν συσκευές από διάφορους κατασκευαστές, καθώς η αναγνώριση και η επικοινωνία μεταξύ τους μπορεί να απαιτεί επιπλέον προσπάθειες και προσαρμογές. Για την αντιμετώπιση αυτού του ζητήματος, η ανάπτυξη διεθνών προτύπων συμβατότητας αποτελεί σημαντικό βήμα προς τη δημιουργία ενός πιο ομαλού και εναρμονισμένου περιβάλλοντος για το IoT.

2.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΕΛΑΤΗ – ΕΞΥΠΗΡΕΤΗΤΗ (CLIENT-SERVER)

Η αρχιτεκτονική client-server σε συνδυασμό με το Internet of Things (IoT) αντιπροσωπεύει ένα σύγχρονο μοντέλο επικοινωνίας όπου οι συσκευές του IoT λειτουργούν ως clients (πελάτες) που αλληλεπιδρούν με έναν κεντρικό διακομιστή [5]. Κάθε συσκευή IoT, εξοπλισμένη με αισθητήρες και άλλες λειτουργίες, μπορεί να συλλέγει δεδομένα και να αποστέλλει αιτήματα στον κεντρικό διακομιστή για επεξεργασία και ανάλυση. Ο κεντρικός διακομιστής (server) λειτουργεί ως κεντρικός κόμβος διαχείρισης, δέχεται τα δεδομένα από τις συσκευές client και παρέχει τις αναγκαίες υπηρεσίες.

Η επικοινωνία ανάμεσα στις συσκευές IoT και τον κεντρικό διακομιστή συνήθως γίνεται μέσω πρωτοκόλλων δικτύου, όπως το HTTP. Η αρχιτεκτονική αυτή επιτρέπει τη συγκέντρωση, την επεξεργασία και τη διαχείριση των δεδομένων από τις πολλές συσκευές IoT σε ένα κεντρικό σημείο. Επίσης, επιτρέπει την αποτελεσματική εκτέλεση εφαρμογών και υπηρεσιών πάνω σε αυτά τα δεδομένα, προσφέροντας συνολική οργάνωση και έλεγχο. Συνοπτικά, η αρχιτεκτονική client-server

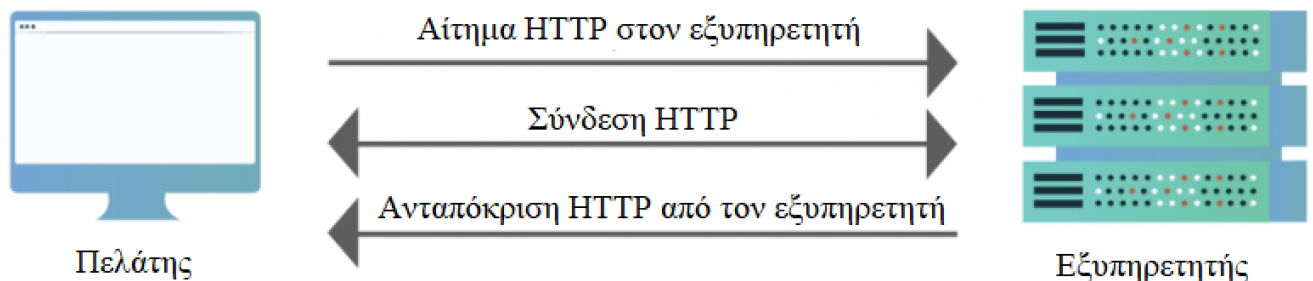
Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

σε συνδυασμό με το IoT δημιουργεί ένα ευέλικτο και κλιμακούμενο περιβάλλον για τη διαχείριση των συσκευών και των δεδομένων του IoT.

Το πρωτόκολλο HTTP (Hypertext Transfer Protocol) αντιπροσωπεύει το θεμέλιο της επικοινωνίας στο World Wide Web. Σχεδιασμένο για τη μεταφορά δεδομένων μεταξύ πελάτη (client) και διακομιστή (server), το HTTP λειτουργεί σε ένα μοντέλο αιτήματος-απόκρισης. Κάθε επικοινωνία αρχίζει με ένα αίτημα από τον πελάτη, που στη συνέχεια ο διακομιστής απαντά με τη σχετική απόκριση. Η διαδικασία περιλαμβάνει τα εξής βήματα: Καταρχάς, ο πελάτης αποστέλλει ένα αίτημα στο διακομιστή για να λάβει κάποιο αρχείο ή να εκτελέσει κάποια ενέργεια, περιλαμβάνοντας τον τύπο της ενέργειας και τυχόν παραμέτρους. Στη συνέχεια, ο διακομιστής απαντά στο πελάτη με την αντίστοιχη απόκριση, η οποία περιλαμβάνει το αποτέλεσμα του αιτήματος, όπως ένα αρχείο ή μια σελίδα HTML, καθώς και έναν τριψήφιο αριθμό κατάστασης πρωτοκόλλου που δείχνει την επιτυχία ή την αποτυχία του αιτήματος.

Το HTTP είναι ανοικτό πρωτόκολλο, παρέχοντας ένα απλό μοντέλο επικοινωνίας και χρησιμοποιείται ευρέως στον κόσμο του διαδικτύου, εξασφαλίζοντας τη μεταφορά δεδομένων με αξιοπιστία και αποτελεσματικότητα.



Εικόνα 2.3: Αρχιτεκτονική client-server και χρήση πρωτοκόλλου HTTP [10] (“*Visualising the client server architecture*” <https://www.toolsqa.com/client-server/client-server-architecture-and-model>)

Η αρχιτεκτονική client-server σε συνδυασμό με το πρωτόκολλο HTTP αντιπροσωπεύει ένα κομβικό στοιχείο στον κόσμο του διαδικτύου. Στη βασική της μορφή, ο πελάτης (client) εκδίδει αιτήματα προς έναν διακομιστή (server), ο οποίος ανταποκρίνεται με τα αντίστοιχα δεδομένα ή υπηρεσίες. Η διαδικασία αυτή βασίζεται σε συγκεκριμένες εντολές. Για παράδειγμα, η εντολή GET

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

χρησιμοποιείται για τη λήψη πληροφοριών από τον διακομιστή, ενώ η εντολή POST χρησιμοποιείται για την υποβολή δεδομένων προς επεξεργασία. Ο διακομιστής απαντά με κατάλληλους κωδικούς κατάστασης, όπως το 200 OK για επιτυχείς αιτήσεις ή το 404 Not Found για ανεπιτυχή αιτήματα. Για την υλοποίηση αυτής της αρχιτεκτονικής, απαιτούνται γνώσεις προγραμματισμού για την ανάπτυξη πελατικών και διακομιστικών εφαρμογών, καθώς και κατανόηση του τρόπου λειτουργίας των δικτύων και των πρωτοκόλλων δικτύου, όπως το HTTP. Επιπλέον, η διαχείριση των κωδικών κατάστασης και η αποτελεσματική διαχείριση των αιτημάτων-αποκρίσεων είναι κρίσιμες γνώσεις για τη σταθερή λειτουργία των συστημάτων. Κατ' επέκταση, οι επαγγελματίες πρέπει να είναι εξοικειωμένοι με το περιβάλλον δικτύου και τον προγραμματισμό, διασφαλίζοντας τη σταθερή και αποδοτική λειτουργία των συστημάτων που χρησιμοποιούν αυτή την αρχιτεκτονική.

Η υλοποίηση του frontend και του backend σε μια σύνδεση client-server απαιτεί τη χρήση διαφόρων γλωσσών προγραμματισμού, καθένα με τον δικό του ρόλο και λειτουργίες. Στο frontend, που αναφέρεται στο τμήμα του χρήστη, χρησιμοποιούνται γλώσσες όπως η HTML, το CSS και η JavaScript. Η HTML χρησιμοποιείται για τη δημιουργία της δομής της ιστοσελίδας, το CSS για τον σχεδιασμό και τη μορφοποίηση, ενώ το JavaScript προσθέτει δυναμική αλληλεπίδραση. Στο backend, που αναφέρεται στο τμήμα του διακομιστή, μπορούν να χρησιμοποιηθούν πολλές γλώσσες, ανάλογα με τις απαιτήσεις του έργου. Για παράδειγμα, η γλώσσα JavaScript (Js), με τη βοήθεια του περιβάλλοντος εκτέλεσης Node.js, μπορεί να χρησιμοποιηθεί και στο backend. Άλλες δημοφιλείς γλώσσες backend περιλαμβάνουν την Java, την Python, την Ruby, την PHP και πολλές άλλες. Η συνδυασμένη χρήση αυτών των γλωσσών επιτρέπει τη δημιουργία λειτουργικών και αποδοτικών εφαρμογών που προσφέρουν μια ολοκληρωμένη εμπειρία στους χρήστες.

2.2.1 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΠΕΛΑΤΗ – ΕΞΥΠΗΡΕΤΗΤΗ

Η αρχιτεκτονική client-server αντιπροσωπεύει ένα εξαιρετικά χρήσιμο μοντέλο στον χώρο της πληροφορικής με πολλά πλεονεκτήματα. Η βασική ιδέα είναι ο διαχωρισμός των καθηκόντων μεταξύ του client, που αναλαμβάνει τη διεπαφή χρήστη, και του server, που είναι υπεύθυνος για τη διαχείριση των αιτημάτων και την παροχή υπηρεσιών*.

Ένα από τα κύρια πλεονεκτήματα της αρχιτεκτονικής client-server είναι η ευελιξία που προσφέρει. Ο διαχωρισμός των επιμέρους λειτουργιών επιτρέπει στους προγραμματιστές να εργαστούν ανεξάρτητα σε κάθε μέρος του συστήματος, διευκολύνοντας έτσι τη συντήρηση και την

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

επέκταση. Ο server, από την άλλη πλευρά, μπορεί να επικεντρωθεί στη διαχείριση των αιτημάτων και την αποτελεσματική κατανομή του φορτίου. Ένα ακόμη σημαντικό πλεονέκτημα είναι η καλύτερη ασφάλεια και διαχείριση των δεδομένων. Ο server μπορεί να υλοποιήσει πολιτικές ασφαλείας, προστατεύοντας τα δεδομένα από ανεπιθύμητη πρόσβαση. Επιπλέον, η αρχιτεκτονική επιτρέπει τον εύκολο έλεγχο των προνομίων πρόσβασης.

Η αρχιτεκτονική client-server έχει επιπλέον πλεονεκτήματα που την καθιστούν ευέλικτη και αποτελεσματική σε πολλές περιπτώσεις. Ένα ακόμη όφελος είναι η εξοικονόμηση πόρων, καθώς οι πελάτες δεν χρειάζεται να διαθέτουν πλήρη εγκατάσταση λογισμικού και δεδομένων, αλλά απλώς να έχουν πρόσβαση στις υπηρεσίες που παρέχονται από τον server. Αυτό επιτρέπει την εξοικονόμηση ενέργειας και αποθηκευτικού χώρου στις συσκευές των χρηστών.

Επιπλέον, η αρχιτεκτονική αυτή προωθεί την εύκολη ενημέρωση και συντήρηση του συστήματος. Αναβαθμίσεις και διορθώσεις σφαλμάτων μπορούν να εφαρμοστούν στον server χωρίς την ανάγκη να επηρεάζουν άμεσα τους πελάτες. Αυτό εξασφαλίζει ότι οι χρήστες θα έχουν πάντα πρόσβαση σε μια ενημερωμένη και λειτουργική έκδοση της υπηρεσίας.

Τέλος, η αρχιτεκτονική client-server υποστηρίζει τη δυνατότητα επεκτασιμότητας. Ο server μπορεί να ανταποκρίνεται σε αυξημένο αριθμό πελατών και να εξυπηρετεί νέες αιτήσεις χωρίς να επηρεάζεται η συνολική απόδοση. Αυτό είναι ιδιαίτερα χρήσιμο σε εφαρμογές που αναπτύσσονται και εξελίσσονται στον χρόνο.

Παρά τα πολλά πλεονεκτήματα, η αρχιτεκτονική client-server διαθέτει και ορισμένα μειονεκτήματα. Ένα από αυτά είναι η επικοινωνία με τον server. Καθώς οι πελάτες εξαρτώνται από τη σύνδεση με τον server για τη λήψη υπηρεσιών και δεδομένων, η απώλεια σύνδεσης ή η κακή απόδοση του server μπορεί να επηρεάσει αρνητικά την εμπειρία του χρήστη. Μία λύση για την αντιμετώπιση των προβλημάτων στην επικοινωνία με τον server είναι η χρήση της ασύγχρονης επικοινωνίας. Στην ασύγχρονη επικοινωνία, ο πελάτης μπορεί να στέλνει αιτήσεις στον server χωρίς να περιμένει ενεργά για την απάντηση. Αυτό επιτρέπει στον πελάτη να συνεχίσει την εκτέλεση άλλων εργασιών ενώ περιμένει την απάντηση από τον server. Η ασύγχρονη επικοινωνία μπορεί να βοηθήσει στην αποφυγή της αδράνειας και της επιβάρυνσης του χρήστη, καθώς η εφαρμογή εξακολουθεί να ανταποκρίνεται σε άλλες εργασίες κατά τη διάρκεια της αναμονής για την ολοκλήρωση των αιτήσεων προς τον εξυπηρετητή.

Επιπλέον, η αποθήκευση δεδομένων σε έναν κεντρικό εξυπηρετητή μπορεί να δημιουργήσει προβλήματα ασφαλείας. Η καταστροφή ή ο χειρισμός των δεδομένων σε μια κεντρική τοποθεσία

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

αυξάνει τον κίνδυνο για απώλεια ή παραβίαση των πληροφοριών, ειδικά όταν πρόκειται για ευαίσθητα ή προσωπικά δεδομένα. Επιπλέον, η επικοινωνία μεταξύ πελάτη και εξυπηρετητή μπορεί να δημιουργήσει φορτία εργασίας στον εξυπηρετητή, ιδίως όταν υπάρχει μεγάλος αριθμός πελατών που κάνουν αιτήσεις ταυτόχρονα. Αυτό μπορεί να οδηγήσει σε καθυστερήσεις ή αργή απόκριση ανάλογα με το φόρτο εργασίας του εξυπηρετητή. Για την αντιμετώπιση των προβλημάτων ασφάλειας και της υπερφόρτωσης του, μια λύση είναι η χρήση προηγμένων συστημάτων ασφαλείας και τεχνικών διαχείρισης φόρτου. Για την προστασία των δεδομένων, η κρυπτογράφηση μπορεί να χρησιμοποιηθεί για να εξασφαλίσει ότι ακόμη και αν παραβιαστεί η ασφάλεια, τα δεδομένα παραμένουν προστατευμένα. Για το θέμα της υπερφόρτωσης, οι τεχνικές λύσεις συμπεριλαμβάνουν τη χρήση εξισορρόπησης φορτίου (load balancing) που κατανέμει τις αιτήσεις των πελατών σε πολλούς εξυπηρετητές, βελτιστοποιώντας έτσι την απόδοση και μειώνοντας τον κίνδυνο υπερφόρτωσης σε έναν συγκεκριμένο εξυπηρετητή. Επιπλέον, η χρήση τεχνικών caching μπορεί να μειώσει τη φόρτωση στον server, αποθηκεύοντας και επαναχρησιμοποιώντας προηγούμενες απαντήσεις για συχνά αιτούμενα δεδομένα, μειώνοντας έτσι τον αριθμό των ερωτημάτων προς τον server.

Τέλος, η αρχιτεκτονική client-server απαιτεί συνεχή σύνδεση στο διαδίκτυο. Σε περιοχές με ασταθή σύνδεση ή όταν ο χρήστης βρίσκεται σε κίνηση, αυτό μπορεί να προκαλέσει προβλήματα αποσύνδεσης και δυσκολίες στη χρήση των υπηρεσιών. Για την αντιμετώπιση των προβλημάτων που προκύπτουν λόγω της απαιτούμενης συνεχούς σύνδεσης στο διαδίκτυο, οι αναπτυσσόμενες τεχνολογίες αναζητούν τρόπους ελαχιστοποίησης των επιπτώσεων από την απώλεια σύνδεσης ή την αστάθεια. Η τεχνολογία "offline mode" αποτελεί μία λύση, επιτρέποντας στους χρήστες να έχουν πρόσβαση σε ορισμένες λειτουργίες της εφαρμογής ακόμη και χωρίς σύνδεση στο διαδίκτυο. Οι τοπικές αποθηκευτικές μονάδες, όπως οι τοπικές βάσεις δεδομένων, μπορούν να χρησιμοποιηθούν για την αποθήκευση πληροφοριών και τη δυνατότητα εργασίας χωρίς σύνδεση. Επιπλέον, η προετοιμασία για αυτόματες συνδέσεις αφού η σύνδεση επανέλθει, γνωστή ως "sync", επιτρέπει τη συγχρονισμένη μεταφορά δεδομένων που έχουν αλλάξει ενώ ο χρήστης ήταν εκτός σύνδεσης. Αυτή η προσέγγιση διασφαλίζει ότι οι αλλαγές που καταγράφηκαν τοπικά θα ενσωματωθούν στο σύστημα όταν επανασυνδεθεί ο χρήστης στο διαδίκτυο.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

2.3 SOCKETS

Τα sockets αντιπροσωπεύουν ένα κλειδί για τη δικτύωση στον κόσμο των υπολογιστών, παρέχοντας ένα δίαυλο επικοινωνίας μεταξύ διαφορετικών υπολογιστικών συστημάτων σε ένα δίκτυο. Επιτρέπουν την αμφίδρομη ανταλλαγή δεδομένων μεταξύ πελάτη και εξυπηρετητή, επιτρέποντας τη δημιουργία σταθερών συνδέσεων και τη μετάδοση πληροφοριών μεταξύ τους. Στο πλαίσιο του Internet of Things (IoT), η χρήση των sockets είναι ουσιαστική για τη διασύνδεση και επικοινωνία μεταξύ των συσκευών. Οι συσκευές IoT χρησιμοποιούν τις τεχνολογίες δικτύου και τα sockets για να στείλουν και να λαμβάνουν δεδομένα. Συνήθως, αυτές οι συσκευές λειτουργούν ως πελάτες που συνδέονται με εξυπηρετητές ή πύλες δεδομένων, με τα sockets να διευκολύνουν την αποτελεσματική και ασφαλή επικοινωνία τους.

Οι εφαρμογές του IoT, όπως τα έξυπνα σπίτια, τα αυτοκίνητα και οι βιομηχανικές συσκευές, εξαρτώνται σημαντικά από τη σωστή λειτουργία των sockets για να ανταλλάσσουν πληροφορίες μεταξύ τους και με τον εξυπηρετητή. Η αξιοποίηση αυτής της τεχνολογίας εξασφαλίζει την αξιόπιστη και αποδοτική λειτουργία των συσκευών IoT, προσφέροντας ταυτόχρονα δυνατότητες ελέγχου και επικοινωνίας σε πραγματικό χρόνο.

2.4 THREADS

Τα threads αποτελούν μια σημαντική έννοια στην προγραμματισμένη πολυεργασία και επιτρέπουν την εκτέλεση πολλαπλών νημάτων εργασίας από την ίδια διεργασία. Κάθε thread αντιπροσωπεύει μια ανεξάρτητη ροή εκτέλεσης, επιτρέποντας την ταυτόχρονη εκτέλεση διαφορετικών εργασιών. Τα threads μοιράζονται τους ίδιους πόρους μνήμης και δίνουν τη δυνατότητα για αποδοτικότερη εκτέλεση πολλαπλών διεργασιών.

Στο πλαίσιο του Internet of Things (IoT), η χρήση threads είναι κρίσιμη για την αποτελεσματική διαχείριση των διαφόρων λειτουργιών που πραγματοποιούνται από τις συσκευές IoT. Οι εφαρμογές IoT πολλές φορές πρέπει να ανταποκρίνονται σε πολλές εργασίες ταυτόχρονα, όπως τη συλλογή δεδομένων από αισθητήρες, την ανάλυση δεδομένων, τη λήψη αποφάσεων και την αλληλεπίδραση με άλλες συσκευές.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Με τη χρήση threads, οι συσκευές IoT μπορούν να διαχειρίζονται αποδοτικά πολλαπλές εργασίες ταυτόχρονα, ενισχύοντας την αποκρισιμότητά τους και επιτρέποντας την ομαλή λειτουργία σε περιβάλλοντα πραγματικού χρόνου. Αυτό είναι κρίσιμο για εφαρμογές που απαιτούν γρήγορη ανταπόκριση, όπως η ελεγχόμενη αυτοκίνηση, τα έξυπνα συστήματα ενέργειας ή η ιατρική παρακολούθηση.

Συνολικά, η ενσωμάτωση της τεχνολογίας threads στο πλαίσιο του IoT συμβάλλει σημαντικά στην αποδοτικότητα και την αξιοπιστία των συσκευών, επιτρέποντας τους να ανταποκρίνονται άμεσα στις απαιτήσεις του περιβάλλοντος τους.

2.5 ΑΙΣΘΗΤΗΡΕΣ

Οι αισθητήρες αποτελούν κρίσιμο συστατικό στον κόσμο του Internet of Things (IoT), παρέχοντας τη δυνατότητα συλλογής δεδομένων από το περιβάλλον τους. Αυτές οι συσκευές μπορούν και είναι συμβατές με πολλούς τύπους δεδομένων, καταγράφοντας πληροφορίες όπως θερμοκρασία, υγρασία, φωτεινότητα, κίνηση, και πολλά άλλα, ανάλογα με τον τύπο τους.

Στο πλαίσιο του IoT, οι αισθητήρες χρησιμοποιούνται ευρέως για τη συλλογή δεδομένων που είναι κρίσιμα για τη λειτουργία του συστήματος. Για παράδειγμα, σε μια έξυπνη πόλη, αισθητήρες μπορεί να καταγράφουν την ποιότητα του αέρα, την κίνηση, και την κατανάλωση ενέργειας. Αυτά τα δεδομένα μεταδίδονται σε πραγματικό χρόνο σε ένα κεντρικό σημείο ελέγχου όπου μπορούν να χρησιμοποιηθούν για λήψη αποφάσεων, όπως η προσαρμογή των συστημάτων κυκλοφορίας ή η βελτίωση της ενεργειακής απόδοσης.

Η σημασία των αισθητήρων στο IoT αντικατοπτρίζεται στη δυνατότητα τους να παρέχουν πρακτικές πληροφορίες για το περιβάλλον, βοηθώντας στην παρακολούθηση, τον έλεγχο, και την εξέλιξη των συνδεδεμένων συσκευών. Αυτή η ανταλλαγή πληροφοριών παίζει καίριο ρόλο σε εφαρμογές όπως έξυπνα σπίτια, έξυπνες πόλεις και βιομηχανικές λύσεις IoT, ενισχύοντας την αυτοματοποίηση και την αποδοτικότητα των συστημάτων.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

2.6 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

Η διπλωματική εργασία βασίστηκε σε μια εκτενή βιβλιογραφική ανασκόπηση, στην οποία χρησιμοποιήθηκαν επιστημονικά άρθρα, διαδικτυακές σελίδες και κάποιες εργασίες διαθέσιμες μέσω του αποθετηρίου “Πολυνόη”. Η εν λόγω ανασκόπηση αποτέλεσε τη θεωρητική βάση για την υλοποίηση της εφαρμογής, καλύπτοντας τόσο το μέρος του frontend όσο και του backend.

Στο κομμάτι του frontend, εξετάστηκαν διάφορα επιστημονικά άρθρα που ασχολούνται με τις σύγχρονες τάσεις και τεχνολογίες στο χώρο του χρήστη (user interface, UI) και του σχεδιασμού εμπειρίας χρήστη (user experience). Παράλληλα, διαδικτυακές σελίδες που παρέχουν οδηγούς και βοήθεια για την υλοποίηση ευέλικτων και ευανάγνωστων διεπαφών επίσης επηρέασαν την διαδικασία σχεδιασμού του frontend.

Στον τομέα του backend, επιστημονικά άρθρα που εξετάζουν τις προηγμένες αρχιτεκτονικές και τεχνικές για την ανάπτυξη αποτελεσματικών και ασφαλών εξυπηρετητών (servers) προσέφεραν σημαντικές γνώσεις. Όπως ακόμη και πηγές που προέκυψαν από εργασίες που παρείχαν πρακτικές συμβουλές και τεχνικές για την υλοποίηση του backend συστήματος. Με αυτήν την πλούσια βιβλιογραφική στήριξη, η διπλωματική εργασία ανέπτυξε μια ολοκληρωμένη και καλά τεκμηριωμένη προσέγγιση για την υλοποίηση της εφαρμογής σε όλα τα επίπεδα.

Αρχικά το άρθρο με τίτλο “*An IoT-based handheld environmental and air quality monitoring station, M. N. M. Aashiq et al., 2023*” [16] επικεντρώθηκε στην ανάπτυξη ενός βασισμένου στο IoT φορητού gadget παρακολούθησης καιρού που μπορεί να μετρήσει τις παραμέτρους καιρού και αέρα. Ο σταθμός αυτός συγκεντρώνει και αναλύει τις μετρήσεις πολλαπλών παραμέτρων σε διάφορες γεωγραφικές τοποθεσίες με σκοπό την προώθηση ενός υγιεινού περιβάλλοντος. Οι μετρήσεις αυτές βασίζονται σε τεχνολογίες IoT και αποσκοπούν στην καταγραφή και αξιολόγηση των συνθηκών περιβάλλοντος για την προώθηση της υγείας. Συγκεκριμένα κάποια από τα εργαλεία που χρησιμοποιήθηκαν είναι το Arduino σε συνδυασμό με αισθητήρες θερμοκρασίας, υγρασίας κλπ. Το Wi-Fi καθώς και μία εφαρμογή που με το αντίστοιχο GUI δίνει τα δεδομένα που ζητά ο χρήστης. Μετά από διάφορες μετρήσεις τονίζεται η σημασία της συνεχούς τροφοδοσίας ενέργειας και προτείνεται περαιτέρω έρευνα για πιο αποτελεσματικές και φορητές πηγές ενέργειας. Ωστόσο αναφέρονται και κάποια μειονεκτήματα, όπως τα προβλήματα στις μετρήσεις με τον αισθητήρα θερμοκρασίας-υγρασίας όπου προτείνονται νέες μετρήσεις με πιθανόν διαφορετικό αισθητήρα. Τα συμπεράσματα της υλοποίησης αυτού του gadget δείχνουν την εφαρμογή αυτή να ανοίγει δρόμο για

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

την κατανόηση και την εφαρμογή συστημάτων IoT, ενώ παράλληλα να προσφέρει βάση για περαιτέρω εξελίξεις στη δημιουργία κεντρικών συστημάτων ελέγχου του κλίματος.

Μία επίσης εργασία που μελετήθηκε ήταν αυτή με τίτλο “*IOT Based Weather Monitoring System Using Arduino-Uno, Hashmi et al., 2022*” [31] που πραγματευόταν τη δημιουργία μιας μεγάλης βάσης δεδομένων συνδυάζοντας τους δύο τομείς, των συστημάτων ελέγχου και της τεχνικής συλλογής δεδομένων. Και σε αυτή τη μελέτη, χρησιμοποιήθηκαν εξίσου αισθητήρες θερμοκρασίας και υγρασίας για τον σχεδιασμό ενός αποτελεσματικού συστήματος παρακολούθησης του καιρού. Το έργο αποσκοπεί στον πραγματικό χρόνο παρακολούθησης του καιρού μέσω μιας εφαρμογής κινητού τηλεφώνου, χρησιμοποιώντας το Arduino UNO ως πλατφόρμα για τη σύνδεση ηλεκτρονικών συσκευών και αισθητήρων μέσω του Διαδικτύου των Πραγμάτων (IoT). Όσον αφορά τα συμπεράσματα, διαπιστώθηκε ότι το παρόν σύστημα είναι καλύτερο και πιο αποδοτικό από άλλα συστήματα, ότι είναι εξαιρετικά συμβατό και μειώνει την ανθρώπινη προσπάθεια. Αναφέρθηκε ότι το παρόν έργο είναι μια τεράστια επιτυχία και θα παρέχει έναν σημαντικό τρόπο για την αποθήκευση παραμέτρων καιρού σε πραγματικό χρόνο, βοηθώντας αγρότες, βιομηχανίες, κανονικούς ανθρώπους και άλλους που έχουν καθημερινή σχέση με τον καιρό και τις παραμέτρους του. Επίσης, επισημάνθηκε ότι μπορεί να χρησιμοποιηθεί για την αναζήτηση πληροφοριών για κάθε περιοχή για πολλά χρόνια και ότι οι πληροφορίες αυτές μπορούν να χρησιμοποιηθούν για την καλύτερη ανάπτυξη φυτών στη γεωργία.

Η εργασία με τίτλο επίσης “*IOT BASED WEATHER MONITORING USING ARDUINO UNO, Sanket Gosavi et al., 2022*” [36] αναφέρεται στη δημιουργία ενός έξυπνου περιβάλλοντος με χρήση αισθητήρων, μικροελεγκτών και λογισμικών προγραμμάτων για αυτόματη παρακολούθηση και αυτοάμυνα. Συγκεκριμένα, το έργο αποσκοπεί στην πραγματοποίηση πραγματικού χρόνου παρακολούθησης και περιβαλλοντικών συνθηκών μέσω του Internet of Things (IoT) χρησιμοποιώντας αισθητήρες που είναι υπεύθυνοι για τη συλλογή δεδομένων θερμοκρασίας και υγρασίας. Αξίζει να αναφερθεί, πως το έργο ενώνει δύο πεδία έρευνας, τη συλλογή δεδομένων και τα συστήματα ελέγχου, με σκοπό τη δημιουργία ενός μαζικού συστήματος βάσης δεδομένων για τη συλλογή και παρουσίαση πληροφοριών. Το υπάρχον σύστημα είναι ανώτερο, αποτελεσματικότερο, αξιόπιστο και οικονομικότερο σε σχέση με άλλες εναλλακτικές λύσεις ενώ επίσης μειώνεται η ανθρώπινη προσπάθεια. Αυτό σημαίνει ότι η τρέχουσα πρωτοβουλία είναι επιτυχημένη και προσφέρει έναν σημαντικό τρόπο για την αποθήκευση παραμέτρων καιρού πραγματικού χρόνου, υποστηρίζοντας αγρότες, επιχειρήσεις και καθημερινούς ανθρώπους που επηρεάζονται από τον καιρό και τις παραμέτρους του.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Ένα ακόμη άρθρο με τίτλο “*IoT Based Environmental Monitoring System using Arduino UNO and Thingspeak, R. Deekshath et al., 2023*” [22] αναφέρεται στην αυξημένη σημασία της παρακολούθησης του περιβάλλοντος λόγω των αλλαγών στο κλίμα. Η εργασία παρουσιάζει τρεις διαφορετικούς ασύρματους αισθητήρες βασισμένους σε IoT για τον περιβαλλοντικό έλεγχο: έναν που χρησιμοποιεί την επικοινωνία Wi-Fi με βάση το πρωτόκολλο User Datagram Protocol (UDP), έναν που επικοινωνεί μέσω Wi-Fi και του πρωτοκόλλου Hypertext Transfer Protocol (HTTP) και έναν που χρησιμοποιεί την τεχνολογία Bluetooth Smart (BLE). Το σύστημα που βασίζεται σε UDP παρακολουθεί τη θερμοκρασία και τη σχετική υγρασία. Το Wi-Fi στέλνει πακέτα UDP ή HTTP σε μια πλατφόρμα Cloud, τα οποία είναι διαθέσιμα μόνο στον διαχειριστή που αποφασίζει εάν τα δεδομένα πρέπει να είναι δημόσια ή ιδιωτικά. Το BLE αποτελείται από αισθητήρες που τοποθετούνται σε διάφορες περιοχές και παράγουν ένα σήμα όταν λαμβάνουν δεδομένα, και ο εξυπηρετητής λαμβάνει τις πληροφορίες από τους αισθητήρες κάθε φορά που παράγεται το σήμα. Το άρθρο χρησιμοποιεί ένα Arduino UNO και ένα ασύρματο (Wi-Fi) module για την επεξεργασία και τη μεταφορά των ανιχνευμένων δεδομένων στο Cloud της Thingspeak. Έτσι, οι παράμετροι που λαμβάνονται αποθηκεύονται στην πλατφόρμα του Cloud (Thingspeak). Οι αλλαγές στο περιβάλλον ενημερώνονται σε μορφή βάσης δεδομένων μέσω της μεθόδου Cloud Computing. Σκοπός του συστήματος να παρέχει ένα αυτόματο σύστημα παρακολούθησης περιβαλλοντικών συνθηκών που δεν απαιτεί ανθρώπινη παρέμβαση.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

ΚΕΦΑΛΑΙΟ 3: ΜΕΘΟΔΟΛΟΓΙΑ

Στην ακόλουθη ενότητα, θα εξεταστεί η μεθοδολογία που ακολουθήθηκε για τον σχεδιασμό και την ανάπτυξη της εφαρμογής. Παράλληλα, θα διερευνηθούν οι στόχοι, τα προβλήματα και τα εμπόδια που προέκυψαν κατά τη διάρκεια της εξέλιξής της.

3.1 ΣΤΟΧΟΣ ΚΑΙ ΥΠΟΠΡΟΒΛΗΜΑΤΑ

Στην παρούσα εργασία, ο βασικός στόχος ήταν η δημιουργία μίας πλήρως λειτουργικής εφαρμογής τεχνολογίας οικιακού αυτοματισμού (full stack). Χρησιμοποιήθηκαν αλγόριθμοι επανάληψης και συστήματα κανόνων σε συνδυασμό με αισθητήρες όπου τοποθετήθηκαν πάνω σε συσκευή arduino, με σκοπό τη διαχείριση διάφορων έξυπνων συσκευών σε ένα οικιακό περιβάλλον. Στο πλαίσιο του back-end, χρησιμοποιήθηκαν οι γλώσσες προγραμματισμού Python, ενώ για το front-end χρησιμοποιήθηκαν οι JavaScript, HTML και CSS. Επιπλέον, εφαρμόστηκε το Thingspeak για την οπτικοποίηση των τιμών των δεδομένων που συλλέγονται από τους αισθητήρες μέσω διαγραμμάτων, παρέχοντας μία ευκρινή επισκόπηση των πληροφοριών.

Διακρίνονται τα εξής 4 υποπροβλήματα:

1. Την διεπαφή του χρήστη (front-end)
2. Την διαχείριση των δεδομένων από τους αισθητήρες καθώς και την επεξεργασία τους στο back-end
3. Την δυνατότητα εξυπηρέτησης http κλήσεων και μιας σταθερής επικοινωνίας socket στον ίδιο server
4. Την επικοινωνία όλων των παραπάνω με σκοπό την ολοκληρωμένη εμπειρία του χρήστη

Το υποπρόβλημα της ανάπτυξης της διεπαφής του χρήστη, γνωστής και ως front-end, αντιπροσωπεύει ένα σημαντικό κομμάτι της διαδικασίας δημιουργίας της εφαρμογής. Κατά τη διάρκεια της ανάπτυξης του front-end, ακολουθούνται πολλά βήματα για να δημιουργηθεί μια ολοκληρωμένη και χρηστική διεπαφή. Καταρχάς, γίνεται ο σχεδιασμός της διεπαφής, όπου προσδιορίζονται τα γραφικά στοιχεία, η δομή της σελίδας, και ο τρόπος αλληλεπίδρασης του χρήστη. Κατά την ανάπτυξη, χρησιμοποιούνται γλώσσες προγραμματισμού όπως η HTML για τη δομή του

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

περιεχομένου, η CSS για τη μορφοποίηση και τη στιλιστική, και η JavaScript για την υλοποίηση δυναμικών στοιχείων και λειτουργιών. Στη συνέχεια, προχωράμε στην υλοποίηση του σχεδιασμού, δημιουργώντας τις αντίστοιχες σελίδες και διασφαλίζοντας τη σωστή λειτουργία των διαφόρων στοιχείων. Κατά την ανάπτυξη, είναι σημαντικό να λαμβάνονται υπόψη τα πρότυπα σχεδιασμού και η ευχρηστία, προκειμένου να προσφέρεται μια ευχάριστη εμπειρία στο χρήστη. Τέλος, πραγματοποιείται ο έλεγχος και η δοκιμή της διεπαφής για να ανιχνευθούν και να επιλυθούν ενδεχόμενα προβλήματα και να εξασφαλιστεί η ομαλή λειτουργία της. Το αποτέλεσμα είναι μια προσεγμένη και αποτελεσματική διεπαφή που προσφέρει στον χρήστη μία ευχάριστη εμπειρία κατά τη χρήση της εφαρμογής.

Η διαχείριση των δεδομένων από τους αισθητήρες και η επεξεργασία τους στο back-end αποτελούν ένα κρίσιμο υποπρόβλημα κατά την ανάπτυξη της εφαρμογής, ιδιαίτερα όταν αυτή ενσωματώνει συσκευές IoT όπως αισθητήρες. Καταρχάς, πρέπει να σχεδιαστεί και να υλοποιηθεί η συλλογή δεδομένων από τους αισθητήρες, περιλαμβάνοντας την καθορισμένη συχνότητα λήψης δειγμάτων και τη μέθοδο μεταφοράς τους στο back-end. Στη συνέχεια, τα δεδομένα πρέπει να αποθηκεύονται αξιόπιστα και με ασφάλεια στο back-end, ώστε να είναι διαθέσιμα για επεξεργασία και ανάκτηση. Στο back-end, επίσης πρέπει να υλοποιηθούν μηχανισμοί επεξεργασίας δεδομένων που να εξάγουν την απαραίτητη πληροφορία από τα δεδομένα που δίνουν οι αισθητήρες. Αυτό περιλαμβάνει την εφαρμογή αλγορίθμων επεξεργασίας, εξόρυξης δεδομένων, ή άλλων τεχνικών ανάλυσης για την απόκτηση ενδιαφέρουσων πληροφοριών. Επιπλέον, είναι σημαντικό να επιτευχθεί η αποτελεσματική διαχείριση του όγκου των δεδομένων, περιλαμβάνοντας τη συμπίεση, τον έλεγχο ασφαλιμάτων, και την οργάνωση των δεδομένων για γρήγορη ανάκτηση. Συνολικά, η επιτυχημένη διαχείριση των δεδομένων από αισθητήρες και η αποτελεσματική τους επεξεργασία στο back-end είναι ουσιώδης για τη λειτουργία μιας εφαρμογής IoT, εξασφαλίζοντας τη σταθερή και αξιόπιστη λειτουργία του συστήματος.

Η δυνατότητα εξυπηρέτησης τόσο των HTTP κλήσεων όσο και μιας σταθερής επικοινωνίας μέσω socket στον ίδιο server αποτελεί επίσης σημαντική πτυχή της αρχιτεκτονικής ενός συστήματος. Καταρχάς, η δυνατότητα εξυπηρέτησης HTTP κλήσεων σημαίνει ότι ο εξυπηρετητής είναι σε θέση να απαντά σε αιτήσεις που γίνονται μέσω του πρωτοκόλλου HTTP. Αυτό επιτρέπει την ανταπόκριση σε αιτήσεις από πελάτες που χρησιμοποιούν έναν web browser ή άλλα εργαλεία που υποστηρίζουν το HTTP πρωτόκολλο. Παράλληλα, η σταθερή επικοινωνία μέσω socket επιτρέπει την συνεχή, διαρκή επικοινωνία μεταξύ των συσκευών και του εξυπηρετητή. Τα sockets παρέχουν ένα δυναμικό μέσο επικοινωνίας, επιτρέποντας τη μετάδοση δεδομένων σε πραγματικό χρόνο και τη δημιουργία

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

σταθερών συνδέσεων μεταξύ συσκευών και του server. Η συνδυασμένη δυνατότητα χρήσης HTTP και sockets επιτρέπει την ευελιξία και την καλύτερη κάλυψη των αναγκών του συστήματος, καθώς καλύπτει και τις περιπτώσεις που απαιτείται περισσότερο σταθερή και συνεχής επικοινωνία, καθώς και εκείνες που απαιτείται ανταπόκριση σε HTTP αιτήσεις. Αυτή η ολοκληρωμένη προσέγγιση βελτιώνει την απόκριση και την αποδοτικότητα του συστήματος.

Η επικοινωνία όλων των παραπάνω συστατικών στο πλαίσιο της αρχιτεκτονικής σχεδίασης επιδιώκει να παρέχει μια ολοκληρωμένη εμπειρία χρήστη με υψηλή απόδοση και ευελιξία. Στον πυρήνα του συστήματος βρίσκεται ο εξυπηρετητής, ο οποίος λαμβάνει δεδομένα από τους αισθητήρες που συλλέγουν πληροφορίες από το περιβάλλον. Ο server χρησιμοποιεί τόσο τις HTTP κλήσεις όσο και τα sockets για την επικοινωνία με το front-end και τους αισθητήρες. Ο χρήστης αλληλοεπιδρά με το front-end, το οποίο παρέχει μια φιλική προς τον χρήστη διεπαφή. Μέσω αυτού, ο χρήστης μπορεί να στέλνει HTTP αιτήσεις προς τον server για να ανακτήσει δεδομένα ή να εκτελέσει ενέργειες. Ο εξυπηρετητής, με τη σειρά του, διαχειρίζεται αυτές τις αιτήσεις και χρησιμοποιεί τα sockets για την αμφίδρομη επικοινωνία με τους αισθητήρες. Τα δεδομένα που λαμβάνονται από τους αισθητήρες επεξεργάζονται στο back-end, προτού αποσταλούν πίσω στο front-end για προβολή στον χρήστη. Με τον τρόπο αυτό, δημιουργείται ένα ολοκληρωμένο σύστημα όπου ο χρήστης μπορεί να αλληλοεπιδρά με τις συσκευές, να λαμβάνει πληροφορίες σε πραγματικό χρόνο και να εκτελεί ενέργειες, ενώ ταυτόχρονα ο εξυπηρετητής διαχειρίζεται τα δεδομένα με αποδοτικό τρόπο. Αυτή η συνεκτική επικοινωνία διασφαλίζει μια ομαλή και ολοκληρωμένη εμπειρία για τον χρήστη.

3.2 ΠΡΩΙΜΑ ΣΤΑΔΙΑ

Στα πρώτα στάδια υλοποίησης της εφαρμογής μία αρχική ιδέα ήταν να επικοινωνεί το front-end με τον server με τη χρήση socket. Ουσιαστικά το σκεπτικό ήταν να υπάρχει μία διαρκής επικοινωνία μεταξύ του user interface (front-end) και του server (back-end) που θα μεταφέρει δεδομένα από και προς το front-end, αναλόγως κάθε φορά βέβαια την επιλογή του χρήστη. Αντιμετωπίζοντας όμως δυσκολίες στον χειρισμό της σύνδεσης καθώς και αδυναμίες στη μεταφορά των δεδομένων, αποφασίστηκε να αναπτυχθεί μια εναλλακτική λύση χρησιμοποιώντας το πρωτόκολλο HTTP. Η σύνδεση αυτή, σε αντίθεση με αυτή του socket, θα ενεργοποιείται μόνο όταν υπάρχει κάποιο αίτημα από το front κομμάτι (πχ. GET, POST) προς τον server και μόνο τότε. Η ενσωμάτωση του HTTP στη σύνδεση τελικά επέτρεψε μια πιο σταθερή και αξιόπιστη ανταλλαγή δεδομένων μεταξύ του server και του front-end. Η αλλαγή αυτή οδήγησε στην αντιμετώπιση των

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

αρχικών προβλημάτων σύνδεσης, διευκολύνοντας την εύρυθμη λειτουργία της εφαρμογής σε πρώιμο στάδιο ανάπτυξης.

3.3 ΕΜΠΟΔΙΑ ΣΤΗΝ ΥΛΟΠΟΙΗΣΗ

Στη διαδικασία υλοποίησης της εφαρμογής, προέκυψαν ορισμένα εμπόδια που απαιτούσαν την αντιμετώπισή τους με προσοχή και καινοτομία. Ένα από τα κύρια προβλήματα ήταν η ανάγκη για ταυτόχρονη επικοινωνία μεταξύ του Arduino (συνδεδεμένων sensors) και του server, καθώς και του front-end και του server. Όσον αφορά τη σύνδεση μεταξύ του Arduino και του server, η επικοινωνία πρέπει να γίνεται συνεχώς, χωρίς διακοπές, για να διασφαλιστεί η σωστή λειτουργία των δεδομένων που δέχεται ο server από τους αισθητήρες.

Αντίστοιχα, η προσθήκη μιας σελίδας front-end είχε ως αποτέλεσμα την ανάγκη για ταυτόχρονη επικοινωνία μεταξύ του front-end και του server. Αυτό απαιτούσε μια λειτουργική υλοποίηση που να μπορεί να χειρίζεται πολλαπλές αιτήσεις ταυτόχρονα, εξασφαλίζοντας την αποδοτική ανταπόκριση και λειτουργία του συστήματος.

Για την αντιμετώπιση αυτών των προκλήσεων, χρειάστηκε να ενσωματωθούν σχετικά threads, συγκεκριμένα δύο, ένα στον socket server που επικοινωνεί με τα Arduino και ένα στον HTTP server που ανταποκρίνεται στα αιτήματα του front-end. Αυτή η προσέγγιση επιτρέπει την αποτελεσματική διαχείριση των αιτήσεων και την εξασφάλιση της σταθερής λειτουργίας του συστήματος, παρέχοντας ένα ευέλικτο περιβάλλον για την επικοινωνία μεταξύ των διαφόρων στοιχείων της εφαρμογής.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

ΚΕΦΑΛΑΙΟ 4: ΤΟ ΣΥΣΤΗΜΑ

Η διαδικασία ανάπτυξης μιας εφαρμογής αποτελεί μια πολύπλοκη διαδικασία, η οποία απαιτεί προσεκτικό σχεδιασμό, συντονισμένη συνεργασία και χρήση ποικίλων τεχνολογιών και εργαλείων. Στα πρώιμα στάδια, πρέπει να καθοριστούν οι λειτουργικές απαιτήσεις, να σχεδιαστεί η δομή της εφαρμογής, και να επιλεγούν οι τεχνολογίες που θα χρησιμοποιηθούν. Η διαχείριση της επικοινωνίας μεταξύ διαφορετικών συστατικών, όπως η σύνδεση με συσκευές IoT, απαιτεί εξειδικευμένες τεχνολογίες και πρωτόκολλα. Για την διευκόλυνση μίας τέτοιας διαδικασίας χρησιμοποιούνται πρωτόκολλα και τεχνολογίες, που υποστηρίζονται από το κοινό, με σκοπό την ελαχιστοποίηση των προβλημάτων καθώς και τεχνικές που θα βελτιώσουν την λειτουργικότητα και την αποδοτικότητα όπως η καλύτερη οργάνωση των αρχείων.

4.1 ΔΟΜΗ ΑΡΧΕΙΩΝ ΕΦΑΡΜΟΓΗΣ

Στην υλοποίηση της εφαρμογής, για να υπάρχει μία πιο δομημένη εικόνα και για να μην γίνει κάποιος λάθος, το project χωρίστηκε σε 2 φακέλους, ένα για το πίσω μέρος του συστήματος (Backend) και ένα για το μπροστινό (Frontend).

Στον φάκελο Backend, το μοναδικό αρχείο που χρειάζεται είναι το `server.py`, καθώς περιέχει όλη τη λογική που χρειάζεται ο εξυπηρετητής για να επικοινωνήσει ταυτόχρονα και με το Arduino καθώς και με το εμπρόσθιο κομμάτι που έρχεται σε επαφή ο χρήστης (Frontend). Είναι σημαντικό να αναφερθεί πως προαπαιτούνται και κάποιες απαραίτητες βιβλιοθήκες για την ορθή λειτουργία του.

Στον φάκελο Frontend, έχουμε τα εξής αρχεία: `interface.html`, `interface.js`, `interface.css`, `wallpaper.jpg`. Το πρώτο, περιέχει τον κώδικα HTML για το διεπαφικό μέρος της εφαρμογής. Περιλαμβάνει τη δομή της σελίδας, τις φόρμες, τα κουμπιά, τα κείμενα και άλλα στοιχεία που ορίζουν τη γραφική διεπαφή του χρήστη. Το `interface.js` αρχείο περιέχει κώδικα JavaScript, ο οποίος χρησιμοποιείται για την αλληλεπίδραση και τη δυναμική λειτουργία της σελίδας. Περιλαμβάνει λογική κώδικα για την ανταπόκριση σε συμβάντα όπως κλικ κουμπιών κ.λπ. Παίζει σημαντικό ρόλο για την ορθή επικοινωνία και λειτουργικότητα frontend μέσω server. Όσο για το αρχείο `interface.css`, περιέχει κώδικα CSS, ο οποίος καθορίζει την εμφάνιση και τη διάταξη των στοιχείων στη σελίδα. Οι κανόνες CSS περιγράφουν το στυλ, το χρώμα, τη γραμματοσειρά και άλλες γραφικές ιδιότητες.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Τέλος το αρχείο wallpaper.jpg αποτελεί μια εικόνα (στατικό περιεχόμενο) που χρησιμοποιείται ως φόντο (wallpaper) για τη σελίδα που χρησιμοποιεί ο χρήστης (user interface). Είναι μια εικόνα που φορτώνεται στη σελίδα για να προσθέσει εικαστική ή αισθητική αξία.

Backend	Frontend
server.py	interface.html
	interface.js
	interface.css
	wallpaper.jpg

Πίνακας 4.1: Δομή αρχείων εφαρμογής

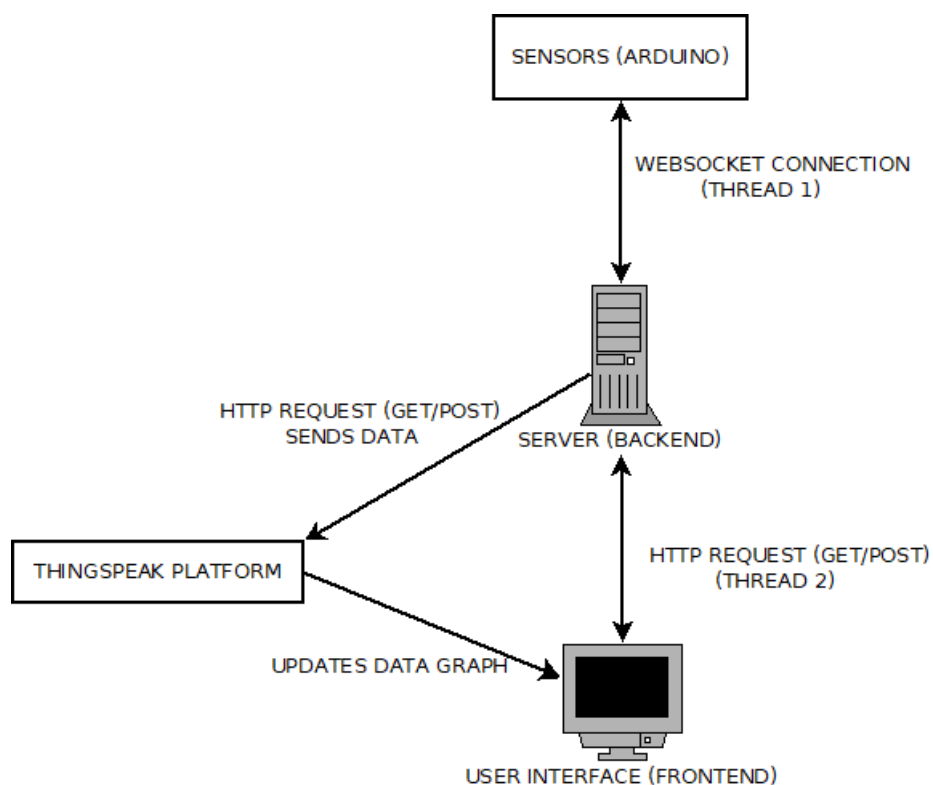
4.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΦΑΡΜΟΓΗΣ

Η αρχιτεκτονική του συστήματος αποτελείται από τρία κύρια μέρη: το Backend, το Frontend και το Arduino δηλαδή τα sensors. Καθένα από αυτά αναλαμβάνει συγκεκριμένο ρόλο για την ομαλή λειτουργία της εφαρμογής. Ο διακομιστής (Backend) αντιπροσωπεύει το κεντρικό “μυαλό” του συστήματος, χειρίζεται τη σύνδεση με το Arduino και αναλαμβάνει τη διαχείριση των αιτημάτων από το Frontend. Ο διακομιστής επεξεργάζεται τα δεδομένα που λαμβάνει από το Arduino και παρέχει μια ασφαλή διασύνδεση για την αποστολή και λήψη πληροφοριών με το Frontend, χρησιμοποιώντας το πρωτόκολλο επικοινωνίας REST API. Το Arduino, από την άλλη, λειτουργεί ως περιφερειακή συσκευή που αναλαμβάνει τη συλλογή δεδομένων από αισθητήρες και συγκεκριμένα θερμοκρασίας, υγρασίας και φωτός. Χρησιμοποιεί τη σειριακή επικοινωνία για τη μεταφορά αυτών των δεδομένων στον διακομιστή ενώ επίσης ανταποκρίνεται σε εντολές που λαμβάνει από το διακομιστή, διασφαλίζοντας την αμοιβαία αλληλεπίδραση με τα υπόλοιπα στοιχεία του συστήματος. Το Frontend αποτελεί το διαδραστικό πρόσωπο του συστήματος, παρουσιάζοντας τα δεδομένα στον χρήστη και διευκολύνοντας τη διαδικασία αλληλεπίδρασης. Συνεργάζεται με τον διακομιστή για τη λήψη και αποστολή δεδομένων, προσφέροντας μια ενιαία και φιλική προς τον χρήστη διεπαφή.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Όσον αφορά τώρα τις τεχνολογίες και βιβλιοθήκες που χρησιμοποιήθηκαν στην ανάπτυξη του συστήματος, επιλέχθηκαν εργαλεία που ενισχύουν την αποδοτικότητα, την αξιοπιστία και την ασφάλεια της εφαρμογής. Ο διακομιστής (server) βασίζεται στο Flask, ένα ευέλικτο και ελαφρύ πλαίσιο (framework) για τον σχεδιασμό των δικτυακών εφαρμογών. Η χρήση του Flask προσφέρει απλότητα στην ανάπτυξη και τη συντήρηση του διακομιστή, ενώ η ενσωμάτωση της βιβλιοθήκης Flask-CORS διευκολύνει την χειρισμό του Cross-Origin Resource Sharing, εξασφαλίζοντας πιο εύκολη επικοινωνία με το Frontend. Για την αμφίδρομη επικοινωνία μεταξύ του διακομιστή και του Arduino, χρησιμοποιήθηκε η βιβλιοθήκη websockets, που υποστηρίζει τη σύνδεση μέσω του πρωτοκόλλου WebSocket. Αυτό διευκολύνει τη διαρκή ανταλλαγή δεδομένων μεταξύ των δύο στοιχείων, διασφαλίζοντας την επικοινωνία σε πραγματικό χρόνο. Επιπλέον, η χρήση της βιβλιοθήκης threading και asyncio επιτρέπει την παράλληλη εκτέλεση διαφορετικών διεργασιών, βελτιώνοντας την απόδοση του συστήματος.



Σχήμα 4.2: Διάγραμμα αρχιτεκτονικής εφαρμογής

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

4.2.1 ΒΙΒΛΙΟΘΗΚΗ FLASK

Η Flask αναλαμβάνει τον ρόλο του βασικού πλαισίου (framework) για τον διακομιστή (Backend) της εφαρμογής, καθιστώντας τον “πυρήνα” πίσω από τη διαχείριση και τη λειτουργία των διαδικτυακών αιτημάτων. Με το χαρακτηριστικό της χαμηλής πολυπλοκότητας και της ευελιξίας, επιτρέπει τη δημιουργία ισχυρών διακομιστών με ελάχιστον κώδικα. Η δομή της Flask προσφέρει μια εύκολη και ευανάγνωστη μέθοδο για τον σχεδιασμό των διαδρομών (routes), οι οποίες αντιπροσωπεύουν τις διάφορες δυνατότητες και λειτουργίες του διακομιστή. Επίσης, διαχειρίζεται ευέλικτα τα αιτήματα HTTP που λαμβάνει από το Frontend, επιτρέποντας την αποτελεσματική επικοινωνία μεταξύ των διαφόρων συστατικών της εφαρμογής. Η Flask συνδυάζει αποδοτικότητα και ευελιξία, προσφέροντας μια απλή δομή για την οργάνωση της επιχειρησιακής λογικής. Με αυτόν τον τρόπο, διευκολύνει τη συντήρηση του κώδικα, καθώς οργανώνει τη λογική του διακομιστή με έναν τρόπο που είναι ευανάγνωστος και ευαίσθητος στις μελλοντικές αλλαγές. Συνολικά, αποτελεί ένα ισχυρό εργαλείο για την ανάπτυξη αποτελεσματικών και ευέλικτων διακομιστών που ανταποκρίνονται αποτελεσματικά στις απαιτήσεις της εφαρμογής μας.

4.2.2 ΒΙΒΛΙΟΘΗΚΗ FLASK – CORS

Η Flask-CORS αναλαμβάνει τον σημαντικό ρόλο της διαχείρισης προβλημάτων Cross-Origin Resource Sharing (CORS) μεταξύ του διακομιστή (Backend) και του Frontend, εξασφαλίζοντας την ασφαλή και αποτελεσματική επικοινωνία μεταξύ δύο διαφορετικών πηγών. Το πρόβλημα του CORS προκύπτει όταν ο Frontend τρέχει σε διαφορετικό domain, πρωτόκολλο ή port από τον διακομιστή. Αυτό μπορεί να δημιουργήσει προκλήσεις ασφαλείας και να εμποδίσει τα αιτήματα AJAX (Asynchronous JavaScript and XML) από το Frontend να εκτελούνται με επιτυχία στον διακομιστή. Τα αιτήματα AJAX αναφέρονται σε τεχνικές που χρησιμοποιούνται στον προγραμματισμό ιστού για να επιτρέψουν την ασύγχρονη ανταλλαγή δεδομένων μεταξύ του περιηγητή του χρήστη και του διακομιστή, χωρίς την ανάγκη για ανανέωση της σελίδας. Η Flask-CORS, λοιπόν, λύνει αυτό το θέμα, επιτρέποντας στον διακομιστή να δίνει τις κατάλληλες άδειες πρόσβασης στο Frontend, εξασφαλίζοντας έτσι τη σωστή ανταλλαγή δεδομένων. Η Flask-CORS επιτρέπει την προσαρμογή και τον έλεγχο των ρυθμίσεων CORS, όπως τις επιτρεπόμενες προελεύσεις, τις υποστηριζόμενες μεθόδους, και τις προσαρμοσμένες κεφαλίδες. Αυτό διασφαλίζει ότι οι αιτήσεις που προέρχονται από το Frontend είναι ασφαλείς και επιτρεπτές, βοηθώντας στην αποτελεσματική επικοινωνία μεταξύ των

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

δύο συστατικών της εφαρμογής. Κατ' αυτόν τον τρόπο, η Flask-CORS συνεισφέρει στη σταθερότητα και στην ασφαλή λειτουργία της εφαρμογής, αποφεύγοντας προβλήματα που σχετίζονται με την ασύμφωνη προέλευση των αιτημάτων.

4.2.3 ΒΙΒΛΙΟΘΗΚΗ WEBSOCKETS

Η βιβλιοθήκη websockets χρησιμοποιείται για την διαχείριση συνεχούς και αμφίδρομης επικοινωνίας μεταξύ του διακομιστή και του Arduino. Εκμεταλλεύεται το πρωτόκολλο WebSocket, το οποίο διασφαλίζει τη μεταφορά δεδομένων σε πραγματικό χρόνο, κάνοντας δυνατή την άμεση ενημέρωση του διακομιστή με τις τρέχουσες τιμές που λαμβάνονται από το Arduino. Μέσω της αρχιτεκτονικής του πρωτοκόλλου WebSocket, η websockets επιτρέπει τη σύνδεση πολλών clients στον διακομιστή, επιτρέποντας έτσι την ταυτόχρονη διαχείριση πολλαπλών συσκευών. Αυτό αποτελεί ουσιαστικό στοιχείο στο πλαίσιο του Internet of Things (IoT), όπου η ανάγκη για συνεχή επικοινωνία και ενημέρωση είναι καίρια. Η δυνατότητα αμφίδρομης επικοινωνίας μεταξύ διακομιστή και Arduino διευκολύνει τη μεταφορά δεδομένων των αισθητήρων και την αποστολή εντολών προς τη συσκευή, επιτρέποντας έτσι τον έλεγχο και την παρακολούθηση σε πραγματικό χρόνο. Η βιβλιοθήκη συνδυάζει την αξιοπιστία του πρωτοκόλλου WebSocket με την ευελιξία προσαρμογής σε διάφορα σενάρια εφαρμογών, καθιστώντας την ένα απαραίτητο εργαλείο για την ανάπτυξη συστημάτων IoT που απαιτούν άμεση αλληλεπίδραση.

4.2.4 ΒΙΒΛΙΟΘΗΚΕΣ THREADING ΚΑΙ ASYNCIO

Στην εφαρμογή, οι βιβλιοθήκες threading και asyncio επιτρέπουν την παράλληλη εκτέλεση διαφορετικών διεργασιών στον διακομιστή, βελτιώνοντας την απόδοση και την αποτελεσματικότητα της εφαρμογής.

Η βιβλιοθήκη threading χρησιμοποιείται για τη δημιουργία νημάτων (threads) που εκτελούν διαφορετικές λειτουργίες ταυτόχρονα. Για παράδειγμα, η ανάγνωση δεδομένων από το Arduino και η επικοινωνία με το Frontend μπορούν να εκτελούνται σε διαφορετικά νήματα, επιτρέποντας στον διακομιστή να ανταποκρίνεται ταυτόχρονα σε πολλές λειτουργίες χωρίς να αποκλείει το ένα το άλλο.

Από την άλλη πλευρά, η βιβλιοθήκη asyncio επιτρέπει την ασύγχρονη εκτέλεση κώδικα, ειδικά όταν υπάρχουν διακοπές στην εκτέλεση, όπως κατά τη διάρκεια αναμονής για δεδομένα από το Arduino ή για αιτήματα από το Frontend. Η ασύγχρονη προγραμματιστική προσέγγιση με τη χρήση

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

του `asyncio` επιτρέπει την αποτελεσματική διαχείριση των διακοπών και των αναμονών, εξασφαλίζοντας ότι ο διακομιστής είναι πάντα ενεργός και αποτελεσματικός.

Συνολικά, η χρήση των βιβλιοθηκών `threading` και `asyncio` συμβάλλει στη δημιουργία ενός αποδοτικού και ανταποκρίνοντος διακομιστή, που μπορεί να διαχειρίζεται παράλληλα πολλαπλές λειτουργίες και αιτήματα, βελτιώνοντας την εμπειρία του χρήστη και εξασφαλίζοντας τη σταθερότητα της εφαρμογής.

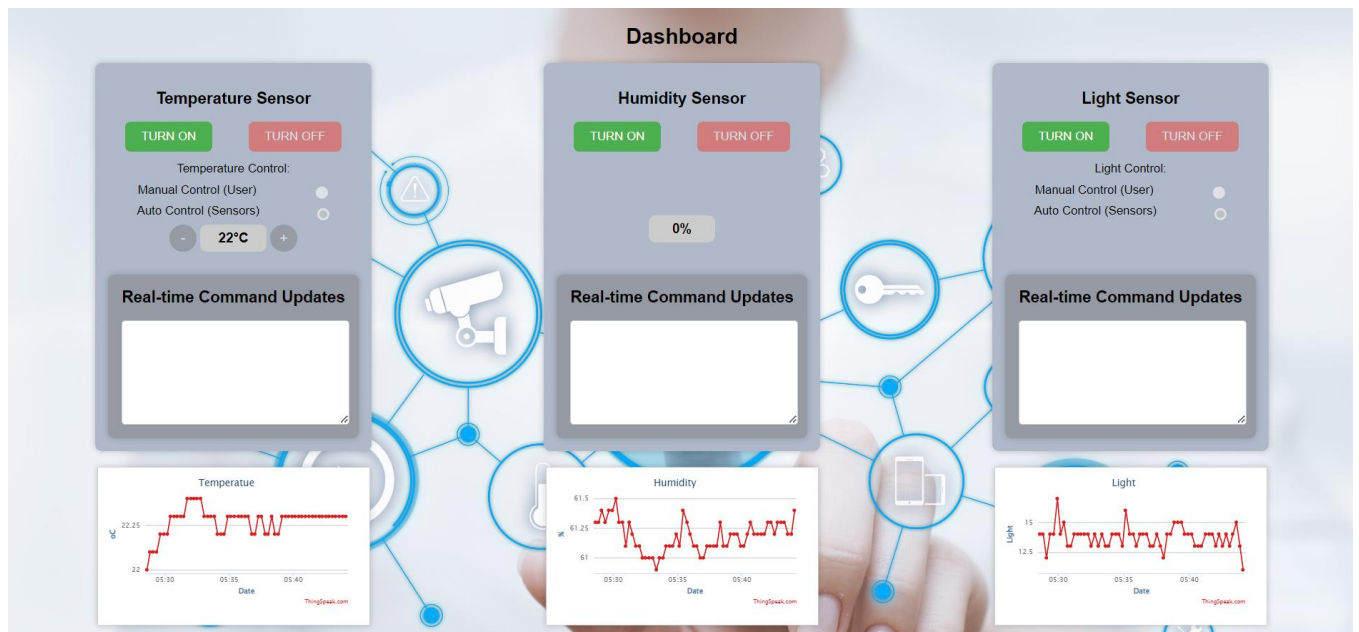
4.3 ΕΜΠΡΟΣΘΙΟ ΜΕΡΟΣ (FRONTEND)

Για το Frontend μέρος της εφαρμογής χρησιμοποιείται μία κεντρική σελίδα ή αλλιώς Interface από το οποίο ο χρήστης έχει τον πλήρη έλεγχο των συνδεδεμένων έξυπνων συσκευών με το λεγόμενο dashboard. Αξίζει να αναφερθεί πως ο χρήστης έχει την δυνατότητα να ελέγχει παράλληλα πολλαπλές συσκευές από το ίδιο μενού (dashboard). Συγκεκριμένα υπάρχουν τρεις κατηγορίες ελέγχου λόγω των `sensors` που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής, βέβαια υπάρχει δυνατότητα επέκτασης της εφαρμογής με παραπάνω αισθητήρες, συσκευές κλπ. Οι τρεις κατηγορίες είναι βασισμένες στο τύπο του κάθε αισθητήρα οπότε έχουμε τις εξής: `Temperature`, `Humidity`, `Light`. Σε κάθε κατηγορία παρουσιάζονται τα γραφήματα (`graphs`) που αντιπροσωπεύουν τις πραγματικές τιμές `real-time` για τον κάθε τύπο `sensor`. Αυτά τα γραφήματα είναι χρήσιμα για την οπτικοποίηση των μετρήσεων και την ευκολία στην αντίληψη των αλλαγών στις τιμές.

Επιπλέον, είναι σημαντικό να σημειωθεί ότι, κατά την εκκίνηση του dashboard, όλες οι συσκευές είναι απενεργοποιημένες, παρέχοντας έναν ασφαλή έλεγχο των συσκευών πριν την ενεργοποίησή τους. Η δομή αυτή του dashboard δίνει στον χρήστη τη δυνατότητα να ελέγχει και να παρακολουθεί τις έξυπνες συσκευές του με άνεση και αποτελεσματικότητα.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων



Εικόνα 4.3.1: Interface εφαρμογής (Dashboard)

4.3.1 ΔΕΔΟΜΕΝΑ ΘΕΡΜΟΚΡΑΣΙΑΣ

Temperature Sensor

Σε αυτή τη κατηγορία αισθητήρα έχει υλοποιηθεί η σύνδεση για την λειτουργία και τον έλεγχο ενός κλιματιστικού (A/C). Η ενεργοποίηση του κλιματιστικού είναι εξαιρετικά απλή, απαιτώντας μόνο το πάτημα του πράσινου κουμπιού "TURN ON". Αμέσως μετά την ενέργεια αυτή, εμφανίζεται κατάλληλο μήνυμα στη διεπαφή προκειμένου να ενημερώσει τον χρήστη για την επιτυχή ενεργοποίηση.

Περίπτωση αυτόματης λειτουργίας

Αφού ενεργοποιηθεί το κλιματιστικό, ξεκινάει λειτουργώντας σε λειτουργία αυτόματου ελέγχου θερμοκρασίας (Auto Control - Προεπιλογή). Αυτή η λειτουργία βασίζεται στα δεδομένα που παρέχονται από τους αισθητήρες, επιτρέποντας στο κλιματιστικό να προσαρμόζει αυτόματα τη θερμοκρασία σύμφωνα με τις συνθήκες περιβάλλοντος. Με αυτόν τον τρόπο, παρέχεται ένα αυτοματοποιημένο και άνετο περιβάλλον, ενώ ο χρήστης ενημερώνεται μέσω της διεπαφής για την τρέχουσα κατάσταση. Αυτή η λειτουργικό-τητα προσφέρει μια ολοκληρωμένη εμπειρία χρήστη και

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

βελτιστοποιεί την απόδοση του κλιματιστικού για την επίτευξη της βέλτιστης θερμοκρασίας στον χώρο. Αυτή η αυτόματη λειτουργία είναι βασισμένη σε ένα σύστημα κανόνων που έχει δημιουργηθεί για τον έλεγχο του κλιματιστικού. Αυτό το σύστημα κανόνων είναι σχεδιασμένο να αναλύει δεδομένα από τους αισθητήρες και να λαμβάνει αποφάσεις που βελτιστοποιούν την άνεση και την απόδοση του κλιματιστικού. Η λειτουργία του συστήματος αυτού αναλύεται ακριβώς παρακάτω.

Λειτουργία του Auto Control (Temperature)

Η λειτουργία του αυτόματου ελέγχου θερμοκρασίας του κλιματιστικού βασίζεται σε ένα σύστημα κανόνων που προσαρμόζει τη θερμοκρασία ανάλογα με τις τρέχουσες συνθήκες. Η θερμοκρασία, μετρούμενη σε βαθμούς Κελσίου (°C), αποτελεί κρίσιμο παράγοντα για την άνεση και την ευημερία σε έναν χώρο.

Σύμφωνα με το σύστημα κανόνων, όταν η θερμοκρασία είναι κάτω από κάποιους βαθμούς Κελσίου, αυξάνεται στο κατώτατο όριο που έχει οριστεί για να διασφαλίσει τη διατήρηση μιας ελάχιστης θερμοκρασίας. Όταν η θερμοκρασία παραμένει σταθερή, σημαίνει πως είναι εντός των καθορισμένων ορίων. Σε περίπτωση που η θερμοκρασία υπερβεί το ανώτατο όριο θερμοκρασίας, μειώνεται στο ανώτατο όριο που έχει οριστεί από το σύστημα, επιδιώκοντας τη διατήρηση μιας ευχάριστης θερμοκρασίας στον χώρο. Κάθε ενέργεια συνοδεύεται από κατάλληλο μήνυμα για ενημέρωση του χρήστη. Αυτή η δυναμική προσαρμογή συμβάλλει στη δημιουργία ενός άνετου και ευχάριστου περιβάλλοντος.

Η λειτουργία του αυτόματου ελέγχου θερμοκρασίας κλιματιστικού βασίστηκε στο εξής σύστημα κανόνων:

- Θερμοκρασία $< 10^{\circ}\text{C}$: Αυξάνεται η θερμοκρασία στο **κατώτατο** καθορισμένο όριο δηλαδή τους 10°C
- $10^{\circ}\text{C} < \text{Θερμοκρασία} < 30^{\circ}\text{C}$: Η θερμοκρασία μένει ίδια καθώς είναι εντός των καθορισμένων ορίων
- Θερμοκρασία $> 30^{\circ}\text{C}$: Μειώνεται η θερμοκρασία στο **ανώτατο** καθορισμένο όριο δηλαδή τους 30°C

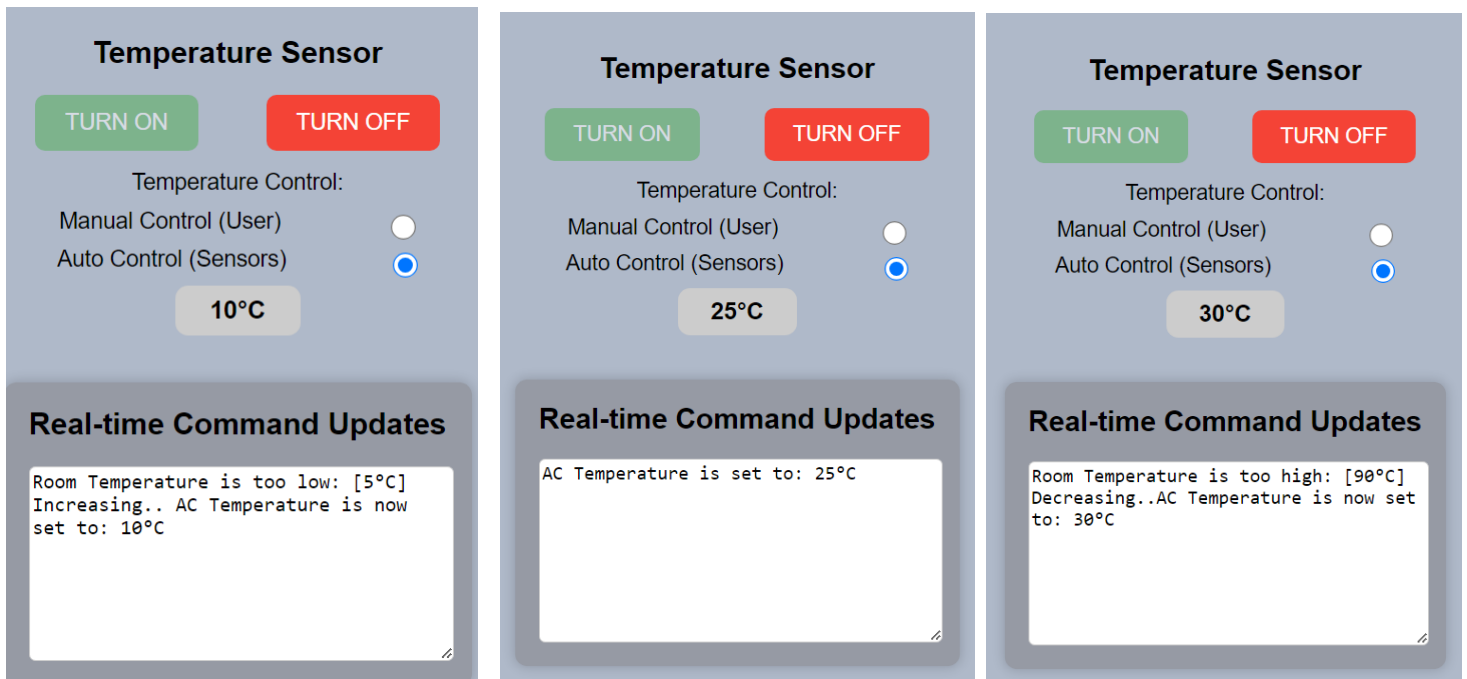
Για κάθε μία ενέργεια από τις παραπάνω εμφανίζεται και ένα κατάλληλο μήνυμα στο χρήστη.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Τα όρια θερμοκρασίας ορίστηκαν με βάση τις τυπικές τιμές που αναμένεται να παρουσιάζει ένα δωμάτιο.

Παρακάτω υπάρχει η απεικόνιση των τριών διαφορετικών περιπτώσεων που αναφέρθηκαν προηγουμένως.



Εικόνα 4.3.2: Διαφορετικές περιπτώσεις ενεργειών αυτόματου ελέγχου κλιματιστικού (Auto Control)

Στην πρώτη εικόνα, η θερμοκρασία του δωματίου είναι πολύ χαμηλή, στους 5°C. Στη συνέχεια το κλιματιστικό αυξάνει τη θερμοκρασία στους 10°C, εντός των ορίων. Ένα μήνυμα εμφανίζεται προς τον χρήστη, ενημερώνοντας ότι η θερμοκρασία του δωματίου ήταν πολύ χαμηλή (Room Temperature is too low: [5°C]) και ότι έχει αυξηθεί, ορίζοντας τώρα τη θερμοκρασία του κλιματιστικού στους 10°C.

Στη δεύτερη εικόνα, η θερμοκρασία στο δωμάτιο είναι κανονική, στα 25°C. Δεν απαιτείται καμία ενέργεια για το κλιματιστικό, και ένα μήνυμα εμφανίζεται ενημερώνοντας τον χρήστη ότι η θερμοκρασία του κλιματιστικού έχει οριστεί στους 25°C.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Στην τρίτη εικόνα, η θερμοκρασία του δωματίου ανέρχεται σε πολύ υψηλά επίπεδα, στους 90°C. Ο χρήστης λαμβάνει ένα μήνυμα που τον ενημερώνει για την υψηλή θερμοκρασία του δωματίου (Room Temperature is too high: [90°C]) και στη συνέχεια η θερμοκρασία του κλιματιστικού ενημερώνεται στους 30°C, εντός των ορίων. Ένα νέο μήνυμα εμφανίζεται, ενημερώνοντας τον χρήστη ότι η θερμοκρασία του κλιματιστικού έχει μειωθεί και είναι πλέον ρυθμισμένη στους 30°C.

Περίπτωση χειροκίνητης λειτουργίας

Στην περίπτωση που ο χρήστης επιθυμεί να αναλάβει τον έλεγχο του κλιματιστικού προσωπικά, μπορεί να επιλέξει τη λειτουργία χειροκίνητου ελέγχου (**Manual Control**). Τα κουμπιά αύξησης (+) και μείωσης (-) της θερμοκρασίας του κλιματιστικού του παρέχουν τη δυνατότητα να προσαρμόσει ελεύθερα τη θερμοκρασία σύμφωνα με τις προτιμήσεις του.

Είναι σημαντικό να σημειωθεί ότι σε αυτήν τη λειτουργία, δεν υπάρχουν περιορισμοί από το σύστημα κανόνων. Ο χρήστης έχει πλήρη έλεγχο και ευθύνη για τη ρύθμιση της θερμοκρασίας, εξασφαλίζοντας έτσι απόλυτη ατομική προσαρμογή. Αυτό δίνει στον χρήστη την ελευθερία να προσαρμόσει το κλίμα σύμφωνα με τις προσωπικές του προτιμήσεις, επιτρέποντάς του να διαμορφώσει το περιβάλλον του χώρου του ακριβώς όπως το επιθυμεί. Η χρήση των κουμπιών (+) και (-) απλοποιεί τη διαδικασία, ενώ ο χρήστης ενημερώνεται μέσω της διεπαφής για την τρέχουσα θερμοκρασία.



Εικόνα 4.3.3: Μήνυμα που λαμβάνει ο χρήστης κατά τον χειροκίνητο έλεγχο θερμοκρασίας (Manual Control)

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εκτός από τις δύο κύριες λειτουργίες της αυτόματης και χειροκίνητης ρύθμισης της θερμοκρασίας, η εφαρμογή προσφέρει επίσης τη δυνατότητα απενεργοποίησης της συσκευής. Η επιλογή απενεργοποίησης επιτρέπει στον χρήστη να τερματίσει το κλιματιστικό όταν δεν απαιτείται η λειτουργία του. Αυτή η επιλογή προσφέρει όχι μόνο εξοικονόμηση ενέργειας, αλλά και τη δυνατότητα ενεργοποίησης της συσκευής μόνο όταν είναι απαραίτητη. Με αυτόν τον τρόπο, ο χρήστης έχει πλήρη έλεγχο του πότε θα χρησιμοποιείται η συσκευή, βελτιστοποιώντας την ενεργειακή απόδοση του κλιματιστικού. Η διαθεσιμότητα αυτής της επιλογής ενισχύει την ολοκληρωμένη και εξατομικευμένη διαχείριση του περιβάλλοντος, προσφέροντας ευελιξία και απόλυτο έλεγχο στον χρήστη.

4.3.2 ΔΕΔΟΜΕΝΑ ΥΓΡΑΣΙΑΣ

Humidity Sensor

Σε αυτή τη κατηγορία αισθητήρα έχει υλοποιηθεί η σύνδεση για την λειτουργία ενός αφυγραντήρα, η ενεργοποίησή του συνδέεται με το κουμπί "TURN ON" στη διεπαφή. Όταν ο χρήστης επιλέγει την εν λόγω λειτουργία, ο αφυγραντήρας ενεργοποιείται, ξεκινώντας τη διαδικασία αφύγρυνσης του χώρου. Ο αφυγραντήρας λειτουργεί με βάση το σύστημα κανόνων που προκαθορίζει την αυτόματη ρύθμιση της υγρασίας στον χώρο. Η λογική πίσω από το σύστημα αυτό βασίζεται σε προκαθορισμένες τιμές υγρασίας, όπου ο αφυγραντήρας ενεργοποιείται ή απενεργοποιείται ανάλογα με τις ανάγκες του περιβάλλοντος.

Το σύστημα κανόνων για τον έλεγχο της υγρασίας βοηθά στη διατήρηση του κατάλληλου επιπέδου υγρασίας στο χώρο, εξασφαλίζοντας ένα άνετο και υγιές περιβάλλον. Επιπλέον, ο χρήστης έχει τη δυνατότητα να απενεργοποιήσει τον αφυγραντήρα, προσφέροντας έτσι επιπλέον έλεγχο και ευελιξία στη χρήση της συσκευής. Με αυτόν τον τρόπο, ο χρήστης μπορεί να προσαρμόσει τη λειτουργία του αφυγραντήρα ανάλογα με τις προτιμήσεις και τις συγκεκριμένες ανάγκες του, επιτρέποντάς του να διαμορφώσει το περιβάλλον του χώρου του.

Λειτουργία του Auto Control (Humidity)

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Το ποσοστό σχετικής υγρασίας (**RH%**) αντιπροσωπεύει το ποσοστό της υγρασίας που περιέχεται στην ατμόσφαιρα σε σχέση με τη μέγιστη ποσότητα υγρασίας που θα μπορούσε να κρατήσει ο αέρας στην τρέχουσα θερμοκρασία. Εκφράζεται σε ποσοστό και χρησιμοποιείται για να περιγράψει την ποσότητα της υγρασίας σε ένα δεδομένο περιβάλλον. Το ποσοστό αυτό είναι κρίσιμο για την άνεση και την υγεία, καθώς επηρεάζει την αναπνοή, την αίσθηση δροσιάς ή υγρασίας, και τη διατήρηση του χώρου.

Η λειτουργία του αυτόματου ελέγχου υγρασίας του αφυγραντήρα βασίζεται σε ένα σύστημα κανόνων που αντιδρά στις τρέχουσες τιμές της υγρασίας. Όταν η υγρασία είναι κάτω από το κατώτατο όριο που έχει οριστεί, ο αφυγραντήρας αυξάνει την υγρασία στο κατώτατο καθορισμένο όριο. Όταν η υγρασία κυμαίνεται ανάμεσα στα όρια που έχουν δοθεί από το σύστημα, ο αφυγραντήρας διατηρεί την υγρασία στα ίδια επίπεδα, ενώ όταν υπερβαίνει το ανώτατο όριο, μειώνει την υγρασία στο ανώτατο καθορισμένο όριο. Αυτή η δυναμική προσαρμογή εξασφαλίζει ότι ο αέρας στον χώρο παραμένει σε επίπεδα υγρασίας που θεωρούνται άνετα και υγιεινά.

Η λειτουργία του αυτόματου ελέγχου υγρασίας αφυγραντήρα βασίστηκε στο εξής σύστημα κανόνων:

- Υγρασία < **30%** : Αυξάνεται η υγρασία στο **κατώτατο** καθορισμένο όριο δηλαδή τους 30%
- **30%** < Υγρασία < **70%** : Η υγρασία μένει ίδια καθώς είναι εντός των καθορισμένων ορίων
- Υγρασία > **70%** : Μειώνεται η υγρασία στο **ανώτατο** καθορισμένο όριο δηλαδή τους 70%

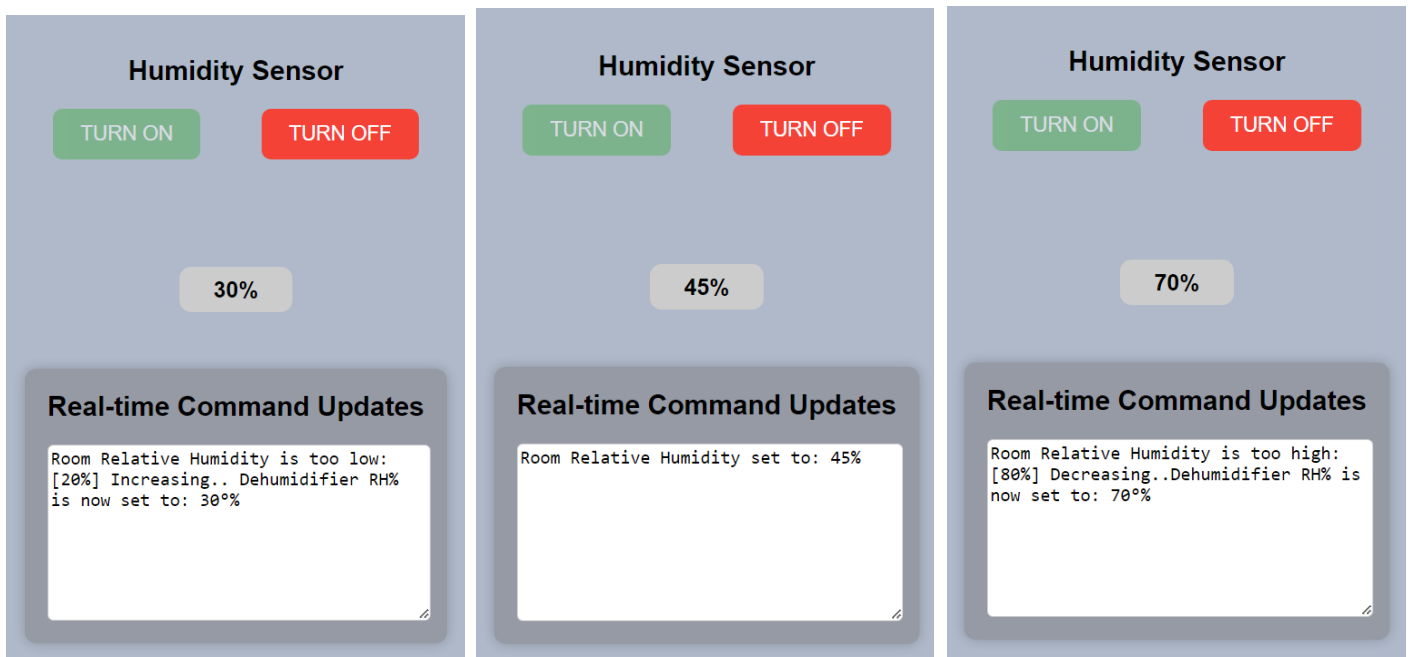
Για κάθε μία ενέργεια από τις παραπάνω εμφανίζεται και ένα κατάλληλο μήνυμα στο χρήστη.

Τα όρια υγρασίας ορίστηκαν με βάση τις τυπικές τιμές που αναμένεται να παρουσιάζει ένα δωμάτιο.

Παρακάτω υπάρχει η απεικόνιση των τριών διαφορετικών περιπτώσεων που αναφέρθηκαν προηγουμένως.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων



Εικόνα 4.3.4: Διαφορετικές περιπτώσεις ενεργειών αυτόματου ελέγχου αφυγραντήρα (Auto Control)

Ακολουθεί η επεξήγηση των παραπάνω εικόνων:

Στην πρώτη εικόνα, παρατηρούμε ότι η υγρασία του δωματίου είναι πολύ χαμηλή (RH = 20%). Σε αυτήν την περίπτωση, ενεργοποιείται ο αφυγραντήρας για να αυξήσει το ποσοστό υγρασίας, καταφέροντας να το αυξήσει στο 30%. Το αντίστοιχο μήνυμα που εμφανίζεται είναι: "Room Relative Humidity is too low: [20%] Increasing.. Dehumidifier RH% is now set to: 30%" (Η υγρασία του δωματίου είναι πολύ χαμηλή: [20%] Αυξάνεται.. Το ποσοστό υγρασίας του αφυγραντήρα τώρα έχει οριστεί σε: 30%).

Στη δεύτερη εικόνα, η υγρασία του δωματίου είναι φυσιολογική (RH = 45%) και, ως εκ τούτου, ο αφυγραντήρας δεν απαιτεί κάποια ενέργεια. Το μήνυμα που εμφανίζεται είναι: "Room Relative Humidity set to: 45%" (Η υγρασία του δωματίου έχει οριστεί σε: 45%).

Στην τρίτη εικόνα, παρατηρούμε ότι η υγρασία του δωματίου είναι πολύ υψηλή (RH = 80%). Εδώ, ενεργοποιείται ο αφυγραντήρας για να μειώσει την υγρασία, φέρνοντάς την στο 70%. Το μήνυμα που εμφανίζεται είναι: "Room Relative Humidity is too high: [80%] Increasing..".

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Dehumidifier RH% is now set to: 70%" (Η υγρασία του δωματίου είναι πολύ υψηλή: [80%]
Μειώνεται.. Το ποσοστό υγρασίας του αφυγραντήρα τώρα έχει οριστεί σε: 70%).

4.3.3 ΔΕΔΟΜΕΝΑ ΦΩΤΕΙΝΟΤΗΤΑΣ

Light Sensor

Σε αυτή την κατηγορία αισθητήρα, η σύνδεση του συστήματος με μία λάμπα ή ένα φωτιστικό επιτρέπει τον έξυπνο έλεγχο του φωτισμού στον χώρο. Κατά την ενεργοποίηση με το κουμπί "TURN ON" στη διεπαφή, το φωτιστικό ξεκινάει με την προεπιλεγμένη ρύθμιση στη λειτουργία "**Auto Control**". Αμέσως μετά την ενέργεια αυτή, εμφανίζεται στη διεπαφή ένα κατάλληλο μήνυμα που ενημερώνει τον χρήστη για την κατάσταση που βρίσκεται.

Περίπτωση αυτόματης λειτουργίας

Ο αισθητήρας φωτός, που μετράει σε **lux** τη φωτεινότητα, αποτελεί κρίσιμο στοιχείο του συστήματος, που σε συνδυασμό με τον αυτόματο έλεγχο φωτισμού βασισμένο σε ένα προκαθορισμένο σύστημα κανόνων, δημιουργεί ένα έξυπνο και αποδοτικό περιβάλλον φωτισμού. Ο αισθητήρας μετρά την ένταση του φωτισμού σε μονάδες lux, προσφέροντας πληροφορίες σχετικά με την ποσότητα του φωτός που φτάνει στον χώρο. Ο συνδυασμός αυτών των μετρήσεων με τον αυτόματο έλεγχο φωτισμού βασισμένο σε συγκεκριμένους κανόνες επιτρέπει στο σύστημα να αντιδρά συνετά στις συνθήκες φωτισμού. Οι κανόνες αυτοί μπορεί να περιλαμβάνουν διάφορες συμπεριφορές ανάλογα με την ένταση του φωτός. Για παράδειγμα, σε υψηλά επίπεδα φωτεινότητας, το σύστημα μπορεί να επιλέξει να απενεργοποιήσει φωτιστικά για εξοικονόμηση ενέργειας ενώ αντίστοιχα, σε χαμηλά επίπεδα φωτεινότητας, το σύστημα μπορεί να ενεργοποιήσει τα φωτιστικά για άνετη χρήση.

Αυτή η έξυπνη αλληλεπίδραση μεταξύ του αισθητήρα φωτός και του συστήματος κανόνων εξασφαλίζει όχι μόνο τη βέλτιστη ορατότητα και άνεση στον χώρο, αλλά και την αποτελεσματική χρήση της ενέργειας.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Λειτουργία του Auto Control (Light)

Στη λειτουργία αυτή, το σύστημα βασίζεται στις μετρήσεις του φωτοαισθητήρα και το σύστημα κανόνων για να προσαρμόσει τον φωτισμό. Αν η φωτεινότητα είναι πολύ υψηλή, υποθέτοντας ότι ο χώρος έχει αρκετό φυσικό φως, το φωτιστικό απενεργοποιείται για εξοικονόμηση ενέργειας. Αντίθετα, όταν η φωτεινότητα είναι χαμηλή, υποθέτοντας ότι ο χώρος χρειάζεται περισσότερο φως, το φωτιστικό ανάβει για να παρέχει τον απαιτούμενο φωτισμό. Η ευφυΐα αυτή λειτουργία εξασφαλίζει ότι ο φωτισμός προσαρμόζεται δυναμικά σύμφωνα με τις συνθήκες του περιβάλλοντος, επιτυγχάνοντας ισορροπία μεταξύ άνεσης και εξοικονόμησης ενέργειας.

Η λειτουργία του αυτόματου ελέγχου έντασης φωτιστικού βασίστηκε στο εξής σύστημα κανόνων:

- Φωτεινότητα δωματίου < **5 Lux** : Φωτεινότητα δωματίου σε χαμηλά επίπεδα, ενεργοποίηση φωτιστικού
- **5 Lux** < Φωτεινότητα δωματίου < **25 Lux** : Φωτεινότητα δωματίου σε φυσιολογικά επίπεδα, καμία ενέργεια φωτιστικού
- Φωτεινότητα δωματίου > **25 Lux** : Φωτεινότητα δωματίου σε υψηλά επίπεδα, απενεργοποίηση φωτιστικού

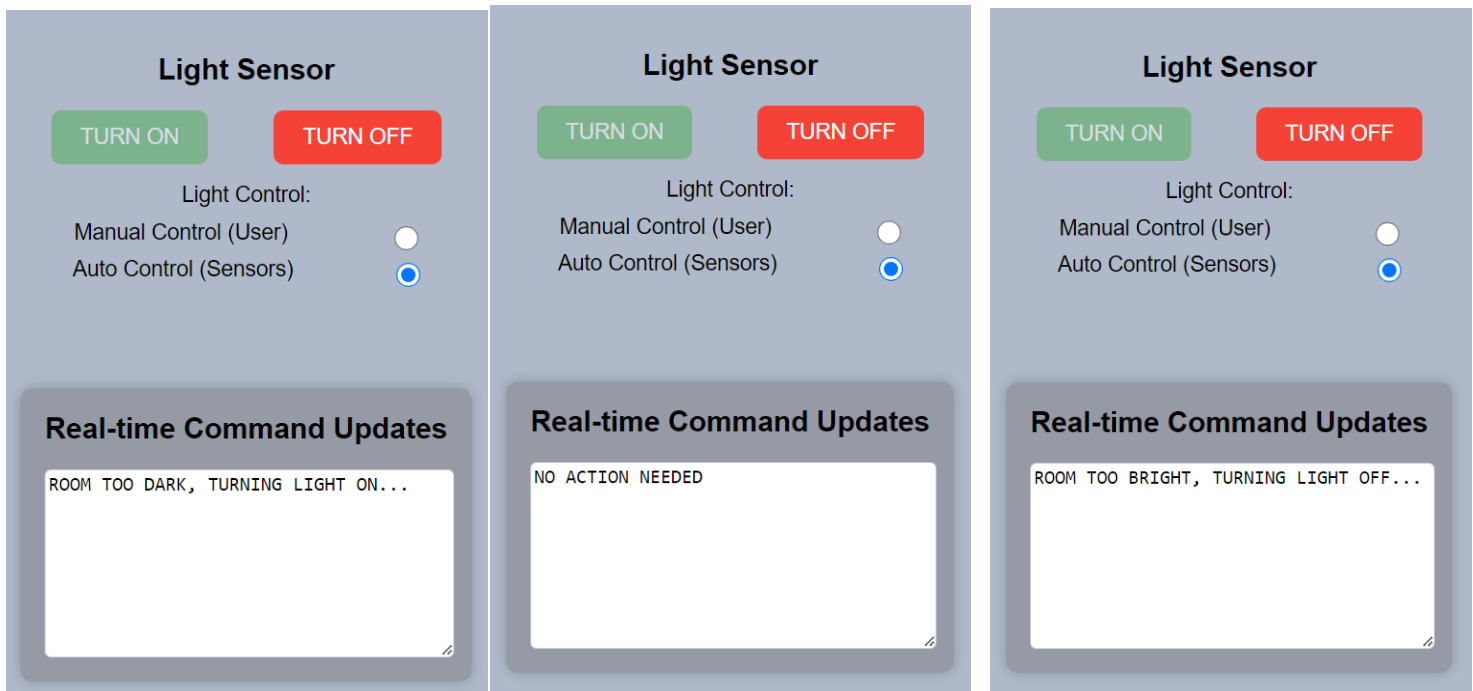
Για κάθε μία ενέργεια από τις παραπάνω εμφανίζεται και ένα κατάλληλο μήνυμα στο χρήστη.

Τα όρια φωτεινότητας ορίστηκαν με βάση τις τυπικές τιμές που αναμένεται να παρουσιάζει ένα δωμάτιο.

Παρακάτω υπάρχει η απεικόνιση των τριών διαφορετικών περιπτώσεων που αναφέρθηκαν προηγουμένως.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων



Εικόνα 4.3.5: Διαφορετικές περιπτώσεις ενεργειών αυτόματου ελέγχου φωτιστικού (Auto Control)

Ακολουθεί η επεξήγηση των παραπάνω εικόνων:

Στην πρώτη εικόνα, παρατηρούμε ένα δωμάτιο που χαρακτηρίζεται ως πολύ σκοτεινό. Σε αυτήν την περίπτωση, το φωτιστικό ενεργοποιείται αυτόματα για να βελτιώσει τη φωτεινότητα. Αυτή η ενέργεια αντανακλάται στο μήνυμα που εμφανίζεται: "ROOM TOO DARK, TURNING LIGHT ON" (Το δωμάτιο πολύ σκοτεινό, ενεργοποίηση φωτιστικού).

Στη δεύτερη εικόνα, το δωμάτιο έχει φυσιολογική φωτεινότητα και δεν απαιτεί κάποια ενέργεια όσον αφορά το φωτιστικό. Αυτό αντικατοπτρίζεται στο μήνυμα: "NO ACTION NEEDED" (Καμία ενέργεια δεν απαιτείται).

Στην τρίτη εικόνα, το δωμάτιο χαρακτηρίζεται ως πολύ φωτεινό. Σε αυτήν την περίπτωση, το φωτιστικό απενεργοποιείται αυτόματα για να μειώσει τη φωτεινότητα. Το μήνυμα που εμφανίζεται είναι: "ROOM TOO BRIGHT, TURNING LIGHT OFF" (Το δωμάτιο πολύ φωτεινό, απενεργοποίηση φωτιστικού).

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Περίπτωση χειροκίνητης λειτουργίας

Η χειροκίνητη λειτουργία παρέχει στον χρήστη τη δυνατότητα να έχει τον πλήρη έλεγχο του επιπέδου φωτισμού για την συγκεκριμένη συσκευή που είναι συνδεδεμένη. Αυτή η λειτουργία υλοποιείται μέσω ενός μενού επιλογών, το οποίο εμφανίζει μια λίστα από επιλογές φωτισμού τύπου **dropdown**. Ο χρήστης μπορεί να επιλέξει το επιθυμητό επίπεδο φωτισμού από τη λίστα, προσφέροντας του έτσι τη δυνατότητα προσαρμογής του φωτισμού στις ατομικές προτιμήσεις και ανάγκες του. Μετά την επιλογή, η εφαρμογή ανταποκρίνεται αμέσως, εφαρμόζοντας την αλλαγή και παρέχοντας επιβεβαιώσεις μέσω κατάλληλων μηνυμάτων στη διεπαφή. Η δυνατότητα αυτή επιτρέπει στον χρήστη να επηρεάζει τον φωτισμό σύμφωνα με τις συγκεκριμένες προτιμήσεις και τις απαιτήσεις του, προσφέροντας έτσι εξατομικευμένη εμπειρία.

Λειτουργία του Manual Control (Light)

Η λειτουργία "Manual Control" επιτρέπει στον χρήστη να αναλάβει τον πλήρη έλεγχο του επιπέδου φωτισμού στο χώρο. Όταν ο χρήστης επιλέγει την επιλογή "Manual Control" και κάνει κλικ στο πλαίσιο που λέει "Choose", εμφανίζεται ένα drop-down μενού που περιλαμβάνει **πέντε** διαφορετικά επίπεδα φωτισμού. Ο χρήστης επιλέγει το επιθυμητό επίπεδο από τη λίστα, και στη συνέχεια εμφανίζεται ένα μήνυμα επιβεβαίωσης που τον ενημερώνει ότι η επιλογή του πραγματοποιήθηκε με επιτυχία. Με αυτόν τον τρόπο, ο χρήστης αποκτά πλήρη έλεγχο του φωτισμού σύμφωνα με τις προσωπικές του προτιμήσεις και ανάγκες.

Η λειτουργία του χειροκίνητου ελέγχου έντασης φωτιστικού βασίστηκε στο εξής σύστημα κανόνων:

- Stage 1 [Low] = Η φωτεινότητα του χώρου ρυθμίζεται στα **5 Lux**
- Stage 2 [Medium] = Η φωτεινότητα του χώρου ρυθμίζεται στα **10 Lux**
- Stage 3 [High] = Η φωτεινότητα του χώρου ρυθμίζεται στα **15 Lux**
- Stage 4 [Very High] = Η φωτεινότητα του χώρου ρυθμίζεται στα **20 Lux**
- Stage 5 [Ultra High] = Η φωτεινότητα του χώρου ρυθμίζεται στα **25 Lux**

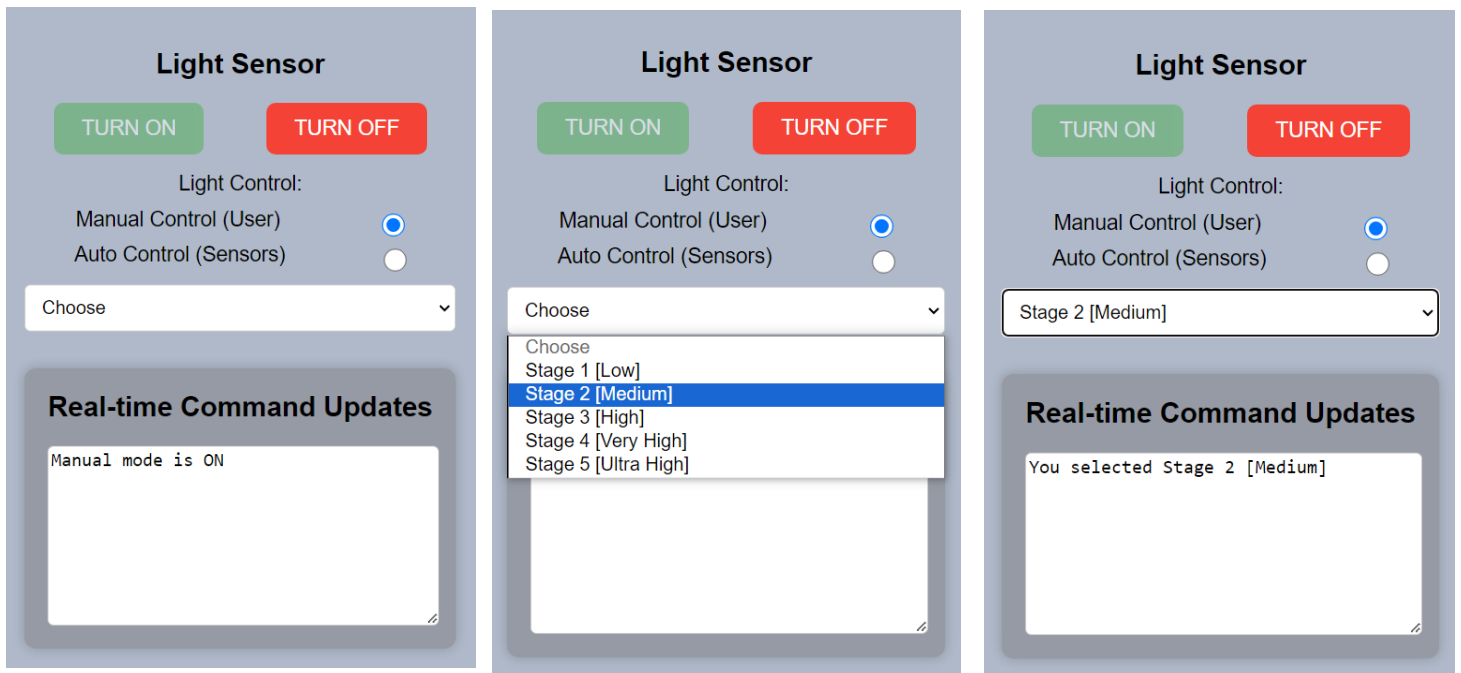
Για κάθε μία ενέργεια από τις παραπάνω εμφανίζεται και ένα κατάλληλο μήνυμα στο χρήστη.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Τα όρια φωτεινότητας ορίστηκαν με βάση τις τοπικές τιμές που αναμένεται να παρουσιάζει ένα δωμάτιο.

Παρακάτω υπάρχει η απεικόνιση των τριών βημάτων για την επιλογή επιπέδου φωτεινότητας όπως αναφέρθηκε προηγουμένως.



Εικόνα 4.3.6: Τα βήματα που ακολουθεί ο χρήστης για την επιλογή επιπέδου φωτισμού (Manual Control)

Όπως παρατηρείται στις παραπάνω εικόνες η διαδικασία επιλογής κάποιου επιπέδου φωτισμού είναι αρκετά εύκολη και φιλική προς τον χρήστη. Συγκεκριμένα:

Στην πρώτη εικόνα, φαίνεται ότι ο χρήστης έχει επιλέξει τη Χειροκίνητη Ελεγχόμενη λειτουργία (Manual Control), και συνεπώς, η λειτουργία χειροκίνητου ελέγχου είναι ενεργοποιημένη. Αυτό φαίνεται από το μήνυμα που εμφανίζεται: "Manual Mode is ON" (Η λειτουργία Χειροκίνητου Ελέγχου είναι Ενεργή).

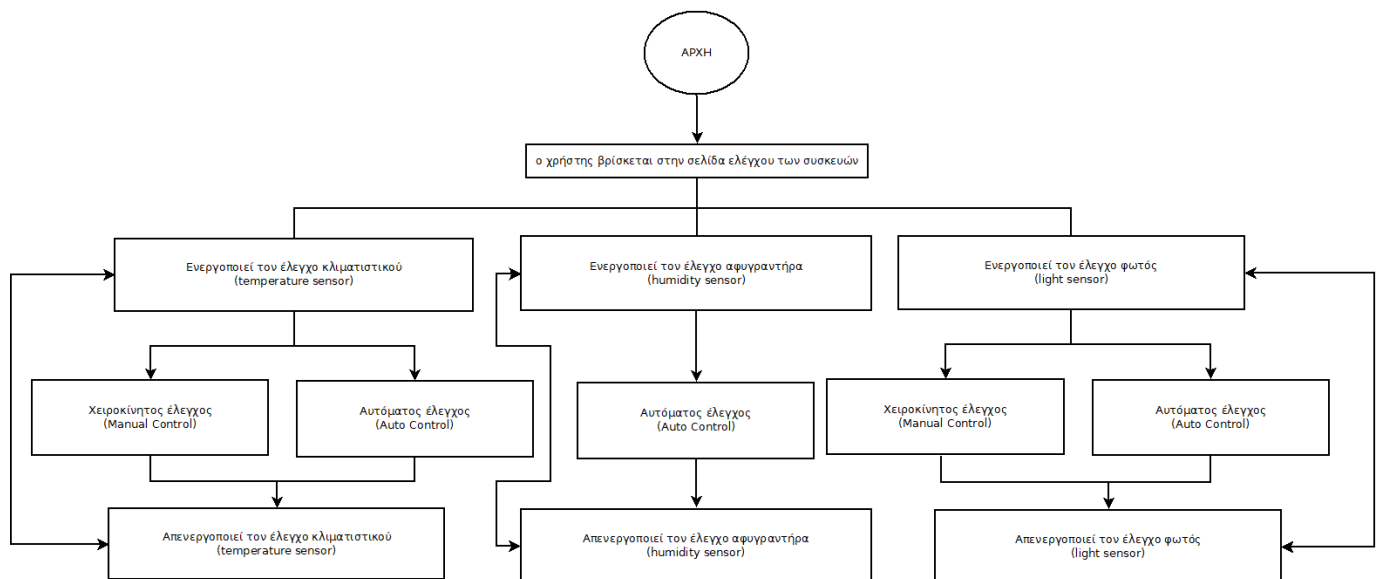
Στη δεύτερη εικόνα, ο χρήστης έχει επιλέξει το δεύτερο επίπεδο λειτουργίας "Medium" (Stage 2 [Medium]). Αυτή η επιλογή αντιπροσωπεύει ένα συγκεκριμένο επίπεδο ισχύος για τη συσκευή.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Στην τρίτη εικόνα, εμφανίζεται ένα μήνυμα για να ενημερώσει τον χρήστη για την επιλογή του: "You selected Stage 2 [Medium]" (Επιλέξατε το Επίπεδο 2 [Μεσαίο]). Αυτό το μήνυμα παρέχει στον χρήστη επιβεβαίωση σχετικά με την επιλογή του δεύτερου επιπέδου λειτουργίας και δηλώνει ότι η συσκευή ή το σύστημα τώρα λειτουργεί στην κατάσταση "Medium".

Εκτός από τους δύο τρόπους ελέγχου της φωτεινότητας, υπάρχει και η δυνατότητα πλήρους απενεργοποίησης, προσφέροντας έτσι στον χρήστη πλήρη έλεγχο και δυνατότητα εξοικονόμησης ενέργειας. Μέσω αυτής της επιλογής, ο χρήστης μπορεί να απενεργοποιήσει τον έλεγχο της φωτεινότητας, επιτρέποντάς του να παρεμβαίνει χειροκίνητα όταν κρίνει απαραίτητο, ενώ ταυτόχρονα εξοικονομεί ενέργεια όταν δεν χρειάζεται ο αυτόματος έλεγχος.



Σχήμα 4.3: Διάγραμμα εφαρμογής Frontend κομματιού

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

4.3.4 ΠΛΑΤΦΟΡΜΑ ΟΠΤΙΚΟΠΟΙΗΣΗΣ ΔΕΔΟΜΕΝΩΝ (THINGSPEAK)

Στην εφαρμογή αυτή, το ThingSpeak χρησιμοποιείται για τη συλλογή και οπτικοποίηση των δεδομένων που παράγονται από τους αισθητήρες του Arduino. Το Arduino, μέσω του αρχείου Server.py που λειτουργεί ως socket server, στέλνει τα δεδομένα στο ThingSpeak. Συγκεκριμένα, τα δεδομένα θερμοκρασίας, υγρασίας και φωτεινότητας συλλέγονται από τους αισθητήρες του Arduino. Τα δεδομένα αυτά, μετά την επεξεργασία και κατηγοριοποίησή τους από τον Server.py, στέλνονται στο ThingSpeak μέσω HTTP requests. Στο ThingSpeak, έχουν δημιουργηθεί κατάλληλα κανάλια για κάθε κατηγορία δεδομένων (θερμοκρασία, υγρασία, φωτεινότητα). Τα δεδομένα εισέρχονται σε αυτά τα κανάλια και απεικονίζονται σε προσαρμοσμένα γραφήματα, τα οποία είναι προσβάσιμα μέσω του ThingSpeak Dashboard. Η χρήση του ThingSpeak βοηθάει στην εξαιρετική οπτικοποίηση των δεδομένων, παρέχοντας σαφή εικόνα της εξέλιξης των συνθηκών του περιβάλλοντος που παρακολουθεί ο χρήστης. Παράλληλα, το ThingSpeak προσφέρει δυνατότητες ανάλυσης και εξαγωγής δεδομένων για περαιτέρω επεξεργασία και χρήση.



Εικόνα 4.3.7: Διαγράμματα (graphs) που απεικονίζουν τις τιμές από κάθε sensor (Interface)

4.4 ΠΙΣΩ ΜΕΡΟΣ (BACKEND)

Το backend της εφαρμογής αποτελεί τον κεντρικό πυρήνα που διαχειρίζεται την αλληλεπίδραση μεταξύ του Arduino, των αισθητήρων και του χρήστη. Αναλαμβάνει τον έλεγχο της συλλογής δεδομένων από τους αισθητήρες, διασφαλίζοντας τη σωστή επικοινωνία με το υλικό. Το backend διαδραματίζει ουσιαστικό ρόλο στην συλλογή και επεξεργασία των δεδομένων, παρέχοντας στο frontend τα απαραίτητα στοιχεία για την παρουσίαση στον χρήστη. Επιπλέον, διαχειρίζεται την εκτέλεση του συστήματος κανόνων, καθορίζοντας πώς θα αντιδράσουν οι συσκευές σε

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

συγκεκριμένες συνθήκες. Το backend εξασφαλίζει επίσης τον έλεγχο της ενέργειας των συσκευών, δίνοντας στον χρήστη τη δυνατότητα πλήρους ελέγχου και εξοικονόμησης ενέργειας. Συνολικά, το backend αποτελεί τον συνδετικό κρίκο που εξασφαλίζει τη συνοχή και αποτελεσματικότητα της εφαρμογής.

Το backend της εφαρμογής βασίζεται σε ένα μόνο αρχείο, το **Server.py**, γραμμένο στη γλώσσα προγραμματισμού Python. Αυτό το αρχείο αντιπροσωπεύει τον κεντρικό server του συστήματος, λειτουργώντας ταυτόχρονα ως socket server για την επικοινωνία με το Arduino καθώς και ως HTTP server για το χρήστη μέσω του περιβάλλοντος χρήστη (frontend).

Στον τομέα της επικοινωνίας με το Arduino, το αρχείο Server.py λειτουργεί ως socket server, λαμβάνοντας δεδομένα από τους αισθητήρες που συνδέονται στο Arduino. Αφού τα δεδομένα φτάσουν στον server, το αρχείο Server.py αναλαμβάνει να τα κατηγοριοποιήσει, δηλαδή να τα διαχωρίσει σε διάφορες κατηγορίες όπως θερμοκρασία, υγρασία, και φωτεινότητα, ανάλογα με τον αισθητήρα που τα παράγει. Έπειτα, προβαίνει σε επεξεργασία των δεδομένων για την προετοιμασία τους για τη μετάδοση προς το Frontend.

Η συνεχής ροή πληροφοριών είναι ουσιώδης για την ασφαλή και ακριβή ενημέρωση του Frontend για τις συνθήκες του περιβάλλοντος. Αυτό επιτυγχάνεται μέσω της σταθερής ακρόασης του socket server για τα νέα δεδομένα που προκύπτουν από το Arduino. Καθώς λαμβάνονται νέα δεδομένα, το Backend ενημερώνει δυναμικά το Frontend, επιτρέποντας στο χρήστη να παρακολουθεί πραγματικό χρόνο τις μετρήσεις των αισθητήρων και την κατάσταση των συσκευών.

Παράλληλα, στον τομέα του χρήστη, το αρχείο Server.py λειτουργεί ως HTTP server, αποστέλλοντας δεδομένα προς το Frontend για την ανανέωση του dashboard και ανταποκρινόμενο στις αιτήσεις του χρήστη, χρησιμοποιώντας τις μεθόδους **GET, POST**.

Οι αιτήσεις **GET** χρησιμοποιούνται για να ανακτήσουν δεδομένα από τον server. Στην περίπτωση του Frontend, αυτές οι αιτήσεις χρησιμοποιούνται για τη λήψη τρεχουσών τιμών αισθητήρων ή καταστάσεων συσκευών για εμφάνιση στο χρήστη. Για παράδειγμα, ένα GET request μπορεί να χρησιμοποιηθεί για τη λήψη τρέχουσας θερμοκρασίας, υγρασίας ή επιπέδου φωτεινότητας.

Οι αιτήσεις **POST** χρησιμοποιούνται για να στείλουν δεδομένα στον server για επεξεργασία ή αποθήκευση. Στην εφαρμογή, αυτές οι αιτήσεις μπορούν να χρησιμοποιηθούν όταν ο χρήστης κάνει ενέργειες, όπως αλλαγή ρυθμίσεων ή ενεργοποίηση/απενεργοποίηση συσκευών. Για παράδειγμα, ένα POST request μπορεί να σταλεί όταν ο χρήστης αλλάξει το επίπεδο φωτεινότητας που επιθυμεί.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Το backend αναλαμβάνει επίσης να ενημερώνει το Frontend, εκπέμποντας εντολές στις έξυπνες συσκευές για την εκτέλεση διαφόρων ενεργειών. Με αυτόν τον τρόπο ο server διασφαλίζει τη συνεχή συνδεσιμότητα, επεξεργασία δεδομένων και αλληλεπίδραση με τις συσκευές για την επίτευξη των λειτουργικών απαιτήσεων του συστήματος.

http://: Η χρήση του πρωτοκόλλου HTTP (Hypertext Transfer Protocol) στο URL "http://" δείχνει ότι το HTTP χρησιμοποιείται για τη μετάδοση δεδομένων μεταξύ του Frontend και του Backend. Το "http://" αναγνωρίζεται ως το πρωτόκολλο που θα χρησιμοποιηθεί κατά την ανταλλαγή πληροφοριών μεταξύ του περιηγητή του χρήστη (Frontend) και του server (Backend). Το HTTP είναι ένα από τα βασικά πρωτόκολλα επικοινωνίας στον Παγκόσμιο Ιστό και χρησιμοποιείται για τη μεταφορά δεδομένων μεταξύ του **client** (π.χ., περιηγητής) και του **server**. Οι διευθύνσεις URL που ξεκινούν με "http://" υποδηλώνουν τη χρήση αυτού του πρωτοκόλλου. Κατά τη διάρκεια της ανταλλαγής δεδομένων μεταξύ του Frontend και του Backend, το HTTP επιτρέπει τη μεταφορά σελίδων HTML, εικόνων, αρχείων CSS, JavaScript, και άλλων πόρων. Μέσω του HTTP, το Frontend μπορεί να ζητά δεδομένα από το Backend και να λαμβάνει απαντήσεις, προσφέροντας έτσι μια διαδραστική εμπειρία χρήστη με την εφαρμογή.

localhost: Η χρήση του "localhost:" αντιπροσωπεύει τη σύνδεση στον τοπικό υπολογιστή, δηλαδή τον υπολογιστή στον οποίο εκτελείται η εφαρμογή, χωρίς την ανάγκη για εξωτερική δικτυακή σύνδεση. Κατά την ανάπτυξη και δοκιμή μιας ιστοσελίδας ή μιας εφαρμογής, η χρήση του "localhost:" προσφέρει τη δυνατότητα πρόσβασης στον τοπικό server χωρίς την ανάγκη για δημόσια δικτυακή σύνδεση. Ο τοπικός υπολογιστής, αναγνωρίζοντας το "localhost:", ανακατευθύνει το αίτημα προς τον εαυτό του, χρησιμοποιώντας τον δικό του συνδυασμό IP διεύθυνσης και θύρας. Αυτό επιτρέπει στους προγραμματιστές και τους διαχειριστές συστημάτων να αναπτύσσουν και να ελέγχουν το λογισμικό τους σε ένα απομονωμένο περιβάλλον, χωρίς να επηρεάζονται από εξωτερικές συνδέσεις ή διακυμάνσεις του διαδικτύου.

5500: Η χρήση του "localhost:5500" αναφέρεται στη σύνδεση στον τοπικό υπολογιστή με χρήση της θύρας 5500. Η θύρα 5500 χρησιμοποιείται ως αριθμός πόρτας για την αντιστοίχιση των εισερχόμενων αιτημάτων στον τοπικό server. Κατά την ανάπτυξη και τον έλεγχο μιας ιστοσελίδας ή μιας εφαρμογής, η εκτέλεση του server στη θύρα 5500 επιτρέπει την πρόσβαση στον τοπικό ιστότοπο μέσω του προγράμματος περιήγησης. Η συγκεκριμένη θύρα επιλέγεται από τον προγραμματιστή κατά τη διάρκεια της ανάπτυξης και έχει ως σκοπό να είναι μια διαθέσιμη θύρα που δεν χρησιμοποιείται ήδη από άλλες υπηρεσίες στον υπολογιστή. Επιτρέπει την αλληλεπίδραση με τον

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

τοπικό server κατά την ανάπτυξη, χωρίς την ανάγκη να καταναλωθεί μια γνωστή ή συγκεκριμένη θύρα.

/path: Χρησιμοποιώντας το **"/Frontend/Interface.html"** στο URL **"http://localhost:5500"**, γίνεται αναφορά σε ένα συγκεκριμένο μονοπάτι (**path**) στον τοπικό διακομιστή της εφαρμογής, το οποίο οδηγεί στο αρχείο "Interface.html" στον φάκελο "Frontend". Αυτό σημαίνει ότι το αρχείο HTML "Interface.html" βρίσκεται μέσα στον φάκελο "Frontend" στον τοπικό server. Όταν ο χρήστης επισκέπτεται το συγκεκριμένο URL, ο περιηγητής (browser) του ερμηνεύει το μονοπάτι **"/Frontend/Interface.html"** και αναζητά το αρχείο "Interface.html" μέσα στον φάκελο "Frontend". Η χρήση μονοπατιών στα URLs είναι ένας τρόπος οργάνωσης των διαδρομών των διαφόρων σελίδων ή υπηρεσιών στην ιεραρχία ενός ιστότοπου. Στη συγκεκριμένη περίπτωση, το **"/Frontend"** λειτουργεί ως υποφάκελος που περιέχει τον φάκελο με τον frontend κώδικα, ενώ **"Interface.html"** είναι το συγκεκριμένο αρχείο HTML που θα φορτωθεί.

Τα αιτήματα που δέχεται το Backend από το Frontend βασίζονται στις παρακάτω διευθύνσεις:

1. Δεδομένα Θερμοκρασίας:

- **/acon:** Το **/acon** αντιπροσωπεύει ένα path στο οποίο το Frontend στέλνει ένα αίτημα τύπου **POST** προς το Backend, με σκοπό να ενεργοποιήσει τον κλιματισμό (A/C). Το αίτημα αυτό προκαλείται όταν ο χρήστης πατάει το κουμπί "TURN ON" στο χρηστικό περιβάλλον της εφαρμογής. Το αίτημα περιλαμβάνει την κατάλληλη πληροφορία ώστε το Backend να αντιληφθεί ότι ο χρήστης επιθυμεί να ενεργοποιήσει τον κλιματισμό.
- **/acoff:** Το **/acoff** αντιστοιχεί σε ένα path στο οποίο το Frontend στέλνει ένα αίτημα τύπου **POST** προς το Backend με σκοπό να απενεργοποιήσει τον κλιματισμό (A/C). Αυτό συμβαίνει όταν ο χρήστης πατάει το κουμπί "TURN OFF" στο χρηστικό περιβάλλον της εφαρμογής. Το αίτημα περιέχει την αντίστοιχη πληροφορία που καθορίζει την επιθυμία του χρήστη για απενεργοποίηση του κλιματισμού. Όταν το Backend λαμβάνει αυτό το αίτημα, εκτελεί τις απαραίτητες ενέργειες για να απενεργοποιήσει τον κλιματισμό, αποστέλλοντας κατάλληλη εντολή προς το Arduino.
- **/aci:** Το **/aci** (increase) αντιστοιχεί σε ένα path στο οποίο το Frontend στέλνει ένα αίτημα τύπου **POST** προς το Backend, όταν ο χρήστης έχει επιλέξει το χειροκίνητο έλεγχο θερμοκρασίας "Manual Control (User)" και πατάει το κουμπί "+" για να αυξήσει τη θερμοκρασία. Το αίτημα περιλαμβάνει την αντίστοιχη πληροφορία που υποδηλώνει την επιθυμία του χρήστη για αύξηση της θερμοκρασίας. Όταν το Backend λαμβάνει αυτό το

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

αίτημα, εκτελεί τις απαραίτητες ενέργειες για να αυξήσει τη θερμοκρασία και συγκεκριμένα προσθέτει στην τωρινή θερμοκρασία 1 βαθμό °C.

- **/acd:** Το /acd (decrease) αντιστοιχεί σε ένα path στο οποίο το Frontend στέλνει ένα αίτημα τύπου POST προς το Backend, όταν ο χρήστης έχει επιλέξει το χειροκίνητο έλεγχο θερμοκρασίας “Manual Control (User)” και πατάει το κουμπί "-" για να μειώσει τη θερμοκρασία. Το αίτημα περιλαμβάνει την αντίστοιχη πληροφορία που υποδηλώνει την επιθυμία του χρήστη για μείωση της θερμοκρασίας. Όταν το Backend λαμβάνει αυτό το αίτημα, εκτελεί τις απαραίτητες ενέργειες για να μειώσει τη θερμοκρασία και συγκεκριμένα αφαιρεί από την τωρινή θερμοκρασία 1 βαθμό °C.
- **/tmpsensordata:** Το /tmpsensordata αντιστοιχεί σε ένα path στο οποίο το Backend διαχειρίζεται τα δεδομένα θερμοκρασίας αυτόματα από τους αισθητήρες όταν πατηθεί το κουμπί “Auto Control (Sensors)”. Κατά το αίτημα τύπου GET σε αυτό το path, το Backend ανακτά τις τρέχουσες μετρήσεις θερμοκρασίας από τους αισθητήρες. Έπειτα, το Backend εφαρμόζει το σύστημα κανόνων που έχει καθοριστεί για τον έλεγχο της θερμοκρασίας. Σύμφωνα με αυτούς τους κανόνες, αν η μετρούμενη θερμοκρασία είναι κάτω των 10°C, το κλιματιστικό αυξάνει τη θερμοκρασία στους 10°C. Αν η θερμοκρασία είναι μεταξύ 10°C και 30°C, το κλιματιστικό διατηρεί την τρέχουσα θερμοκρασία. Για κάθε περίπτωση εμφανίζεται και το αντίστοιχο μήνυμα στον χρήστη. Τέλος, αν η θερμοκρασία ξεπερνά τους 30°C, το κλιματιστικό μειώνει τη θερμοκρασία στους 30°C. Επομένως, το path αυτό συνδέει την ανάκτηση των δεδομένων των αισθητήρων με το σύστημα κανόνων που εφαρμόζεται για τον αυτόματο έλεγχο της θερμοκρασίας. Η ανταλλαγή αυτών των δεδομένων είναι κρίσιμη για τη λειτουργία του κλιματιστικού, διασφαλίζοντας ότι ο χώρος διατηρείται σε ένα επιθυμητό θερμοκρασιακό επίπεδο.

2. Δεδομένα Υγρασίας:

- **/deon:** Το /deon αντιστοιχεί σε ένα path που ενεργοποιεί τον αφυγραντήρα όταν πατηθεί το κουμπί “TURN ON”. Κατά την αποστολή ενός HTTP POST request σε αυτό το path, γίνεται αίτημα προς το Backend για να ενεργοποιηθεί η συσκευή. Το αίτημα αυτό λειτουργεί ως γέφυρα επικοινωνίας μεταξύ του χρήστη και του συστήματος ελέγχου του αφυγραντήρα, επιτρέποντας την ενεργοποίηση της συγκεκριμένης λειτουργίας μέσω της διεπαφής του Frontend

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

- **/deoff:** Το /deoff αντιστοιχεί σε ένα path που χρησιμοποιείται για να απενεργοποιήσει τον αφυγραντήρα. Όταν ο χρήστης πατάει το κουμπί "TURN OFF" στο Frontend, γίνεται αποστολή HTTP POST request στο /deoff. Το Backend λαμβάνει αυτό το αίτημα και στέλνει την κατάλληλη εντολή στο Arduino για να απενεργοποιηθεί ο αφυγραντήρας.
- **/humsensordata:** Το /humsensordata είναι ένα path που χρησιμοποιείται για την αυτόματη διαχείριση της υγρασίας στο χώρο. Όταν λαμβάνεται ένα GET request σε αυτό το path, το Backend αποστέλλει αίτημα στους αισθητήρες υγρασίας που συνδέονται στο Arduino και λαμβάνει τα τρέχοντα δεδομένα υγρασίας. Στη συνέχεια, το Backend εφαρμόζει το σύστημα κανόνων που έχει καθοριστεί. Συγκεκριμένα, εάν η υγρασία είναι κάτω από το 30%RH, η εφαρμογή προσπαθεί να αυξήσει την υγρασία. Αντίστροφα, εάν η υγρασία υπερβεί το 70%RH, προσπαθεί να την μειώσει. Τέλος, αν η τιμή κυμαίνεται ανάμεσα σε αυτά τα ποσοστά τότε δεν γίνεται κάποια ενέργεια. Η λειτουργία αυτή συντελεί στη διατήρηση της επιθυμητής υγρασίας στο χώρο, προσφέροντας έτσι άνεση και ευεξία στον χρήστη. Επιπλέον, βοηθά στην πρόληψη της υπερβολικής ξηρασίας ή υγρασίας, προσφέροντας μια ιδανική περιβαλλοντική εμπειρία.

3. Δεδομένα Φωτεινότητας:

- **/autoon:** Το /autoon αποτελεί ένα path που ενεργοποιείται όταν ο χρήστης πατάει το κουμπί "TURN ON" για το φως. Όταν λαμβάνεται λοιπόν POST request σε αυτό το path από το Frontend, το Backend αντιλαμβάνεται το αίτημα και εκτελεί τη διαδικασία ενεργοποίησης του φωτιστικού. Αρχικά, η προεπιλεγμένη κατάσταση είναι ο αυτόματος έλεγχος, όπως προκαθορίζεται από το σύστημα κανόνων. Όταν ο χρήστης ενεργοποιεί το φως, το σύστημα παραμένει σε αυτή την κατάσταση αλλά η λάμπα/φωτιστικό ανάβει αν οι μετρήσεις του photosensor δείχνουν ότι το περιβάλλον χρειάζεται φωτισμό. Επομένως, το σύστημα εξακολουθεί να χρησιμοποιεί τον αυτόματο έλεγχο για να προσαρμόζει τον φωτισμό ανάλογα με τις συνθήκες. Με τον τρόπο αυτό, επιτυγχάνεται μια συμβιβαστική λύση όπου ο χρήστης μπορεί να ανοίξει το φως με το "TURN ON" κουμπί, αλλά το σύστημα συνεχίζει να επιβλέπει και να προσαρμόζει τον φωτισμό αυτόματα.
- **/autooff:** Το /autooff αντιπροσωπεύει ένα path που ενεργοποιείται όταν ο χρήστης πατάει το κουμπί "TURN OFF" ή το "Manual Control" για το φως. Με τη λήψη ενός POST request σε

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

αυτό το path από το Frontend, το Backend αντιλαμβάνεται το αίτημα και εκτελεί τη διαδικασία απενεργοποίησης του φωτιστικού ή τον χειροκίνητο έλεγχο.

- **/automode:** Το /automode αντιπροσωπεύει ένα path στο Backend που ενεργοποιείται όταν ο χρήστης επιλέγει το κουμπί “Auto Control (Sensors)” για τον αυτόματο έλεγχο φωτισμού. Αυτό σημαίνει ότι ο χρήστης επιλέγει να αφήσει το σύστημα να ρυθμίζει αυτόματα τον φωτισμό, βασιζόμενο στις μετρήσεις του φωτοαισθητήρα και στο σύστημα κανόνων. Όταν ο χρήστης κάνει αυτή την επιλογή στο Frontend, γίνεται ένα GET request προς το /automode. Το Backend ανταποκρίνεται σε αυτό το αίτημα ενεργοποιώντας τη λειτουργία του αυτόματου ελέγχου φωτισμού. Στη συνέχεια, το σύστημα ξεκινά να παρακολουθεί τις μετρήσεις του φωτοαισθητήρα και να ρυθμίζει δυναμικά τον φωτισμό ανάλογα με τις συνθήκες του περιβάλλοντος. Συγκεκριμένα, η λειτουργία αυτή είναι προγραμματισμένη να ανταποκρίνεται σε συγκεκριμένες τιμές φωτεινότητας (Lux). Αν η φωτεινότητα είναι κάτω από 5 Lux, το φωτιστικό ανάβει για να παρέχει επαρκή φωτισμό. Αν η φωτεινότητα είναι πάνω από 25 Lux, το φωτιστικό σβήνει για εξοικονόμηση ενέργειας. Όταν η φωτεινότητα βρίσκεται ανάμεσα σε αυτά τα δύο όρια, δεν γίνονται αλλαγές και το φωτιστικό διατηρεί την τρέχουσα κατάσταση.
- **/manualon:** Το /manualon είναι ένα path που ενεργοποιείται όταν ο χρήστης επιλέγει το κουμπί "Manual Control" στο Frontend. Με αυτήν την επιλογή, ο χρήστης επιθυμεί να έχει χειροκίνητο έλεγχο επί του φωτισμού. Όταν ο χρήστης κάνει αυτή την επιλογή, γίνεται ένα POST request προς το /manualon. Το Backend ανταποκρίνεται εμφανίζοντας το αντίστοιχο μήνυμα και παρέχοντας τη λίστα με τα επίπεδα φωτισμού. Η λίστα αυτή εμφανίζεται στο Frontend, όπου ο χρήστης μπορεί να επιλέξει το επιθυμητό επίπεδο φωτισμού από μια λίστα επιλογών.
- **/manualoff:** Το /manualoff είναι το path που ενεργοποιείται όταν ο χρήστης επιλέγει να απενεργοποιήσει τον χειροκίνητο έλεγχο του φωτισμού ή να αλλάξει στον αυτόματο έλεγχο. Επομένως όταν πατάει το κουμπί "TURN OFF" ή το “Auto Control” στο Frontend, στέλνεται ένα αίτημα τύπου POST προς τον server, και το σύστημα απενεργοποιείται ή επαναφέρεται στη λειτουργία του αυτόματου ελέγχου.
- **/lightstage:** Το /lightstage αφορά τη λειτουργία όταν είναι ενεργοποιημένος ο χειροκίνητος έλεγχος φωτισμού και ο χρήστης επιλέγει ένα από τα πέντε διαθέσιμα επίπεδα φωτεινότητας. Με το αίτημα POST που αποστέλλεται στον server μετά την επιλογή του επιπέδου φωτισμού από το χρήστη, το σύστημα προσαρμόζει την ένταση του φωτισμού ανάλογα με την επιλογή

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

που έγινε. Ουσιαστικά, το /lightstage επιτρέπει στο χρήστη να έχει άμεσο έλεγχο της φωτεινότητας στον χώρο, ανεξάρτητα από το αυτόματο σύστημα ελέγχου, και να επιλέγει το επίπεδο φωτισμού που τον εξυπηρετεί την στιγμή εκείνη.

Ένα σημαντικό χαρακτηριστικό της εφαρμογής είναι η συχνή ενημέρωση των τιμών των αισθητήρων προς το backend. Κάθε 2 δευτερόλεπτα, οι αισθητήρες ενημερώνουν το backend μέσω τριών καθολικών (**global**) μεταβλητών, συγκεκριμένα της **global_temperature**, **global_humidity**, και **global_light**. Αυτές οι μεταβλητές λειτουργούν ως αποθηκευτικά χώρου που διατηρούν την τρέχουσα τιμή των αντίστοιχων μετρήσεων. Ανεξαρτήτως εάν η αλλαγή των τιμών προέρχεται από το frontend ή από τους αισθητήρες, τα global αυτά ανανεώνονται διαρκώς. Σε περίπτωση που ο χρήστης προβεί σε αλλαγή μέσω του frontend, οι τιμές των global μεταβλητών ενημερώνονται αμέσως, αντανakλώντας τη νέα κατάσταση. Το ίδιο ισχύει και σε περίπτωση που υπάρχει μια αλλαγή στις τιμές που καταγράφονται από τους αισθητήρες. Καθώς η εφαρμογή είναι συνεχώς συνδεδεμένη με τους αισθητήρες, οι τιμές των global μεταβλητών προσαρμόζονται δυναμικά, παρέχοντας μια ακριβή αναπαράσταση της πραγματικής κατάστασης του περιβάλλοντος. Το backend, εφόσον λάβει τις νέες τιμές από τους αισθητήρες ή από το frontend, επεξεργάζεται αυτές τις πληροφορίες και ενημερώνει με τη σειρά του το frontend καθώς και τις συσκευές που είναι συνδεδεμένες. Αυτός ο διπλός τρόπος ενημέρωσης διασφαλίζει ότι ο χρήστης θα έχει πρόσβαση σε έγκυρα δεδομένα κάθε στιγμή, επιτυγχάνοντας ένα σταθερό και αξιόπιστο περιβάλλον ελέγχου.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

/acon	Path για την ενεργοποίηση κλιματισμού
/acoff	Path για την απενεργοποίηση κλιματισμού
/aci	Path για την χειροκίνητη αύξηση θερμοκρασίας
/acd	Path για την χειροκίνητη μείωση θερμοκρασίας
/tmpsensordata	Path για την αυτόματη ρύθμιση θερμοκρασίας
/deon	Path για την ενεργοποίηση αφυγραντήρα
/deoff	Path για την απενεργοποίηση αφυγραντήρα
/humsensordata	Path για την αυτόματη ρύθμιση υγρασίας
/autoon	Path για την ενεργοποίηση αυτόματης ρύθμισης φωτισμού
/autooff	Path για την απενεργοποίηση αυτόματης ρύθμισης φωτισμού
/automode	Path για την αυτόματη ρύθμιση φωτισμού
/manualon	Path για την ενεργοποίηση χειροκίνητης ρύθμισης φωτισμού
/manualoff	Path για την απενεργοποίηση χειροκίνητης ρύθμισης φωτισμού
/lightstage	Path για την χειροκίνητη ρύθμιση φωτισμού

Πίνακας 4.3: Αναφορικός πίνακας διευθύνσεων (paths) αιτημάτων στο Backend από το Frontend

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

ΚΕΦΑΛΑΙΟ 5: ΑΞΙΟΛΟΓΗΣΗ

Ο απώτερος σκοπός της εφαρμογής είναι να προσφέρει μια εύχρηστη, φιλική προς το χρήστη πλατφόρμα που επιτρέπει στον χρήστη να έχει πλήρη έλεγχο επί των διαφόρων συσκευών και εξοπλισμών στον χώρο του. Μέσω μιας ευαίσθητης και εύκολα κατανοητής διεπαφής, ο χρήστης έχει τη δυνατότητα να παρακολουθεί και να ρυθμίζει τις διάφορες λειτουργίες, προσφέροντας του ένα αίσθημα πλήρους ελέγχου και ενεργού συμμετοχής στο περιβάλλον του.

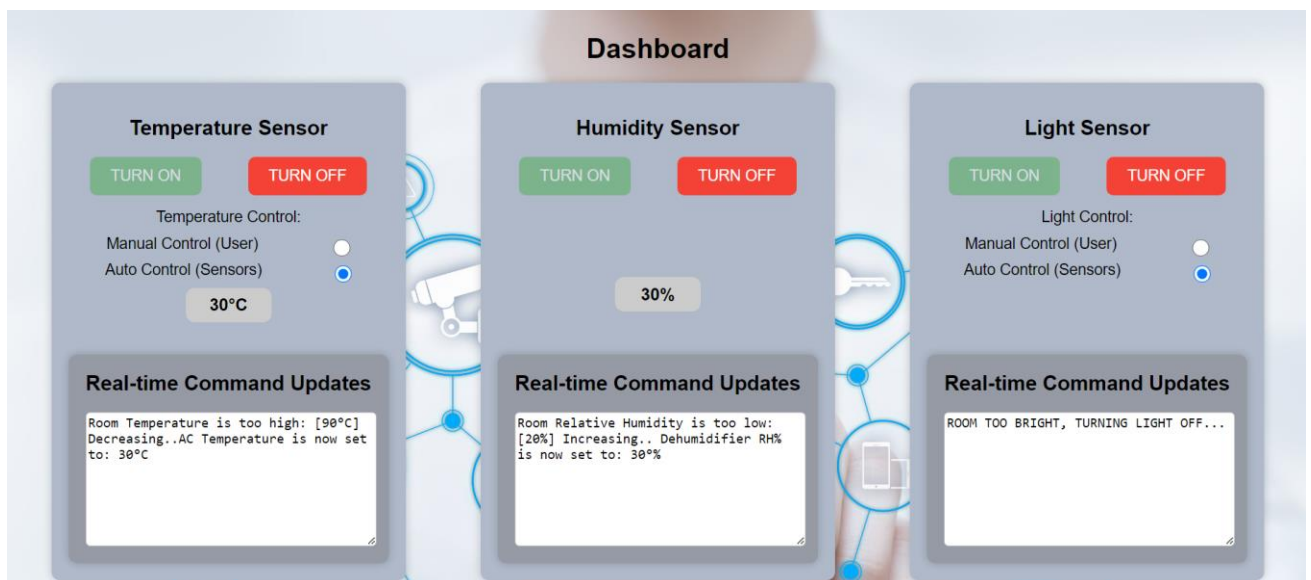
5.1 ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΗΣ

Στο τελευταίο στάδιο της εφαρμογής, δοκιμάστηκαν διάφορες περιπτώσεις επιλογών του μενού, όπου η εφαρμογή ανταποκρίθηκε ως εξής:

Χρήση 1^η

Ενεργοποίηση ελέγχου και των τριών διαφορετικών τύπων δεδομένων:

- Temperature (Auto Control) = Θερμοκρασία πολύ υψηλή, ρύθμιση στους 30 βαθμούς κελσίου
- Humidity = Χαμηλά ποσοστά υγρασίας, αύξηση της σχετικής υγρασίας σε 30%
- Light (Auto Control) = Χώρος πολύ φωτεινός, απενεργοποίηση συσκευής φωτισμού



Εικόνα 5.1.1: Πρώτη περίπτωση χρήσης εφαρμογής

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

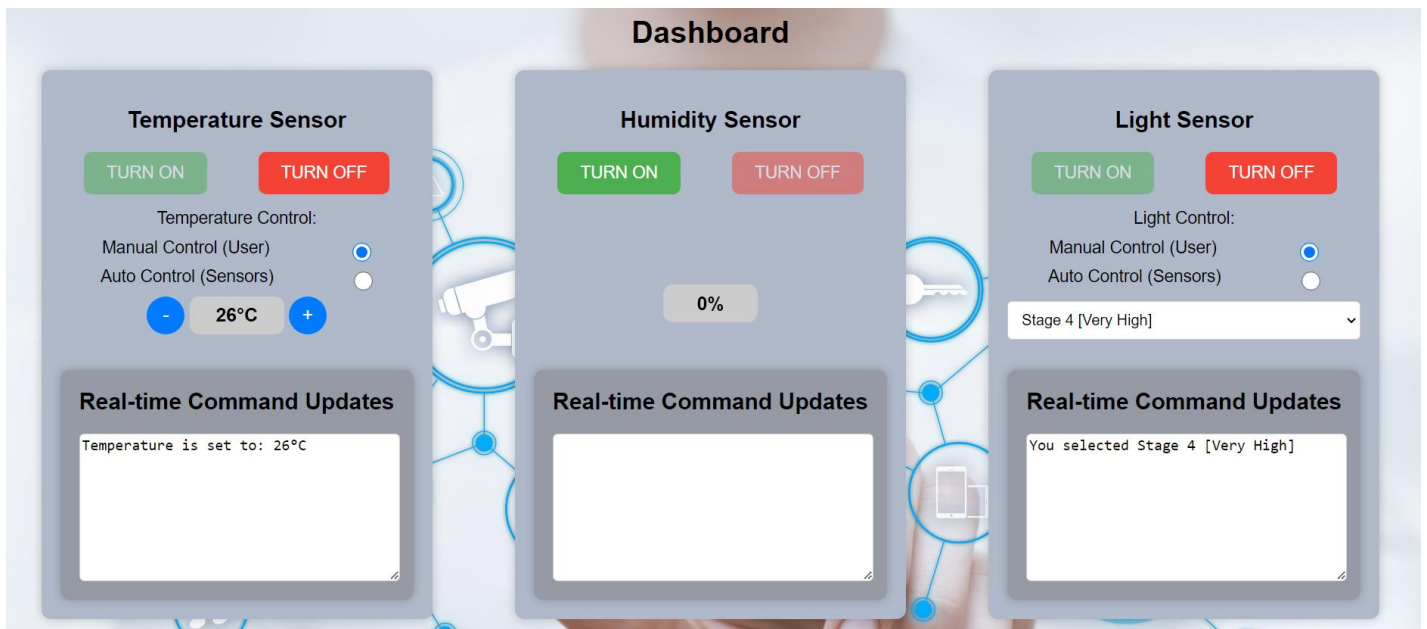
Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Η περίπτωση με τα Auto Control φαίνεται να ανταποκρίθηκε σωστά. (Να σημειωθεί πώς ο έλεγχος υγρασίας λειτουργεί μόνο σε Auto Control)

Χρήση 2^η

Ενεργοποίηση ελέγχου τύπων δεδομένων θερμοκρασίας, φωτισμού:

- Temperature (Manual Control) = Ρύθμιση στους 26 βαθμούς κελσίου
- Light (Manual Control) = Επιλογή του επιπέδου 4 φωτισμού (Very High)



Εικόνα 5.1.2: Δεύτερη περίπτωση χρήσης εφαρμογής

Η περίπτωση με τα Manual Control φαίνεται να ανταποκρίθηκε σωστά.

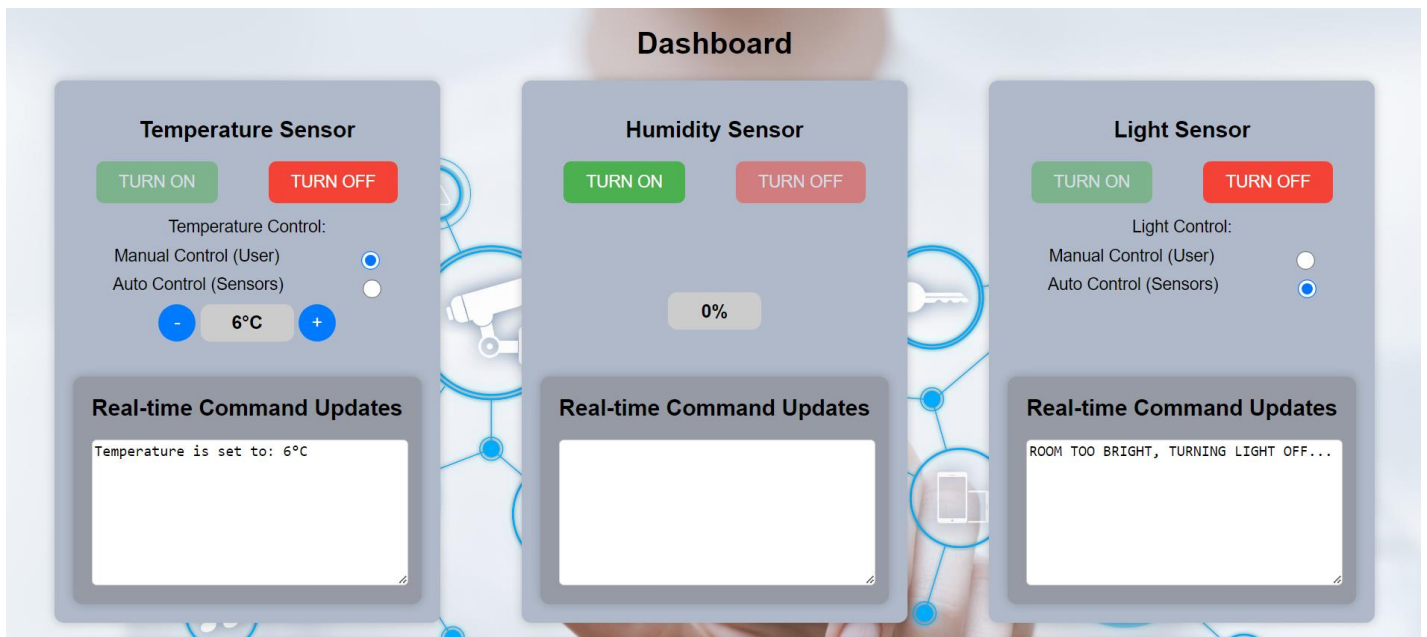
Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Χρήση 3^η

Ενεργοποίηση ελέγχου τύπων δεδομένων θερμοκρασίας, φωτισμού:

- Temperature (Manual Control) = Ρύθμιση στους 6 βαθμούς κελσίου
- Light (Auto Control) = Χώρος πολύ φωτεινός, απενεργοποίηση συσκευής φωτισμού



Εικόνα 5.1.3: Τρίτη περίπτωση χρήσης εφαρμογής

Η περίπτωση με χρήση Manual Control για θερμοκρασία και Auto Control για φωτισμό φαίνεται να ανταποκρίθηκε σωστά.

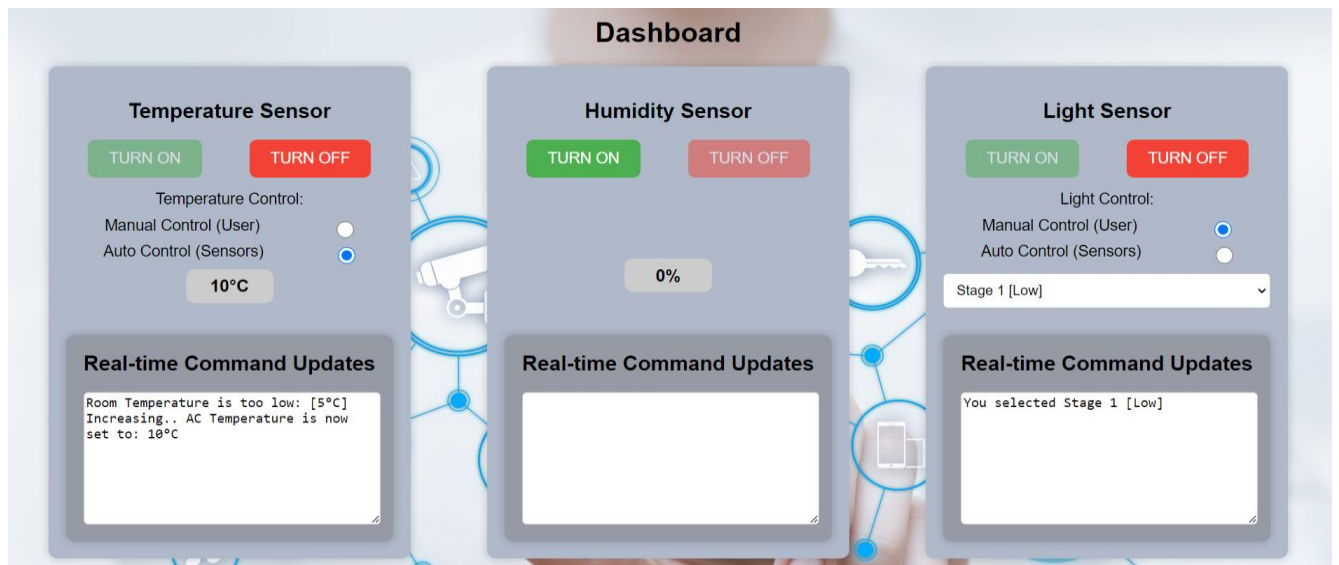
Χρήση 4^η

Ενεργοποίηση ελέγχου τύπων δεδομένων θερμοκρασίας, φωτισμού:

- Temperature (Auto Control) = Ρύθμιση στους 10 βαθμούς κελσίου
- Light (Manual Control) = Επιλογή του επιπέδου 1 φωτισμού (Low)

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων



Εικόνα 5.1.4: Τέταρτη περίπτωση χρήσης εφαρμογής

Η περίπτωση με χρήση Auto Control για θερμοκρασία και Manual Control για φωτισμό φαίνεται να ανταποκρίθηκε σωστά.

5.2 ΑΞΙΟΛΟΓΗΣΗ ΑΠΟ ΧΡΗΣΤΕΣ

Η αξιολόγηση της εφαρμογής πραγματοποιήθηκε με τη συμμετοχή τεσσάρων ατόμων, τα οποία ήταν διαφόρων ηλικιών. Τα άτομα αυτά κλήθηκαν να απαντήσουν διάφορες ερωτήσεις σχετικές με την εφαρμογή και συγκεκριμένα να αξιολογήσουν τη λειτουργικότητα, την αξιοπιστία και την αισθητική της. Για τον καλύτερο καθορισμό των όρων που αναφέρθηκαν θα αναλυθούν παρακάτω.

Η **λειτουργικότητα** της εφαρμογής αποτελεί τον πυρήνα της χρησιμότητάς της, καθορίζοντας την ικανότητά της να εκτελεί τις βασικές λειτουργίες που έχει σχεδιαστεί να υποστηρίζει. Αυτό περιλαμβάνει τη σωστή μεταφορά και εμφάνιση δεδομένων από τους αισθητήρες στον εξυπηρετητή, τη σωστή απόκριση σε ενέργειες του χρήστη και τη γενική αξιοπιστία των λειτουργιών της. Η ευκολία χρήσης, η σαφής δομή μενού και η αποτελεσματική διαχείριση των δεδομένων συμβάλλουν στη διασφάλιση της λειτουργικότητας της εφαρμογής, προσφέροντας την αναμενόμενη εμπειρία στον χρήστη.

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Η **αξιοπιστία** μιας εφαρμογής μετριέται από τη συνέπεια και την ακρίβεια της απόκρισής της σε διάφορες συνθήκες. Αυτό συμπεριλαμβάνει την αξιοπιστία στη μεταφορά δεδομένων, την ακρίβεια των μετρήσεων αισθητήρων και τη γενική της σταθερότητα. Μια αξιόπιστη εφαρμογή διασφαλίζει ότι οι χρήστες μπορούν να βασίζονται στις πληροφορίες που παρέχει, καθιστώντας την χρήσιμη και αξιόπιστη για τον σκοπό της.

Η **αισθητική** της εφαρμογής αναφέρεται στην οπτική της εμφάνιση και τον τρόπο με τον οποίο παρουσιάζει τις πληροφορίες στον χρήστη. Συμπεριλαμβάνει τη χρήση συνεπών στοιχείων σχεδιασμού, ευανάγνωστης γραμματοσειράς, εναρμονισμένων χρωμάτων και γενικά μια ευχάριστη και εύκολη για τον χρήστη αισθητική. Η ευπροσάρμοστη και καλαίσθητη διεπαφή βοηθά στη δημιουργία μιας θετικής εμπειρίας χρήστη και στην αποδοχή της εφαρμογής.

Μετά την κατανόηση των παραπάνω όρων, οι χρήστες εκτέλεσαν την εφαρμογή, ερωτήθηκαν για την εμπειρία τους κατά τη χρήση, και ζητήθηκε από αυτούς να αξιολογήσουν την λειτουργικότητα, την αξιοπιστία και την αισθητική της. Οι απόψεις και οι παρατηρήσεις τους θα παρουσιαστούν πιο αναλυτικά στη συνέχεια της αξιολόγησης.

5.2.1 ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ

Για την αξιολόγηση της λειτουργικότητας της εφαρμογής, τα άτομα που συμμετείχαν ερωτήθηκαν για διάφορα θέματα που καλύπτουν τη συνολική εμπειρία τους κατά τη χρήση της. Τα ερωτήματα είναι:

- Πώς βαθμολογείτε τη γενική ευχρηστία της εφαρμογής;
- Είναι εύκολο να πλοηγηθείτε στο μενού και να βρείτε τις επιθυμητές λειτουργίες;
- Πώς αντιλαμβάνεστε την ταχύτητα απόκρισης της εφαρμογής κατά τη διάρκεια της χρήσης;
- Παρατηρήσατε κάποια ασυνήθιστη συμπεριφορά ή προβλήματα κατά τη χρήση;
- Πώς αξιολογείτε τη συνολική επίδοση της εφαρμογής κατά την μεταφορά δεδομένων από τους αισθητήρες στον διακομιστή;

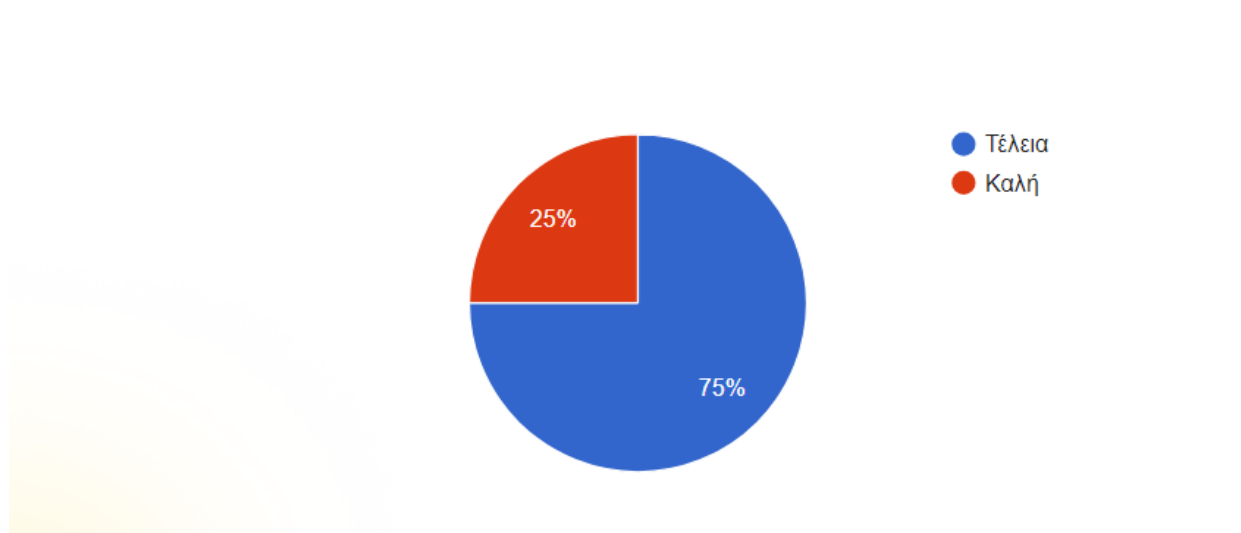
Η αξιολόγηση της εφαρμογής από τα τρία άτομα ήταν εξαιρετικά θετική, ενώ ένα άτομο εξέφρασε ικανοποίηση με επιθυμία για περαιτέρω λειτουργίες. Τα τρία ικανοποιημένα άτομα

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

υπογράμμισαν την αποτελεσματική μεταφορά δεδομένων από τους αισθητήρες στον διακομιστή και την ομαλή λειτουργία της εφαρμογής. Ωστόσο, το ένα άτομο που εξέφρασε μικρότερη ικανοποίηση έκανε συγκεκριμένες προτάσεις για βελτιώσεις. Αυτό το άτομο επιθυμούσε επιπλέον λειτουργίες σχετικά με τον έλεγχο της υγρασίας, και πρότεινε την προσθήκη χειροκίνητου ελέγχου υγρασίας, όπως υπάρχει στον έλεγχο της θερμοκρασίας και του φωτισμού. Επίσης, πρότεινε τη δυνατότητα ελέγχου για περισσότερους τύπους δεδομένων, όπως ο αισθητήρας κίνησης.

Λειτουργικότητα



Σχήμα 5.2.1: Στατιστικά αξιολόγησης λειτουργικότητας

5.2.2 ΑΞΙΟΠΙΣΤΙΑ

Για την αξιολόγηση της αξιοπιστίας της εφαρμογής, τα άτομα που συμμετείχαν ερωτήθηκαν για διάφορα θέματα που καλύπτουν τη σταθερότητα και την ακρίβεια της λειτουργίας της. Τα ερωτήματα είναι:

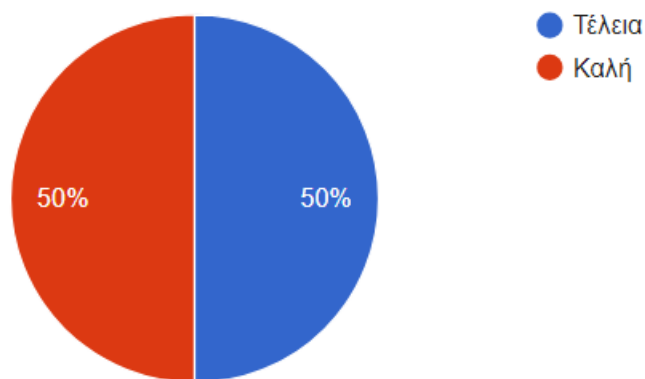
- Πόσο αξιόπιστα μεταφέρονται τα δεδομένα από τους αισθητήρες στην εφαρμογή;
- Πώς αντιμετωπίζει η εφαρμογή αλλαγές στις συνθήκες λειτουργίας, όπως π.χ. απώλεια σύνδεσης με το δίκτυο;
- Υπήρχαν περιπτώσεις σφαλμάτων ή απώλειας δεδομένων κατά τη λειτουργία της εφαρμογής;

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Κατά την αξιολόγηση της αξιοπιστίας της εφαρμογής, δύο από τα τέσσερα άτομα ανέφεραν ότι είχαν τελείως ικανοποιητική εμπειρία, επισημαίνοντας τη σταθερότητα και την ακρίβεια των δεδομένων. Τα άλλα δύο άτομα, παρόλο που ήταν ικανοποιημένα, πρότειναν μερικές αλλαγές για τη βελτίωση της αξιοπιστίας. Συγκεκριμένα, αναφέρθηκαν σε πιθανές βελτιώσεις όσον αφορά την ευαισθησία των αισθητήρων και τη δυνατότητα ενσωμάτωσης περισσότερων ελέγχων για την ποιότητα των δεδομένων. Επίσης, προέτρεψαν για την ανάληψη πρωτοβουλιών προς την κατεύθυνση ενδεχόμενων βελτιώσεων στην ακρίβεια των μετρήσεων.

Αξιοπιστία



Σχήμα 5.2.2: Στατιστικά αξιολόγησης αξιοπιστίας

5.2.3 ΑΙΣΘΗΤΙΚΗ

Για την αξιολόγηση της αισθητικής της εφαρμογής, τα άτομα που συμμετείχαν ερωτήθηκαν για διάφορα θέματα που καλύπτουν τη χρήση και το σχεδιασμό της διεπαφής. Τα ερωτήματα είναι:

- Πώς αξιολογείτε τη γενική εμπειρία χρήσης της εφαρμογής;

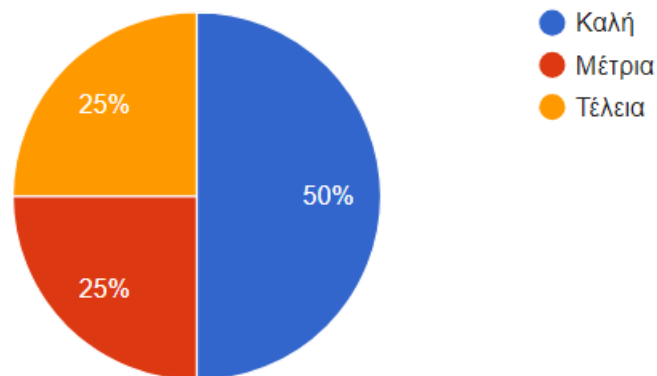
Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

- Είναι η διεπαφή εύχρηστη και εύκολα κατανοητή;
- Πώς αντιλαμβάνεστε τον γενικό σχεδιασμό και τον τρόπο παρουσίασης των δεδομένων;
- Θα προτιμούσατε κάποιες αλλαγές στο χρωματικό σχήμα ή τη διάταξη των στοιχείων;

Κατά την αξιολόγηση της αισθητικής της εφαρμογής, τα αποτελέσματα έδειξαν ποικιλία απόψεων. Δύο από τα τέσσερα άτομα ανέφεραν ότι είχαν μια καλή αισθητική εμπειρία, εκφράζοντας ικανοποίηση για τη σχεδίαση της εφαρμογής. Ένα άλλο άτομο περιέγραψε την εμπειρία του ως μετριότητα, ενώ το τέταρτο άτομο την χαρακτήρισε τελείως ικανοποιητική, εκφράζοντας την αναγνώριση της προσεγμένης αισθητικής. Οι προτάσεις για αλλαγές επεκτείνονται σε θέματα που βελτιώνουν τη γενική αισθητική, όπως η προσθήκη επιπλέον γραφικών στοιχείων ή η ενσωμάτωση περισσότερων επιλογών προσαρμογής για τον χρήστη.

Αισθητική



Σχήμα 5.2.3: Στατιστικά αξιολόγησης αισθητικής

ΚΕΦΑΛΑΙΟ 6: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ

6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Η Εφαρμογή διαχείρισης δεδομένων προσφέρει μια πολύ αποτελεσματική και ευέλικτη λύση για τον έλεγχο του περιβάλλοντος, ενσωματώνοντας έξυπνους αισθητήρες και ελεγκτές. Με βάση τη σχεδίαση και τη λειτουργικότητα της εφαρμογής, προκύπτουν πολλά θετικά συμπεράσματα.

Καταρχάς, η επικοινωνία του backend με το Arduino και η συλλογή δεδομένων από αισθητήρες επιτρέπει τον συνεχή και ακριβή έλεγχο του περιβάλλοντος. Για παράδειγμα, η δυναμική προσαρμογή του φωτισμού βάσει της φωτεινότητας που υπάρχει στο χώρο αποτελεί ένα έξυπνο χαρακτηριστικό που συνδυάζει άνεση και εξοικονόμηση ενέργειας. Επίσης, η εφαρμογή προσφέρει δύο τρόπους ελέγχου του περιβάλλοντος: τον αυτόματο και τον χειροκίνητο. Ο χρήστης έχει τη δυνατότητα να παρεμβαίνει άμεσα ή να εμπιστεύεται το σύστημα για την αυτόματη ρύθμιση, παρέχοντας έτσι επιπλέον ευελιξία. Παράλληλα, το frontend της εφαρμογής παρέχει ένα φιλικό προς τον χρήστη περιβάλλον, επιτρέποντας την καλύτερη παρακολούθηση και έλεγχο των συσκευών ενώ επίσης η οπτικοποίηση δεδομένων μέσω γραφημάτων βελτιώνει την κατανόηση των πληροφοριών.

Παρατηρήθηκε ότι η εφαρμογή λειτουργεί με αξιοσημείωτη αποτελεσματικότητα, διακρίνοντας ο χρήστης άμεση ανταπόκριση κατά την αλλαγή των περιβαλλοντικών συνθηκών. Η έλλειψη καθυστέρησης κατά την ενημέρωση των τιμών και των μηνυμάτων αντανακλά την επίδοση του συστήματος, προσφέροντας στον χρήστη άμεση ενημέρωση. Αυτή η γρήγορη ανταπόκριση προσθέτει στην ομαλή και φυσική εμπειρία χρήστη, εξασφαλίζοντας ότι οι αλλαγές στο περιβάλλον αντανακλώνται άμεσα στον χρήστη. Αυτό το χαρακτηριστικό αποτελεί κομβικό πλεονέκτημα, καθώς εξασφαλίζει όχι μόνο την ακρίβεια στις πληροφορίες που παρουσιάζονται, αλλά και την αμεσότητα στην αλληλεπίδραση με το περιβάλλον.

Το κυρίως χαρακτηριστικό που διακρίνει την εφαρμογή όμως είναι το σύστημα κανόνων που την καθοδηγεί. Αυτό το σύστημα αποτελεί τον πυρήνα της λειτουργίας της εφαρμογής, προσφέροντας αξιόπιστη και αυτόνομη λήψη αποφάσεων. Μέσω του συστήματος κανόνων, η εφαρμογή αντιλαμβάνεται δυναμικά τις συνθήκες του περιβάλλοντος και προβαίνει σε προσαρμογές που ισορροπούν μεταξύ της εξοικονόμησης ενέργειας και της άνεσης του χρήστη. Αυτή η αυτοματοποιημένη λειτουργία επιτρέπει στην εφαρμογή να αντιδρά άμεσα σε μεταβολές όπως η φωτεινότητα, η θερμοκρασία, και η υγρασία, προσφέροντας ένα ομαλό και έξυπνο περιβάλλον. Οι κανόνες εκτελούν κρίσιμο ρόλο σε κάθε πτυχή της εφαρμογής, από τον έλεγχο του φωτισμού μέχρι

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

τη διαχείριση της θερμοκρασίας και την επικοινωνία με τον χρήστη. Η ύπαρξη της δυνατότητας απενεργοποίησης του συνολικού ελέγχου επιτρέπει στον χρήστη να έχει πλήρη έλεγχο και επιρροή, παρέχοντας προσαρμοστικότητα και ευελιξία. Συνοπτικά, το σύστημα κανόνων αναδεικνύει την ευφυΐα της εφαρμογής, παρέχοντας στον χρήστη μια σύγχρονη και αποτελεσματική εμπειρία διαχείρισης του περιβάλλοντος του.

6.2 ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ

Η εφαρμογή παρουσιάζει σημαντική επίτευξη στην ενοποίηση του Internet of Things (IoT) με τον έλεγχο και τη διαχείριση του περιβάλλοντος. Για μελλοντικές εξελίξεις και βελτιώσεις, υπάρχουν πολλές δυνατότητες που μπορούν να εξερευνηθούν.

Μια προοπτική είναι η επέκταση των δυνατοτήτων της εφαρμογής προσθέτοντας περισσότερους αισθητήρες και λειτουργίες, όπως αισθητήρες κίνησης, ήχου ή ακόμη και ενσωματώνοντας τεχνολογίες όπως την τεχνητή νοημοσύνη για εξατομικευμένες ρυθμίσεις. Η επέκταση των δυνατοτήτων της εφαρμογής με την προσθήκη περισσότερων αισθητήρων και λειτουργιών ανοίγει τον δρόμο για μια πιο εξελιγμένη και πολυποίκιλη εμπειρία. Προσθέτοντας αισθητήρες κίνησης ή ήχου, η εφαρμογή μπορεί να ανταποκριθεί σε επιπλέον παραμέτρους του περιβάλλοντος, επιτρέποντας πιο λεπτομερείς και εξατομικευμένες ρυθμίσεις. Η ενσωμάτωση τεχνολογιών όπως η τεχνητή νοημοσύνη μπορεί να επιτρέψει στην εφαρμογή να μαθαίνει από τις προτιμήσεις και τις συνήθειες του χρήστη, προσαρμόζοντας αυτόματα τις ρυθμίσεις για μια εξατομικευμένη εμπειρία. Αυτό μπορεί να περιλαμβάνει την προετοιμασία του περιβάλλοντος πριν την άφιξη του χρήστη ή ακόμη και την πρόβλεψη των αναγκών του με βάση τα προηγούμενα πρότυπα.

Επίσης, μια δυνατότητα είναι η περαιτέρω ενίσχυση του συστήματος κανόνων για πιο προηγμένες προβλέψεις και αντιδράσεις, λαμβάνοντας υπόψη περισσότερες παραμέτρους και συνθήκες. Ενώ ακόμα, η ενίσχυση της οπτικοποίησης των δεδομένων, προσθέτοντας πιο λεπτομερή γραφήματα και γραφικές παραστάσεις για μια ευκρινέστερη κατανόηση των αλλαγών. Ακόμα, μπορεί να εξεταστεί η δυνατότητα επέκτασης της εφαρμογής σε διαφορετικά περιβάλλοντα και σενάρια χρήσης, όπως εφαρμογές για επαγγελματικούς χώρους, εκπαιδευτικά περιβάλλοντα ή ακόμη και σε βιομηχανικές εφαρμογές, προσφέροντας προηγμένες λειτουργίες και προσαρμοσμένες λύσεις. Τέλος, με την εξέλιξη της τεχνολογίας, η εφαρμογή θα μπορούσε να επεκταθεί σε διάφορα περιβάλλοντα και

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

Εφαρμογή διαχείρισης δεδομένων από αισθητήρες και λήψη αποφάσεων

σενάρια, όπως στον επαγγελματικό χώρο, την υγεία ή ακόμη και σε έξυπνα κτίρια, ενώ ακόμα με περισσότερες λειτουργίες και δυνατότητες, η εφαρμογή θα πρόσφερε ολοκληρωμένες και εξατομικευμένες λύσεις για την ευεξία και την άνεση του χρήστη.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Michael Aaron Dennis, Robert Kahn, “Encyclopaedia Britannica”,
<https://www.britannica.com/technology/Internet/Foundation-of-the-Internet>

- [2] Alexander S. Gillis “What is the internet of things” ,
<https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT>

- [3] Alessandro Bassi, Martin Bauer, Martin Fiedler, Thorsten Kramp, Rob Kranenburg, Sebastian Lange, Stefan Meissner, “Designing IoT solutions with the IoT Architectural Reference Model“, <https://link.springer.com/book/10.1007/978-3-642-40403-0>

- [4] Sachin Kumar, Prayag Tiwari, Mikhail Zymbler , “Journal of Big Data”,
<https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0268-2>

- [5] S. Medhat, "Client/server computing-an engine for change and growth," International Seminar on Client/Server Computing. Key Note Addresses, La Hulpe, Belgium, 1995, pp. 1/1-120

- [6] Britannica, The Editors of Encyclopaedia. "client-server architecture". Encyclopedia Britannica, 3 Nov. 2023, <https://www.britannica.com/technology/client-server-architecture>.

- [7] S. Medhat, "Client/server computing-an engine for change and growth," International Seminar on Client/Server Computing. Key Note Addresses, La Hulpe, Belgium, 1995, pp. 1/1-120

- [8] C. Skelton, "Rhe Client/Server Challenge," International Seminar on Client/Server Computing. Key Note Addresses, La Hulpe, Belgium, 1995

- [9] H. Patni, C. Henson and A. Sheth, "Linked sensor data," 2010 International Symposium on Collaborative Technologies and Systems, Chicago, IL, USA, 2010, pp. 362-370
- [10] "Visualising the client server architecture" <https://www.toolsqa.com/client-server/client-server-architecture-and-model>
- [11] Krishnamurthi, Rajalakshmi, et al. "An overview of IoT sensor data processing, fusion, and analysis techniques." *Sensors* 20.21 (2020)
- [12] Van Rossum, Guido. *An introduction to Python*. Ed. Fred L. Drake. Bristol: Network Theory Ltd., 2003.
- [13] Jensen, Simon Holm, Anders Møller, and Peter Thiemann. "Type analysis for JavaScript." *Static Analysis: 16th International Symposium, SAS 2009, Los Angeles, CA, USA, August 9-11, 2009. Proceedings 16*. Springer Berlin Heidelberg, 2009.
- [14] Macaulay, Michael. *Introduction to web interaction design: With Html and Css*. CRC Press, 2017.
- [15] Ferreira, Hiro & Canedo, E.D. & de Sousa Junior, Rafael. (2013). IoT architecture to enable intercommunication through REST API and UPnP using IP, ZigBee and arduino. *International Conference on Wireless and Mobile Computing, Networking and Communications*. 53-60.
- [16] M. N. M. Aashiq, W. T. C. C. Kurera, M. G. S. P. Thilekaratne, A. M. A. Saja, M. R. M. Rouzin, Navod Neranjan, Hayti Yassin, An IoT-based handheld environmental and air quality monitoring station, *Acta IMEKO*, vol. 12, no. 3, article 6, September 2023, identifier: IMEKO-ACTA-12 (2023)-03-06

- [17] S. B. Kamble, P. R. P. Rao, A. S. Pingalkar, G. S. Chayal, IoT Based Weather Monitoring System, IJARIE, vol. 3, no. 2, 2017, pp. 2886–2891. Online [Accessed 26 July 2023]
http://ijarie.com/AdminUploadPdf/IoT_Baerd_Weather_Monitoring_System_ijarie4557.pdf
- [18] M. Alvarez-Campana, G. López, E. Vázquez, V. A. Villagrà, J. Berrocal, Smart CEI Moncloa: An IoT-based Platform for People Flow and Environmental Monitoring on a Smart University Campus, Sensors, vol. 17, no. 12, Dec. 2017, Art. no. 12. DOI: [10.3390/s17122856](https://doi.org/10.3390/s17122856)
- [19] S. R. Shinde, A. H. Karode, S. R. Suralkar, Review on - IoT Based Environment Monitoring System, Int. J. Electron. Commun. Eng. Technol. IJECET, vol. 8, no. 2, Apr. 2017, pp. 103–108. Online [Accessed 26 July 2023]
https://iaeme.com/MasterAdmin/Journal_uploads/IJECET/VOLUME_8_ISSUE_2/IJECET_08_02_014.pdf
- [20] S. Zafar, G. Miraj, R. Baloch, D. Murtaza, K. Arshad, An IoT Based Real-Time Environmental Monitoring System Using Arduino and Cloud Service, Eng. Technol. Appl. Sci. Res., vol. 8, no. 4, Aug. 2018, pp. 3238–3242. DOI: [10.48084/etasr.2144](https://doi.org/10.48084/etasr.2144)
- [21] F. J. J. Joseph, IoT Based Weather Monitoring System for Effective Analytics, Int. J. Eng. Adv. Technol. IJEAT, vol. 8, no.4, pp. 311–315, Apr. 2019. Online [Accessed 26 July 2023] <https://www.ijeat.org/wpcontent/uploads/papers/v8i4/D6100048419.pdf>
- [22] R. Deekshath, P. Dharanya, K. R. Dimpil Kabadia, G. Deepak Dinakaran, S. Shanthini, IoT Based Environmental Monitoring System using Arduino UNO and Thingspeak, Int. J. Sci. Technol. Eng., vol. 4, no. 9, Mar. 2018, pp. 68–75. Online [Accessed 26 July 2023] <http://www.ijste.org/articles/IJSTEV4I9025.pdf>

- [23] Arduino, 2013. Available at : <<http://www.arduino.cc/>> accessed at jun 7, 2013
- [24] M. Taştan, H. Gökozan, Real-Time Monitoring of Indoor Air Quality with Internet of Things-Based E-Nose, Appl. Sci., vol. 9, no. 16, Jan. 2019, Art. no. 16. DOI: 10.3390/app9163435
- [25] A. López-Vargas, M. Fuentes, M. V. García, F. J. MuñozRodríguez, Low-Cost Datalogger Intended for Remote Monitoring of Solar Photovoltaic Standalone Systems Based on Arduino™, IEEE Sens. J., vol. 19, no. 11, Jun. 2019, pp. 4308– 4320. DOI:10.1109/JSEN.2019.2898667
- [26] Λιάκος Χρήστος, Παναγόπουλος Αιμίλιος Χρήστος, “*Internet of Things Industry 4.0*” <https://polynoe.lib.uniwa.gr/xmlui/handle/11400/2603>
- [27] Kusuma S M, Assistant Professor, Department of telecommunication, MSRIT, Bangalore, India. “Home Automation Using Internet of Things
- [28] Niharika Shrotriya, Anjali Kulkarni, Priti Gadhave, International Journal of Science, Engineering and Technology Research (IJSETR), “SMART HOME USING WI-FI”
- [29] Microsoft, “Why visual studio code”, March 30, 2022. [Online]. Available: <https://code.visualstudio.com/docs/editor/whyvscode>. [Accessed Feb. 16, 2022].
- [30] Redhat, 2020. “What is a REST API?”, May 8, 2020. [Online]. Available: <https://www.redhat.com/>. [Accessed Feb. 20, 2022].
- [31] Hashmi, Shifa and Pawar, Jayshree and Manegopale, Parmeshwar and Wagh, Kavita, IOT Based Weather Monitoring System Using Arduino-Uno (April 8, 2022). Available at SSRN: <https://ssrn.com/abstract=4113699>

- [32] Yan-e, D., 2011, March. Design of intelligent weather management information system based on IoT. In Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on (Vol. 1, pp. 1045-1049). IEEE.
- [33] Kamilaris, A., Gao, F., Prenafeta-Boldú, F.X. and Ali, M.I., 2016, December. AgriIoT: A semantic framework for Internet of Things-enabled smart monitoring applications. In Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on (pp. 442-447).IEEE
- [34] Arduino Based Weather Monitoring System .AUTHOR : Karthik Krishnamurthi, Suraj Thapa, Lokesh Kothari, Arun Prakash
- [35]] Nashwa El-Bendary, Mohamed Mostafa M. Fouad, Rabie A. Ramadan, Soumya Banerjee and Aboul Ella Hassanien, “Smart Environmental Monitoring Using Wireless Sensor Networks”, K15146_C025.indd, 2013
- [36] “IOT BASED WEATHER MONITORING USING ARDUINO UNO” : Sanket Gosavi, Devraj Bhosale, Sourabh Bhide, Prof. A.A. Sutar 2022
- [37] Sean Dieter Tebje Kelly, Nagender Kumar Suryadevara et al., “Towards the Implementation of IoT for Environmental Condition Monitoring in Homes”, IEEE Sensors Journal, Vol, 13, No. 10, October 2013.
- [38] George Mois, SilviuFolea, et al., “Analysis of Three IoT-Based Wireless Sensors for Environmental Monitoring”, IEEE Transaction on Instrumentation and Measurement 2017.

- [39] DivyavaniPalle, ArunaKommu, “Design and Development of CC3200-based Cloud IoT for measuring Humidity and Temperature”, International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)-2016.
- [40] Shiny Abraham and Joshua Beard, “Remote Environmental Monitoring Using Internet of Things” IEEE Transaction on Instrumentation and Measurement 2017.
- [41] Shifeng Fang, Li Da Xu et al., “An Integrated System for Regional Environmental Monitoring and Management Based on Internet of Things”, IEEE Transactions on Industrial Informatics, Vol. 10, No.2, May 2014.
- [42] J.Cabra, D.Castro et al., “An IoT approach for Wireless Sensor Networks applied to e-health environmental monitoring”, 2017 IEEE International Conference on Internet of Things and IEEE Green Computing and Communications and IEEE Cyber, Physical and Social computing and IEEE Smart Data.
- [43] Nikolas Vidakis et al., “Environmental Monitoring through Embedded System and Sensors”, 52nd International conference on Power Engineering Universities, 2017.
- [44] HakanUcgun, et al., “Arduino Based Weather Forecasting Station”, 2nd International Conference on Computer Science and Engineering 2017.
- [45] S. Tozlu, M. Senel, W. Mao and A. Keshavarzian "Wi-Fi enabled sensors for internet of things: A practical approach", IEEE Commun. Mag., vol. 50, no. 6, pp.134 -143 2012.
- [46] Somansh Kumar, “Air Quality Monitoring System Based on IoT using Raspberry Pi”, International Conference on Computing, Communication and Automation (ICCCA 2017).

[47] IoT based Data Logger System for weather monitoring using Wireless sensor networks, [online] Available: www.ijettjournal.org/2016/volume-32/number-2/IJETT-V32P213.pdf.

[48] D. Larios, J. Barbancho, G. Rodríguez, J. Sevillano, F. Molina and C. León "Energy efficient wireless sensor network communications based on computational intelligent data fusion for environmental monitoring", IET Commun., vol. 6, no. 14, pp.2189 -2197 2012

[49] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista and Michele Zorzi, "Internet of Things for Smart Cities", IEEE Internet of Things Journal, vol. 1, no. 1, February 2014.

[50] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis and M. Welsh "Wireless sensor networks for healthcare", Proc. IEEE, vol. 98, no. 11, pp.1947 -1960 2010

[51] Kshitij shingal, Arti Noor, Neelam Srivastava and Raghuvir Singh, Intelligent Humidity Sensor for Wireless Sensor Network Agricultural Application, vol. 3, no. 1, February 2011.

[52] Norberto Barroca, Luís M. Borges, Fernando J. Velez, Filipe Monteiro, Marcin Górski and João Castro-Gomes, "Wireless sensor networks for temperature and humidity monitoring with concrete structures", Construction and Building materials, pp. 1156-1166, December 2015.

[53] Wu, W.; Cao, J.; Wu, H.; Li, J. Robust and dynamic data aggregation in wireless sensor networks: A cross-layer approach. Comput. Netw. 2013, 57, 3929–3940.

- [54] He, J.; Wei, J.; Chen, K.; Tang, Z.; Zhou, Y.; Zhang, Y. Multitier Fog Computing With Large-Scale IoT Data Analytics for Smart Cities. *IEEE Internet Things J.* 2018, 5, 677–686.
- [55] Wang, M.; Perera, C.; Jayaraman, P.P.; Zhang, M.; Strazdins, P.; Shyamsundar, R.K.; Ranjan, R. City data fusion: Sensor data fusion in the internet of things. *Int. J. Distrib. Syst. Technol.* 2016, 7, 15–36.
- [56] Hromic, H.; Le Phuoc, D.; Serrano, M.; AntoniĆ, A.; Žarko, I.P.; Hayes, C.; Decker, S. Real time analysis of sensor data for the Internet of Things by means of clustering and event processing. In *Proceedings of the IEEE International Conference on Communications 2015, London, UK, 8–12 June 2015*; pp. 685–691.
- [57] Aggarwal, C.C. An introduction to sensor data analytics. In *Managing and Mining Sensor Data*; Springer: Boston, MA, USA, 2013; pp. 1–8.
- [58] Raafat, H.M.; Hossain, M.S.; Essa, E.; Elmougy, S.; Tolba, A.S.; Muhammad, G.; Ghoneim, A. Fog Intelligence for Real-Time IoT Sensor Data Analytics. *IEEE Access* 2017, 5, 24062–24069.
- [59] Chen, D.; Wu, B.; Chen, T.; Dong, J. Development of distributed data sharing platform for multi-source IOT sensor data of agriculture and forestry. *Nongye Gongcheng Xuebao/Trans. Chin. Soc. Agric. Eng.* 2017, 33, 300–307.
- [60] Hua, C.; Yum, T.S.P. Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks. *IEEE/ACM Trans. Netw.* 2008, 16, 892–903.