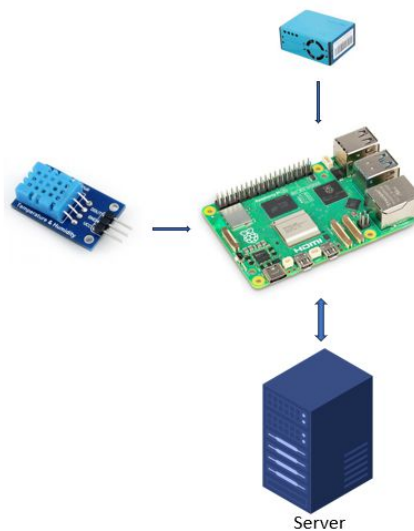


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Σύστημα παρακολούθησης ποιότητας αέρα και συνθήκες
άνεσης σε κλειστούς χώρους»



Φοιτητής

Μήτος Νικόλαος

516084

Επιβλέπων

Δρ. Κυριάκος Τσιακμάκης

Σεπτέμβριος 2024

Σύστημα παρακολούθησης ποιότητας αέρα και συνθήκες άνεσης σε κλειστούς χώρους

Κωδικός: 23306

Φοιτητής: Μήτος Νικόλαος

Εισηγητής: Δρ Κυριάκος Τσιακμάκης

Ημερομηνία ανάληψης Π.Ε. 31-10-2023

Ημερομηνία περάτωσης Π.Ε. 10-09-2024

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή **Μήτου Νικολάου** που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Περίληψη

Η εργασία αυτή στοχεύει στην ανάπτυξη ενός συστήματος παρακολούθησης της ποιότητας αέρα και των συνθηκών άνεσης σε κλειστούς χώρους, χρησιμοποιώντας το Raspberry Pi, τον αισθητήρα σωματιδίων PMS5003 και τον αισθητήρα θερμοκρασίας και υγρασίας DHT11. Τα δεδομένα θα αποστέλλονται σε έναν server που λειτουργεί με Laravel, όπου αποθηκεύονται σε βάση δεδομένων MySQL. Οι χρήστες μπορούν να συνδέονται σε μια ιστοσελίδα για να παρακολουθούν γραφήματα με τις συνθήκες που επικρατούν στον χώρο, καθώς και να έχουν μια γρήγορη εικόνα για την καταλληλότητα του χώρου όσον αφορά την ποιότητα αέρα και τις κλιματικές συνθήκες. Το σύστημα στοχεύει στη βελτίωση της άνεσης και της υγείας σε εσωτερικούς χώρους.

« Air quality monitoring system and indoor comfort conditions »

Abstract

This work aims to develop a system for monitoring indoor air quality and comfort conditions using Raspberry Pi, PMS5003 particle sensor and DHT11 temperature and humidity sensor. The data will be sent to a server running Laravel, where it is stored in a MySQL database. Users can log into a web page to view graphs of site conditions, as well as get a quick picture of site suitability in terms of air quality and climate conditions. The system aims to improve indoor comfort and health.

Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου προς τους γονείς μου για την αμέριστη συμπαράσταση και στήριξή τους καθ' όλη τη διάρκεια αυτής της προσπάθειας. Επίσης, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, για την πολύτιμη παιδαγωγική καθοδήγηση, τις επιστημονικές του συμβουλές, καθώς και την καθοριστική συνεισφορά του στην ανάπτυξη του κώδικα.

Περιεχόμενα

Περίληψη	iv
Abstract	v
Ευχαριστίες	vi
Περιεχόμενα	vii
Κατάλογος Σχημάτων	ix
Κεφάλαιο 1ο: Εισαγωγή.....	9
1.1 Εισαγωγή.....	9
1.2 Συνεισφορά της εργασίας.....	10
1.3 Δομή της εργασίας	11
Κεφάλαιο 2ο: Βιβλιογραφική Ανασκόπηση και Παρόμοια Συστήματα	12
2.1 Εισαγωγή.....	12
2.2 Βιβλιογραφική Ανασκόπηση	12
2.3 Παρόμοια Συστήματα	17
2.3.1 Awair Element	17
2.3.2 IQAir AirVisual Pro.....	18
2.3.3 Foobot	18
2.3.4 uHoo.....	19
2.3.5 Netatmo	20
Κεφάλαιο 3ο: Τεχνολογικά εργαλεία και συσκευές.....	21
3.1 Τεχνολογικά εργαλεία	21
3.1.1 PHP	21
3.1.2 Laravel.....	25
3.1.3 MySQL.....	32
3.1.4 Python	33
3.1.5 Javascript.....	34
3.1.6 Apache.....	36
3.2 Συσκευές – Αισθητήρες	36
3.2.1 Raspberry Pi.....	36
3.2.2 PMS5003.....	38

3.2.3	DHT11	39
Κεφάλαιο 4ο:	Το σύστημα παρακολούθησης ποιότητας αέρα και συνθήκες άνεσης σε κλειστούς χώρους	41
4.1	Εισαγωγή στο συνολικό σύστημα	41
4.2	Το σύστημα από την πλευρά της συσκευής	43
4.3	Το σύστημα από την πλευρά του server.....	45
4.3.1	Από την πλευρά του κόμβου	47
4.3.2	API για τα requests.....	50
4.3.3	Laravel για εξυπηρέτηση της ιστοσελίδας	52
4.3.4	Η βάση του συστήματος.....	57
4.4	Ασφάλεια Συστήματος	59
4.4.1	Ασφάλεια στη μετάδοση δεδομένων από τη συσκευή στον Server	60
4.4.2	Ασφάλεια στη πρόσβαση στην ιστοσελίδα	60
Κεφάλαιο 5ο:	Συμπεράσματα και προτάσεις βελτίωσης.....	61
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		62
ΠΑΡΑΡΤΗΜΑ Α		65

Κατάλογος Σχημάτων

Εικόνα 3.1: PHP	21
Εικόνα 3.2: Laravel	26
Εικόνα 3.3: Raspberry Pi 3	37
Εικόνα 3.4: PMS5003	39
Εικόνα 3.5: DHT11	40
Εικόνα 4.1: Το συνολικό σύστημα	41
Εικόνα 4.2: Το διάγραμμα ροής για το συνολικό σύστημα	42
Εικόνα 4.3: Το διάγραμμα ροής από την πλευρά της συσκευής-κόμβου	43
Εικόνα 4.4: Το διάγραμμα ροής από την πλευρά του API-server	45
Εικόνα 4.5: Το διάγραμμα ροής από την πλευρά του server - Ιστοσελίδα	46
Εικόνα 4.6: Σελίδα υποδοχής.....	52
Εικόνα 4.7: Σύνδεση χρήστη	53
Εικόνα 4.8: Προβολή places του χρήστη.....	53
Εικόνα 4.9: Σελίδα δημιουργίας του νέου χώρου	54
Εικόνα 4.10: Σελίδα τροποποίησης του χώρου	54
Εικόνα 4.11: Σελίδα προβολής των δεδομένων του κόμβου- ενιαίο γράφημα.....	55
Εικόνα 4.12: Σελίδα προβολής των δεδομένων του κόμβου- ξεχωριστά γραφήματα	56
Εικόνα 4.13: Σελίδα για Κατάσταση Αισθητήρων - Δείκτης Καταλληλότητας	57

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η ανάγκη για υγιείς και άνετους εσωτερικούς χώρους γίνεται ολοένα και πιο σημαντική, καθώς οι περισσότεροι άνθρωποι περνούν μεγάλο μέρος της ημέρας τους σε κλειστά περιβάλλοντα. Οι κλειστοί χώροι, είτε είναι οικιακοί, επαγγελματικοί ή δημόσιοι, μπορούν να επηρεάσουν σημαντικά την υγεία και την ευεξία των ανθρώπων. Παράγοντες όπως η ποιότητα του αέρα, η θερμοκρασία και η υγρασία διαδραματίζουν καθοριστικό ρόλο στην άνεση και την ασφάλεια των εσωτερικών χώρων. Η ανάπτυξη συστημάτων παρακολούθησης αυτών των παραμέτρων προσφέρει τη δυνατότητα για συνεχή έλεγχο και βελτίωση των συνθηκών διαβίωσης, προστατεύοντας τους ανθρώπους από πιθανούς κινδύνους, όπως η ατμοσφαιρική ρύπανση ή οι ακατάλληλες θερμοκρασίες.

Ένα σύστημα παρακολούθησης της ποιότητας αέρα και των συνθηκών άνεσης σε κλειστούς χώρους μπορεί να ενσωματώσει σύγχρονη τεχνολογία για τη συλλογή, ανάλυση και παρουσίαση δεδομένων σε πραγματικό χρόνο. Χρησιμοποιώντας αισθητήρες όπως ο PMS5003 για τη μέτρηση σωματιδίων στον αέρα και ο DHT11 για τη μέτρηση θερμοκρασίας και υγρασίας, τα δεδομένα μπορούν να αποστέλλονται σε έναν κεντρικό server για περαιτέρω επεξεργασία. Οι χρήστες έχουν τη δυνατότητα να παρακολουθούν την κατάσταση του χώρου μέσω μιας φιλικής προς το χρήστη διαδικτυακής πλατφόρμας, λαμβάνοντας έτσι έγκαιρες πληροφορίες για την ποιότητα του περιβάλλοντος τους. Αυτό όχι μόνο συμβάλλει στην καλύτερη διαχείριση των εσωτερικών συνθηκών, αλλά ενισχύει και την ευαισθητοποίηση σχετικά με την υγεία και την ασφάλεια σε καθημερινό επίπεδο.

Η συνεχής παρακολούθηση της ποιότητας αέρα και των συνθηκών άνεσης σε κλειστούς χώρους έχει γίνει κρίσιμη σε ένα κόσμο όπου η υγεία και η ευεξία είναι πρωταρχικής σημασίας. Η ποιότητα του αέρα μπορεί να επηρεαστεί από διάφορους παράγοντες, όπως η συγκέντρωση ρύπων, η κυκλοφορία του αέρα και η παρουσία αλλεργιογόνων, ενώ η θερμοκρασία και η υγρασία επηρεάζουν άμεσα την άνεση και τη λειτουργικότητα των εσωτερικών χώρων. Η παρακολούθηση αυτών των παραμέτρων βοηθά στην έγκαιρη ανίχνευση προβλημάτων και στην αποφυγή καταστάσεων που θα μπορούσαν να επηρεάσουν αρνητικά την υγεία ή την παραγωγικότητα των ανθρώπων που διαμένουν ή εργάζονται σε αυτούς τους χώρους.

Τέτοια συστήματα συνήθως συνδυάζουν προηγμένη τεχνολογία για τη συλλογή και ανάλυση δεδομένων, παρέχοντας στους χρήστες την δυνατότητα να έχουν ακριβείς πληροφορίες σε πραγματικό χρόνο. Μέσω αυτών των πληροφοριών, οι υπεύθυνοι των χώρων μπορούν να λαμβάνουν τεκμηριωμένες αποφάσεις για τη βελτίωση των συνθηκών, ενώ οι κάτοικοι ή οι εργαζόμενοι μπορούν

να προσαρμόζουν τη συμπεριφορά τους ανάλογα με τις ανάγκες τους. Με αυτό τον τρόπο, ενισχύεται η ευεξία και η ασφάλεια, προάγοντας ένα πιο υγιεινό και άνετο περιβάλλον διαβίωσης και εργασίας.

Ο κύριος στόχος αυτής της εργασίας είναι Η ανάπτυξη ενός ολοκληρωμένου συστήματος παρακολούθησης της ποιότητας αέρα και των συνθηκών άνεσης σε κλειστούς χώρους, με δυνατότητα άμεσης ενημέρωσης και ανάλυσης δεδομένων, προκειμένου να βελτιωθεί η υγεία και η ευεξία των ανθρώπων που διαμένουν ή εργάζονται σε αυτούς τους χώρους.

Συγκεκριμένα, οι επιμέρους στόχοι της εργασίας είναι:

- **Ακριβής συλλογή δεδομένων:** Εξασφάλιση της ακριβούς και συνεχούς συλλογής δεδομένων σχετικά με την ποιότητα αέρα, τη θερμοκρασία και την υγρασία σε πραγματικό χρόνο.
- **Ανάλυση και παρουσίαση δεδομένων:** Ανάπτυξη μιας διαδικτυακής πλατφόρμας που θα επιτρέπει στους χρήστες να παρακολουθούν εύκολα και κατανοητά τα δεδομένα μέσω γραφημάτων και αναφορών.
- **Βελτίωση της ευαισθητοποίησης:** Προώθηση της ευαισθητοποίησης σχετικά με τη σημασία της ποιότητας αέρα και των κλιματικών συνθηκών σε κλειστούς χώρους, παρέχοντας πληροφορίες για την υγεία και την άνεση.

1.2 Συνεισφορά της εργασίας

Η συνεισφορά της εργασίας αυτής είναι πολυεπίπεδη, συνδυάζοντας την τεχνολογία με την ευημερία των ανθρώπων σε καθημερινό επίπεδο. Αρχικά, παρέχει ένα καινοτόμο εργαλείο για την παρακολούθηση της ποιότητας αέρα και των συνθηκών άνεσης σε κλειστούς χώρους, το οποίο μπορεί να χρησιμοποιηθεί σε διάφορους τομείς, όπως κατοικίες, γραφεία, σχολεία και δημόσιους χώρους. Μέσω της ακριβούς συλλογής και ανάλυσης δεδομένων, το σύστημα αυτό συμβάλλει στην προληπτική διαχείριση της ποιότητας αέρα και της άνεσης, προλαμβάνοντας προβλήματα υγείας και βελτιώνοντας την καθημερινή εμπειρία των ανθρώπων.

Επιπλέον, η εργασία αυτή ενισχύει την ευαισθητοποίηση γύρω από τις συνθήκες διαβίωσης σε κλειστούς χώρους, προωθώντας τη σημασία της συνεχούς παρακολούθησης και της άμεσης επέμβασης όταν απαιτείται. Τέλος, το έργο μπορεί να αποτελέσει βάση για περαιτέρω έρευνα και ανάπτυξη σε τομείς όπως η έξυπνη διαχείριση κτιρίων και η βελτίωση της ποιότητας ζωής σε αστικά περιβάλλοντα, προσφέροντας ένα πλαίσιο για μελλοντικές καινοτομίες.

1.3 Δομή της εργασίας

Στο πρώτο κεφάλαιο παρουσιάζεται η εισαγωγή της εργασίας και οι στόχοι της.

Επίσης, αναφέρεται η συνεισφορά της εργασίας.

Στο δεύτερο κεφάλαιο παρουσιάζεται η βιβλιογραφική ανασκόπηση παρόμοιων συστημάτων.

Στο τρίτο κεφάλαιο περιγράφονται τα τεχνολογικά που χρησιμοποιήθηκαν.

Στο τέταρτο κεφάλαιο αναλύεται η εφαρμογή από την μεριά του Raspberry και των αισθητήρων του και η εφαρμογή-ιστοσελίδα στο server.

Στο πέμπτο κεφάλαιο παρουσιάζονται τα συμπεράσματα της εργασίας και θέματα για βελτιώσεις.

Στο τέλος της εργασίας παρατίθεται το παράρτημα με κώδικες που χρησιμοποιήθηκαν στην εργασία.

Κεφάλαιο 2ο: Βιβλιογραφική Ανασκόπηση και Παρόμοια Συστήματα

2.1 Εισαγωγή

Το κεφάλαιο αυτό θα επικεντρωθεί στη ανασκόπηση της υπάρχουσας βιβλιογραφίας και των παρόμοιων συστημάτων που έχουν αναπτυχθεί για την παρακολούθηση της ποιότητας αέρα και των συνθηκών άνεσης σε κλειστούς χώρους. Αρχικά, θα εξεταστούν οι βασικές θεωρητικές προσεγγίσεις και τα μοντέλα που έχουν προταθεί στη βιβλιογραφία για την παρακολούθηση και βελτίωση της ποιότητας του αέρα, της θερμοκρασίας και της υγρασίας. Στη συνέχεια, θα γίνει αναφορά στις τεχνολογικές λύσεις και τις συσκευές που χρησιμοποιούνται σε διάφορα συστήματα, περιλαμβάνοντας αναφορές σε αισθητήρες, λογισμικό επεξεργασίας δεδομένων, και μεθόδους παρουσίασης πληροφοριών στους χρήστες.

Επιπλέον, το κεφάλαιο θα περιγράψει και θα συγκρίνει υπάρχοντα εμπορικά και ερευνητικά συστήματα παρακολούθησης, αναλύοντας τα πλεονεκτήματα και τα μειονεκτήματά τους, καθώς και την αποτελεσματικότητά τους σε διάφορες εφαρμογές. Η σύγκριση αυτή θα βοηθήσει στην ανάδειξη των κενών και των αναγκών που υπάρχουν στην τρέχουσα αγορά και στη βιβλιογραφία, παρέχοντας το πλαίσιο για την ανάπτυξη του προτεινόμενου συστήματος στο πλαίσιο της εργασίας αυτής. Τέλος, θα εξεταστούν οι μελλοντικές τάσεις στον τομέα, προβάλλοντας πιθανές βελτιώσεις και καινοτομίες που θα μπορούσαν να ενσωματωθούν σε νέα συστήματα παρακολούθησης ποιότητας αέρα και άνεσης.

2.2 Βιβλιογραφική Ανασκόπηση

Ένα από τα παρόμοια συστήματα που έχουν αναπτυχθεί για την παρακολούθηση της ποιότητας αέρα σε κλειστούς χώρους είναι το σύστημα που περιγράφεται στο άρθρο "Reengineering Indoor Air Quality Monitoring Systems to Improve End-User Experience". [1]

Το σύστημα αυτό έχει σχεδιαστεί με γνώμονα τη βελτίωση της εμπειρίας του τελικού χρήστη και επικεντρώνεται στην παρακολούθηση της ποιότητας του αέρα σε εσωτερικούς χώρους χωρίς αυτόματο σύστημα εξαερισμού. Χρησιμοποιεί αισθητήρες χαμηλού κόστους και πολυπλοκότητας, επιτρέποντας την εύκολη υλοποίηση σε πολλαπλές ζώνες μέσα σε ένα κτίριο. Επιπλέον, προσφέρει γραφική απεικόνιση των δεδομένων σε πραγματικό χρόνο και δυνατότητα πρόβλεψης της εξέλιξης της ποιότητας του αέρα, κάτι που βοηθά τους χρήστες να αποφασίσουν αν πρέπει να αερίσουν τον χώρο τους. Αυτό το σύστημα είναι ένα παράδειγμα της τάσης προς τη δημιουργία προσιτών και φιλικών προς

το χρήστη λύσεων, που ενσωματώνουν σύγχρονη τεχνολογία για τη βελτίωση της καθημερινής ζωής των ανθρώπων που ζουν ή εργάζονται σε κλειστούς χώρους.

Ένα άλλο σχετικό σύστημα είναι αυτό που περιγράφεται στο άρθρο "Building an indoor air quality monitoring system based on the architecture of the Internet of Things". [2]

Το σύστημα αυτό χρησιμοποιεί την αρχιτεκτονική του Internet of Things (IoT) για την παρακολούθηση της ποιότητας του αέρα σε εσωτερικούς χώρους. Το σύστημα αυτό ενσωματώνει αισθητήρες για την ανίχνευση διάφορων ρύπων, όπως CO₂ και μικροσωματίδια (PM_{2.5}), και στέλνει τα δεδομένα σε πραγματικό χρόνο σε μια πλατφόρμα cloud. Οι χρήστες μπορούν να παρακολουθούν τις συνθήκες αέρα μέσω μιας διαδικτυακής διεπαφής και να λαμβάνουν ειδοποιήσεις όταν οι συνθήκες ξεπερνούν τα ασφαλή όρια.

Αυτό το σύστημα είναι ιδιαίτερα χρήσιμο για περιβάλλοντα με μεγάλες συγκεντρώσεις ανθρώπων ή σε περιοχές όπου η ποιότητα του αέρα μπορεί να επηρεάσει σημαντικά την υγεία, προσφέροντας μια λύση που ενσωματώνει σύγχρονες τεχνολογίες και διασύνδεση με το IoT για καλύτερη διαχείριση των εσωτερικών συνθηκών.

Ένα άλλο σχετικό σύστημα παρουσιάζεται στο άρθρο "A comprehensive review on indoor air quality monitoring systems for enhanced public health". [3]

Το άρθρο αυτό παρέχει μια εκτενή ανασκόπηση των συστημάτων παρακολούθησης της ποιότητας του αέρα σε εσωτερικούς χώρους, δίνοντας έμφαση στην επίδραση αυτών των συστημάτων στη δημόσια υγεία. Το σύστημα που περιγράφεται ενσωματώνει πολλούς αισθητήρες για την ανίχνευση ποικίλων ρύπων, όπως PM_{2.5}, VOCs, CO, και NO_x, και στοχεύει στη μείωση των κινδύνων που συνδέονται με τη μακροχρόνια έκθεση σε επιβλαβείς αέριους ρύπους σε εσωτερικούς χώρους. Το σύστημα παρακολουθεί συνεχώς την ποιότητα του αέρα και παρέχει δεδομένα σε πραγματικό χρόνο στους χρήστες μέσω μιας εύχρηστης διεπαφής.

Αυτό το σύστημα υπογραμμίζει τη σημασία της ενσωμάτωσης των τεχνολογιών παρακολούθησης αέρα σε καθημερινά περιβάλλοντα, προκειμένου να προστατευθεί η υγεία των κατοίκων από σοβαρές αναπνευστικές και καρδιαγγειακές παθήσεις.

Το άρθρο "A survey on air quality monitoring systems for indoor environments", δημοσιευμένο στο περιοδικό *Computer Networks*, [4] παρέχει μια εκτενή ανασκόπηση των συστημάτων παρακολούθησης ποιότητας αέρα σε εσωτερικούς χώρους, εστιάζοντας σε σύγχρονες τεχνολογικές λύσεις, αρχιτεκτονικές συστημάτων και προκλήσεις που υπάρχουν στον τομέα αυτό. Το άρθρο αναλύει τις διάφορες τεχνολογίες που χρησιμοποιούνται στα συστήματα παρακολούθησης ποιότητας αέρα, όπως οι αισθητήρες για την ανίχνευση σωματιδίων και αερίων (π.χ., CO₂, VOCs, PM_{2.5}). Επιπλέον, εξετάζεται η χρήση της τεχνολογίας IoT για τη διασύνδεση των αισθητήρων με πλατφόρμες cloud, επιτρέποντας τη συνεχή παρακολούθηση και αποθήκευση δεδομένων. Οι συντάκτες του άρθρου

συζητούν τις μεθόδους ανάλυσης δεδομένων που χρησιμοποιούνται για την επεξεργασία των πληροφοριών που συλλέγονται από τους αισθητήρες. Περιλαμβάνουν την εφαρμογή αλγορίθμων μηχανικής μάθησης για την πρόβλεψη της εξέλιξης της ποιότητας αέρα, δίνοντας τη δυνατότητα για προληπτικές ενέργειες από τους χρήστες. Το άρθρο επισημαίνει τις προκλήσεις που παραμένουν στην ανάπτυξη αυτών των συστημάτων, όπως η ακρίβεια των αισθητήρων, η διαλειτουργικότητα μεταξύ διαφορετικών συστημάτων, και η ανάγκη για προσαρμοσμένες λύσεις ανάλογα με τον τύπο του κλειστού χώρου. Τέλος, συζητούνται οι μελλοντικές τάσεις, με έμφαση στη βελτίωση των αισθητήρων και των τεχνικών ανάλυσης δεδομένων, καθώς και στην αυξανόμενη ενσωμάτωση των συστημάτων αυτών στην καθημερινή ζωή.

Το άρθρο "An IoT-Based Air Quality Monitoring System with Predictive Analysis" [5]

που δημοσιεύτηκε στο *Electronics* παρέχει μια λεπτομερή περιγραφή ενός συστήματος παρακολούθησης ποιότητας αέρα βασισμένου στο Internet of Things (IoT). Το σύστημα αυτό είναι σχεδιασμένο για να παρακολουθεί τις συνθήκες αέρα σε πραγματικό χρόνο, με την ικανότητα να κάνει προγνωστική ανάλυση για μελλοντικές συνθήκες ποιότητας αέρα. Το άρθρο εξετάζει την αρχιτεκτονική του συστήματος, περιλαμβάνοντας τη χρήση αισθητήρων για τη συλλογή δεδομένων όπως τα επίπεδα PM2.5, CO2, και άλλων ρύπων.

Οι πληροφορίες αυτές μεταδίδονται σε ένα κεντρικό σύστημα μέσω IoT, όπου αναλύονται με τη χρήση αλγορίθμων μηχανικής μάθησης για την πρόβλεψη των μελλοντικών συνθηκών. Το άρθρο τονίζει τη σημασία της προγνωστικής ανάλυσης, καθώς αυτή επιτρέπει στους χρήστες να λαμβάνουν προληπτικά μέτρα για τη βελτίωση της ποιότητας του αέρα, όπως ο εξαερισμός ή η ενεργοποίηση καθαριστών αέρα πριν τα επίπεδα ρύπων γίνουν επικίνδυνα.

Το σύστημα αυτό είναι ιδιαίτερα χρήσιμο για περιβάλλοντα όπου η ποιότητα του αέρα μπορεί να έχει σημαντικές επιπτώσεις στην υγεία, όπως γραφεία, σχολεία, και βιομηχανικούς χώρους.

Το άρθρο "Achieving better indoor air quality with IoT systems for future buildings: Opportunities and challenges" [6]

που δημοσιεύτηκε στο περιοδικό *Science of The Total Environment* εξετάζει τη χρήση συστημάτων Internet of Things (IoT) για τη βελτίωση της ποιότητας αέρα σε εσωτερικούς χώρους, ειδικά σε μελλοντικά κτίρια. Το άρθρο αναλύει τις ευκαιρίες που προσφέρουν τα IoT συστήματα για τη συνεχή παρακολούθηση και διαχείριση της ποιότητας αέρα σε πραγματικό χρόνο, επιτρέποντας την πρόληψη και αντιμετώπιση προβλημάτων που σχετίζονται με την υγεία και την άνεση των κατοίκων.

Επίσης, το άρθρο συζητά τις προκλήσεις που αντιμετωπίζουν τα IoT συστήματα, όπως η ασφάλεια των δεδομένων, η ανάγκη για διαλειτουργικότητα μεταξύ διαφορετικών συσκευών και συστημάτων, και η ενεργειακή αποδοτικότητα των αισθητήρων και των άλλων συστατικών. Αυτές οι προκλήσεις πρέπει

να αντιμετωπιστούν προκειμένου τα IoT συστήματα να αποτελέσουν έναν βασικό πυλώνα στη βελτίωση της ποιότητας αέρα σε κλειστούς χώρους στο μέλλον.

Το άρθρο "Indoor Air Quality Monitoring Systems Based on Internet of Things: A Systematic Review" [7] που δημοσιεύτηκε στο περιοδικό *International Journal of Environmental Research and Public Health* εστιάζει στη χρήση συστημάτων παρακολούθησης ποιότητας αέρα εσωτερικών χώρων βασισμένων στο Internet of Things (IoT). Το άρθρο αναλύει πώς τα IoT συστήματα μπορούν να συμβάλουν στην ενίσχυση της δημόσιας υγείας σε έξυπνες πόλεις, παρακολουθώντας συνεχώς τη συγκέντρωση επικίνδυνων ρύπων και παρέχοντας δεδομένα σε πραγματικό χρόνο. Αυτά τα δεδομένα επιτρέπουν την άμεση λήψη αποφάσεων για τη βελτίωση της ποιότητας του αέρα και την προστασία των κατοίκων από τους κινδύνους της ατμοσφαιρικής ρύπανσης.

Το άρθρο διερευνά επίσης τις τεχνολογικές προκλήσεις και τις απαιτήσεις για την αποτελεσματική εφαρμογή τέτοιων συστημάτων, όπως η ενεργειακή αποδοτικότητα, η ακρίβεια των αισθητήρων, και η διαχείριση μεγάλων δεδομένων (Big Data). Οι συντάκτες επισημαίνουν τη σημασία της διαλειτουργικότητας και της κλιμάκωσης αυτών των συστημάτων, ειδικά καθώς οι έξυπνες πόλεις συνεχίζουν να αναπτύσσονται και να υιοθετούν νέες τεχνολογίες.

Το άρθρο "Development of low-cost indoor air quality monitoring devices: Recent advancements"

[8] που δημοσιεύτηκε στο περιοδικό *Science of The Total Environment* εξετάζει τις πρόσφατες εξελίξεις στην ανάπτυξη χαμηλού κόστους συσκευών για την παρακολούθηση της ποιότητας αέρα σε εσωτερικούς χώρους. Το άρθρο εστιάζει στις τεχνολογίες που χρησιμοποιούνται για την κατασκευή αυτών των συσκευών, την ακρίβεια των αισθητήρων, καθώς και στις προκλήσεις και τις μελλοντικές κατευθύνσεις στον τομέα.

Αναλύονται οι προοπτικές για τη χρήση αυτών των συσκευών σε εφαρμογές ευρείας κλίμακας, καθώς και οι επιπτώσεις τους στη βελτίωση της δημόσιας υγείας και της ποιότητας ζωής σε αστικές και αγροτικές περιοχές.

Το άρθρο "A comprehensive review on indoor air quality monitoring systems for enhanced public health" [9] που δημοσιεύτηκε στο *Sustainable Environment Research* παρέχει μια εκτενή ανασκόπηση των συστημάτων παρακολούθησης ποιότητας αέρα σε εσωτερικούς χώρους, με έμφαση στη βελτίωση της δημόσιας υγείας. Το άρθρο εξετάζει τις διάφορες τεχνολογίες και τις αρχιτεκτονικές συστημάτων που έχουν αναπτυχθεί για την παρακολούθηση της ποιότητας αέρα, περιλαμβάνοντας αναφορές σε αισθητήρες, συστήματα IoT, και τις προκλήσεις που αντιμετωπίζουν αυτά τα συστήματα στην πράξη.

Επιπλέον, το άρθρο αναλύει τη σημασία της συνεχούς παρακολούθησης της ποιότητας αέρα για τη μείωση των κινδύνων που συνδέονται με την ατμοσφαιρική ρύπανση σε κλειστούς χώρους, καθώς και τις στρατηγικές που μπορούν να υιοθετηθούν για τη βελτίωση της ποιότητας του αέρα σε αστικά και αγροτικά περιβάλλοντα.

Το άρθρο "A Cost-effective Wireless Sensor Network System for Indoor Air Quality Monitoring Applications" [10] που δημοσιεύτηκε στο *Procedia Computer Science* εξετάζει την ανάπτυξη ενός οικονομικά αποδοτικού ασύρματου δικτύου αισθητήρων για εφαρμογές παρακολούθησης της ποιότητας του αέρα σε εσωτερικούς χώρους. Το άρθρο περιγράφει τη σχεδίαση, την υλοποίηση και την αξιολόγηση ενός συστήματος που χρησιμοποιεί χαμηλού κόστους αισθητήρες και ασύρματη τεχνολογία για τη συλλογή και μετάδοση δεδομένων σχετικά με την ποιότητα του αέρα.

Οι συγγραφείς αναλύουν τα πλεονεκτήματα ενός τέτοιου συστήματος, συμπεριλαμβανομένης της ευκολίας εγκατάστασης, της επεκτασιμότητας και της δυνατότητας ενσωμάτωσης σε υπάρχουσες υποδομές χωρίς να απαιτούνται σημαντικοί οικονομικοί πόροι. Το σύστημα έχει σχεδιαστεί για να προσφέρει σε πραγματικό χρόνο δεδομένα που μπορούν να χρησιμοποιηθούν για τη βελτίωση της ποιότητας του αέρα σε κλειστούς χώρους, συμβάλλοντας έτσι στη βελτίωση της υγείας και της άνεσης των κατοίκων.

Το άρθρο "A Modular IoT Platform for Real-Time Indoor Air Quality Monitoring"

[11] που δημοσιεύτηκε στο περιοδικό *Sensors* περιγράφει την ανάπτυξη μιας αρθρωτής πλατφόρμας IoT (Internet of Things) για την παρακολούθηση της ποιότητας του αέρα σε εσωτερικούς χώρους σε πραγματικό χρόνο. Η πλατφόρμα αυτή είναι σχεδιασμένη με τρόπο που επιτρέπει την εύκολη προσαρμογή και επέκταση ανάλογα με τις ανάγκες του χρήστη, χρησιμοποιώντας μια σειρά από αισθητήρες για την ανίχνευση διάφορων παραμέτρων ποιότητας αέρα, όπως τα επίπεδα CO₂, PM_{2.5}, και άλλων ρύπων.

Το σύστημα επιτρέπει τη συλλογή και μετάδοση δεδομένων σε πραγματικό χρόνο, τα οποία μπορούν να αναλυθούν και να παρουσιαστούν στους χρήστες μέσω μιας φιλικής προς το χρήστη διεπαφής. Η αρθρωτή φύση της πλατφόρμας διευκολύνει την ενσωμάτωση επιπλέον αισθητήρων ή την προσαρμογή του συστήματος για διαφορετικά περιβάλλοντα ή απαιτήσεις.

Το άρθρο "Development of an IoT-Based Indoor Air Quality Monitoring Platform" [12]

που δημοσιεύτηκε στο *Hindawi's Journal of Sensors* περιγράφει την ανάπτυξη μιας πλατφόρμας παρακολούθησης της ποιότητας αέρα σε εσωτερικούς χώρους, η οποία βασίζεται στην τεχνολογία Internet of Things (IoT). Η πλατφόρμα αυτή είναι σχεδιασμένη για να παρέχει συνεχή παρακολούθηση των επιπέδων ατμοσφαιρικών ρύπων όπως το CO₂, οι πτητικές οργανικές ενώσεις (VOCs), και άλλα σημαντικά συστατικά που επηρεάζουν την ποιότητα του αέρα. Αναλύει την αρχιτεκτονική του συστήματος, περιλαμβάνοντας τους αισθητήρες, τα δίκτυα δεδομένων, και τις τεχνολογίες ανάλυσης που χρησιμοποιούνται για να διασφαλιστεί η ακριβής και αξιόπιστη μέτρηση της ποιότητας του αέρα. Επιπλέον, εξετάζονται οι δυνατότητες του συστήματος για την παροχή ειδοποιήσεων σε πραγματικό χρόνο στους χρήστες, ώστε να λαμβάνουν άμεσα μέτρα για τη βελτίωση της ποιότητας του αέρα, αν χρειαστεί. Τονίζει επίσης τα πλεονεκτήματα και τις προκλήσεις που σχετίζονται με την υιοθέτηση

τέτοιων συστημάτων, ειδικά σε περιβάλλοντα με αυξημένες απαιτήσεις όπως γραφεία, σχολεία, και κατοικίες.

2.3 Παρόμοια Συστήματα

2.3.1 Awair Element

Το **Awair Element** [13είναι ένα προηγμένο σύστημα παρακολούθησης ποιότητας αέρα, σχεδιασμένο για να βοηθά τους χρήστες να διατηρούν καθαρό και υγιεινό αέρα στους εσωτερικούς τους χώρους. Το Awair Element μετρά κρίσιμες παραμέτρους όπως τα επίπεδα των αιωρούμενων σωματιδίων PM2.5, τις πτητικές οργανικές ενώσεις (VOCs), το διοξείδιο του άνθρακα (CO2), τη θερμοκρασία και την υγρασία. Τα δεδομένα που συλλέγονται στέλνονται σε πραγματικό χρόνο σε έναν κεντρικό server, όπου οι χρήστες μπορούν να τα παρακολουθούν μέσω μιας φιλικής προς τον χρήστη εφαρμογής για κινητά ή μιας διαδικτυακής πλατφόρμας. Αυτή η δυνατότητα προσφέρει μια ολοκληρωμένη εικόνα των συνθηκών αέρα στο σπίτι ή στον χώρο εργασίας, επιτρέποντας στους χρήστες να λαμβάνουν τεκμηριωμένες αποφάσεις για τη βελτίωση της ποιότητας του αέρα.

Το Awair Element παρέχει όχι μόνο απλή παρακολούθηση αλλά και ενημερωτικές συμβουλές για το πώς να βελτιωθεί η ποιότητα του αέρα με βάση τα δεδομένα που συλλέγει. Για παράδειγμα, αν το επίπεδο των VOCs είναι υψηλό, το σύστημα μπορεί να προτείνει αερισμό του χώρου ή περιορισμό της χρήσης χημικών προϊόντων. Η εφαρμογή επιτρέπει στους χρήστες να βλέπουν ιστορικά δεδομένα και να αναλύουν τις τάσεις στη διάρκεια του χρόνου, κάτι που είναι εξαιρετικά χρήσιμο για την κατανόηση των παραγόντων που επηρεάζουν την ποιότητα του αέρα. Επιπλέον, η πλατφόρμα μπορεί να συνδεθεί με άλλες έξυπνες συσκευές, επιτρέποντας αυτοματοποιημένες ενέργειες, όπως η ενεργοποίηση ενός καθαριστή αέρα όταν οι συνθήκες αέρα χειροτερεύουν.

Το σύστημα Awair Element είναι ιδιαίτερα δημοφιλές λόγω της ευκολίας χρήσης του και της υψηλής του ακρίβειας. Με τη δυνατότητα πρόσβασης στα δεδομένα από οπουδήποτε μέσω της εφαρμογής, οι χρήστες μπορούν να παρακολουθούν και να βελτιώνουν την ποιότητα του αέρα σε πραγματικό χρόνο, ακόμη και όταν δεν βρίσκονται στον χώρο. Το Awair Element απευθύνεται σε όσους ενδιαφέρονται για την υγεία και την ευεξία τους, προσφέροντας μια εύκολη λύση για τη διαχείριση της ποιότητας του αέρα και την αποφυγή προβλημάτων υγείας που συνδέονται με την ατμοσφαιρική ρύπανση στους εσωτερικούς χώρους.

2.3.2 IQAir AirVisual Pro

Το **IQAir AirVisual Pro** [14] είναι ένα ολοκληρωμένο σύστημα παρακολούθησης ποιότητας αέρα που συνδυάζει υψηλή ακρίβεια με ευκολία στη χρήση. Το σύστημα είναι εξοπλισμένο με αισθητήρες που μετρούν διάφορες παραμέτρους όπως τα επίπεδα PM2.5, το διοξείδιο του άνθρακα (CO₂), τη θερμοκρασία και την υγρασία, τόσο για εσωτερικούς όσο και για εξωτερικούς χώρους. Τα δεδομένα που συλλέγει το AirVisual Pro αποστέλλονται σε πραγματικό χρόνο σε έναν cloud server, επιτρέποντας στους χρήστες να παρακολουθούν τις συνθήκες αέρα μέσω μιας διαδικτυακής πλατφόρμας ή μιας εφαρμογής για κινητά. Αυτή η δυνατότητα παρέχει μια λεπτομερή εικόνα των συνθηκών αέρα στον χώρο, προσφέροντας στους χρήστες τη δυνατότητα να λαμβάνουν τεκμηριωμένες αποφάσεις για τη βελτίωση της ποιότητας του αέρα.

Το AirVisual Pro παρέχει επίσης τη δυνατότητα σύγκρισης των δεδομένων από τον εσωτερικό χώρο με δεδομένα εξωτερικού περιβάλλοντος, χρησιμοποιώντας τοπικούς μετεωρολογικούς σταθμούς. Αυτό επιτρέπει στους χρήστες να κατανοήσουν καλύτερα πώς επηρεάζονται οι εσωτερικές συνθήκες από τις εξωτερικές, και να προσαρμόσουν ανάλογα τις ενέργειές τους. Η πλατφόρμα παρέχει αναλυτικά γραφήματα και ιστορικά δεδομένα που βοηθούν τους χρήστες να παρακολουθούν τις τάσεις και να εντοπίζουν παράγοντες που επηρεάζουν αρνητικά την ποιότητα του αέρα. Επίσης, το σύστημα μπορεί να δώσει προειδοποιήσεις και συμβουλές όταν οι τιμές των ρύπων υπερβαίνουν τα ασφαλή επίπεδα.

Ένα από τα βασικά πλεονεκτήματα του IQAir AirVisual Pro είναι η δυνατότητα να λειτουργεί τόσο ως αυτόνομο σύστημα όσο και σε συνδυασμό με άλλες έξυπνες συσκευές. Οι χρήστες μπορούν να το ενσωματώσουν σε έξυπνα οικιακά συστήματα, επιτρέποντας την αυτόματη ενεργοποίηση καθαριστών αέρα ή άλλων συσκευών όταν οι συνθήκες το απαιτούν. Αυτό καθιστά το AirVisual Pro ιδιαίτερα χρήσιμο για όσους ζουν σε περιοχές με υψηλή ατμοσφαιρική ρύπανση ή για εκείνους που έχουν ιδιαίτερες ανάγκες υγείας και χρειάζονται ακριβή παρακολούθηση της ποιότητας αέρα στους χώρους που διαμένουν ή εργάζονται.

2.3.3 Foobot

Το **Foobot** [15] είναι ένα σύστημα παρακολούθησης ποιότητας αέρα σε εσωτερικούς χώρους που επικεντρώνεται στην παροχή πληροφοριών σε πραγματικό χρόνο για την υγεία και την άνεση των κατοίκων. Το Foobot μετρά βασικές παραμέτρους όπως τα επίπεδα PM2.5, τις πτητικές οργανικές ενώσεις (VOCs), το διοξείδιο του άνθρακα (CO₂), τη θερμοκρασία και την υγρασία. Τα δεδομένα αυτά αποστέλλονται συνεχώς σε έναν cloud server, επιτρέποντας στους χρήστες να παρακολουθούν τις συνθήκες αέρα μέσω μιας εφαρμογής για κινητά ή μέσω διαδικτύου. Η φιλική προς τον χρήστη διεπαφή

παρέχει τόσο τρέχουσες μετρήσεις όσο και ιστορικά δεδομένα, επιτρέποντας στους χρήστες να κατανοήσουν τις τάσεις και να εντοπίσουν πηγές μόλυνσης.

Το σύστημα Foobot ξεχωρίζει για την ικανότητά του να δίνει προτάσεις για τη βελτίωση της ποιότητας του αέρα στον χώρο σας. Για παράδειγμα, αν ανιχνεύει υψηλά επίπεδα VOCs, μπορεί να προτείνει το άνοιγμα παραθύρων ή τη χρήση ενός καθαριστή αέρα. Επιπλέον, το Foobot μπορεί να συνδεθεί με άλλες έξυπνες οικιακές συσκευές, επιτρέποντας την αυτόματη ενεργοποίηση του εξαερισμού ή άλλων συστημάτων βελτίωσης αέρα όταν ανιχνεύονται επικίνδυνες συνθήκες. Αυτή η δυνατότητα καθιστά το Foobot ιδιαίτερα ευέλικτο και χρήσιμο σε περιβάλλοντα όπου η ποιότητα του αέρα είναι κρίσιμη για την υγεία των κατοίκων.

Το Foobot είναι ιδανικό για χρήση σε οικιακά περιβάλλοντα, γραφεία και άλλους χώρους όπου η ποιότητα του αέρα μπορεί να επηρεάσει σημαντικά την υγεία και την ευεξία των ανθρώπων. Χάρη στην ικανότητά του να παρακολουθεί και να αναλύει τα δεδομένα σε πραγματικό χρόνο, παρέχει στους χρήστες τα εργαλεία που χρειάζονται για να διατηρούν καθαρό και υγιεινό αέρα στον χώρο τους. Το σύστημα είναι επίσης εύκολο στη ρύθμιση και τη χρήση, καθιστώντας το μια προσιτή και αποτελεσματική λύση για όσους ενδιαφέρονται για την υγεία και την ποιότητα του περιβάλλοντος στο οποίο ζουν ή εργάζονται.

2.3.4 uHoo

Το **uHoo** [16] είναι ένα εξαιρετικά εξελιγμένο σύστημα παρακολούθησης ποιότητας αέρα που παρακολουθεί έως και εννέα διαφορετικούς παράγοντες που επηρεάζουν την ποιότητα του αέρα σε εσωτερικούς χώρους. Αυτές οι μετρήσεις περιλαμβάνουν τη θερμοκρασία, την υγρασία, το διοξείδιο του άνθρακα (CO₂), τα αιωρούμενα σωματίδια (PM_{2.5}), τα επίπεδα πτητικών οργανικών ενώσεων (VOCs), τη συγκέντρωση μονοξειδίου του άνθρακα (CO), την πίεση αέρα, και τα επίπεδα όζοντος (O₃). Τα δεδομένα από αυτούς τους αισθητήρες αποστέλλονται σε πραγματικό χρόνο σε έναν cloud server, όπου οι χρήστες μπορούν να τα παρακολουθούν μέσω μιας εφαρμογής για κινητά ή ενός web interface. Η πλατφόρμα προσφέρει αναλυτικά γραφήματα και ιστορικά δεδομένα, βοηθώντας τους χρήστες να κατανοήσουν τις τάσεις και να εντοπίσουν προβλήματα στην ποιότητα του αέρα.

Ένα από τα χαρακτηριστικά που ξεχωρίζουν το uHoo από άλλα συστήματα είναι η δυνατότητα παροχής προσωπικών προτάσεων με βάση τα δεδομένα που συλλέγει. Το uHoo χρησιμοποιεί τεχνητή νοημοσύνη για να αναλύει τα δεδομένα και να προτείνει συγκεκριμένες ενέργειες για τη βελτίωση της ποιότητας του αέρα, όπως η αύξηση του εξαερισμού, η χρήση καθαριστών αέρα ή η αποφυγή συγκεκριμένων δραστηριοτήτων που παράγουν ρύπους. Η εφαρμογή του uHoo στέλνει επίσης ειδοποιήσεις όταν οι μετρήσεις ξεπερνούν τα ασφαλή όρια, δίνοντας στους χρήστες τη δυνατότητα να ενεργούν άμεσα για να διατηρήσουν έναν υγιεινό εσωτερικό χώρο.

Το uHoo είναι επίσης συμβατό με άλλες έξυπνες οικιακές συσκευές, επιτρέποντας την αυτόματη διαχείριση του περιβάλλοντος του σπιτιού. Για παράδειγμα, μπορεί να συνδεθεί με συστήματα έξυπνου θερμοστάτη, καθαριστές αέρα και συστήματα εξαερισμού, ώστε να ενεργοποιούνται αυτόματα όταν οι συνθήκες αέρα απαιτούν προσαρμογές. Αυτή η δυνατότητα το καθιστά ιδιαίτερα χρήσιμο για οικογένειες με ευαισθησίες, όπως άτομα με άσθμα ή αλλεργίες, που χρειάζονται συνεχή έλεγχο της ποιότητας του αέρα. Το uHoo είναι ιδανικό για όσους θέλουν να διασφαλίσουν ότι ο αέρας που αναπνέουν είναι όσο το δυνατόν πιο καθαρός και ασφαλής, προσφέροντας προηγμένες δυνατότητες παρακολούθησης και προληπτικής διαχείρισης της ποιότητας του αέρα.

2.3.5 Netatmo

Το Netatmo Smart Indoor Air Quality Monitor [17] είναι ένα ολοκληρωμένο σύστημα σχεδιασμένο για την παρακολούθηση και βελτίωση της ποιότητας του αέρα σε εσωτερικούς χώρους. Το σύστημα παρακολουθεί τέσσερις βασικές παραμέτρους: την ποιότητα του αέρα μέσω της μέτρησης CO₂, την υγρασία, τη θερμοκρασία και τον θόρυβο. Αυτές οι πληροφορίες αποστέλλονται σε πραγματικό χρόνο σε έναν κεντρικό server και είναι διαθέσιμες στους χρήστες μέσω μιας εφαρμογής για κινητά ή ενός web interface. Η πλατφόρμα προσφέρει λεπτομερή δεδομένα και ιστορικά γραφήματα, βοηθώντας τους χρήστες να εντοπίσουν προβλήματα στην ποιότητα του αέρα και να προσαρμόσουν το περιβάλλον τους αναλόγως.

Ένα από τα κύρια πλεονεκτήματα του Netatmo Smart Indoor Air Quality Monitor είναι η δυνατότητα αποστολής προειδοποιήσεων στους χρήστες όταν οι μετρήσεις ξεπερνούν τα καθορισμένα όρια. Για παράδειγμα, αν τα επίπεδα CO₂ γίνουν υψηλά, το σύστημα θα ειδοποιήσει τους χρήστες να αερίσουν τον χώρο για να επανέλθει η ποιότητα του αέρα σε ασφαλή επίπεδα. Αυτή η δυνατότητα είναι ιδιαίτερα χρήσιμη για την άμεση διαχείριση των συνθηκών αέρα, αποτρέποντας πιθανά προβλήματα υγείας που συνδέονται με την κακή ποιότητα αέρα.

Το Netatmo Smart Indoor Air Quality Monitor είναι επίσης συμβατό με άλλες έξυπνες οικιακές συσκευές, όπως θερμοστάτες και συστήματα εξαερισμού, επιτρέποντας την αυτοματοποίηση της διαχείρισης του εσωτερικού περιβάλλοντος. Η εγκατάσταση και η χρήση του είναι απλή και φιλική προς τον χρήστη, καθιστώντας το μια εξαιρετική επιλογή για όσους θέλουν να παρακολουθούν και να βελτιώνουν την ποιότητα του αέρα στον χώρο τους με εύκολο και αποτελεσματικό τρόπο. Το σύστημα είναι επίσης ιδανικό για οικογένειες, γραφεία, και άλλους εσωτερικούς χώρους όπου η ποιότητα του αέρα είναι κρίσιμη για την ευεξία των ανθρώπων.

Κεφάλαιο 3ο: Τεχνολογικά εργαλεία και συσκευές

Στο κεφάλαιο αυτό θα περιγραφούν τα τεχνολογικά εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της εργασίας.

3.1 Τεχνολογικά εργαλεία

3.1.1 PHP

Η PHP (Hypertext Preprocessor) είναι μια από τις πιο διαδεδομένες γλώσσες προγραμματισμού που χρησιμοποιούνται για την ανάπτυξη δυναμικών ιστοσελίδων και εφαρμογών στο διαδίκτυο. Η γλώσσα δημιουργήθηκε το 1994 από τον Rasmus Lerdorf και από τότε έχει εξελιχθεί σημαντικά, παρέχοντας στους προγραμματιστές ένα ισχυρό εργαλείο για την ανάπτυξη web εφαρμογών. Η PHP είναι μια γλώσσα ανοιχτού κώδικα, πράγμα που σημαίνει ότι είναι ελεύθερα διαθέσιμη και υποστηρίζεται από μια μεγάλη κοινότητα προγραμματιστών. Αυτή η μεγάλη βάση χρηστών έχει οδηγήσει στην ανάπτυξη πολυάριθμων βιβλιοθηκών, εργαλείων και frameworks που διευκολύνουν την ανάπτυξη εφαρμογών και προσφέρουν στους προγραμματιστές ένα ευρύ φάσμα επιλογών. [18-20]



Εικόνα 3.1: PHP

[<https://www.valuecoders.com/blog/technology-and-apps/php-guide-vital-things-you-should-know-about-php/>]

Ένα από τα κύρια πλεονεκτήματα της PHP είναι η ευκολία χρήσης και η καμπύλη εκμάθησης. Ακόμα και νέοι προγραμματιστές μπορούν να δημιουργήσουν λειτουργικές web εφαρμογές με σχετική ευκολία, χρησιμοποιώντας τη γλώσσα. Η PHP είναι επίσης ιδιαίτερα συμβατή με πολλούς web servers και μπορεί να ενσωματωθεί εύκολα με άλλες τεχνολογίες, όπως HTML, CSS και JavaScript, κάνοντας την ιδανική για την ανάπτυξη ολοκληρωμένων web λύσεων. Επιπλέον, η PHP προσφέρει ισχυρή υποστήριξη για τη διαχείριση βάσεων δεδομένων, επιτρέποντας τη διασύνδεση με σχεδόν οποιοδήποτε

σύστημα διαχείρισης βάσεων δεδομένων (DBMS), συμπεριλαμβανομένων των MySQL, PostgreSQL, και SQLite.

Η PHP έχει διαδραματίσει καθοριστικό ρόλο στην εξέλιξη του διαδικτύου, αποτελώντας τη βάση για πολλές δημοφιλείς πλατφόρμες, όπως το WordPress, το Drupal και το Joomla. Αυτές οι πλατφόρμες χρησιμοποιούν την PHP για να διαχειρίζονται περιεχόμενο, να αλληλεπιδρούν με βάσεις δεδομένων και να προσφέρουν στους χρήστες πλούσιες και δυναμικές εμπειρίες. Καθώς το διαδίκτυο συνεχίζει να εξελίσσεται, η PHP παραμένει μια από τις κύριες επιλογές για την ανάπτυξη web εφαρμογών, λόγω της ευελιξίας της, της ευρείας υποστήριξης και της συνεχιζόμενης εξέλιξης και βελτίωσής της από την κοινότητα των προγραμματιστών.

Η PHP συνεχίζει να εξελίσσεται και να προσαρμόζεται στις ανάγκες της σύγχρονης ανάπτυξης λογισμικού, με την κυκλοφορία νέων εκδόσεων και τη συνεχή βελτίωση των χαρακτηριστικών της. Κάθε νέα έκδοση της PHP προσφέρει βελτιώσεις στην απόδοση, ασφάλεια και λειτουργικότητα, καθιστώντας την πιο αποτελεσματική και ισχυρή για τους προγραμματιστές. Για παράδειγμα, οι πρόσφατες εκδόσεις της PHP έχουν εισαγάγει σημαντικές βελτιώσεις στην ταχύτητα εκτέλεσης κώδικα, καθώς και νέες δυνατότητες όπως ο τύπος δεδομένων `typed properties`, τα βελτιωμένα `error handling` και η υποστήριξη `preloading`, που επιτρέπουν τη φόρτωση κώδικα στη μνήμη κατά την εκκίνηση του server, επιταχύνοντας τις μελλοντικές εκτελέσεις.

Ένα από τα σημαντικά πλεονεκτήματα της PHP είναι η δυνατότητά της να ενσωματώνεται εύκολα με ποικίλες τεχνολογίες και εργαλεία, επιτρέποντας τη δημιουργία πολύπλοκων και λειτουργικών εφαρμογών. Η PHP υποστηρίζει την επικοινωνία με μια ευρεία γκάμα βάσεων δεδομένων, προσφέροντας στους προγραμματιστές την ευελιξία να επιλέξουν το κατάλληλο σύστημα διαχείρισης βάσεων δεδομένων για τις ανάγκες τους. Επιπλέον, η PHP συνεργάζεται άψογα με άλλες γλώσσες προγραμματισμού και τεχνολογίες, όπως το JavaScript, το οποίο χρησιμοποιείται για την ανάπτυξη διεπαφών χρήστη, και το HTML/CSS για τη δομή και την παρουσίαση των ιστοσελίδων.

Παράλληλα, η PHP έχει αναπτύξει ένα ισχυρό οικοσύστημα εργαλείων και πλαισίων (frameworks) που επιταχύνουν και διευκολύνουν την ανάπτυξη εφαρμογών. Πλαίσια όπως το Laravel, το Symfony και το CodeIgniter παρέχουν προκατασκευασμένες λύσεις για κοινές ανάγκες, όπως η διαχείριση χρηστών, η επικοινωνία με βάσεις δεδομένων και η επεξεργασία αιτημάτων HTTP. Αυτά τα frameworks όχι μόνο εξοικονομούν χρόνο στους προγραμματιστές, αλλά και ενισχύουν την ασφάλεια και τη συντηρησιμότητα των εφαρμογών. Χάρη σε αυτά τα εργαλεία, η PHP παραμένει μια από τις πιο δημοφιλείς επιλογές για την ανάπτυξη web εφαρμογών, προσφέροντας ένα ισχυρό, ευέλικτο και υποστηριζόμενο περιβάλλον ανάπτυξης.

3.1.1.1 Πλεονεκτήματα της PHP

Η PHP είναι γνωστή για την απλότητα και την εύκολη καμπύλη εκμάθησής της. Ακόμα και αρχάριοι προγραμματιστές μπορούν να ξεκινήσουν να αναπτύσσουν δυναμικές ιστοσελίδες και εφαρμογές γρήγορα. Ο απλός συντακτικός κανόνας και η εκτεταμένη τεκμηρίωση καθιστούν την PHP ιδανική για όσους ξεκινούν στον τομέα του web development .

Η PHP διαθέτει μια τεράστια κοινότητα προγραμματιστών που συνεισφέρουν σε ανοιχτού κώδικα βιβλιοθήκες, frameworks και εργαλεία. Αυτό προσφέρει πλούσιο υλικό μάθησης, πόρους και βοήθεια για επίλυση προβλημάτων. Πλατφόρμες όπως το WordPress και το Drupal, που βασίζονται σε PHP, έχουν ενισχύσει τη δημοτικότητα της γλώσσας, καθιστώντας την ευρέως υποστηριζόμενη σε διάφορους web servers .

Η PHP είναι συμβατή με σχεδόν όλους τους web servers (Apache, Nginx, IIS) και υποστηρίζει πολυάριθμες βάσεις δεδομένων όπως MySQL, PostgreSQL, Oracle, και SQLite. Αυτή η ευελιξία την καθιστά μια πολύπλευρη επιλογή για web development, ειδικά για εφαρμογές που απαιτούν διαφορετικές υποδομές. [21]

3.1.1.2 Μειονεκτήματα της PHP

Η PHP έχει συχνά επικριθεί για ζητήματα ασφάλειας. Λόγω της δημοτικότητας της γλώσσας, έχει υπάρξει στόχος πολλών επιθέσεων. Ενώ η γλώσσα έχει εξελιχθεί και έχουν ενσωματωθεί καλύτερες πρακτικές ασφάλειας, πολλές παλιές εφαρμογές είναι ευάλωτες σε κοινές επιθέσεις όπως SQL injection και cross-site scripting (XSS) .

Μια συχνή κριτική προς την PHP είναι η έλλειψη συνέπειας στη σύνταξη και στη διαχείριση των λειτουργιών. Αυτή η ασυνέπεια μπορεί να προκαλέσει σύγχυση στους προγραμματιστές και να κάνει τη συντήρηση του κώδικα πιο δύσκολη, ειδικά σε μεγάλα έργα. Επιπλέον, η παλαιότερη φύση της PHP σημαίνει ότι υπάρχουν πολλές παλιές πρακτικές που δεν είναι πλέον βέλτιστες .

Ενώ η PHP είναι κατάλληλη για μικρές και μεσαίες εφαρμογές, μπορεί να αντιμετωπίσει προβλήματα απόδοσης σε πολύ μεγάλες και απαιτητικές εφαρμογές. Παρά τις βελτιώσεις στις πρόσφατες εκδόσεις, άλλες γλώσσες όπως η Node.js ή η Python με συγκεκριμένα frameworks μπορεί να προσφέρουν καλύτερη απόδοση και δυνατότητες κλιμάκωσης σε μεγάλης κλίμακας έργα.

3.1.1.3 Χρήση της PHP για λήψη δεδομένων από Requests

Παρατίθεται ένα παράδειγμα κώδικα PHP που διαχειρίζεται την αποδοχή δεδομένων από τις μεθόδους GET και POST: [22]

```
<?php
// Έλεγχος αν τα δεδομένα έχουν σταλεί μέσω GET
if ($_SERVER["REQUEST_METHOD"] == "GET") {
    if (isset($_GET['username'])) {
        $username = htmlspecialchars($_GET['username']);
        echo "Welcome, " . $username . "!";
    } else {
        echo "No username provided via GET.";
    }
}

// Έλεγχος αν τα δεδομένα έχουν σταλεί μέσω POST
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST['username'])) {
        $username = htmlspecialchars($_POST['username']);
        echo "Hello, " . $username . ", you've sent data via POST!";
    } else {
        echo "No username provided via POST.";
    }
}
?>
```

Διαχείριση GET Μεθόδου:

Ο κώδικας ξεκινά ελέγχοντας τη μέθοδο που χρησιμοποιήθηκε για την αποστολή των δεδομένων μέσω του `$_SERVER["REQUEST_METHOD"]`.

Αν η μέθοδος είναι GET, τότε ο κώδικας ελέγχει αν το πεδίο username είναι ορισμένο στο `$_GET` array.

Αν υπάρχει το πεδίο username, η τιμή του αποθηκεύεται στη μεταβλητή `$username` και εμφανίζεται ένα μήνυμα καλωσορίσματος.

Η συνάρτηση `htmlspecialchars()` χρησιμοποιείται για να αποτρέψει επιθέσεις XSS, μετατρέποντας ειδικούς χαρακτήρες HTML σε οντότητες.

Διαχείριση POST Μεθόδου:

Με τον ίδιο τρόπο, ο κώδικας ελέγχει αν η μέθοδος είναι POST.

Αν η μέθοδος είναι POST, ελέγχεται αν το πεδίο username είναι ορισμένο στο `$_POST` array.

Αν υπάρχει, η τιμή του αποθηκεύεται στη μεταβλητή `$username` και εμφανίζεται ένα μήνυμα που επιβεβαιώνει την αποστολή των δεδομένων μέσω POST.

Ασφάλεια και Χρήση:

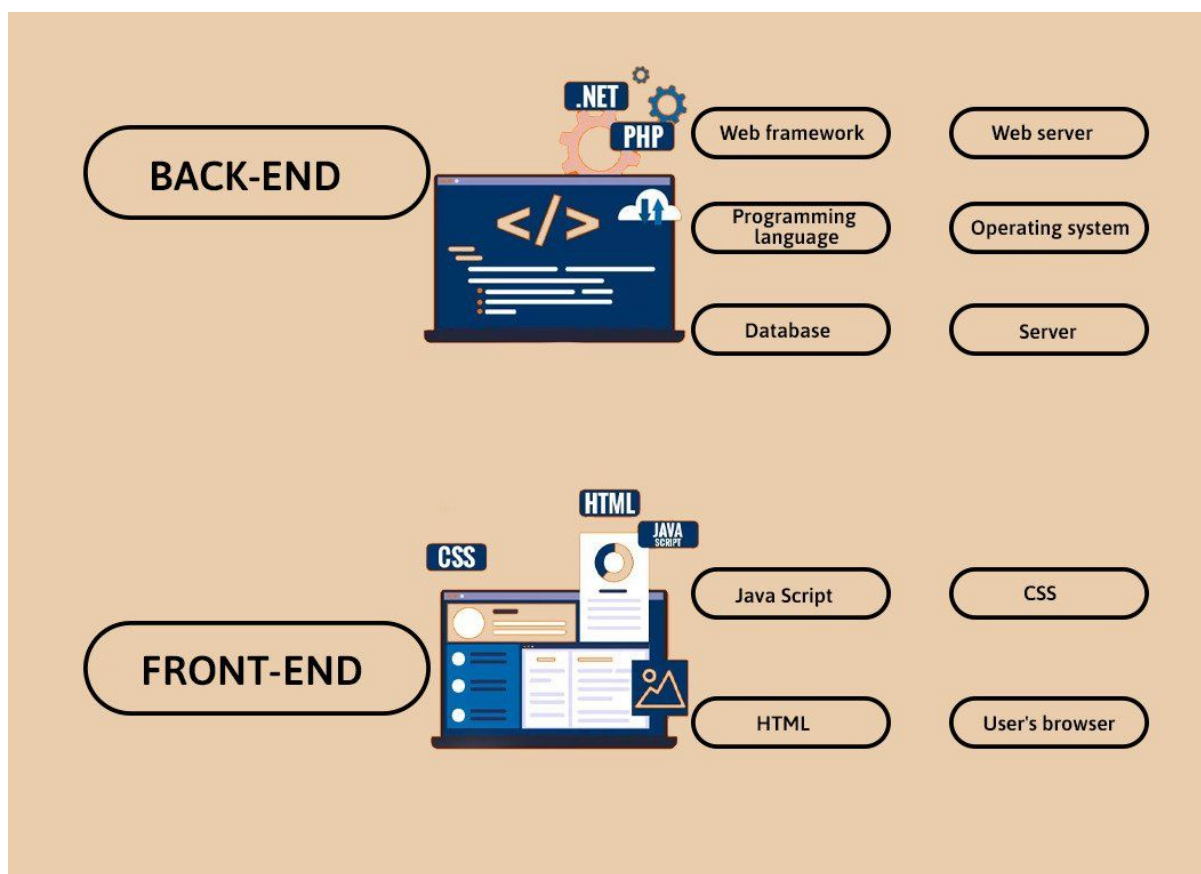
Ο κώδικας είναι γραμμένος με τέτοιο τρόπο ώστε να χειρίζεται δεδομένα από τις δύο κύριες μεθόδους αποστολής φόρμας στο web, δηλαδή GET και POST.

Είναι σημαντικό να σημειωθεί ότι οι GET παράμετροι εμφανίζονται στη γραμμή διευθύνσεων του προγράμματος περιήγησης, ενώ τα δεδομένα που στέλνονται μέσω POST δεν εμφανίζονται.

Η χρήση της `htmlspecialchars()` προστατεύει από βασικές επιθέσεις XSS, καθιστώντας το παράδειγμα πιο ασφαλές για πρακτική χρήση.

3.1.2 Laravel

Το Laravel είναι ένα από τα πιο δημοφιλή PHP frameworks που χρησιμοποιούνται για την ανάπτυξη web εφαρμογών. Δημιουργήθηκε από τον Taylor Otwell το 2011 και από τότε έχει εξελιχθεί σε ένα πλήρες οικοσύστημα εργαλείων και βιβλιοθηκών που επιτρέπουν στους προγραμματιστές να κατασκευάζουν ασφαλείς και ευέλικτες εφαρμογές με ευκολία. Το Laravel βασίζεται στο μοντέλο MVC (Model-View-Controller), το οποίο διαχωρίζει τη λογική της εφαρμογής από την παρουσίαση και τη διαχείριση των δεδομένων, κάνοντας τον κώδικα πιο οργανωμένο και συντηρήσιμο. Ένα από τα βασικά χαρακτηριστικά του Laravel είναι η φιλική προς τον χρήστη σύνταξη και η εκτεταμένη τεκμηρίωση, που διευκολύνουν την εκμάθηση και την ανάπτυξη εφαρμογών. [23]



Εικόνα 3.2: Laravel

[<https://www.tisdigitech.com/blog/laravel-framework-best-choice-for-php-web-development/>]

Ένα από τα ισχυρά πλεονεκτήματα του Laravel είναι η πλούσια βιβλιοθήκη έτοιμων εργαλείων και λειτουργιών που περιλαμβάνει, όπως το Eloquent ORM για διαχείριση βάσεων δεδομένων, το Blade για τη δημιουργία δυναμικών templates, και το Artisan για τη διαχείριση της γραμμής εντολών. Αυτά τα εργαλεία επιτρέπουν στους προγραμματιστές να επικεντρωθούν στη λογική της εφαρμογής, εξοικονομώντας χρόνο από τις επαναλαμβανόμενες διαδικασίες ανάπτυξης. Το Laravel διαθέτει επίσης ένα ισχυρό σύστημα διαχείρισης εξαρτήσεων μέσω του Composer, που διευκολύνει την εγκατάσταση και ενημέρωση πακέτων, κάνοντας την ανάπτυξη λογισμικού πιο αποτελεσματική.

Εκτός από τις βασικές λειτουργίες, το Laravel προσφέρει επίσης εξαιρετική υποστήριξη για σύνθετα χαρακτηριστικά όπως η ασφάλεια, η διαχείριση ταυτότητας και ο έλεγχος πρόσβασης, η διαχείριση API, και η δοκιμή εφαρμογών. Το Laravel είναι σχεδιασμένο για να διευκολύνει την ανάπτυξη και συντήρηση εφαρμογών σε μεγάλες κλίμακες, καθιστώντας το ιδανικό για μικρά και μεγάλα έργα. Η ενεργή κοινότητα και η συνεχής ανάπτυξη νέων χαρακτηριστικών κάνουν το Laravel μια από τις πιο δυναμικές επιλογές για PHP developers που θέλουν να χτίσουν μοντέρνες και αποδοτικές web εφαρμογές.

Το Laravel χρησιμοποιείται ευρέως σε διάφορους τομείς ανάπτυξης web εφαρμογών λόγω της ευελιξίας και της ισχυρής δομής του. [24-27]

1. Ανάπτυξη Εφαρμογών Διαχείρισης Περιεχομένου (CMS):

Το Laravel χρησιμοποιείται συχνά για την ανάπτυξη συστημάτων διαχείρισης περιεχομένου (CMS) που επιτρέπουν στους χρήστες να διαχειρίζονται και να δημοσιεύουν περιεχόμενο σε ιστοσελίδες με ευκολία. Με την υποστήριξη του Eloquent ORM και του Blade templating engine, οι προγραμματιστές μπορούν να δημιουργήσουν δυναμικά, ασφαλή και επεκτάσιμα CMS που προσαρμόζονται εύκολα στις ανάγκες των χρηστών. Παραδείγματα δημοφιλών CMS που βασίζονται σε Laravel περιλαμβάνουν το OctoberCMS και το PyroCMS.

2. Δημιουργία API:

Το Laravel είναι ιδανικό για τη δημιουργία RESTful APIs που επιτρέπουν στις εφαρμογές να επικοινωνούν μεταξύ τους ή με εξωτερικές υπηρεσίες. Η ενσωματωμένη υποστήριξη για API development στο Laravel περιλαμβάνει εργαλεία όπως το Laravel Passport και το Laravel Sanctum, τα οποία προσφέρουν ασφαλή διαχείριση ταυτότητας και ελέγχου πρόσβασης για APIs. Αυτή η δυνατότητα καθιστά το Laravel εξαιρετικά χρήσιμο για την ανάπτυξη μικροϋπηρεσιών (microservices) και άλλων εφαρμογών που απαιτούν ασφαλή και αξιόπιστη επικοινωνία μεταξύ συστημάτων.

3. Ανάπτυξη Εμπορικών Πλατφορμών:

Το Laravel είναι ιδανικό για την ανάπτυξη ηλεκτρονικών καταστημάτων και άλλων εμπορικών πλατφορμών λόγω των ισχυρών εργαλείων που προσφέρει για τη διαχείριση χρηστών, την επεξεργασία πληρωμών, και την ασφάλεια δεδομένων. Πλατφόρμες όπως το Laravel Cashier διευκολύνουν την ενσωμάτωση υπηρεσιών πληρωμών, ενώ το Laravel Echo υποστηρίζει τη διαχείριση πραγματικού χρόνου επικοινωνιών και ειδοποιήσεων, κάτι που είναι σημαντικό για τη βελτίωση της εμπειρίας χρήστη σε εμπορικές εφαρμογές.

4. Ανάπτυξη Εφαρμογών Διαχείρισης Σχέσεων Πελατών (CRM):

Το Laravel χρησιμοποιείται συχνά για την ανάπτυξη συστημάτων διαχείρισης σχέσεων πελατών (CRM). Αυτές οι εφαρμογές βοηθούν τις επιχειρήσεις να διαχειρίζονται τις αλληλεπιδράσεις με τους πελάτες τους, παρακολουθώντας πληροφορίες, επικοινωνίες και συναλλαγές. Με το Laravel, οι προγραμματιστές μπορούν να δημιουργήσουν προσαρμοσμένα CRM συστήματα που ενσωματώνονται εύκολα με άλλες πλατφόρμες και εργαλεία, όπως το email marketing και τα συστήματα διαχείρισης

έργων. Το Laravel προσφέρει την ευελιξία να δημιουργήσετε εξατομικευμένες λύσεις που καλύπτουν τις ειδικές ανάγκες των επιχειρήσεων.

5. Ανάπτυξη Κοινωνικών Δικτύων:

Η δημιουργία κοινωνικών δικτύων απαιτεί εργαλεία που μπορούν να διαχειριστούν μεγάλες ποσότητες δεδομένων χρηστών, τη ροή ειδήσεων, την ανταλλαγή μηνυμάτων σε πραγματικό χρόνο και την ασφάλεια των δεδομένων. Το Laravel υποστηρίζει αυτές τις ανάγκες με τα ισχυρά χαρακτηριστικά του, όπως η διαχείριση συνεδριών, η ταυτοποίηση χρηστών, και η κρυπτογράφηση δεδομένων. Χρησιμοποιώντας το Laravel, είναι εφικτό να αναπτυχθούν κοινωνικά δίκτυα που υποστηρίζουν εκατομμύρια χρήστες, διασφαλίζοντας ταυτόχρονα την ασφάλεια και την απόδοση της εφαρμογής.

6. Ανάπτυξη Συστημάτων Εκμάθησης και Εκπαίδευσης:

Το Laravel είναι επίσης ιδανικό για την ανάπτυξη πλατφορμών ηλεκτρονικής μάθησης (e-learning) που επιτρέπουν στους χρήστες να έχουν πρόσβαση σε μαθήματα, τεστ και εκπαιδευτικό υλικό. Η δυνατότητα του Laravel να χειρίζεται πολυπλοκότητα στη διαχείριση χρηστών, να προσφέρει εξατομικευμένες εμπειρίες και να ενσωματώνεται με εργαλεία πολυμέσων, το καθιστά ιδανικό για την ανάπτυξη πλατφορμών μάθησης. Επιπλέον, οι δυνατότητες κλιμάκωσης του Laravel επιτρέπουν σε αυτές τις πλατφόρμες να διαχειρίζονται μεγάλο αριθμό μαθητών και εκπαιδευτικών με μεγάλη αποδοτικότητα.

7. Ανάπτυξη Διαδικτυακών Φόρουμ:

Το Laravel είναι ιδανικό για την ανάπτυξη διαδικτυακών φόρουμ, όπου οι χρήστες μπορούν να αλληλεπιδρούν, να ανταλλάσσουν πληροφορίες και να συζητούν διάφορα θέματα. Με το Laravel, οι προγραμματιστές μπορούν να δημιουργήσουν συστήματα που διαχειρίζονται χρήστες, κατηγορίες, θέματα και απαντήσεις, προσφέροντας παράλληλα χαρακτηριστικά όπως ειδοποιήσεις, αναφορές, και διαχείριση χρηστών. Η δυνατότητα επέκτασης και προσαρμογής του Laravel το καθιστά κατάλληλο για την ανάπτυξη φόρουμ που μπορούν να υποστηρίξουν κοινότητες με εκατομμύρια χρήστες.

8. Διαχείριση Συμβάντων και Κρατήσεων:

Το Laravel χρησιμοποιείται ευρέως για την ανάπτυξη συστημάτων διαχείρισης συμβάντων και κρατήσεων. Αυτά τα συστήματα επιτρέπουν στους χρήστες να δημιουργούν, να προωθούν και να διαχειρίζονται εκδηλώσεις, καθώς και να διευκολύνουν τη διαδικασία κρατήσεων. Με ενσωματωμένα εργαλεία για τη διαχείριση χρηστών, πληρωμών και ειδοποιήσεων, το Laravel προσφέρει μια ολοκληρωμένη λύση για την ανάπτυξη εφαρμογών που απαιτούν σύνθετη διαχείριση πόρων και

κρατήσεων. Παραδείγματα περιλαμβάνουν πλατφόρμες κρατήσεων εισιτηρίων ή συστήματα διαχείρισης συνεδρίων.

9. Ανάπτυξη Blogs και Προσωπικών Ιστοσελίδων:

Το Laravel είναι μια εξαιρετική επιλογή για την ανάπτυξη blogs και προσωπικών ιστοσελίδων που απαιτούν δυναμική διαχείριση περιεχομένου. Με το Blade templating engine και το Eloquent ORM, οι προγραμματιστές μπορούν να δημιουργήσουν γρήγορα και εύκολα ιστοσελίδες που είναι εύκολες στη συντήρηση και ενημέρωση. Επιπλέον, οι δυνατότητες SEO (Search Engine Optimization) που προσφέρει το Laravel το καθιστούν ιδανικό για blogs που στοχεύουν σε υψηλή ορατότητα στις μηχανές αναζήτησης.

3.1.2.1 Παράδειγμα για τη δημιουργία ενός απλού API χρησιμοποιώντας το Laravel, που χειρίζεται αιτήματα GET και POST

Δημιουργία ενός νέου Laravel Project:

Αρχικά, δημιουργήστε ένα νέο Laravel project χρησιμοποιώντας το Composer:

```
composer create-project --prefer-dist laravel/laravel my-api
```

Ορισμός Routes:

Στο αρχείο routes/api.php, ορίστε τις διαδρομές για τα αιτήματα GET και POST. [28]

```
use Illuminate\Http\Request;

use Illuminate\Support\Facades\Route;

// Διαδρομή για GET αίτημα
Route::get('/user/{id}', function ($id) {
    return response()->json([
        'id' => $id,
        'name' => 'User ' . $id,
        'email' => 'user' . $id . '@example.com',
    ]);
});
```

```
});  
  
// Διαδρομή για POST αίτημα  
Route::post('/user', function (Request $request) {  
    $data = $request->all();  
    return response()->json([  
        'message' => 'User created successfully!',  
        'data' => $data,  
    ]);  
});
```

Διαδρομή για GET Αίτημα:

Η διαδρομή GET `/api/user/{id}` αποδέχεται ένα αναγνωριστικό χρήστη (ID) ως παράμετρο.

Επιστρέφει μια JSON απόκριση με το αναγνωριστικό του χρήστη, το όνομα και το email του. Το `response()->json()` χρησιμοποιείται για να διασφαλίσει ότι τα δεδομένα θα επιστραφούν σε μορφή JSON, η οποία είναι κοινή μορφή για API απαντήσεις.

Διαδρομή για POST Αίτημα:

Η διαδρομή POST `/api/user` αποδέχεται δεδομένα από ένα POST αίτημα, που μπορεί να περιλαμβάνει λεπτομέρειες χρήστη.

Χρησιμοποιώντας το Request αντικείμενο, ο κώδικας λαμβάνει όλα τα δεδομένα που στέλνονται με το POST αίτημα μέσω της μεθόδου `all()`.

Επιστρέφεται μια JSON απόκριση με μήνυμα επιτυχίας και τα δεδομένα που στάλθηκαν, για επιβεβαίωση της δημιουργίας του χρήστη.

Δοκιμή του API:

Μπορείτε να χρησιμοποιήσετε εργαλεία όπως το Postman ή το cURL για να δοκιμάσετε το API σας.

GET Αίτημα:

```
curl http://localhost:8000/api/user/1
```

POST Αίτημα:

```
curl -X POST -d "name=John Doe&email=johndoe@example.com" http://localhost:8000/api/user
```

3.1.2.2 Μειονεκτήματα του Laravel

Απόδοση σε Μεγάλες Κλίμακες:

Αν και το Laravel είναι πολύ αποτελεσματικό για μικρά και μεσαία έργα, μπορεί να αντιμετωπίσει προβλήματα απόδοσης σε πολύ μεγάλες εφαρμογές με μεγάλο όγκο χρηστών και δεδομένων. Αυτό οφείλεται κυρίως στην επιπλέον "φορτίο" που φέρνει η χρήση ενός πλήρους framework με πολλές ενσωματωμένες λειτουργίες, οι οποίες μπορεί να μην είναι όλες απαραίτητες σε μεγάλες κλίμακες.

Καμπύλη Εκμάθησης:

Αν και το Laravel προσφέρει πολλές αυτοματοποιήσεις και εργαλεία, η εκμάθηση του πλαισίου μπορεί να είναι δύσκολη για νέους προγραμματιστές ή για όσους δεν είναι εξοικειωμένοι με τη φιλοσοφία των frameworks. Η καμπύλη εκμάθησης μπορεί να είναι απότομη, ιδιαίτερα όταν προσπαθείτε να κατανοήσετε πλήρως όλα τα χαρακτηριστικά και τις βέλτιστες πρακτικές που προωθεί το Laravel.

Βαθιά Σύνδεση με το Eloquent ORM:

Το Eloquent ORM του Laravel είναι ισχυρό και ευέλικτο, αλλά η βαθιά του ενσωμάτωση στο πλαίσιο μπορεί να δημιουργήσει προβλήματα όταν εργάζεστε με πολύπλοκες βάσεις δεδομένων ή χρειάζεστε εξαιρετικά προσαρμοσμένες λύσεις SQL. Αν και το Laravel υποστηρίζει τη χρήση άλλων εργαλείων διαχείρισης βάσεων δεδομένων, η απομάκρυνση από το Eloquent μπορεί να οδηγήσει σε πρόσθετη πολυπλοκότητα και αύξηση του χρόνου ανάπτυξης.

Συχνές Ενημερώσεις και Ασυμβατότητες:

Το Laravel εξελίσσεται γρήγορα με συχνές ενημερώσεις και νέες εκδόσεις. Αν και αυτό είναι γενικά καλό για την ενσωμάτωση νέων χαρακτηριστικών και βελτιώσεων ασφαλείας, μπορεί να δημιουργήσει

προβλήματα για τις υπάρχουσες εφαρμογές, καθώς οι ενημερώσεις μπορεί να προκαλέσουν ασυμβατότητες ή να απαιτήσουν σημαντικές αλλαγές στον κώδικα.

Βαρύ Πλαίσιο:

Το Laravel, ως ένα ολοκληρωμένο πλαίσιο, έρχεται με πολλά προεγκατεστημένα χαρακτηριστικά και βιβλιοθήκες. Αν και αυτό μπορεί να είναι χρήσιμο για την ταχεία ανάπτυξη, μπορεί επίσης να οδηγήσει σε εφαρμογές που είναι πιο "βαριές" από ό,τι είναι απαραίτητο, με συνέπεια την αύξηση των απαιτήσεων σε πόρους, όπως μνήμη και CPU. [29-30]

3.1.3 MySQL

Η MySQL αποτελεί μία από τις πιο δημοφιλείς και ισχυρές βάσεις δεδομένων ανοιχτού κώδικα που χρησιμοποιούνται παγκοσμίως. Η δημοτικότητά της οφείλεται κυρίως στην αξιοπιστία της, την ευελιξία της και την ευρεία υποστήριξή της από την κοινότητα των προγραμματιστών. Η MySQL είναι μία σχεσιακή βάση δεδομένων (RDBMS) που επιτρέπει στους χρήστες να διαχειρίζονται και να οργανώνουν δεδομένα με δομημένο και αποδοτικό τρόπο. Η χρήση της είναι εκτεταμένη σε μικρές και μεγάλες επιχειρήσεις, καθώς και σε διαδικτυακές πλατφόρμες που απαιτούν διαχείριση μεγάλου όγκου δεδομένων με υψηλή απόδοση και ασφάλεια. [31]

Ένα από τα κύρια χαρακτηριστικά της MySQL είναι η υποστήριξή της για διάφορους τύπους δεδομένων και οι ισχυρές δυνατότητες ερωτημάτων που προσφέρει. Η MySQL επιτρέπει την εκτέλεση πολύπλοκων ερωτημάτων SQL που μπορούν να συνδυάσουν πολλαπλούς πίνακες και να εκτελούν λειτουργίες όπως φίλτρα, ομαδοποιήσεις και ταξινομήσεις με υψηλή ταχύτητα. Επιπλέον, υποστηρίζει συναλλαγές και απομόνωση δεδομένων, κάτι που είναι κρίσιμο για εφαρμογές που απαιτούν αξιοπιστία και ακεραιότητα δεδομένων. Η ευκολία στη χρήση της και η δυνατότητα σύνδεσης με διάφορες γλώσσες προγραμματισμού, όπως PHP, Java, και Python, την καθιστούν ιδανική για ανάπτυξη διαδικτυακών εφαρμογών.

Ως προς τα πλεονεκτήματά της, η MySQL ξεχωρίζει για την ταχύτητά της στην εκτέλεση ερωτημάτων και την αποτελεσματική διαχείριση πόρων, κάτι που την καθιστά κατάλληλη για εφαρμογές υψηλών επιδόσεων. Είναι ευέλικτη και μπορεί να κλιμακωθεί ώστε να εξυπηρετεί εφαρμογές από μικρές ιστοσελίδες μέχρι μεγάλες διαδικτυακές πλατφόρμες με εκατομμύρια χρήστες. Το γεγονός ότι είναι ανοιχτού κώδικα συνεπάγεται ότι υπάρχει ευρεία κοινότητα προγραμματιστών που συνεισφέρουν στη βελτίωσή της και παρέχουν υποστήριξη. Η MySQL προσφέρει επίσης ισχυρά εργαλεία διαχείρισης και βελτιστοποίησης βάσεων δεδομένων, επιτρέποντας στους διαχειριστές να διατηρούν την απόδοση της βάσης δεδομένων σε υψηλά επίπεδα.

Παρόλα αυτά, υπάρχουν και μειονεκτήματα στη χρήση της MySQL. Ένα από τα βασικά μειονεκτήματα είναι η περιορισμένη υποστήριξη για ορισμένες προηγμένες λειτουργίες SQL, όπως αποθηκευμένες διαδικασίες και triggers, σε σχέση με άλλες RDBMS όπως η PostgreSQL. Επιπλέον, παρόλο που η MySQL είναι εξαιρετικά γρήγορη για αναγνώσεις, οι επιδόσεις της μπορεί να μειωθούν κατά τη διαχείριση σύνθετων ή πολύπλοκων συναλλαγών που απαιτούν υψηλή συγγραφή δεδομένων. Επίσης, ενώ η MySQL είναι εξαιρετικά αποδοτική σε απλές εφαρμογές, μπορεί να απαιτεί πρόσθετη προσαρμογή και βελτιστοποίηση σε μεγάλης κλίμακας εφαρμογές που απαιτούν πολύ υψηλές επιδόσεις και αξιοπιστία.

Η σύνδεση της MySQL με το Laravel, ένα από τα πιο δημοφιλή PHP frameworks, είναι φυσική και πολύ αποτελεσματική. Το Laravel παρέχει ενσωματωμένη υποστήριξη για MySQL μέσω του Eloquent ORM (Object-Relational Mapping), καθιστώντας την αλληλεπίδραση με τη βάση δεδομένων απλή και αποδοτική. Το Laravel διευκολύνει την κατασκευή σύνθετων ερωτημάτων και την εκτέλεση μεταναστεύσεων της βάσης δεδομένων, επιτρέποντας στους προγραμματιστές να διαχειρίζονται τη MySQL με εύκολο και δομημένο τρόπο. Η MySQL, σε συνδυασμό με το Laravel, επιτρέπει την ανάπτυξη εφαρμογών που είναι τόσο ισχυρές όσο και επεκτάσιμες, καλύπτοντας τις ανάγκες μιας ευρείας γκάμας έργων, από μικρές ιστοσελίδες μέχρι μεγάλες επιχειρηματικές εφαρμογές.

3.1.4 Python

Η Python είναι μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού στον κόσμο, γνωστή για την απλότητα και την ευελιξία της. Αναπτύχθηκε στα τέλη της δεκαετίας του 1980 από τον Guido van Rossum και έκτοτε έχει γίνει απαραίτητο εργαλείο για προγραμματιστές σε διάφορους τομείς, από την ανάπτυξη εφαρμογών μέχρι την επιστημονική έρευνα και την ανάλυση δεδομένων. Η Python είναι γλώσσα υψηλού επιπέδου, πράγμα που σημαίνει ότι ο κώδικας είναι εύκολα κατανοητός και διαβάζεται σχεδόν σαν φυσική γλώσσα, καθιστώντας την ιδανική για αρχάριους, αλλά και για έμπειρους προγραμματιστές που επιθυμούν να αναπτύξουν σύνθετες εφαρμογές με λιγότερο κώδικα. [32-23]

Ένα από τα κύρια χαρακτηριστικά της Python είναι η εκτεταμένη βιβλιοθήκη της, η οποία περιλαμβάνει εργαλεία και πακέτα για σχεδόν κάθε πιθανή εφαρμογή. Από την επιστημονική ανάλυση δεδομένων με βιβλιοθήκες όπως οι NumPy και Pandas, μέχρι την ανάπτυξη ιστοσελίδων με το Django και το Flask, η Python προσφέρει εργαλεία που καλύπτουν ευρεία γκάμα αναγκών. Επιπλέον, η Python είναι αντικειμενοστραφής γλώσσα, υποστηρίζοντας κληρονομικότητα, πολυμορφισμό και άλλες έννοιες που καθιστούν τη διαχείριση του κώδικα πιο δομημένη και επεκτάσιμη. Η Python είναι επίσης δυναμική γλώσσα, επιτρέποντας στους προγραμματιστές να γράφουν και να εκτελούν κώδικα χωρίς την ανάγκη για προηγούμενη δήλωση τύπων δεδομένων, γεγονός που επιταχύνει τη διαδικασία ανάπτυξης.

Η γλώσσα είναι διάσημη για την απλότητά της και την ευκολία μάθησής της. Ο κώδικας σε Python είναι συνήθως πιο σύντομος και πιο ευανάγνωστος σε σύγκριση με άλλες γλώσσες προγραμματισμού, γεγονός που επιτρέπει στους προγραμματιστές να εστιάζουν στην επίλυση προβλημάτων αντί στην επίλυση σφαλμάτων σύνταξης. Επιπλέον, η Python είναι πλατφόρμα ανεξάρτητη, πράγμα που σημαίνει ότι ο κώδικας που γράφεται σε Python μπορεί να εκτελεστεί σε διαφορετικά λειτουργικά συστήματα χωρίς αλλαγές. Η μεγάλη και ενεργή κοινότητα της Python εξασφαλίζει ότι υπάρχουν πληθώρα πόρων, εγχειριδίων και βιβλιοθηκών διαθέσιμα, υποστηρίζοντας έτσι την ανάπτυξη καινοτόμων λύσεων.

Ένα από τα κύρια μειονεκτήματα της Python είναι η ταχύτητά της σε σχέση με γλώσσες χαμηλότερου επιπέδου όπως η C ή η Java. Επειδή η Python είναι ερμηνευμένη γλώσσα, οι εφαρμογές της μπορεί να εκτελούνται πιο αργά, ειδικά σε περιπτώσεις όπου απαιτείται μεγάλη επεξεργαστική ισχύς. Επίσης, η διαχείριση της μνήμης στην Python δεν είναι τόσο αποτελεσματική όσο σε άλλες γλώσσες, κάτι που μπορεί να αποτελέσει πρόβλημα σε εφαρμογές με μεγάλες απαιτήσεις μνήμης. Παρόλο που η Python είναι εξαιρετικά ευέλικτη, μπορεί να μην είναι η καλύτερη επιλογή για πολύπλοκες εφαρμογές σε πραγματικό χρόνο ή για εφαρμογές που απαιτούν άμεση πρόσβαση στο υλικό.

Γενικά η Python είναι μία γλώσσα προγραμματισμού που συνδυάζει απλότητα, ευελιξία και ισχύ, καθιστώντας την ιδανική για ένα ευρύ φάσμα εφαρμογών. Παρόλο που μπορεί να μην είναι η καλύτερη επιλογή για εφαρμογές υψηλών επιδόσεων, η ευκολία στη χρήση της και η υποστήριξή της από μία τεράστια κοινότητα την καθιστούν ακαταμάχητη επιλογή για πολλές εφαρμογές, από την ανάπτυξη ιστοσελίδων μέχρι την επιστημονική ανάλυση δεδομένων και τη μηχανική μάθηση.

3.1.5 Javascript

Η JavaScript είναι μία από τις πιο διαδεδομένες γλώσσες προγραμματισμού παγκοσμίως και θεωρείται η «γλώσσα του διαδικτύου» λόγω της εκτεταμένης χρήσης της στην ανάπτυξη ιστοσελίδων. Αναπτύχθηκε αρχικά από τον Brendan Eich το 1995 και από τότε έχει εξελιχθεί σε μια από τις πιο ισχυρές και ευέλικτες γλώσσες, ικανή να εκτελείται τόσο στην πλευρά του πελάτη (client-side) όσο και στην πλευρά του διακομιστή (server-side). Η JavaScript είναι γλώσσα υψηλού επιπέδου, δυναμικά δακτυλογραφημένη και υποστηρίζει αντικειμενοστραφή προγραμματισμό, γεγονός που την καθιστά ιδιαίτερα προσιτή για τους νέους προγραμματιστές, ενώ παράλληλα παρέχει τη δύναμη που απαιτείται για την ανάπτυξη σύνθετων εφαρμογών. [34]

Ένα από τα κύρια χαρακτηριστικά της JavaScript είναι η ευελιξία της και η εκτεταμένη υποστήριξη για τη διαχείριση των στοιχείων του DOM (Document Object Model), επιτρέποντας στους προγραμματιστές να δημιουργούν δυναμικά και διαδραστικά περιβάλλοντα χρήστη. Η JavaScript μπορεί να εκτελεστεί απευθείας στον browser του χρήστη, κάτι που επιτρέπει την άμεση αντίδραση

στις ενέργειες του χρήστη, χωρίς την ανάγκη για επαναφόρτωση της σελίδας. Επιπλέον, με την εισαγωγή του Node.js, η JavaScript επεκτάθηκε στην πλευρά του διακομιστή, δίνοντας τη δυνατότητα στους προγραμματιστές να γράφουν πλήρως ενοποιημένες εφαρμογές χρησιμοποιώντας μία μόνο γλώσσα. Η γλώσσα διαθέτει επίσης πλούσια οικοσυστήματα βιβλιοθηκών και πλαισίων εργασίας, όπως το React, το Angular, και το Vue.js, που διευκολύνουν την ανάπτυξη σύγχρονων εφαρμογών μιας σελίδας (single-page applications - SPAs). Η γλώσσα είναι εξαιρετικά προσαρμοστική και υποστηρίζει πολλαπλά παραδείγματα προγραμματισμού, όπως ο αντικειμενοστραφής, ο λειτουργικός και ο διαδικαστικός προγραμματισμός. Η JavaScript είναι επίσης ασύγχρονη από τη φύση της, υποστηρίζοντας λειτουργίες όπως οι callbacks, οι promises και οι async/await, που διευκολύνουν την εκτέλεση ασύγχρονων εργασιών όπως η φόρτωση δεδομένων από έναν διακομιστή. Αυτή η ικανότητα καθιστά τη JavaScript ιδανική για την ανάπτυξη εφαρμογών που απαιτούν ταχύτατες αντιδράσεις σε πραγματικό χρόνο, όπως οι διαδραστικοί χάρτες, τα chat applications και τα ηλεκτρονικά παιχνίδια.

Επειδή εκτελείται στον browser, η ασφάλεια είναι μια σημαντική πρόκληση, καθώς ο κώδικας μπορεί να εκτεθεί σε κακόβουλες επιθέσεις ή να παραποιηθεί από τον χρήστη. Επιπλέον, η πολυπλοκότητα της γλώσσας μπορεί να αυξηθεί σημαντικά σε μεγάλες εφαρμογές, όπου η διαχείριση του κώδικα γίνεται δυσκολότερη. Η ασύγχρονη φύση της JavaScript, αν και ισχυρή, μπορεί να δυσκολέψει τον εντοπισμό σφαλμάτων και την αποσφαλμάτωση, ειδικά για προγραμματιστές που δεν είναι εξοικειωμένοι με τις έννοιες της ασύγχρονης εκτέλεσης.

Η χρήση της JavaScript σε συνδυασμό με το Laravel, ένα δημοφιλές PHP framework, είναι μια εξαιρετικά αποτελεσματική πρακτική στην ανάπτυξη σύγχρονων διαδικτυακών εφαρμογών. Το Laravel, αν και ισχυρό από μόνο του στην πλευρά του διακομιστή, επωφελείται από τη χρήση της JavaScript για την ανάπτυξη πλούσιων διεπαφών χρήστη. Η ενσωμάτωση της JavaScript με το Laravel μπορεί να γίνει με διάφορους τρόπους. Για παράδειγμα, το Laravel παρέχει ενσωματωμένη υποστήριξη για Vue.js, καθιστώντας εύκολη τη δημιουργία δυναμικών components απευθείας στις προβολές του Laravel. Επίσης, η χρήση της Axios, μιας δημοφιλούς βιβλιοθήκης JavaScript για την εκτέλεση AJAX αιτημάτων, μπορεί να διευκολύνει τη διαχείριση δεδομένων σε πραγματικό χρόνο, επικοινωνώντας με τα API που δημιουργούνται στο Laravel. [35]

Με την αυξανόμενη δημοτικότητα των SPA (single-page applications), η συνδυαστική χρήση της JavaScript με το Laravel επιτρέπει την κατασκευή εφαρμογών που είναι ταυτόχρονα αποδοτικές και εύκολες στη χρήση. Το Laravel λειτουργεί ως ισχυρό backend, παρέχοντας τα API και την επιχειρηματική λογική, ενώ η JavaScript διαχειρίζεται το frontend, δημιουργώντας μια απρόσκοπτη και διαδραστική εμπειρία για τον χρήστη. Αυτός ο συνδυασμός επιτρέπει την ανάπτυξη εφαρμογών που είναι ταυτόχρονα ισχυρές, επεκτάσιμες και διατηρήσιμες, καλύπτοντας ένα ευρύ φάσμα αναγκών, από μικρές επιχειρηματικές λύσεις μέχρι σύνθετες, πολυεπίπεδες πλατφόρμες.

3.1.6 Apache

Ο Apache HTTP Server, γνωστός απλώς ως Apache, είναι ένας από τους πιο δημοφιλείς και ευρέως χρησιμοποιούμενους web servers στον κόσμο. Αναπτύχθηκε από το Apache Software Foundation και είναι ανοιχτού κώδικα, κάτι που σημαίνει ότι μπορεί να χρησιμοποιηθεί, να τροποποιηθεί και να διανεμηθεί ελεύθερα. Ο Apache αποτελεί τη βάση για εκατομμύρια ιστοσελίδες και διαδικτυακές εφαρμογές, προσφέροντας ευελιξία, αξιοπιστία και εξαιρετική υποστήριξη από την κοινότητα των προγραμματιστών. [36]

Ο Apache υποστηρίζει μια ποικιλία λειτουργικών συστημάτων, όπως Linux, Windows, και macOS, και μπορεί να διαχειριστεί έναν μεγάλο αριθμό ταυτόχρονων συνδέσεων, καθιστώντας τον ιδανικό για μικρές έως μεγάλες επιχειρήσεις. Ένα από τα κύρια χαρακτηριστικά του Apache είναι η αρθρωτή αρχιτεκτονική του, που επιτρέπει στους διαχειριστές να ενεργοποιούν ή να απενεργοποιούν διάφορες λειτουργίες μέσω των modules. Αυτή η ευελιξία επιτρέπει στον Apache να προσαρμόζεται σε διάφορα περιβάλλοντα και ανάγκες, όπως η διαχείριση SSL/TLS για ασφαλείς συνδέσεις, η διαχείριση αρχείων μέσω του mod_rewrite, ή η σύνδεση με βάσεις δεδομένων.

Η δύναμη του Apache έγκειται επίσης στη δυνατότητα του να εξυπηρετεί δυναμικό περιεχόμενο, υποστηρίζοντας γλώσσες προγραμματισμού όπως PHP, Python, και Perl. Επιπλέον, η ενσωμάτωση με εργαλεία όπως το mod_proxy και το mod_cache επιτρέπει την κλιμάκωση της απόδοσης και την αποτελεσματική διαχείριση των πόρων του server. Ο Apache προσφέρει επίσης εκτεταμένες δυνατότητες παραμετροποίησης μέσω αρχείων ρυθμίσεων, όπως το httpd.conf, επιτρέποντας στους διαχειριστές να προσαρμόσουν τη συμπεριφορά του server στις ανάγκες τους.

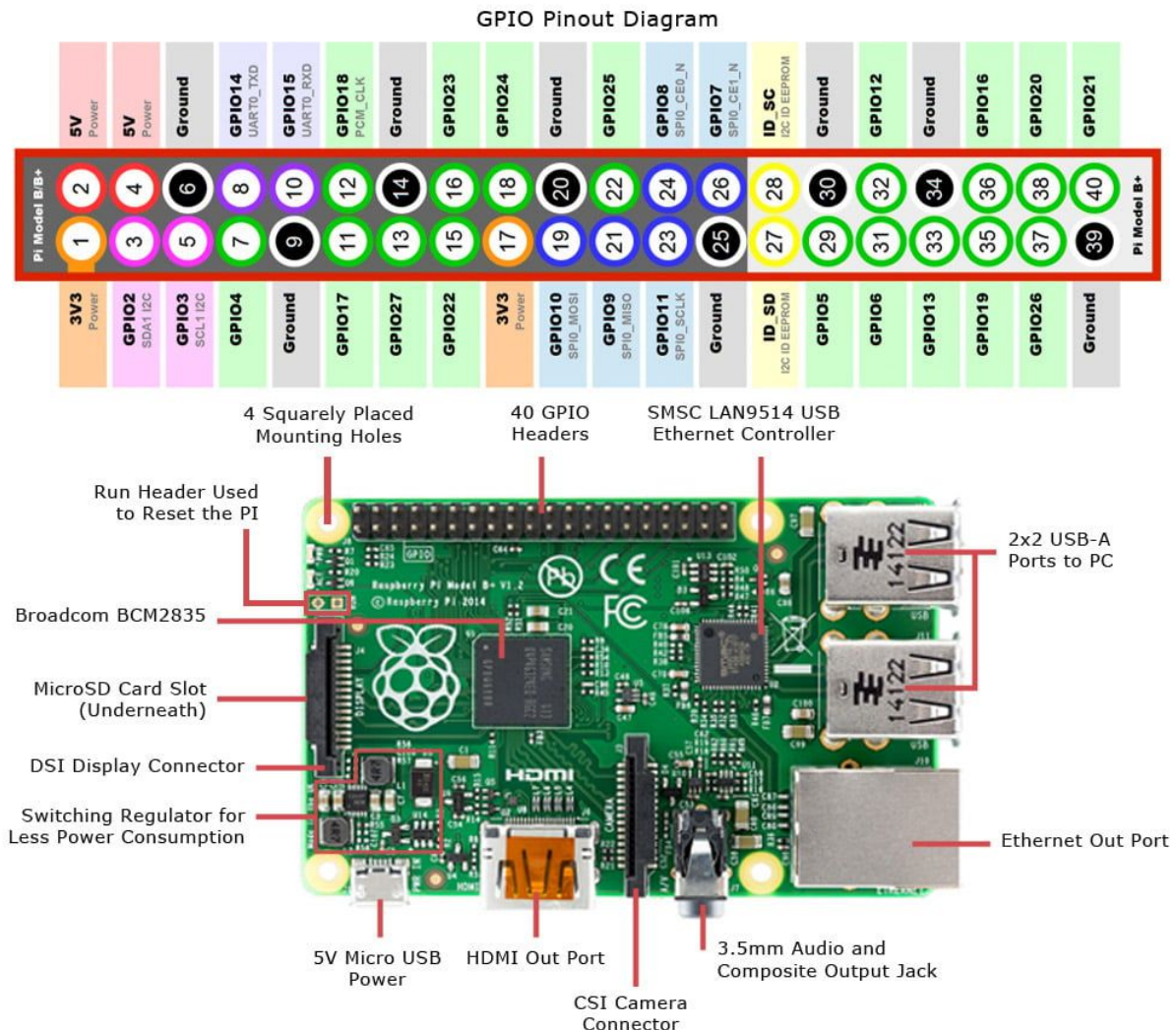
Διακρίνεται για τη σταθερότητα και αξιοπιστία του, καθιστώντας τον έναν από τους πιο αξιόπιστους web servers για εφαρμογές που απαιτούν υψηλή διαθεσιμότητα. Ωστόσο, όπως κάθε τεχνολογία, ο Apache έχει και κάποιες προκλήσεις. Για παράδειγμα, μπορεί να απαιτήσει περισσότερη διαχείριση σε σύγκριση με νεότερους web servers όπως το Nginx, ειδικά όταν πρόκειται για βελτιστοποίηση της απόδοσης σε περιβάλλοντα με υψηλό φορτίο. Παρόλα αυτά, η ευρεία υιοθέτηση και η εκτεταμένη τεκμηρίωση καθιστούν τον Apache μια σίγουρη επιλογή για όσους αναζητούν έναν δοκιμασμένο και ευέλικτο web server.

3.2 Συσκευές – Αισθητήρες

3.2.1 Raspberry Pi

Το Raspberry Pi είναι ένας μικρός, χαμηλού κόστους υπολογιστής που έχει κατακτήσει τον κόσμο της τεχνολογίας με τις ευέλικτες δυνατότητές του. Αναπτύχθηκε από το Raspberry Pi Foundation με σκοπό

την προώθηση της διδασκαλίας της πληροφορικής και των βασικών εννοιών προγραμματισμού σε σχολεία και αναπτυσσόμενες χώρες. Παρά το μικρό του μέγεθος και την προσιτή τιμή του, το Raspberry Pi έχει τη δυνατότητα να λειτουργήσει ως πλήρης υπολογιστής, με πλήρη υποστήριξη για λειτουργικά συστήματα, δικτύωση, και συνδεσιμότητα με διάφορα περιφερειακά. [37]



Εικόνα 3.3: Raspberry Pi 3

[<https://www.jameco.com/Jameco/workshop/CircuitNotes/Raspberry-Pi-3-pinout.jpg>]

Το Raspberry Pi έρχεται σε διάφορα μοντέλα, το καθένα με διαφορετικές προδιαγραφές, όπως τα μοντέλα Raspberry Pi 3, 4, και Zero. Αυτές οι μικρές πλακέτες διαθέτουν επεξεργαστή ARM, μνήμη RAM, θύρες USB, HDMI, και GPIO (General Purpose Input/Output) για σύνδεση με εξωτερικές συσκευές και αισθητήρες. Ένα από τα σημαντικότερα χαρακτηριστικά του Raspberry Pi είναι η δυνατότητά του να εκτελεί διάφορα λειτουργικά συστήματα, όπως το Raspberry Pi OS (πρώην Raspbian), καθώς και άλλες διανομές Linux, ακόμα και Windows IoT Core. Η ευελιξία αυτή επιτρέπει τη χρήση του Raspberry Pi σε μια πληθώρα εφαρμογών, από απλές εκπαιδευτικές δραστηριότητες μέχρι προηγμένα έργα IoT (Internet of Things) και αυτοματισμού.

Τα πλεονεκτήματα του Raspberry Pi είναι πολλά και ποικίλα. Πρώτον, η προσβασιμότητα και το χαμηλό κόστος το καθιστούν ιδανικό εργαλείο για εκπαίδευση και πειραματισμό. Οι μαθητές και οι χομπίστες μπορούν να ξεκινήσουν εύκολα να μαθαίνουν προγραμματισμό, ρομποτική, και ηλεκτρονικά, χωρίς να χρειάζεται να επενδύσουν σε ακριβά εξαρτήματα. Επιπλέον, το Raspberry Pi είναι εξαιρετικά ενεργειακά αποδοτικό, κάτι που το καθιστά κατάλληλο για έργα που απαιτούν συνεχή λειτουργία με χαμηλή κατανάλωση ενέργειας. Η ενεργή κοινότητα γύρω από το Raspberry Pi προσφέρει τεράστια υποστήριξη, με αμέτρητα έργα, οδηγούς και φόρουμ που βοηθούν τους χρήστες να αξιοποιήσουν πλήρως τις δυνατότητες της πλατφόρμας.

Το Raspberry Pi, λόγω του μικρού του μεγέθους και της χαμηλής ισχύος, δεν είναι τόσο ισχυρό όσο ένας κανονικός υπολογιστής ή ένας server, κάτι που σημαίνει ότι μπορεί να αντιμετωπίσει δυσκολίες σε εφαρμογές που απαιτούν υψηλή επεξεργαστική ισχύ ή ταχύτητες δικτύου. Επιπλέον, ο αποθηκευτικός χώρος είναι περιορισμένος και βασίζεται κυρίως σε κάρτες SD, που μπορεί να είναι πιο αργές και λιγότερο αξιόπιστες από τους σκληρούς δίσκους. Παρόλα αυτά, για τις περισσότερες DIY εφαρμογές και έργα IoT, το Raspberry Pi προσφέρει περισσότερο από αρκετή ισχύ και λειτουργικότητα.

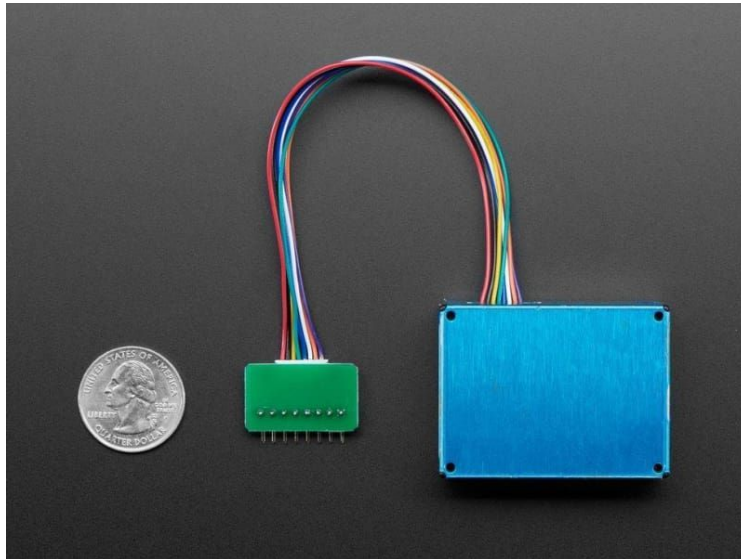
Το Raspberry Pi έχει αποδείξει ότι μπορεί να ξεπεράσει κατά πολύ τον αρχικό του σκοπό ως ένα εκπαιδευτικό εργαλείο και έχει καθιερωθεί ως ένα απαραίτητο εργαλείο για προγραμματιστές, χομπίστες και επαγγελματίες σε όλο τον κόσμο. Είτε χρησιμοποιείται για προσωπικά έργα είτε για επαγγελματικές λύσεις, το Raspberry Pi παραμένει μια από τις πιο ισχυρές και προσιτές πλατφόρμες ανάπτυξης στη σύγχρονη εποχή.

3.2.2 PMS5003

Το PMS5003 είναι ένας αισθητήρας σωματιδίων αέρα που χρησιμοποιείται ευρέως για την ανίχνευση και την παρακολούθηση της ποιότητας του αέρα. Παράγεται από την εταιρεία Plantower και είναι γνωστός για την ακρίβειά του στην ανίχνευση λεπτών σωματιδίων, όπως η σκόνη και άλλοι ρύποι, που έχουν διάμετρο από 0,3 έως 10 μικρόμετρα. Ο αισθητήρας χρησιμοποιεί τεχνολογία σκέδασης φωτός για να μετρήσει την ποσότητα και το μέγεθος των σωματιδίων στον αέρα, παρέχοντας δεδομένα σε πραγματικό χρόνο που μπορούν να χρησιμοποιηθούν σε συστήματα παρακολούθησης της ποιότητας του αέρα, σε καθαριστές αέρα, και σε έργα IoT (Internet of Things). [38]

Ένα από τα κύρια πλεονεκτήματα του PMS5003 είναι η ευκολία χρήσης και η αξιοπιστία του. Ο αισθητήρας παρέχει δεδομένα μέσω μιας σειριακής διεπαφής (UART), καθιστώντας τον εύκολα ενσωματώσιμο σε διάφορες πλατφόρμες όπως το Arduino, το Raspberry Pi και άλλα μικροελεγκτές. Επιπλέον, το PMS5003 είναι αρκετά μικρό και ελαφρύ, κάτι που το καθιστά ιδανικό για ενσωμάτωση σε φορητές συσκευές ή συστήματα με περιορισμένο χώρο. Ωστόσο, όπως κάθε τεχνολογία, ο

αισθητήρας αυτός απαιτεί τακτική συντήρηση και σωστή τοποθέτηση για να διασφαλιστεί ότι τα δεδομένα που παρέχει είναι ακριβή και αξιόπιστα.



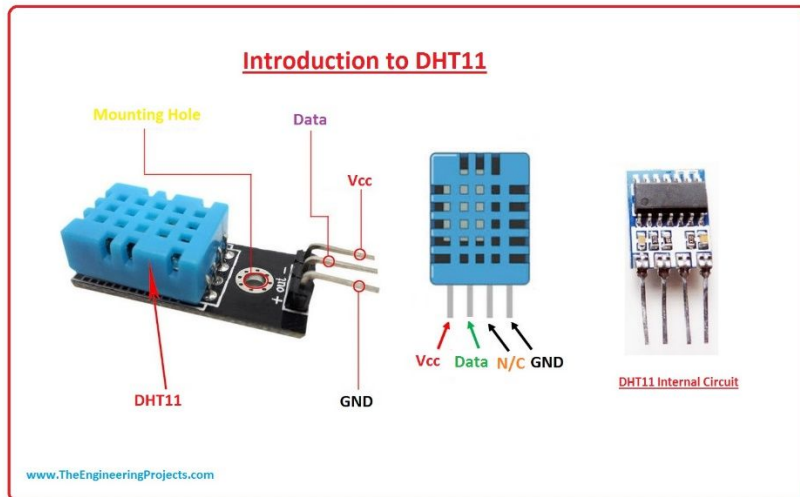
Εικόνα 3.4: PMS5003

[https://coolcomponents.co.uk/cdn/shop/products/pm2-5-air-quality-sensor-and-breadboard-adapter-kit-pms5003-id-3686-atmospheric-sensors-adafruit-cool-components_630.jpg?v=1537307014&width=1214]

3.2.3 DHT11

Ο DHT11 είναι ένας δημοφιλής αισθητήρας χαμηλού κόστους που χρησιμοποιείται για τη μέτρηση της θερμοκρασίας και της υγρασίας σε διάφορες εφαρμογές. Χρησιμοποιείται συχνά σε έργα DIY (Do It Yourself) και σε εφαρμογές αυτοματισμού λόγω της απλότητάς του και της ευκολίας ενσωμάτωσής σε μικροελεγκτές όπως το Arduino και το Raspberry Pi. Ο DHT11 μπορεί να μετρήσει θερμοκρασίες από 0°C έως 50°C και υγρασία από 20% έως 90%, με μια μέτρια ακρίβεια. Αν και δεν είναι ο πιο ακριβής αισθητήρας στην αγορά, είναι ιδανικός για έργα όπου η ακρίβεια δεν είναι η κύρια προτεραιότητα, αλλά απαιτείται μια οικονομική και εύκολη λύση. [39]

Ένα από τα κύρια πλεονεκτήματα του DHT11 είναι η απλότητά του στη χρήση. Ο αισθητήρας περιλαμβάνει ένα ενσωματωμένο ADC (Analog-to-Digital Converter) και παρέχει τα δεδομένα σε ψηφιακή μορφή μέσω μίας σειριακής διεπαφής, γεγονός που διευκολύνει την επικοινωνία με τον μικροελεγκτή. Παρόλο που ο DHT11 έχει περιορισμένη ακρίβεια και εύρος μέτρησης σε σύγκριση με άλλους αισθητήρες, η αξιοπιστία του και το χαμηλό κόστος τον καθιστούν κατάλληλο για εφαρμογές όπου η χαμηλή κατανάλωση ενέργειας και η απλότητα της εγκατάστασης είναι κρίσιμες. Ωστόσο, για πιο απαιτητικές εφαρμογές όπου απαιτείται υψηλότερη ακρίβεια, ίσως να χρειαστεί η χρήση ενός πιο προηγμένου αισθητήρα, όπως ο DHT22.

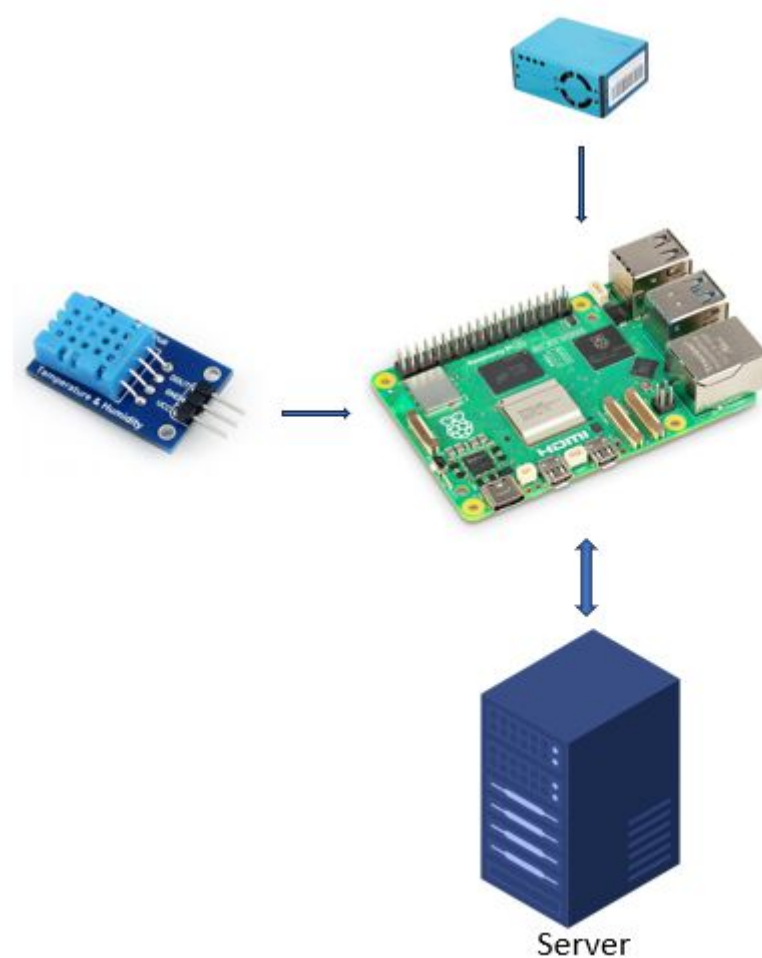


Εικόνα 3.5: DHT11

[<https://images.theengineeringprojects.com/image/main/2019/02/dht11.jpg>]

Κεφάλαιο 4ο: Το σύστημα παρακολούθησης ποιότητας αέρα και συνθήκες άνεσης σε κλειστούς χώρους

4.1 Εισαγωγή στο συνολικό σύστημα



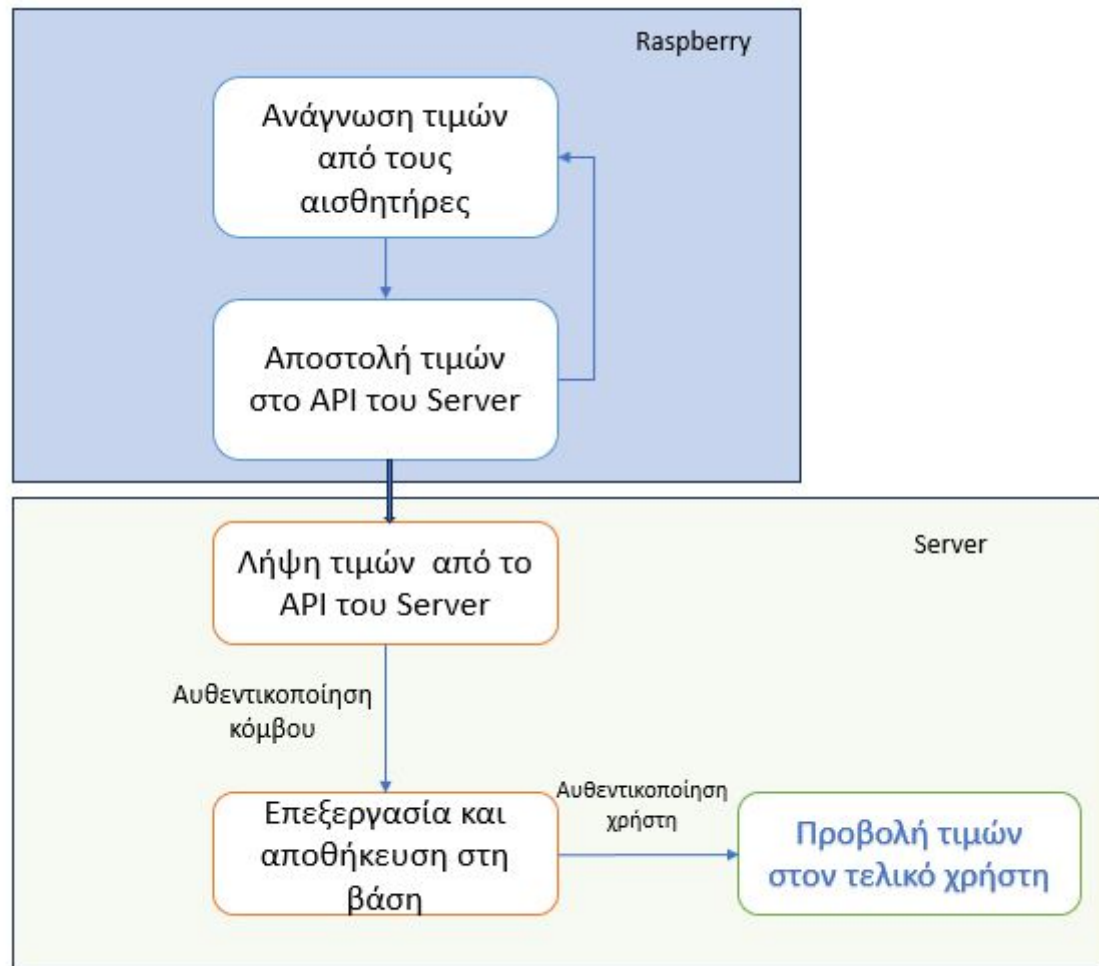
Εικόνα 4.1: Το συνολικό σύστημα

Στην Εικόνα 4.1 παρουσιάζεται το συνολικό σύστημα παρακολούθησης ποιότητας αέρα και συνθήκες άνεσης σε κλειστούς χώρους.

Η Εικόνα 4.2 που παρουσιάζεται απεικονίζει τη ροή δεδομένων από έναν αισθητήρα που είναι συνδεδεμένος με ένα Raspberry Pi μέχρι την τελική προβολή αυτών των δεδομένων από τον χρήστη μέσω ενός διακομιστή (server).

1. Ανάγνωση Τιμών από τους Αισθητήρες (Raspberry Pi)

Το Raspberry Pi είναι συνδεδεμένο με διάφορους αισθητήρες που μετρούν παραμέτρους όπως η θερμοκρασία, η υγρασία, και η συγκέντρωση σωματιδίων. Σε αυτό το στάδιο, οι τιμές που συλλέγονται από τους αισθητήρες διαβάζονται και προετοιμάζονται για αποστολή.



Εικόνα 4.2: Το διάγραμμα ροής για το συνολικό σύστημα

2. Αποστολή Τιμών στο API του Server (Raspberry Pi)

Αφού διαβαστούν οι τιμές από τους αισθητήρες, το Raspberry Pi αποστέλλει αυτά τα δεδομένα μέσω ενός HTTP GET αιτήματος στο API του διακομιστή. Το μήνυμα περιλαμβάνει τόσο τις μετρήσεις όσο και πληροφορίες για τον κόμβο που αποστέλλει τα δεδομένα, όπως το αναγνωριστικό του κόμβου.

3. Λήψη Τιμών από το API του Server (Server)

Ο διακομιστής λαμβάνει το αίτημα από το Raspberry Pi και αποσπά τις τιμές που περιλαμβάνονται στο μήνυμα. Ακολουθεί η διαδικασία αυθεντικοποίησης του κόμβου, όπου ο διακομιστής ελέγχει αν ο κόμβος είναι καταχωρισμένος στη βάση δεδομένων.

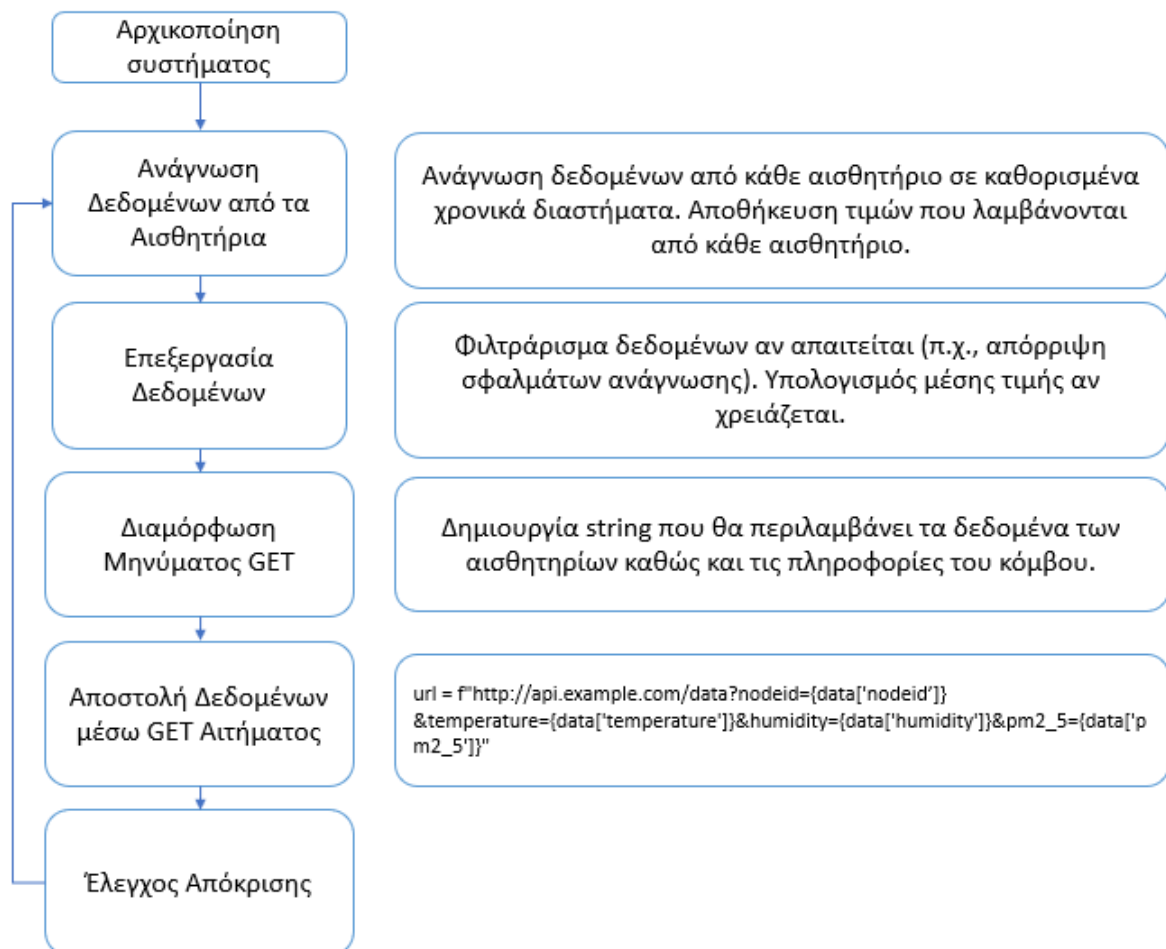
4. Επεξεργασία και Αποθήκευση στη Βάση (Server)

Αφού επιβεβαιωθεί ότι ο κόμβος είναι έγκυρος, τα δεδομένα αποθηκεύονται στη βάση δεδομένων του διακομιστή για μελλοντική επεξεργασία και ανάλυση.

5. Προβολή Τιμών στον Τελικό Χρήστη (Server)

Οι εξουσιοδοτημένοι χρήστες μπορούν να συνδεθούν στο σύστημα και να δουν τα δεδομένα που έχουν συλλεχθεί από τους κόμβους τους. Οι χρήστες λαμβάνουν τις κατάλληλες πληροφορίες και μηνύματα σχετικά με τις συνθήκες που επικρατούν στο χώρο που αντιστοιχεί στον εκάστοτε κόμβο, όπως για παράδειγμα προειδοποιήσεις για κακή ποιότητα αέρα.

4.2 Το σύστημα από την πλευρά της συσκευής



Εικόνα 4.3: Το διάγραμμα ροής από την πλευρά της συσκευής-κόμβου

Η Εικόνα 4.3 που παρουσιάζεται περιγράφει τη διαδικασία που εκτελείται από ένα σύστημα που χρησιμοποιεί Raspberry Pi για να διαβάσει δεδομένα από αισθητήρες και να τα αποστέλλει μέσω ενός HTTP GET αιτήματος σε έναν server.

1. Αρχικοποίηση Συστήματος:

Το σύστημα ξεκινά με την αρχικοποίηση, όπου οι απαραίτητες παράμετροι και οι βιβλιοθήκες φορτώνονται, και το Raspberry Pi ετοιμάζεται να διαβάσει δεδομένα από τους συνδεδεμένους αισθητήρες.

2. Ανάγνωση Δεδομένων από τα Αισθητήρια:

Σε αυτό το βήμα, τα δεδομένα διαβάζονται από κάθε αισθητήρα σε καθορισμένα χρονικά διαστήματα. Οι τιμές αποθηκεύονται προσωρινά για επεξεργασία και αποστολή.

3. Επεξεργασία Δεδομένων:

Τα δεδομένα που συλλέγονται από τους αισθητήρες υφίστανται επεξεργασία. Αυτό περιλαμβάνει την απόρριψη τυχόν σφαλμάτων ανάγνωσης και τον υπολογισμό μέσων τιμών εάν απαιτείται.

4. Διαμόρφωση Μηνύματος GET:

Δημιουργείται ένα string που περιλαμβάνει τα δεδομένα των αισθητήρων καθώς και τις πληροφορίες του κόμβου, όπως το node_id. Αυτή η συμβολοσειρά θα χρησιμοποιηθεί για την αποστολή του αιτήματος GET στον server.

5. Αποστολή Δεδομένων μέσω GET Αιτήματος:

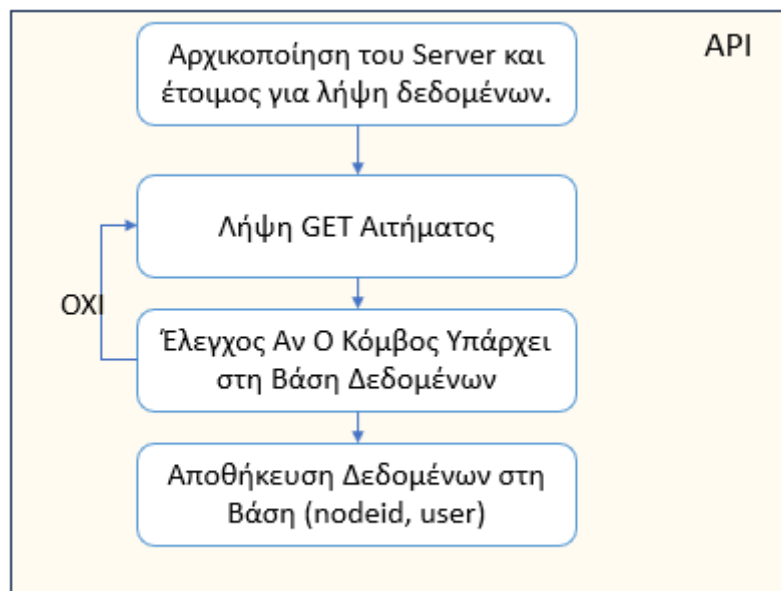
Το διαμορφωμένο μήνυμα αποστέλλεται στον server μέσω ενός αιτήματος GET. Το μήνυμα περιλαμβάνει όλες τις σχετικές πληροφορίες από τα αισθητήρια και τον κόμβο.

6. Έλεγχος Απόκρισης:

Μετά την αποστολή του αιτήματος, το σύστημα λαμβάνει και ελέγχει την απόκριση του server για να επιβεβαιώσει ότι τα δεδομένα αποθηκεύτηκαν επιτυχώς ή για να αναγνωρίσει τυχόν σφάλματα.

Αυτό το διάγραμμα ροής απεικονίζει τη διαδικασία συλλογής και αποστολής δεδομένων σε πραγματικό χρόνο από αισθητήρες μέσω ενός συστήματος Raspberry Pi προς έναν απομακρυσμένο server, διασφαλίζοντας την αξιόπιστη παρακολούθηση των περιβαλλοντικών συνθηκών.

4.3 Το σύστημα από την πλευρά του server



Εικόνα 4.4: Το διάγραμμα ροής από την πλευρά του API-server

Στη Εικόνα 4.4 παρουσιάζεται περιγραφή της διαδικασίας λήψης και επεξεργασίας δεδομένων από έναν server που λειτουργεί ως API για την αποθήκευση δεδομένων από αισθητήρες συνδεδεμένους σε Raspberry Pi ή άλλους κόμβους.

1. Αρχικοποίηση του Server:

Ο server αρχικοποιείται και τίθεται σε ετοιμότητα για να λάβει δεδομένα από εξωτερικούς κόμβους. Σε αυτό το στάδιο, φορτώνονται οι απαραίτητες βιβλιοθήκες και εργαλεία, και ο server προετοιμάζεται για την επεξεργασία αιτημάτων GET.

2. Λήψη GET Αιτήματος:

Ο server λαμβάνει ένα GET αίτημα από έναν κόμβο που περιλαμβάνει δεδομένα αισθητήρων και άλλες σχετικές πληροφορίες, όπως το αναγνωριστικό του κόμβου (nodeid).

3. Έλεγχος Αν Ο Κόμβος Υπάρχει στη Βάση Δεδομένων:

Ο server ελέγχει αν το nodeid που περιλαμβάνεται στο αίτημα υπάρχει ήδη στη βάση δεδομένων.

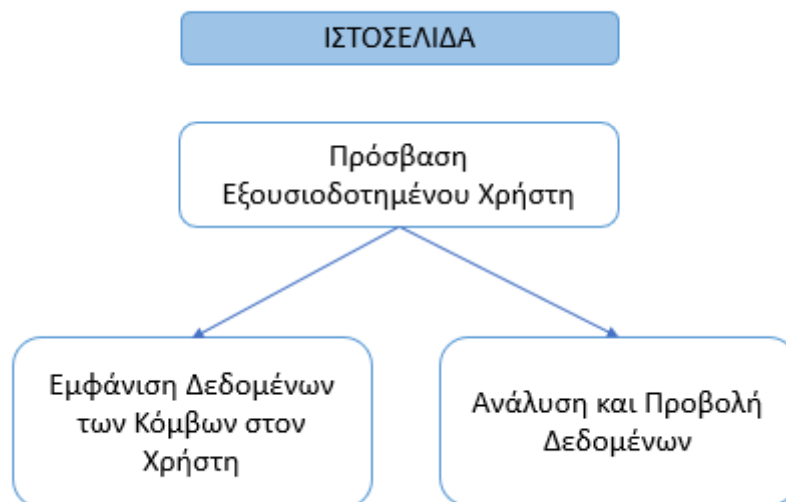
Ναι: Προχωρά στο βήμα 4.

Όχι: Επιστρέφει μήνυμα σφάλματος και σταματά τη διαδικασία (η ροή επιστρέφει στον προηγούμενο βρόχο).

4. Αποθήκευση Δεδομένων στη Βάση Δεδομένων:

Εάν ο κόμβος είναι καταχωρισμένος, ο server αποθηκεύει τα δεδομένα στη βάση δεδομένων. Η αποθήκευση περιλαμβάνει το nodeid, τις τιμές που λαμβάνονται από τα αισθητήρια και τυχόν άλλες σχετικές πληροφορίες, όπως ο χρήστης που είναι συνδεδεμένος με τον κόμβο.

Αυτό το διάγραμμα ροής περιγράφει έναν τυπικό μηχανισμό λήψης και αποθήκευσης δεδομένων από έναν server που λειτουργεί ως κεντρικό σημείο συλλογής δεδομένων από καταναμημένους κόμβους, διασφαλίζοντας ότι μόνο εγγεγραμμένοι και εξουσιοδοτημένοι κόμβοι μπορούν να αποστέλλουν δεδομένα για αποθήκευση.



Εικόνα 4.5: Το διάγραμμα ροής από την πλευρά του server - Ιστοσελίδα

Η Εικόνα 4.5 που παρουσιάζεται περιγράφει τη διαδικασία που λαμβάνει χώρα σε μια ιστοσελίδα που επιτρέπει σε εξουσιοδοτημένους χρήστες να έχουν πρόσβαση και να αναλύουν δεδομένα από κόμβους που παρακολουθούν συνθήκες σε διάφορους χώρους.

1. Πρόσβαση Εξουσιοδοτημένου Χρήστη:

Ο εξουσιοδοτημένος χρήστης εισέρχεται στην ιστοσελίδα χρησιμοποιώντας τα διαπιστευτήριά του. Με την επιτυχή σύνδεση, ο χρήστης έχει πρόσβαση στα δεδομένα των κόμβων που είναι συνδεδεμένοι με τον λογαριασμό του.

2. Εμφάνιση Δεδομένων των Κόμβων στον Χρήστη:

Μετά τη σύνδεση, ο χρήστης μπορεί να δει τα δεδομένα που συλλέγονται από τους κόμβους του. Αυτά τα δεδομένα μπορεί να περιλαμβάνουν μετρήσεις όπως η θερμοκρασία, η υγρασία, και τα επίπεδα ατμοσφαιρικών ρύπων.

Η προβολή αυτών των δεδομένων επιτρέπει στον χρήστη να παρακολουθεί τις συνθήκες που επικρατούν στους χώρους που ελέγχει.

3. Ανάλυση και Προβολή Δεδομένων:

Ο χρήστης έχει επίσης τη δυνατότητα να αναλύει τα δεδομένα μέσω της ιστοσελίδας. Αυτό μπορεί να περιλαμβάνει τη δημιουργία γραφημάτων, την παρακολούθηση ιστορικών δεδομένων και την προβολή τάσεων με την πάροδο του χρόνου.

Η ανάλυση αυτή επιτρέπει στον χρήστη να βγάλει συμπεράσματα για τη διαχείριση των συνθηκών στο χώρο του και να λάβει τις απαραίτητες ενέργειες για τη βελτίωση της ποιότητας του αέρα ή άλλων παραμέτρων.

Αυτό το διάγραμμα απεικονίζει πώς ένας εξουσιοδοτημένος χρήστης μπορεί να αλληλεπιδρά με την ιστοσελίδα για να διαχειριστεί και να αναλύσει τα δεδομένα που προέρχονται από κόμβους αισθητήρων.

4.3.1 Από την πλευρά του κόμβου

Παρακάτω παρατίθεται ο κώδικας Python με σχόλια για το Raspberry Pi, ο οποίος διαβάζει τις τιμές από τους αισθητήρες (όπως το DHT11 για θερμοκρασία και υγρασία και το PMS5003 για σωματίδια PM2.5, PM1.0, PM10) και στέλνει τα δεδομένα αυτά σε έναν API server μέσω ενός GET αιτήματος.

```
import time # Βιβλιοθήκη για τον έλεγχο χρόνου

import requests # Βιβλιοθήκη για να στέλνουμε HTTP αιτήματα

import Adafruit_DHT # Βιβλιοθήκη για τον αισθητήρα DHT

from pms5003 import PMS5003, ReadTimeoutError # Βιβλιοθήκη για τον αισθητήρα PMS5003


# Αρχικοποίηση αισθητήρων

DHT_SENSOR = Adafruit_DHT.DHT11

DHT_PIN = 4 # GPIO pin που είναι συνδεδεμένος ο DHT11

pms5003 = PMS5003()
```

```

# Διεύθυνση URL του API server όπου θα στέλνουμε τα δεδομένα

API_URL = "http://api.example.com/data"

# Αναγνωριστικό του κόμβου (node_id) που θα περιλαμβάνεται στο αίτημα

NODE_ID = "RaspberryPi_001"

LOCATION = "Lab1"

def get_sensor_data():
    """
    Λειτουργία που διαβάζει τις τιμές από τους αισθητήρες DHT11 και PMS5003.
    """
    # Ανάγνωση τιμών θερμοκρασίας και υγρασίας από τον αισθητήρα DHT11
    humidity, temperature = Adafruit_DHT.read(DHT_SENSOR, DHT_PIN)

    try:
        # Ανάγνωση τιμών σωματιδίων από τον αισθητήρα PMS5003
        pm_data = pms5003.read()

        pm1_0 = pm_data.pm_ug_per_m3(1.0)
        pm2_5 = pm_data.pm_ug_per_m3(2.5)
        pm10 = pm_data.pm_ug_per_m3(10)

    except ReadTimeoutError:
        print("Σφάλμα: Ο αισθητήρας PMS5003 δεν αποκρίνεται.")
        pm1_0 = None
        pm2_5 = None
        pm10 = None

    return temperature, humidity, pm1_0, pm2_5, pm10

def send_data_to_server(temperature, humidity, pm1_0, pm2_5, pm10):
    """
    Λειτουργία που στέλνει τα δεδομένα αισθητήρων στον API server μέσω ενός GET αιτήματος.
    """
    # Δημιουργία του URL με τα δεδομένα ως query string

```

```

url =

f"{API_URL}?nodeid={NODEID}& temperature={temperature}&humidity={humidity}&pm1_0={pm1_0}&pm2_5={pm2_5}&pm10={pm10}"

try:

    # Στέλνουμε το GET αίτημα στον server

    response = requests.get(url)

    # Έλεγχος αν το αίτημα ήταν επιτυχές

    if response.status_code == 200:

        print("Τα δεδομένα στάλθηκαν επιτυχώς.")

    else:

        print(f"Αποτυχία αποστολής δεδομένων. Κωδικός σφάλματος: {response.status_code}")

except requests.exceptions.RequestException as e:

    # Εκτύπωση του σφάλματος σε περίπτωση αποτυχίας

    print(f"Σφάλμα κατά την αποστολή δεδομένων: {e}")

def main():

    """

    Κύρια λειτουργία που εκτελείται συνεχώς και στέλνει δεδομένα σε προκαθορισμένα διαστήματα.

    """

    while True:

        # Ανάγνωση δεδομένων από τους αισθητήρες

        temperature, humidity, pm1_0, pm2_5, pm10 = get_sensor_data()

        # Έλεγχος αν η ανάγνωση από τον αισθητήρα DHT ήταν επιτυχής

        if temperature is not None and humidity is not None:

            # Στέλνουμε τα δεδομένα στον server

            send_data_to_server(temperature, humidity, pm1_0, pm2_5, pm10)

        else:

            print("Σφάλμα ανάγνωσης δεδομένων από τον αισθητήρα DHT.")

        # Αναμονή για 10 λεπτά πριν την επόμενη αποστολή δεδομένων

        time.sleep(600)

```

```
if __name__ == "__main__":  
  
    main()
```

Ο αισθητήρας DHT11 χρησιμοποιείται για τη μέτρηση της θερμοκρασίας και της υγρασίας, ενώ ο αισθητήρας PMS5003 χρησιμοποιείται για τη μέτρηση των συγκεντρώσεων PM1.0, PM2.5 και PM10.

Συνάρτηση `get_sensor_data()`: Αυτή η συνάρτηση διαβάζει τις τιμές από τους αισθητήρες DHT11 και PMS5003. Τα δεδομένα επιστρέφονται για περαιτέρω επεξεργασία.

Σε περίπτωση που ο PMS5003 δεν αποκρίνεται, η συνάρτηση θα επιστρέψει `None` για τις τιμές PM.

Συνάρτηση `send_data_to_server()`: Η συνάρτηση αυτή δημιουργεί ένα GET αίτημα με τα δεδομένα των αισθητήρων και τις πληροφορίες του κόμβου και τα στέλνει στον API server. Τα δεδομένα περιλαμβάνουν πλέον τις τιμές PM1.0, PM2.5 και PM10.

Κύρια Συνάρτηση `main()`: Αυτή η συνάρτηση τρέχει συνεχώς και στέλνει τα δεδομένα στον server κάθε 10 λεπτά.

Αυτός ο κώδικας επιτρέπει τη συνεχή παρακολούθηση και αποστολή των δεδομένων PM1.0, PM2.5, PM10, θερμοκρασίας και υγρασίας από τους αισθητήρες στο Raspberry Pi σε έναν απομακρυσμένο server για περαιτέρω επεξεργασία και ανάλυση.

4.3.2 API για τα requests

Αυτά είναι τα Routes που χρησιμοποιήσαμε:

```
Route::get('/', function () {  
    return view('welcome');  
});  
Route::get('/about', [MainController::class, 'about']);  
  
//API για να λάβουμε τιμές απο το Raspberry και να τις αποθηκεύσουμε στη βάση  
Route::post('/sensors/store', [MainController::class, 'storeSensorData']->name('sensors.store'));  
  
// Προβολή όλων των places για τον χρήστη στο dashboard  
Route::get('/dashboard', [MainController::class, 'dashboard']->name('dashboard'));  
  
// Προβολή σελίδας για τη δημιουργία νέου place  
Route::get('/place/create', [MainController::class, 'create']->name('place.create'));
```

```

// Αποθήκευση νέου place στη βάση δεδομένων
Route::post('/place/store', [MainController::class, 'store']->name('place.store'));

// Προβολή σελίδας για επεξεργασία ενός συγκεκριμένου place
Route::get('/place/{id}/edit', [MainController::class, 'edit']->name('place.edit'));

// Ενημέρωση ενός place μετά την επεξεργασία
Route::post('/place/{id}/update', [MainController::class, 'update']->name('place.update'));

// Προβολή των δεδομένων αισθητήρων για ένα συγκεκριμένο place - Όλα μαζί
Route::get('/place/{id}', [MainController::class, 'show']->name('place.show'));
// Προβολή των δεδομένων αισθητήρων για ένα συγκεκριμένο place- Ξεχωριστά
Route::get('/place2/{id}', [MainController::class, 'show2']->name('place.show2'));
// Προβολή των δεδομένων αισθητήρων για ένα συγκεκριμένο place - Δείκτης Καταλληλότητας
Route::get('/place3/{id}', [MainController::class, 'show3']->name('place.show3'));

```

και η συνάρτηση για την αποδοχή των τιμών από το Raspberry

```

class MainController extends Controller
{

    public function storeSensorData(Request $request)
    {
        try {
            Log::info(json_encode('storeSensorData'));

            //Επικύρωση των δεδομένων που λαμβάνονται από το αίτημα
            $validatedData = $request->validate([
                'userid' => 'required|integer',
                'placeid' => 'required|integer',
                'temp' => 'required|numeric',
                'hum' => 'required|numeric',
                'pm1' => 'nullable|numeric',
                'pm2_5' => 'required|numeric',
                'pm10' => 'required|numeric',
            ]);

            // Αποθήκευση των δεδομένων στη βάση δεδομένων
            DB::table('sensors')->insert([
                'userid' => $request['userid'],
                'placeid' => $request['placeid'],
            ]);
        } catch (\Illuminate\Http\Exceptions\ValidationException $e) {
            // Επεξεργασία σφάλματος
        }
    }
}

```

```

        'temp' => $request['temp'],
        'hum' => $request['hum'],
        'pm1' => $request['pm1'],
        'pm2_5' => $request['pm2_5'],
        'pm10' => $request['pm10'],
        'created' => now(),
    ]);

    return response()->json(['message' => 'Τα δεδομένα αποθηκεύτηκαν επιτυχώς.'], 200);
} catch (\Exception $e) {
    // Εκτύπωση του σφάλματος
    return response()->json(['error' => $e->getMessage()], 500);
}
}

```

4.3.3 Laravel για εξυπηρέτηση της ιστοσελίδας



Εικόνα 4.6: Σελίδα υποδοχής

Η σελίδα υποδοχής παρέχει ένα κεντρικό σημείο για την πλοήγηση στο σύστημα. Εδώ ο χρήστης έχει τη δυνατότητα να συνδεθεί ή να προχωρήσει σε εγγραφή για να αποκτήσει πρόσβαση στις υπηρεσίες του συστήματος.

Σύστημα παρακολούθησης ποιότητας αέρα και συνθήκες άνεσης σε κλειστούς χώρους

Σύνδεση

Email

nikos

Pass

....

Σύνδεση

Εικόνα 4.7: Σύνδεση χρήστη

Η σελίδα σύνδεσης ζητάει από τον χρήστη τα διαπιστευτήριά του (όνομα χρήστη και κωδικό πρόσβασης) για να του επιτρέψει την πρόσβαση στον λογαριασμό του και στις υπηρεσίες του συστήματος.



Σύστημα παρακολούθησης ποιότητας αέρα και
συνθήκες άνεσης σε κλειστούς χώρους

[Logout](#)

[nikos]

Τα Μέρη Σας

Προσθήκη Νέου Μέρους

Εργαστήριο Δ4

Κωδικός Χρήστη: 1

Επεξεργασία

Προβολή Αισθητήρων Μαζί

Προβολή Αισθητήρων

Δείκτης

Αίθουσα B1

Κωδικός Χρήστη: 1

Επεξεργασία

Προβολή Αισθητήρων Μαζί

Προβολή Αισθητήρων

Δείκτης

Εικόνα 4.8: Προβολή places του χρήστη

Επιλογές για δημιουργία νέου χώρου, επεξεργασία χώρου, προβολή γραφημάτων τιμών αισθητήρων – σε ένα ενιαίο ή σε ξεχωριστά γραφήματα, προβολή δείκτη καταλληλότητας.

Σε αυτήν τη σελίδα ο χρήστης μπορεί να δει όλα τα διαθέσιμα μέρη (places) που σχετίζονται με τον λογαριασμό του. Υπάρχουν επιλογές για τη δημιουργία νέου χώρου, την επεξεργασία των υπαρχόντων χώρων, καθώς και την προβολή δεδομένων αισθητήρων μέσω γραφημάτων (είτε σε ένα ενιαίο είτε σε ξεχωριστά γραφήματα). Επίσης, παρέχεται η δυνατότητα προβολής του δείκτη καταλληλότητας του χώρου.

Προσθήκη Νέου Μέρους

Όνομα Μέρους

Εισαγάγετε όνομα μέρους

Προσθήκη

Επιστροφή στο Dashboard

Εικόνα 4.9: Σελίδα δημιουργίας του νέου χώρου

Η σελίδα αυτή επιτρέπει στον χρήστη να δημιουργήσει έναν νέο χώρο (place). Ο χρήστης εισάγει το όνομα του χώρου και στη συνέχεια αποθηκεύει τα στοιχεία του για μελλοντική χρήση.

Επεξεργασία Μέρους

Όνομα Μέρους

Εργαστήριο Δ4

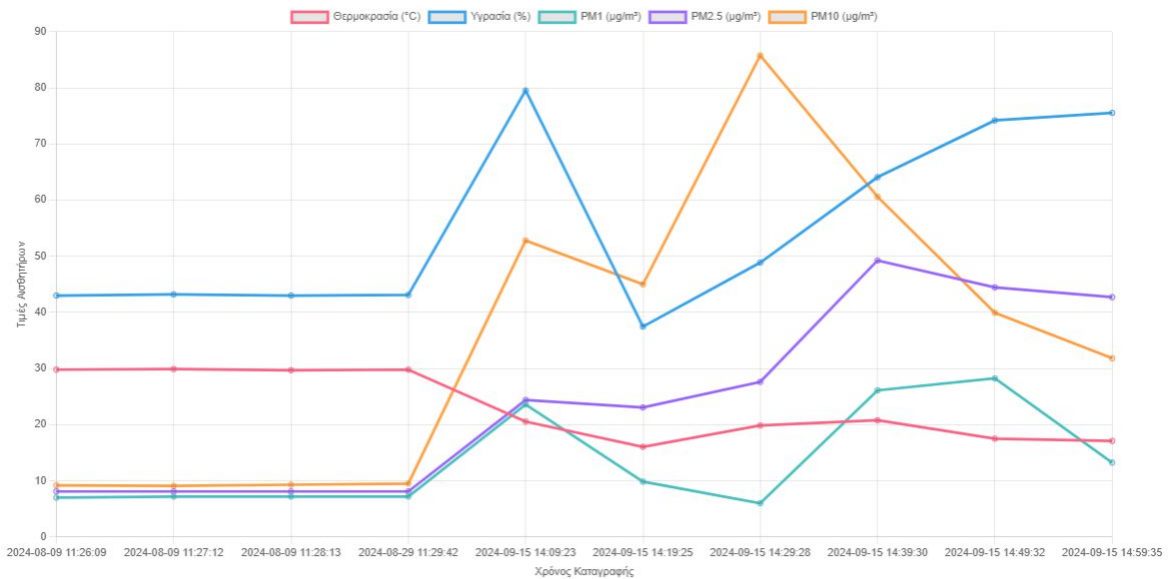
Αποθήκευση

Επιστροφή στο Dashboard

Εικόνα 4.10: Σελίδα τροποποίησης του χώρου

Η σελίδα τροποποίησης του χώρου επιτρέπει στον χρήστη να επεξεργαστεί τα στοιχεία ενός υπάρχοντος χώρου. Ο χρήστης μπορεί να αλλάξει το όνομα του χώρου και να αποθηκεύσει τις αλλαγές.

Δεδομένα Αισθητήρων



[Επιστροφή στο Dashboard](#)

Εικόνα 4.11: Σελίδα προβολής των δεδομένων του κόμβου- ενιαίο γράφημα

Αυτή η σελίδα παρέχει μια ενιαία προβολή όλων των δεδομένων των αισθητήρων (θερμοκρασία, υγρασία, PM1, PM2.5, PM10) του συγκεκριμένου κόμβου σε ένα κοινό γράφημα, επιτρέποντας στον χρήστη να δει συνολικά τις τιμές των αισθητήρων με την πάροδο του χρόνου.

Η σελίδα 4.12 παρουσιάζει τα δεδομένα των αισθητήρων προβάλλοντας σε ξεχωριστά γραφήματα. Για κάθε αισθητήρα (π.χ., θερμοκρασία, υγρασία, PM1, PM2.5, PM10) υπάρχει διαφορετικό γράφημα, παρέχοντας λεπτομερέστερη ανάλυση των τιμών.



Εικόνα 4.12: Σελίδα προβολής των δεδομένων του κόμβου- ξεχωριστά γραφήματα

Κατάσταση Αισθητήρων

Αισθητήρας	Τελευταία Τιμή	Μονάδα
Θερμοκρασία	29.8 °C	°C
Υγρασία	43.1 %	%
PM1	7.2 µg/m ³	µg/m ³
PM2.5	8.1 µg/m ³	µg/m ³
PM10	9.5 µg/m ³	µg/m ³

Δείκτης Καταλληλότητας

Η τρέχουσα κατάσταση: **Μη Κατάλληλη**
Σχόλιο: Η θερμοκρασία δεν είναι εντός των επιθυμητών ορίων.

Όρια Τιμών και Υπολογισμός Δείκτη Καταλληλότητας

Αισθητήρας	Κατάλληλες Τιμές	Μη Κατάλληλες Τιμές
Θερμοκρασία	18 - 24 °C	Κάτω από 18 °C ή πάνω από 24 °C
Υγρασία	30% - 60%	Κάτω από 30% ή πάνω από 60%
PM1	< 15 µg/m ³	Πάνω από 15 µg/m ³
PM2.5	< 25 µg/m ³	Πάνω από 25 µg/m ³
PM10	< 50 µg/m ³	Πάνω από 50 µg/m ³

Ο δείκτης καταλληλότητας βασίζεται σε αυτές τις τιμές. Αν η θερμοκρασία, η υγρασία ή τα επίπεδα των σωματιδίων βρίσκονται εκτός των παραπάνω ορίων, η συνθήκη θεωρείται μη κατάλληλη για διαβίωση.

[Επιστροφή στο Dashboard](#)

Εικόνα 4.13: Σελίδα για Κατάσταση Αισθητήρων - Δείκτης Καταλληλότητας

Αυτή η σελίδα δείχνει την τρέχουσα κατάσταση των αισθητήρων (τελευταίες τιμές) μαζί με έναν δείκτη που ενημερώνει τον χρήστη εάν οι συνθήκες είναι κατάλληλες για διαβίωση. Στο τέλος της σελίδας, υπάρχει πίνακας που εξηγεί τα όρια των τιμών των αισθητήρων και τον τρόπο υπολογισμού του δείκτη καταλληλότητας.

4.3.4 Η βάση του συστήματος

Η βάση δεδομένων του έργου αποτελεί έναν κεντρικό πυλώνα για τη λειτουργία του συστήματος παρακολούθησης συνθηκών αέρα και άνεσης. Στηρίζεται σε μια MySQL βάση δεδομένων, η οποία περιέχει διάφορους πίνακες για την αποθήκευση των χρηστών, των χώρων (places), των αισθητήρων και των τιμών που συλλέγονται από αυτούς. Παρακάτω παρατίθεται μια αναλυτική περιγραφή της δομής της βάσης:

Πίνακες της Βάσης Δεδομένων

user:

Περιγραφή: Ο πίνακας αυτός αποθηκεύει πληροφορίες για τους χρήστες του συστήματος.

Πεδία:

id: Το μοναδικό αναγνωριστικό του χρήστη.

username: Όνομα χρήστη για σύνδεση.

password: Κωδικός πρόσβασης (συνήθως αποθηκευμένος με κρυπτογράφηση).

lastname: Επώνυμο του χρήστη.

firstname: Όνομα του χρήστη.

kind: Ο τύπος του χρήστη (π.χ. administrator, απλός χρήστης).

created: Η ημερομηνία δημιουργίας του λογαριασμού.

place:

Περιγραφή: Ο πίνακας αυτός αποθηκεύει τους χώρους που σχετίζονται με κάθε χρήστη.

Πεδία:

id: Το μοναδικό αναγνωριστικό του χώρου.

userid: Το αναγνωριστικό του χρήστη στον οποίο ανήκει ο χώρος (σύνδεση με τον πίνακα user).

placename: Το όνομα του χώρου (π.χ. "Γραφείο", "Αίθουσα B1").

sensors:

Περιγραφή: Αποθηκεύει τις τιμές που συλλέγονται από τους αισθητήρες σε κάθε χώρο.

Πεδία:

id: Το μοναδικό αναγνωριστικό της εγγραφής.

userid: Το αναγνωριστικό του χρήστη που συνδέεται με τον αισθητήρα.

placeid: Το αναγνωριστικό του χώρου στον οποίο ανήκουν οι αισθητήρες.

temp: Η καταγεγραμμένη θερμοκρασία από τον αισθητήρα.

hum: Το ποσοστό υγρασίας που καταγράφηκε.

pm1: Τα επίπεδα σωματιδίων PM1 που ανιχνεύθηκαν.

pm2_5: Τα επίπεδα σωματιδίων PM2.5 που ανιχνεύθηκαν.

pm10: Τα επίπεδα σωματιδίων PM10 που ανιχνεύθηκαν.

created: Η ημερομηνία και ώρα της εγγραφής των δεδομένων.

Λειτουργία της Βάσης Δεδομένων

Συσχέτιση χρηστών και χώρων: Κάθε χρήστης μπορεί να έχει πολλούς χώρους (places), οι οποίοι είναι αποθηκευμένοι στον πίνακα place. Η συσχέτιση γίνεται με το πεδίο userid.

Καταγραφή τιμών αισθητήρων: Σε κάθε χώρο συνδέονται αισθητήρες, και τα δεδομένα που συλλέγονται από αυτούς (θερμοκρασία, υγρασία, σωματίδια) αποθηκεύονται στον πίνακα sensors. Η συσχέτιση του πίνακα sensors με τον πίνακα place γίνεται μέσω του placeid, που δείχνει σε ποιο χώρο ανήκουν οι αισθητήρες.

Σχέσεις και Ακεραιότητα Δεδομένων

Πίνακας user → Πίνακας place: Σχέση 1 προς πολλά (ένας χρήστης μπορεί να έχει πολλούς χώρους).

Πίνακας place → Πίνακας sensors: Σχέση 1 προς πολλά (ένας χώρος μπορεί να έχει πολλές εγγραφές αισθητήρων).

Παραδείγματα Χρήσης

Εισαγωγή Νέων Δεδομένων Αισθητήρων: Όταν καταγράφονται νέες τιμές από τους αισθητήρες, αυτές αποθηκεύονται στον πίνακα sensors. Για κάθε νέα εγγραφή αποθηκεύονται η θερμοκρασία, η υγρασία και τα επίπεδα σωματιδίων PM για το συγκεκριμένο χώρο.

Προβολή Δεδομένων: Όταν ο χρήστης συνδέεται στην εφαρμογή και θέλει να δει τα δεδομένα του, το σύστημα ανακτά τα places που ανήκουν στον χρήστη από τον πίνακα place και τις αντίστοιχες εγγραφές από τον πίνακα sensors.

4.4 Ασφάλεια Συστήματος

Η ασφάλεια ενός συστήματος είναι ένα κρίσιμο στοιχείο για την προστασία των δεδομένων και των χρηστών του από κακόβουλες επιθέσεις και διαρροές πληροφοριών. Καθώς το σύστημα που αναπτύσσεται περιλαμβάνει τη συλλογή, την αποστολή και την αποθήκευση ευαίσθητων δεδομένων από αισθητήρες σε έναν κεντρικό server, καθώς και την πρόσβαση σε μια ιστοσελίδα διαχείρισης, η ασφάλεια πρέπει να είναι υψηλής προτεραιότητας. Αυτό περιλαμβάνει την ασφάλεια κατά τη μετάδοση

των δεδομένων από τη συσκευή (Raspberry Pi) στον server καθώς και την προστασία της πρόσβασης στην ιστοσελίδα από μη εξουσιοδοτημένους χρήστες.

Επιπλέον, η συνεχής αναβάθμιση του συστήματος ασφαλείας με τις τελευταίες πρακτικές και η εφαρμογή αυστηρών πολιτικών πρόσβασης και διαχείρισης των δικαιωμάτων χρήσης είναι απαραίτητα για την εξασφάλιση της ακεραιότητας του συστήματος. Η πρόληψη και η προστασία από επιθέσεις, όπως η υποκλοπή δεδομένων και οι επιθέσεις DDoS, είναι επίσης σημαντικά σημεία που πρέπει να λαμβάνονται υπόψη κατά την ανάπτυξη και τη λειτουργία του συστήματος.

4.4.1 Ασφάλεια στη μετάδοση δεδομένων από τη συσκευή στον Server

Η ασφάλεια στη μετάδοση δεδομένων από τη συσκευή (π.χ., Raspberry Pi) στον server είναι κρίσιμη για την αποτροπή της υποκλοπής και της αλλοίωσης των δεδομένων κατά τη μεταφορά τους. Για να διασφαλιστεί ότι τα δεδομένα παραμένουν ασφαλή, η χρήση κρυπτογράφησης είναι απαραίτητη. Η κρυπτογράφηση μέσω SSL/TLS (Secure Sockets Layer/Transport Layer Security) παρέχει ένα ασφαλές κανάλι επικοινωνίας μεταξύ της συσκευής και του server, προστατεύοντας τα δεδομένα. Δεν εφαρμόστηκε στο έργο μας διότι είναι πολύ δύσκολη εγκατάσταση certifications και από τις δυο πλευρές.

Γενικά είναι σημαντικό να γίνεται έλεγχος ταυτότητας τόσο στη συσκευή όσο και στον server πριν από κάθε επικοινωνία, διασφαλίζοντας ότι τα δεδομένα προέρχονται από εξουσιοδοτημένες πηγές και καταλήγουν στον σωστό προορισμό.

4.4.2 Ασφάλεια στη πρόσβαση στην ιστοσελίδα

Η ασφάλεια στην πρόσβαση στην ιστοσελίδα είναι εξίσου σημαντική για την προστασία των ευαίσθητων δεδομένων και τη διατήρηση της εμπιστευτικότητας των πληροφοριών των χρηστών. Η χρήση αυθεντικοποίησης μέσω ισχυρών κωδικών πρόσβασης και η εφαρμογή επιπρόσθετων μέτρων όπως η επαλήθευση δύο παραγόντων (2FA) μπορούν να ενισχύσουν σημαντικά την ασφάλεια.

Επιπλέον, θα πρέπει να εφαρμοστούν πολιτικές περιορισμένης πρόσβασης, όπου μόνο εξουσιοδοτημένοι χρήστες έχουν πρόσβαση σε συγκεκριμένα τμήματα της ιστοσελίδας, με βάση τα δικαιώματα που τους έχουν ανατεθεί. Η χρήση HTTPS για την πρόσβαση στην ιστοσελίδα διασφαλίζει ότι τα δεδομένα μεταξύ του χρήστη και του server είναι κρυπτογραφημένα και δεν μπορούν να υποκλαπούν. Τέλος, είναι σημαντικό να υπάρχει συνεχής παρακολούθηση για ασυνήθιστες δραστηριότητες και να εφαρμόζονται μηχανισμοί άμεσης αντίδρασης σε πιθανές παραβιάσεις ασφαλείας.

Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης

Το έργο αυτό, που αφορούσε την ανάπτυξη ενός ολοκληρωμένου συστήματος παρακολούθησης της ποιότητας του αέρα και των συνθηκών άνεσης σε κλειστούς χώρους, επιβεβαιώνει τη σημασία της τεχνολογίας στην προαγωγή της υγείας και της ευημερίας των ανθρώπων. Με την υλοποίηση αυτού του συστήματος, επιτεύχθηκε ο στόχος της καταγραφής και της ανάλυσης κρίσιμων περιβαλλοντικών παραμέτρων, όπως η θερμοκρασία, η υγρασία και τα επίπεδα σωματιδίων PM, παρέχοντας στους χρήστες πολύτιμες πληροφορίες σε πραγματικό χρόνο μέσω μιας φιλικής προς τον χρήστη ιστοσελίδας. Η ενσωμάτωση της τεχνολογίας Raspberry Pi με αισθητήρες, καθώς και η χρήση του πλαισίου Laravel για την ανάπτυξη του server και της ιστοσελίδας, επέτρεψαν τη δημιουργία ενός αποδοτικού και επεκτάσιμου συστήματος που μπορεί να χρησιμοποιηθεί σε διάφορες εφαρμογές.

Το σύστημα αυτό αναδεικνύει τη σημασία της ασφάλειας στη μετάδοση και αποθήκευση δεδομένων. Η αυθεντικοποίηση των χρηστών για την πρόσβαση στην ιστοσελίδα, διασφαλίζουν την προστασία των δεδομένων και των προσωπικών πληροφοριών των χρηστών. Η υλοποίηση αυτών των μέτρων ασφαλείας ενισχύει την αξιοπιστία και την αποδοχή του συστήματος από τους χρήστες, καθιστώντας το ένα σημαντικό εργαλείο για τη διαχείριση της ποιότητας του αέρα σε κλειστούς χώρους.

Η δυνατότητα του συστήματος να επεκταθεί και να προσαρμοστεί σε διαφορετικές ανάγκες και εφαρμογές είναι ένα από τα μεγαλύτερα πλεονεκτήματά του. Η αρχιτεκτονική του συστήματος επιτρέπει την προσθήκη νέων αισθητήρων, την ενσωμάτωση με άλλες πλατφόρμες και τη χρήση του σε μεγαλύτερης κλίμακας εφαρμογές, όπως βιομηχανικές εγκαταστάσεις ή σχολικά συγκροτήματα. Επιπλέον, η χρήση εργαλείων όπως τα DataTables και το Chart.js για την ανάλυση και την απεικόνιση των δεδομένων προσφέρει στους χρήστες μια ολοκληρωμένη εμπειρία διαχείρισης και κατανόησης των περιβαλλοντικών συνθηκών στους χώρους τους.

Συνολικά, το έργο αυτό όχι μόνο επιτυγχάνει τους τεχνικούς στόχους του αλλά και συμβάλλει ουσιαστικά στην προαγωγή της ασφάλειας και της υγείας των ανθρώπων μέσω της παρακολούθησης της ποιότητας του αέρα. Η εφαρμογή των τεχνολογιών που αναπτύχθηκαν προσφέρει μια καινοτόμο προσέγγιση στη διαχείριση των εσωτερικών περιβαλλοντικών συνθηκών, η οποία μπορεί να βελτιώσει την ποιότητα ζωής και την αποδοτικότητα των ανθρώπων στους χώρους όπου ζουν και εργάζονται. Το έργο αυτό μπορεί να αποτελέσει τη βάση για περαιτέρω έρευνα και ανάπτυξη σε τομείς που αφορούν την έξυπνη διαχείριση του περιβάλλοντος και την ευημερία των χρηστών.

BIBΛΙΟΓΡΑΦΙΑ

- [1] Pietraru, R.N.; Olteanu, A.; Adochiei, I.-R.; Adochiei, F.-C. Reengineering Indoor Air Quality Monitoring Systems to Improve End-User Experience. *Sensors* 2024, 24, 2659. <https://doi.org/10.3390/s24082659>
- [2] Sung, WT., Hsiao, SJ. Building an indoor air quality monitoring system based on the architecture of the Internet of Things. *J Wireless Com Network* 2021, 153 (2021). <https://doi.org/10.1186/s13638-021-02030-1>
- [3] Saini, J., Dutta, M. & Marques, G. A comprehensive review on indoor air quality monitoring systems for enhanced public health. *Sustain Environ Res* 30, 6 (2020). <https://doi.org/10.1186/s42834-020-0047-y>
- [4] Sassi, Mohamed Saifeddine Hadj, and Lamia Chaari Fourati. "Comprehensive survey on air quality monitoring systems based on emerging computing and communication technologies." *Computer Networks* 209 (2022): 108904.
- [5] Mumtaz R, Zaidi SMH, Shakir MZ, Shafi U, Malik MM, Haque A, Mumtaz S, Zaidi SAR. Internet of Things (IoT) Based Indoor Air Quality Sensing and Predictive Analytic—A COVID-19 Perspective. *Electronics*. 2021; 10(2):184. <https://doi.org/10.3390/electronics10020184>
- [6] Dai, X., Shang, W., Liu, J., Xue, M., & Wang, C. (2023). Achieving better indoor air quality with IoT systems for future buildings: Opportunities and challenges. *Science of The Total Environment*, 895, 164858.
- [7] Saini J, Dutta M, Marques G. Indoor Air Quality Monitoring Systems Based on Internet of Things: A Systematic Review. *International Journal of Environmental Research and Public Health*. 2020; 17(14):4942. <https://doi.org/10.3390/ijerph17144942>
- [8] https://www.sciencedirect.com/science/article/pii/S0048969720318982?casa_token=9l3IsjsnlfUAAAAA:8AxP2AyjxbpbrCon7EmDkROD3QdRS29d0NmnYYGcvYnkpmgjn7NRclNgbq4U7YdjNRAt8F_d
- [9] Saini, J., Dutta, M., & Marques, G. (2020). A comprehensive review on indoor air quality monitoring systems for enhanced public health. *Sustainable environment research*, 30, 1-12.
- [10] Abraham, S., & Li, X. (2014). A cost-effective wireless sensor network system for indoor air quality monitoring applications. *Procedia Computer Science*, 34, 165-171.
- [11] Benammar M, Abdaoui A, Ahmad SHM, Touati F, Kadri A. A Modular IoT Platform for Real-Time Indoor Air Quality Monitoring. *Sensors*. 2018; 18(2):581. <https://doi.org/10.3390/s18020581>

- [12] Jo, J., Jo, B., Kim, J., Kim, S., & Han, W. (2020). Development of an IoT-based indoor air quality monitoring platform. *Journal of Sensors*, 2020(1), 8749764.
- [13] <https://uk.getawair.com/>
- [14] <https://www.iqair.com/>
- [15] <https://foobot.io/>
- [16] <https://getuhoo.com/>
- [17] <https://www.netatmo.com/smart-indoor-air-quality-monitor>
- [18] Rasmussen, M. (2022). *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. O'Reilly Media.
- [19] Welling, L., & Thomson, L. (2016). *PHP and MySQL Web Development (5th ed.)*. Addison-Wesley Professional.
- [20] <https://www.php.net/manual/en/>
- [21] <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-php/>
- [22] <https://www.php.net/manual/en/language.variables.superglobals.php>
- [23] Laravel Documentation. (2023). Introduction to Laravel. Διαθέσιμο στο: <https://laravel.com/docs>
- [24] Laravel Documentation. (2023). API Authentication. Διαθέσιμο στο: <https://laravel.com/docs/api-authentication>
- [25] Ecommerce with Laravel. (2022). Building Scalable Ecommerce Platforms with Laravel. Διαθέσιμο στο: <https://laravel-ecommerce.com>
- [26] McFarlin, J. (2021). Laravel 8 from Scratch. Laracasts. Διαθέσιμο στο: <https://laracasts.com/series/laravel-8-from-scratch>
- [27] Laravel Documentation. (2023). Laravel Packages & Tools. Διαθέσιμο στο: <https://laravel.com/docs>
- [28] Laravel Documentation. (2023). Routing - Basic Routing. Διαθέσιμο στο: <https://laravel.com/docs/routing>
- [29] Laravel Documentation. (2023). Upgrade Guide. Διαθέσιμο στο: <https://laravel.com/docs/upgrade>
- [30] CodeSource.io. (2020). Disadvantages of Laravel You Need to Know. Διαθέσιμο στο: <https://codesource.io/>
- [31] <https://dev.mysql.com/doc/>
- [32] https://www.python.org/doc/essays/blurb/?external_link=true
- [33] <https://www.w3schools.com/python/>
- [34] <https://www.javascript.com/>
- [35] <https://laravel.com/docs/5.3/frontend>
- [36] <https://httpd.apache.org/>
- [37] https://www.tutorialspoint.com/raspberry_pi/index.htm

[38] https://www.aqmd.gov/docs/default-source/aq-spec/resources-page/plantower-pms5003-manual_v2-3.pdf

[39] https://wiki.dfrobot.com/DHT11_Temperature_and_Humidity_Sensor__SKU__DFR0067_

ΠΑΡΑΡΤΗΜΑ Α

Στο παράρτημα αυτό αναφέρονται τα βασικά κομμάτια του κώδικα που χρησιμοποιήθηκε.

dashboard

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dashboard</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
  <div class="container mt-5">
    <!-- Header -->
    @include('header')

    <!-- Κεντρικό περιεχόμενο -->
    <div class="mt-4">
      <h1>Τα Μέρη Σας</h1>
      <a href="{{ route('place.create') }}" class="btn btn-success mb-4">Προσθήκη Νέου
Μέρους</a>

      <!-- Εμφάνιση των places του χρήστη -->
      @if (count($places) > 0)
        <div class="list-group">
          @foreach ($places as $place)
            <div class="list-group-item d-flex justify-content-between align-items-
center">
              <div>
                <h5 class="mb-1">{{ $place->placename }}</h5>
                <p class="mb-1">Κωδικός Χρήστη: {{ $place->userid }}</p>
              </div>
              <div>
                <a href="{{ route('place.edit', $place->id) }}" class="btn btn-
warning">Επεξεργασία</a>
                <a href="{{ route('place.show', $place->id) }}" class="btn btn-
primary">Προβολή Αισθητήρων Μαζί</a>
                <a href="{{ route('place.show2', $place->id) }}" class="btn btn-
primary">Προβολή Αισθητήρων</a>
                <a href="{{ route('place.show3', $place->id) }}" class="btn btn-
primary">Δείκτης</a>
              </div>
            </div>
          @endforeach
        </div>
      @else
        <p>Δεν έχετε προσθέσει μέρη ακόμη.</p>
      @endif
    </div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

View - μαζί

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Προβολή Δεδομένων Αισθητήρων</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
<div class="container mt-5">
<h1>Δεδομένα Αισθητήρων</h1>

<!-- Εμφάνιση του chart -->
<div class="mt-4">
<canvas id="sensorDataChart"></canvas>
</div>

<!-- Κουμπί επιστροφής -->
<a href="{{ route('dashboard') }}" class="btn btn-secondary mt-5">Επιστροφή στο Dashboard</a>
</div>

<script>
// Λήψη δεδομένων από το Laravel controller μέσω Blade directives
const sensorData = @json($sensors);

const labels = sensorData.map(item => item.created); // Λήψη των ημερομηνιών καταγραφής
const tempData = sensorData.map(item => item.temp); // Λήψη δεδομένων θερμοκρασίας
const humData = sensorData.map(item => item.hum); // Λήψη δεδομένων υγρασίας
const pm1Data = sensorData.map(item => item.pm1); // Λήψη δεδομένων PM1
const pm25Data = sensorData.map(item => item.pm2_5); // Λήψη δεδομένων PM2.5
const pm10Data = sensorData.map(item => item.pm10); // Λήψη δεδομένων PM10

// Δημιουργία του chart με Chart.js
const ctx = document.getElementById('sensorDataChart').getContext('2d');
const sensorChart = new Chart(ctx, {
type: 'line', // Τύπος διαγράμματος
data: {
labels: labels, // Ο χρόνος (created_at)
datasets: [
{
label: 'Θερμοκρασία (°C)',
data: tempData,
borderColor: 'rgba(255, 99, 132, 1)',
fill: false
},
{
label: 'Υγρασία (%)',
data: humData,

```

```

        borderColor: 'rgba(54, 162, 235, 1)',
        fill: false
    },
    {
        label: 'PM1 (µg/m³)',
        data: pm1Data,
        borderColor: 'rgba(75, 192, 192, 1)',
        fill: false
    },
    {
        label: 'PM2.5 (µg/m³)',
        data: pm25Data,
        borderColor: 'rgba(153, 102, 255, 1)',
        fill: false
    },
    {
        label: 'PM10 (µg/m³)',
        data: pm10Data,
        borderColor: 'rgba(255, 159, 64, 1)',
        fill: false
    }
]
},
options: {
    responsive: true,
    scales: {
        x: {
            title: {
                display: true,
                text: 'Χρόνος Καταγραφής'
            }
        },
        y: {
            title: {
                display: true,
                text: 'Τιμές Αισθητήρων'
            }
        }
    }
}
});
</script>

```

```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js"></script>
</body>

```

```
</html>
```

Δείκτης

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Κατάσταση Αισθητήρων και Δείκτης Καταλληλότητας</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
  <div class="container mt-5">
    <h1>Κατάσταση Αισθητήρων</h1>

    <!-- Πίνακας με τις τελευταίες τιμές των αισθητήρων -->
    <table class="table table-bordered mt-4">
      <thead>
        <tr>
          <th>Αισθητήρας</th>
          <th>Τελευταία Τιμή</th>
          <th>Μονάδα</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Θερμοκρασία</td>
          <td>{{ $latestSensorData->temp }} °C</td>
          <td>°C</td>
        </tr>
        <tr>
          <td>Υγρασία</td>
          <td>{{ $latestSensorData->hum }} %</td>
          <td>%</td>
        </tr>
        <tr>
          <td>PM1</td>
          <td>{{ $latestSensorData->pm1 }} μg/m³</td>
          <td>μg/m³</td>
        </tr>
        <tr>
```

```

        <td>PM2.5</td>

        <td>{{ $latestSensorData->pm2_5 }} μg/m³</td>

        <td>μg/m³</td>
    </tr>
    <tr>
        <td>PM10</td>

        <td>{{ $latestSensorData->pm10 }} μg/m³</td>

        <td>μg/m³</td>
    </tr>
</tbody>
</table>

<!-- Δείκτης καταλληλότητας -->
<div class="mt-5">
    <h3>Δείκτης Καταλληλότητας</h3>
    <div class="alert alert-{{ $suitabilityIndex['status'] == 'Κατάλληλη' ? 'success' : 'danger'
    }}" role="alert">
        Η τρέχουσα κατάσταση: <strong>{{ $suitabilityIndex['status'] }}</strong><br>
        Σχόλιο: {{ $suitabilityIndex['comment'] }}
    </div>
</div>

<!-- Εξήγηση των ορίων και του δείκτη -->
<div class="mt-5">
    <h4>Όρια Τιμών και Υπολογισμός Δείκτη Καταλληλότητας</h4>
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>Αισθητήρας</th>
                <th>Κατάλληλες Τιμές</th>
                <th>Μη Κατάλληλες Τιμές</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>Θερμοκρασία</td>
                <td>18 - 24 °C</td>
                <td>Κάτω από 18 °C ή πάνω από 24 °C</td>
            </tr>
            <tr>
                <td>Υγρασία</td>
                <td>30% - 60%</td>
                <td>Κάτω από 30% ή πάνω από 60%</td>
            </tr>
            <tr>
                <td>PM1</td>

```

```

        <td>< 15 µg/m³</td>
        <td>Πάνω από 15 µg/m³</td>
    </tr>
    <tr>
        <td>PM2.5</td>
        <td>< 25 µg/m³</td>
        <td>Πάνω από 25 µg/m³</td>
    </tr>
    <tr>
        <td>PM10</td>
        <td>< 50 µg/m³</td>
        <td>Πάνω από 50 µg/m³</td>
    </tr>
</tbody>
</table>

    <p>0 δείκτης καταλληλότητας βασίζεται σε αυτές τις τιμές. Αν η θερμοκρασία, η υγρασία ή τα
    επίπεδα των σωματιδίων βρίσκονται εκτός των παραπάνω ορίων, η συνθήκη θεωρείται μη κατάλληλη για
    διαβίωση.</p>
</div>

    <!-- Κουμπί επιστροφής -->
    <a href="{{ route('dashboard') }}" class="btn btn-secondary mt-5">Επιστροφή στο Dashboard</a>
</div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

Και το αντίστοιχο του Controller

```

//Δείκτης
public function show3($placeId)
{
    // Ανάκτηση της τελευταίας εγγραφής από τους αισθητήρες
    $latestSensorData = DB::table('sensors')
        ->where('placeid', $placeId)
        ->orderBy('created', 'desc')
        ->first();

    // Υπολογισμός της καταλληλότητας των συνθηκών
    $suitabilityIndex = $this->calculateSuitability($latestSensorData);

    // Επιστροφή των δεδομένων στο view
    return view('view3', [

```

```

        'latestSensorData' => $latestSensorData,
        'suitabilityIndex' => $suitabilityIndex,
    ]));
}

private function calculateSuitability($sensorData)
{
    $status = 'Κατάλληλη';
    $comment = 'Οι συνθήκες είναι εντός των ορίων.';

    // Έλεγχος θερμοκρασίας
    if ($sensorData->temp < 18 || $sensorData->temp > 24) {
        $status = 'Μη Κατάλληλη';
        $comment = 'Η θερμοκρασία δεν είναι εντός των επιθυμητών ορίων.';
    }

    // Έλεγχος υγρασίας
    if ($sensorData->hum < 30 || $sensorData->hum > 60) {
        $status = 'Μη Κατάλληλη';
        $comment = 'Η υγρασία δεν είναι εντός των επιθυμητών ορίων.';
    }

    // Έλεγχος PM1
    if ($sensorData->pm1 > 15) {
        $status = 'Μη Κατάλληλη';
        $comment = 'Τα επίπεδα σωματιδίων PM1 είναι υψηλά.';
    }

    // Έλεγχος PM2.5
    if ($sensorData->pm2_5 > 25) {
        $status = 'Μη Κατάλληλη';
        $comment = 'Τα επίπεδα σωματιδίων PM2.5 είναι υψηλά.';
    }

    // Έλεγχος PM10
    if ($sensorData->pm10 > 50) {
        $status = 'Μη Κατάλληλη';
        $comment = 'Τα επίπεδα σωματιδίων PM10 είναι υψηλά.';
    }

    return [
        'status' => $status,
        'comment' => $comment
    ];
}

```