



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Τίτλος Πτυχιακής Εργασίας	Δημιουργία Ιστότοπου συλλογής, επεξεργασίας και προβολής δεδομένων Development of a website to collect, process and present data
Όνοματεπώνυμο Φοιτητή	Σπυροπούλου Κανέλλα
Πατρώνυμο	Αθανάσιος
Αριθμός Μητρώου	Π20179
Επιβλέπων	Δουληγέρης Χρήστος



Copyright ©

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.



Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες προς όλους όσους συνέβαλαν στην ολοκλήρωση της παρούσας εργασίας.

Πρώτα απ' όλα, ένα μεγάλο ευχαριστώ στον κ. Γκοτσιόπουλο και την κ. Σεραλίδου, οι οποίοι με στήριξαν σε όλη αυτή τη διαδρομή, προσφέροντάς μου τις πολύτιμες γνώσεις και την καθοδήγησή τους. Η βοήθεια και οι συμβουλές τους υπήρξαν ανεκτίμητες για την πορεία και την ολοκλήρωση του έργου αυτού.

Θα ήθελα, επίσης, να εκφράσω την ιδιαίτερη ευγνωμοσύνη μου στον επιβλέποντα καθηγητή μου, κ. Δουληγέρη.

Τέλος, ευχαριστώ θερμά όλους όσους στάθηκαν δίπλα μου με οποιονδήποτε τρόπο κατά τη διάρκεια αυτής της διαδικασίας.



Περίληψη

Η παρούσα εργασία ασχολείται με την ανάπτυξη ενός ιστοτόπου που βοηθά στη διαχείριση, ανάλυση και απεικόνιση δεδομένων αισθητήρων. Βασικός στόχος αυτού είναι να δώσει στους χρήστες ένα εύκολο εργαλείο για να λαμβάνουν, να ανεβάζουν, να οργανώνουν και να αναλύουν αρχεία δεδομένων. Οι χρήστες μπορούν να εγγράφονται στον ιστοτόπο, να συνδέονται στον ιστοτόπο, να ανεβάζουν αρχεία σε διάφορες μορφές (όπως .xls, .xlsx, .csv) και να τα διαχειρίζονται. Οι βασικές δυνατότητες που δίνονται στους χρήστες του ιστοτόπου περιλαμβάνουν δημιουργία συνδέσεων σε MQTT broker θέματα (topics) όπου θα δημοσιεύονται δεδομένα αισθητήρων, η παρακολούθηση των τιμών των αισθητήρων, η εξαγωγή τους σε αρχεία και η αποθήκευση τους, και η επιλογή των αρχείων δεδομένων για την δημιουργία διαγραμμάτων. Πολλές ήταν οι προκλήσεις που αντιμετωπίστηκαν κατά την δημιουργία του ιστοτόπου. Πολλές ξεπεράστηκαν και άλλες, όπως θέματα ασφάλειας δεδομένων και θέματα στην δημιουργία διαγραμμάτων επιλέχθηκαν ως μελλοντικά βήματα επέκτασης της λειτουργικότητας του ιστοτόπου.

Λέξεις κλειδιά: Ανάλυση δεδομένων, IoT, MQTT, αισθητήρες.



Abstract

This paper deals with the development of a website that helps in the management, analysis and visualization of sensor data. The main goal of this is to give users an easy tool to receive, upload, organize and analyze data files. Users can register on the site, log in to the site, upload files in various formats (such as .xls, .xlsx, .csv) and manage them. The main features given to the users of the website include establishing connections in MQTT broker topics where sensor data will be published, tracking sensor values, exporting them to files and saving them, and selecting data files for charting. Many were the challenges faced during the creation of the website. Many were overcome and others such as data security issues and charting issues were selected as future steps to expand the site's functionality.

Keywords: Data analysis, IoT, MQTT, sensors.



Περιεχόμενα

Περίληψη	4
Abstract.....	5
1. Εισαγωγή.....	7
2. Θεωρητικό Υπόβαθρο.....	8
2.1. Συλλογή Δεδομένων Αισθητήρων	8
2.2. Τεχνολογίες Δημιουργίας του ιστότοπου	9
2.2.1. Διεπαφή χρήστη – Frontend	9
2.2.2. Λειτουργίες ιστότοπου και υποστήριξη δημιουργίας περιεχομένου στην διεπαφή χρήστη – Backend	10
2.2.3. Σύστημα διαχείρισης βάσεων δεδομένων	11
2.2.4. Λογισμικό υποστήριξη της απαιτούμενης λειτουργικότητας κατά την ανάπτυξη του ιστότοπου.....	12
3. Ανάπτυξη του Ιστοτόπου	12
3.1. Περιγραφή Απαιτήσεων	12
3.2. Ανάλυση απαιτήσεων	13
3.3. Αρχιτεκτονική λειτουργίας του συστήματος	14
3.4. Σχεδίαση και Υλοποίηση της βάσης δεδομένων.....	16
3.5. Σχεδίαση και Περιγραφή λειτουργιών της Διεπαφής Χρήστη του Ιστοτόπου	17
3.5.1. Αρχική Σελίδα.....	18
3.5.2. Σελίδες Εγγραφής, Σύνδεσης και Αλλαγής κωδικού	18
3.5.3. Σελίδα Προβολής των Εγγεγραμμένων Αισθητήρων	21
3.5.4. Σελίδα Φόρμας Προσθήκης Αισθητήρα	22
3.5.5. Σελίδα Πίνακας Ελέγχου.....	23
3.5.6. Σελίδα Προβολής Αρχείων Χρήστη και η Σελίδα Ανέβασμα Αρχείων	25
3.5.7. Σελίδα Προβολής Δεδομένων και Στατιστικών Αρχείου Χρήστη & Σελίδα Προβολής Διαγράμματος Αρχείου.....	27
3.6. Υλοποίηση των Λειτουργιών του Ιστοτόπου	30
3.7. Λειτουργίες του Backend.....	37
4. Έλεγχος Εφαρμογής.....	41
5. Συζήτηση	43
5.1. Σύγκριση με άλλες λύσεις.....	43
5.2. Περιορισμοί της εφαρμογής	46
6. Συμπέρασμα.....	47
7. Βιβλιογραφία.....	49
8. Παράρτημα	51



Δημιουργία βάσης δεδομένων.....51

1. Εισαγωγή

Το αντικείμενο που πραγματεύεται η παρούσα πτυχιακή εργασία ανήκει κυρίως στον τομέα της Επιστήμης των Υπολογιστών, με ιδιαίτερη έμφαση στο Web Development και την Επεξεργασία Δεδομένων. Λόγω της αύξησης της χρήσης του IoT (Internet of Things) και της ανάγκης για εργαλεία παρακολούθησης και ανάλυσης των δεδομένων τους σε πραγματικό χρόνο, αποφασίστηκε η ανάπτυξη ενός ιστότοπου που θα συνδυάζει τη συλλογή δεδομένων αισθητήρων, την παρακολούθηση και την βασική επεξεργασία αυτών. Έτσι, ως σκοπός αυτής της εργασίας ορίστηκε η δημιουργία ενός ιστότοπου συλλογής, επεξεργασίας και εμφάνισης δεδομένων. Πιο συγκεκριμένα, ο σκοπός της εργασίας είναι η ανάπτυξη και η σχεδίαση ενός ιστότοπου, ο οποίος θα επιτρέπει στους χρήστες:

- να παρακολουθούν ζωντανά δεδομένα αισθητήρων σε ένα πίνακα ελέγχου
- και να έχουν τη δυνατότητα να τα αποθηκεύουν σε αρχεία δεδομένων,
- να εισάγουν εξωτερικά αρχεία δεδομένων αισθητήρων
- καθώς και να προβάλλουν τα αρχεία δεδομένων με τη χρήση πινάκων και
- διαγραμματικών απεικονίσεων καθώς και βασικών στατιστικών αυτών.

Ένας από τους βασικούς στόχους για την επίτευξη των παραπάνω ήταν η ανάπτυξη ενός περιβάλλοντος φιλικού προς τον χρήστη, που θα επιτρέπει την εύκολη χρήση του ιστότοπου. Επίσης, κρίθηκε αναγκαία η ανάπτυξη ενός μηχανισμού διαχείρισης των χρηστών του ιστότοπου καθώς και των δεδομένων τους. Για την επεξεργασία των δεδομένων αισθητήρων, δημιουργήθηκε ένας αλγόριθμος οργάνωσης, και ανάλυσης της πληροφορίας. Τέλος, έλαβε χώρα και έλεγχος της λειτουργίας του ιστότοπου αλλά όχι αξιολόγηση της λειτουργίας του συστήματος με βάση την ανατροφοδότηση των χρηστών, που αποτελεί έναν μελλοντικό στόχο για την εξασφάλιση της λειτουργικότητας, της αποτελεσματικότητας και την περαιτέρω βελτίωσή του ιστότοπου.



2. Θεωρητικό Υπόβαθρο

Για την ανάπτυξη, τον έλεγχο και την λειτουργία του προαναφερόμενου ιστότοπου, είναι απαραίτητη η εξοικείωση με τεχνολογίες φιλοξενίας και υποστήριξης λειτουργίας ιστοτόπων όπως και γνώσεις τεχνολογιών ανάπτυξης ιστοσελίδων. Επίσης η επιλογή του τρόπου συλλογής δεδομένων αισθητήρων καθόρισε και τρόπο υλοποίησης του με την επιλογή των κατάλληλων τεχνολογιών. Έτσι, μετά τον σχεδιασμό και τον καθορισμό των λειτουργικών απαιτήσεων και πριν την έναρξη δημιουργίας του ιστότοπου, η επιλογή των κατάλληλων τεχνολογιών αποτελούσε ένα πολύ σημαντικό στάδιο για την επίτευξη των στόχων του ιστότοπου. Παρακάτω ακολουθεί μια ανασκόπηση των τεχνολογιών που εξετάστηκαν και χρησιμοποιήθηκαν για την υλοποίηση του ιστότοπου.

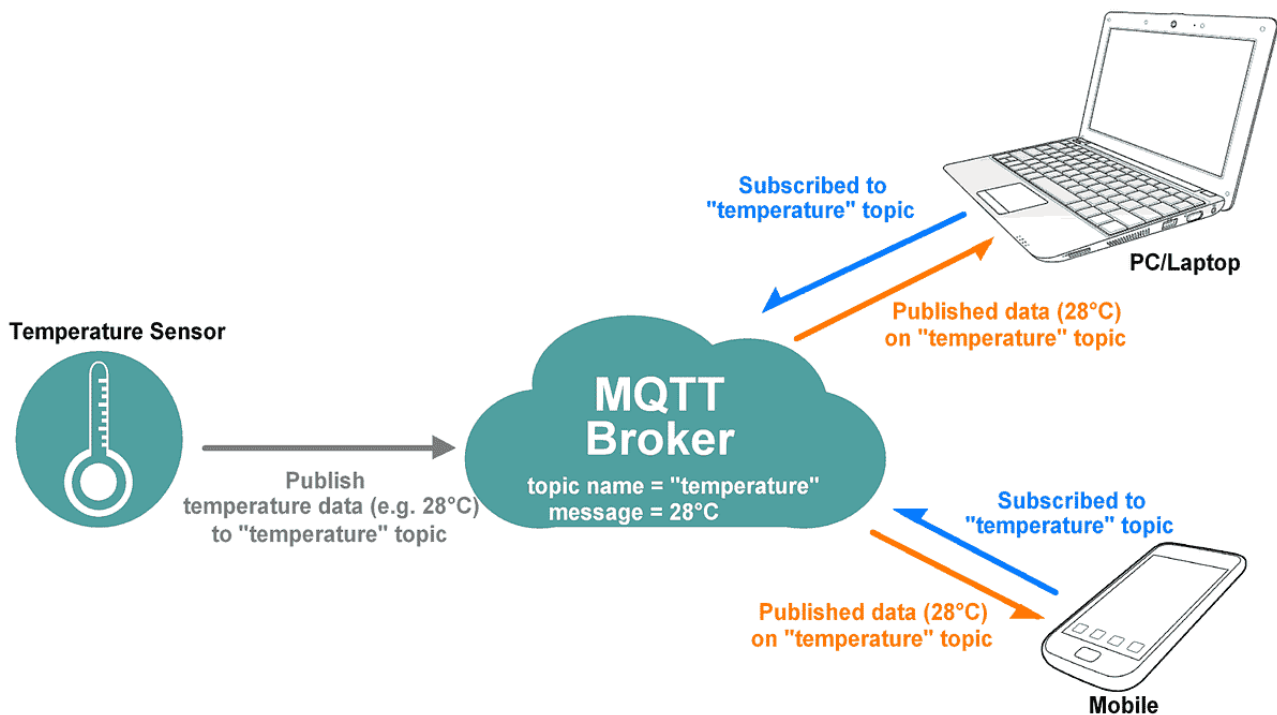
2.1. Συλλογή Δεδομένων Αισθητήρων

Το MQTT (MQ Telemetry Transport)^[1] είναι ένα ελαφρύ πρωτόκολλο μεταφοράς μηνυμάτων σχεδιασμένο για μικρές συσκευές και περιβάλλοντα μικρού εύρους ζώνης και υψηλής καθυστέρησης, όπως το Internet of Things (IoT). Πιο αναλυτικά, το MQTT έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι ελαφρύ και αποδοτικό, χρησιμοποιώντας ελάχιστους πόρους, γεγονός που το καθιστά κατάλληλο για περιορισμένα δίκτυα και συσκευές. Χαρακτηριστική λειτουργία του αποτελεί η διαδικασία δημοσίευσης/εγγραφής (Publish/Subscribe) σε θέματα (topics), η οποία επιτρέπει στους συνδρομητές (subscribers) να λαμβάνουν μηνύματα χωρίς να γνωρίζουν απαραίτητα τους εκδότες (publishers) (Εικόνα 1). Σχετικά με την Ποιότητα Υπηρεσίας (Quality of Service, QoS), ο MQTT υποστηρίζει τρεις βαθμούς για την αποστολή μηνυμάτων, αποστολή μηνύματος μια φορά χωρίς επιβεβαίωση, αποστολή μηνύματος μια τουλάχιστον φορά με επιβεβαίωση λήψης και αποστολή μηνύματος μια μόνο φορά με μια περίπλοκη διαδικασία επιβεβαίωσης. Το μέγεθος των MQTT μηνυμάτων είναι σχετικά μικρό (συνήθως 256 MB), γεγονός που σημαίνει ότι περιέχουν μόνο τα απαραίτητα δεδομένα. Τέλος, είναι σημαντικό να αναφερθεί ότι το MQTT υποστηρίζει διάφορες μεθόδους ασφάλειας, όπως την κρυπτογράφηση TLS/SSL.

Το MQTT είναι ευέλικτο, καθώς υποστηρίζει πολλές αρχιτεκτονικές, όπως το IoT και εφαρμογές cloud. Η κλιμακωτή λειτουργικότητά του είναι υποδειγματική για εφαρμογές που απαιτούν τη σύνδεση πολυάριθμων συσκευών. Έτσι, μπορεί να το συναντήσουμε στη διαχείριση έξυπνων συσκευών (Smart Home), στη σύνδεση μηχανών και συστημάτων σε βιομηχανικά περιβάλλοντα (Βιομηχανική Αυτοματοποίηση) και, τέλος, σε συσκευές παρακολούθησης και σε ιατρικά εργαλεία. Όπως προαναφέρθηκε, πολύ συχνά το συναντάμε στο IoT, καθώς είναι κατάλληλο για τη σύνδεση αισθητήρων, εφαρμογών και συσκευών. Ένα από τα πιο τυπικά σενάρια χρήσης του μπορεί να είναι η παρακολούθηση των καιρικών συνθηκών και των περιβαλλοντικών παραμέτρων με τη βοήθεια αισθητήρων. Πιο συγκεκριμένα, μέσω του MQTT publisher, ένας αισθητήρας υγρασίας, για παράδειγμα, μετρά την τρέχουσα υγρασία και στέλνει τα δεδομένα στο broker σε ένα θέμα (topic π.χ. Υγρασία). Στη συνέχεια, ο MQTT Broker λαμβάνει και διαχειρίζεται τα δεδομένα που αποστέλλονται στο συγκεκριμένο topic, μέσω του MQTT Subscriber, ενός μηχανισμού παρακολούθησης των δεδομένων, ώστε να μεταβιβάζονται οι πληροφορίες στην εφαρμογή



διαχείρισης αυτής της πληροφορίας.



Εικόνα 1. Παράδειγμα MQTT Broker
(από <https://medium.com/@jaydev.dave93/what-is-mqtt-protocol-c6a0cafffa8c>)

2.2. Τεχνολογίες Δημιουργίας του ιστότοπου

2.2.1. Διεπαφή χρήστη – Frontend

HTML/CSS: Η HTML ^[2] (Hypertext Markup Language), ή γλώσσα σήμανσης υπερκειμένου, είναι η τυπική γλώσσα σήμανσης για ιστοσελίδες που έχει σχεδιαστεί για την εμφάνιση περιεχομένου σε προγράμματα περιήγησης του ιστού. Καθορίζει το περιεχόμενο και τη δομή του περιεχομένου μιας ιστοσελίδας. Το CSS ^[3] (Cascading Style Sheets) είναι μια τρόπος για την περιγραφή της παρουσίασης ενός αρχείου που είναι σε μορφή γλώσσας σήμανσης. Είναι, δηλαδή, αυτό που καθορίζει πως εμφανίζονται στον επισκέπτη μιας σελίδας τα διάφορα στοιχεία της HTML. Προστέθηκε στην HTML 4.0 για να επιλύσει το πρόβλημα της μορφοποίησης των σελίδων, σώζοντας τους σχεδιαστές από πολύ κόπο και πολύ χρόνο, μειώνοντας σημαντικά τον όγκο της εργασίας.

JavaScript: Η JavaScript ^[4] η οποία συναντάται συχνά και ως JS, είναι μια δυναμική γλώσσα προγραμματισμού υψηλού επιπέδου που χρησιμοποιείται κυρίως για τη δημιουργία διαδραστικών διεπαφών web εφαρμογών χρήστη. Έχει στοιχεία scripting, functional και αντικειμενοστραφούς προγραμματισμού. Η JavaScript ανήκει στις γλώσσες προγραμματισμού οι οποίες μεταγλωττίζονται κατά την εκτέλεση και έχει φτιαχτεί για να λειτουργεί με το Document Object More (DOM), ένα πρότυπο που αλληλεπιδράσης με στοιχεία της HTML.



2.2.2. Λειτουργίες ιστότοπου και υποστήριξη δημιουργίας περιεχομένου στην διεπαφή χρήστη – BackEnd

PHP: Η PHP ^[5] είναι μια γλώσσα προγραμματισμού ανοιχτού κώδικα για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού, ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που είτε θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML ή θα επεξεργασθεί τις εισόδους δίχως να προβάλλει την έξοδο στο χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script. Είναι, δηλαδή, μια γλώσσα η οποία εκτελείται στον διακομιστή και το τελικό αποτέλεσμα που παράγει αποστέλλεται στον χρήστη ως HTML.

Python: Η Python ^[6] είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού (object-oriented) υψηλού επιπέδου, με ενσωματωμένες δομές δεδομένων και δυναμικές ιδιότητες. Αυτά την καθιστούν κατάλληλη για ταχύτατο χτίσιμο εφαρμογών, πόσο μάλλον αν ένας προγραμματιστής ψάχνει για μια εύκολη στη χρήση γλώσσα. Ακόμα, η φιλικότητα που έχει προς τον χρήστη, καθώς και η παροχή πολυάριθμων libraries επιτρέπει την εύκολη και ταχεία ανάπτυξη εφαρμογών.

Python Libraries

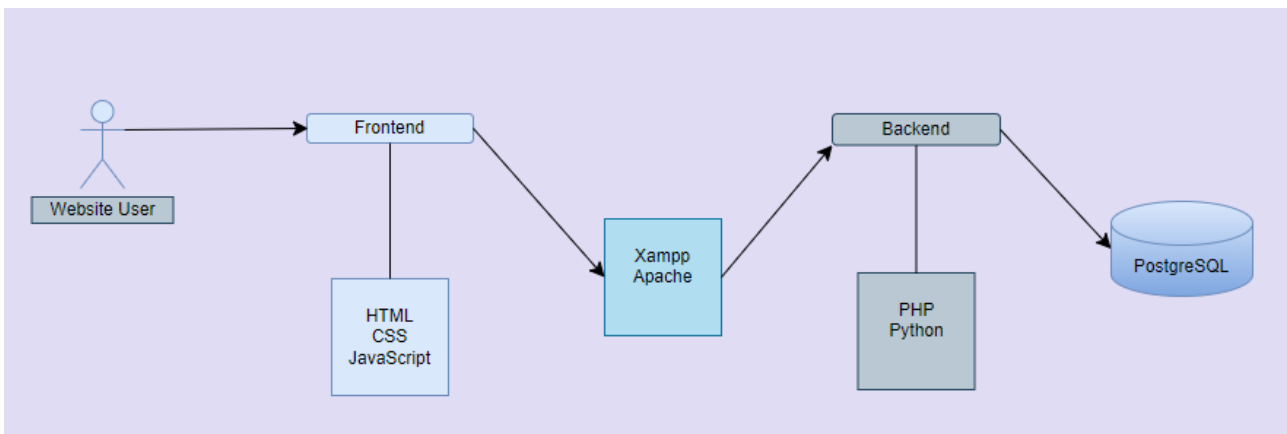
- **Flask:** Το Flask ^[7] είναι ένα microframework που βασίζεται στην Python για την ανάπτυξη εφαρμογών ιστού. Ταξινομείται ως microframework, καθότι δεν βασίζεται σε εξωτερικές βιβλιοθήκες για την εκτέλεση εργασιών. Επιπλέον, είναι ευέλικτο, ανεξάρτητο και γενικότερα προσφέρει περισσότερο έλεγχο στη διαδικασία ανάπτυξης. Εστιάζει στην απλότητα και το μινιμαλισμό και είναι κατάλληλο για single-page applications, καθώς και έρχεται με ένα ενσωματωμένο debugger που προσφέρει εικονικό debugging (εντοπισμό σφαλμάτων).
- **Flask-CORS:** Το Flask-CORS ^[8] (Cross-Origin Resource Sharing) είναι μια απλή επέκταση του Flask που επιτρέπει την υποστήριξη της ασφαλούς κοινής χρήσης και ανταλλαγής πόρων μεταξύ των διαφόρων μερών της εφαρμογής.
- **Matplotlib:** Το Matplotlib ^[9] είναι μια ολοκληρωμένη βιβλιοθήκη για τη δημιουργία στατικών, κινούμενων και διαδραστικών απεικονίσεων στην Python.
- **Pandas:** Το Pandas ^[10] είναι μια βιβλιοθήκη προγραμματισμού ανοιχτού κώδικα που προσφέρει στους προγραμματιστές που εργάζονται στην Python έναν πιο αποτελεσματικό τρόπο χειρισμού και ανάλυσης μεγάλου όγκου δεδομένων. Επίσης, λόγω του παραπάνω αυτή η βιβλιοθήκη χρησιμοποιείται και για την υποστήριξη διαδικασιών μηχανικής εκμάθησης, επιτρέποντάς την προετοιμασία των δεδομένων για την εκπαίδευση μοντέλων μηχανικής εκμάθησης.
- **NumPy:** Η βιβλιοθήκη NumPy ^[11] χρησιμοποιείται για επιστημονικούς υπολογισμούς. Είναι μια βιβλιοθήκη Python που παρέχει ένα πολυδιάστατο αντικείμενο πίνακα, μαζί με μια ποικιλία από ρουτίνες για γρήγορες λειτουργίες σε



πίνακες, συμπεριλαμβανομένων μαθηματικών, λογικών υπολογισμών όπως και πολλών άλλων.

- **Seaborn:** ^[12] Πρόκειται για ένα εργαλείο οπτικοποίησης δεδομένων, το οποίο βασίζεται στο Matplotlib. Σχεδιάστηκε για να διευκολύνει την παραγωγή όμορφων και πολύπλοκων διαγραμμάτων με λιγότερο κώδικα. Επικεντρώνεται στη στατιστική ανάλυση και οπτικοποίηση, καθιστώντας το ιδανικό για την παρουσίαση δεδομένων που περιέχουν στατιστικές σχέσεις.
- **Plotly:** Η Plotly ^[13] είναι μια διαδραστική ανοιχτού κώδικα βιβλιοθήκη δημιουργίας γραφημάτων. Προσφέρει δυνατότητες δημιουργίας γραφημάτων σε Python εφαρμογές και με την συνεργασία της βιβλιοθήκης JavaScript plotly.js την δημιουργία διαδραστικών οπτικοποιήσεων. Η βιβλιοθήκη διατίθεται με περισσότερους από 30 τύπους γραφημάτων, συμπεριλαμβανομένων επιστημονικών γραφημάτων, τρισδιάστατων γραφημάτων, στατιστικών διαγραμμάτων, οικονομικών διαγραμμάτων όπως και πολλά άλλα.
- **Paho:** Η Paho βιβλιοθήκη αποτελεί έργο της Eclipse Foundation και παρέχει ένα σύνολο βιβλιοθηκών και εργαλείων για την υποστήριξη των λειτουργιών του MQTT πρωτοκόλλου σε Python εφαρμογές.

Στη παρακάτω εικόνα (Εικόνα 2) φαίνεται η ροή της εφαρμογής, όπου ο χρήστης αλληλεπιδρά με το Frontend. Αυτό συνδέεται μέσω Xampp με το Backend. Αυτή η αρχιτεκτονική εξασφαλίζει την ομαλή λειτουργία και επικοινωνία μεταξύ όλων των τμημάτων της εφαρμογής.



Εικόνα 2. Χρήση τεχνολογιών για την επίτευξη της λειτουργικότητας του ιστότοπου

2.2.3. Σύστημα διαχείρισης βάσεων δεδομένων

PostgreSQL: Η PostgreSQL ^[14] είναι η πιο προχωρημένη open source σχεσιακή βάση δεδομένων. Υποστηρίζει τα χαρακτηριστικά ACID όπως και τα υπόλοιπα RDBMS με τα δεδομένα να είναι συσχετισμένα σε πίνακες αλλά και NoSQL (non-relational) με τα



δεδομένα να είναι σε αρχεία με δομή όπως τα JSON αρχεία. Χρησιμοποιείται ευρέως σε εφαρμογές οι οποίες χρειάζονται μεγάλη ευελιξία, επεκτασιμότητα και ασφάλεια.

2.2.4. Λογισμικό υποστήριξη της απαιτούμενης λειτουργικότητας κατά την ανάπτυξη του ιστότοπου

Xampp: Το Xampp ^[15] είναι ένα δημοφιλές πακέτο λογισμικού το οποίο επιτρέπει την εύκολη εγκατάσταση σε τοπικό διακομιστή, υπηρεσιών όπως του Apache HTTP διακομιστή και της MySQL καθιστώντας έτσι δυνατή την ανάπτυξη ιστοσελίδων και διαδικτυακών εφαρμογών. Το Xampp επιτρέπει τη εργασία σε ιστοσελίδες και εφαρμογές τοπικά, χωρίς να χρειάζεται η πρόσβαση σε διακομιστή στο διαδίκτυο. Επίσης, παρέχει ένα απλό περιβάλλον ελέγχου των διαφόρων υπηρεσιών που προσφέρει.

3. Ανάπτυξη του Ιστοτόπου

3.1. Περιγραφή Απαιτήσεων

Ο ιστότοπος θα πρέπει να σχεδιαστεί με τέτοιο τρόπο ώστε να προσφέρει ένα ενιαίο περιβάλλον εργασίας προσφέροντας ένα σύνολο λειτουργιών στον χρήστη.
Επιγραμματικά:

- Η πλατφόρμα θα πρέπει να είναι φιλική προς τον χρήστη, προσφέροντας εύκολη πλοήγηση, ευχρηστία και συμβατότητα με διάφορες συσκευές και προγράμματα περιήγησης.
- Θα πρέπει να δίνεται η δυνατότητα εγγραφής και σύνδεσης των χρηστών.
 - Κατά τη δημιουργία ενός προσωπικού λογαριασμού, ο χρήστης θα αποκτά πρόσβαση μέσω μιας σελίδας σε έναν ιδιωτικό φάκελο, εκεί όπου θα αποθηκεύονται τα αρχεία που ανεβάζει.
 - Παράλληλα, θα υπάρχουν σελίδες διαχείρισης των αρχείων για κάθε χρήστη με επιλογές διαγραφής, προβολής και ανάλυσης των δεδομένων που περιέχουν.
 - Στη σελίδα ανάλυσης θα επιτρέπεται στους χρήστες, εκτός από τα δεδομένα των αρχείων, να προβάλουν στατιστικά στοιχεία σε μορφή πινάκων και να δημιουργούν προσαρμόσιμα γραφήματα,.
 - Ο ιστότοπος θα πρέπει να επιτρέπει τη σύνδεση με MQTT topics που δημοσιεύονται δεδομένα αισθητήρων, σε δημόσιους ή και σε ιδιωτικούς MQTT brokers, δίνοντας στον χρήστη τη δυνατότητα να προσθήκης και διαχείρισης των συνδέσεων αυτών μέσα από μια σελίδα ελέγχου. Σε μια σελίδα προβολής των ήδη εγγεγραμμένων συνδέσεων, θα μπορούν να βλέπουν πληροφορίες για την σύνδεση όπως και να διαγράφουν τις



συνδέσεις που δεν χρειάζονται. Στην ίδια σελίδα θα επιλέγουν ποιον ή ποιους από τους αισθητήρες θέλουν να απεικονίσουν διαγραμματικά στην σελίδα του πίνακα ελέγχου. Η απόκριση του συστήματος θα πρέπει να είναι άμεση, με ταχύτητα στη φόρτωση των αρχείων και στη δημιουργία γραφημάτων.

- Τέλος, οι χρήστες θα πρέπει να μπορούν να αποθηκεύουν αυτά τα γραφήματα που δημιουργούνται και να εξάγουν δεδομένα αισθητήρων σε αρχεία Excel για περαιτέρω χρήση.
- Για την υλοποίηση όλων των παραπάνω λειτουργιών, θα πρέπει να υπάρχει αποτελεσματική επικοινωνία με μια βάση δεδομένων. Σε αυτήν την βάση και στους αντίστοιχους πίνακες θα αποθηκεύονται όλες τις πληροφορίες σχετικά με το εκάστοτε χρήστη, τα αρχεία δεδομένων τους, και τυχόν παραμετροποιήσεις που έχουν λάβει χώρα από κάθε χρήστη.

3.2. Ανάλυση απαιτήσεων

Για τη δημιουργία του ιστότοπου, οι λειτουργικές απαιτήσεις περιλαμβάνουν:

1. Διαχείριση Χρηστών

- Εγγραφή και Σύνδεση: Ο χρήστης πρέπει να μπορεί να εγγραφεί με ένα απλό σύστημα καταχώρησης στοιχείων (όνομα, επίθετο, όνομα χρήστη, κωδικός, ημερομηνία γέννησης). Πρέπει να μπορεί επίσης να συνδέεται και να αποσυνδέεται από τον λογαριασμό του με ασφάλεια.
- Δημιουργία Προσωπικού Φακέλου: Κατά την εγγραφή, ο χρήστης θα πρέπει αυτόματα να αποκτά έναν μοναδικό προσωπικό φάκελο, στον οποίο θα αποθηκεύονται όλα τα αρχεία που ανεβάζει.

2. Σύνδεση και Διαχείριση Αισθητήρων

- Σύνδεση Αισθητήρων: Ο χρήστης θα πρέπει να έχει τη δυνατότητα να ορίζει στοιχεία σύνδεσης με θεματικές ενότητες, όπου δημοσιεύονται στοιχεία αισθητήρων, σε MQTT brokers της επιλογής του.
- Προβολή και Διαγραφή Αισθητήρων: Ο χρήστης θα βλέπει τη λίστα με όλες τις συνδέσεις που έχει προσθέσει και θα μπορεί να διαγράψει όποιες δεν χρειάζεται.

3. Πίνακας Ελέγχου και Οπτικοποίηση Δεδομένων

- Πίνακας Ελέγχου Αισθητήρων: Θα πρέπει να υπάρχει μια διεπαφή (Πίνακας Ελέγχου) όπου ο χρήστης θα παρακολουθεί γραφικά δεδομένα από τους αισθητήρες του. Ο χρήστης θα μπορεί να επιλέγει ποιους αισθητήρες θέλει να εμφανίζονται σε μορφή γραφημάτων στον Πίνακα Ελέγχου από τη Διαχείριση αισθητήρων.

4. Διαχείριση Αρχείων

- Ανέβασμα Αρχείων: Οι χρήστες θα πρέπει να μπορούν να ανεβάζουν αρχεία

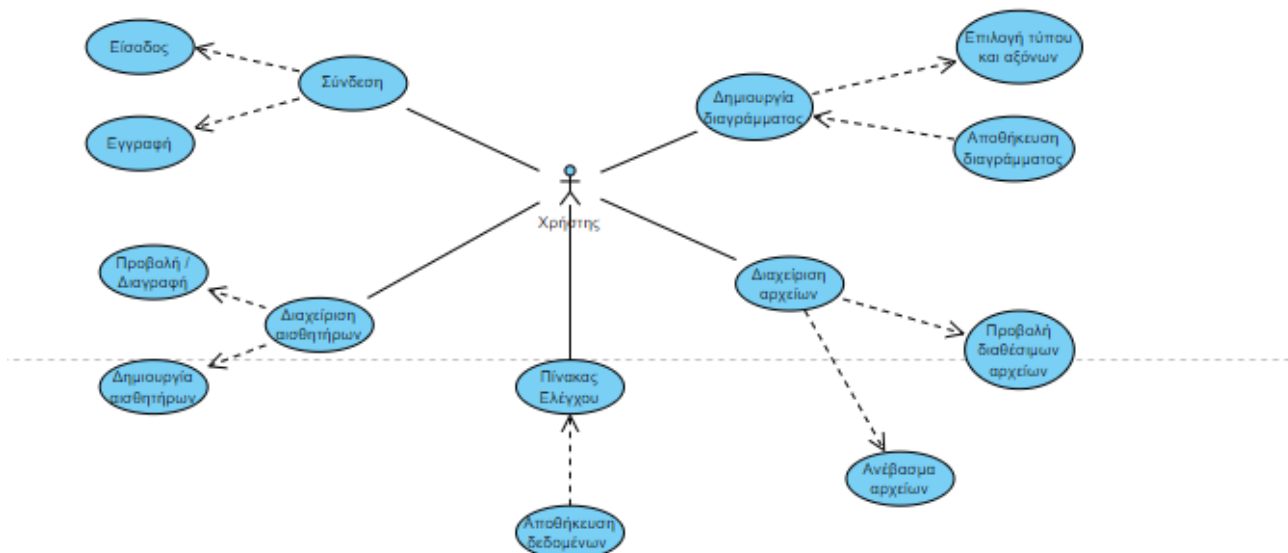


- Excel στον προσωπικό τους φάκελο μέσω μιας απλής διεπαφής.
- Προβολή και Διαγραφή Αρχείων: Ο χρήστης θα μπορεί να βλέπει όλα τα αρχεία που έχει ανεβάσει σε μια συγκεκριμένη σελίδα και να διαγράψει όποια δεν χρειάζεται.
- Ανάλυση Δεδομένων: Οι χρήστες πρέπει να μπορούν να ανοίξουν οποιοδήποτε αρχείο Excel για να δουν τα δεδομένα του σε διαγράμματα και πίνακες μαζί με κάποια βασικά στατιστικά στοιχεία.

5. Εξαγωγή και Αποθήκευση Δεδομένων

- Εξαγωγή Αρχείων Excel: Ο χρήστης πρέπει να μπορεί να εξάγει τα δεδομένα των αισθητήρων του σε αρχεία Excel και να τα κατεβάζει στη συσκευή του.
- Αποθήκευση Διαγραμμάτων: Τα γραφήματα που δημιουργεί ο χρήστης θα μπορούν να αποθηκεύονται ως εικόνες για κατέβασμα στη συσκευή του.

Αυτές οι λειτουργικές απαιτήσεις διασφαλίζουν ότι ο ιστότοπος θα παρέχει μια πλήρη εμπειρία χρήστη, δίνοντας τη δυνατότητα στους εμπλεκόμενους χρήστες να διαχειρίζονται αρχεία και δεδομένα με ασφάλεια, ευκολία και αποτελεσματικότητα (Εικόνα 3).



Εικόνα 3. Διάγραμμα περιπτώσεων χρήσης

3.3. Αρχιτεκτονική λειτουργίας του συστήματος

Η αρχιτεκτονική τριών επιπέδων (η οποία χρησιμοποιείται και στην παρούσα εργασία) ^[16] είναι μια αρχιτεκτονική εφαρμογών λογισμικού που οργανώνει τις εφαρμογές σε τρία λογικά επίπεδα: το επίπεδο παρουσίασης ή τη διεπαφή χρήστη, το επίπεδο εφαρμογής, όπου γίνεται η επεξεργασία των δεδομένων, και το επίπεδο δεδομένων, όπου



αποθηκεύονται και γίνεται η διαχείριση των δεδομένων της εφαρμογής (Εικόνα 4). Το κύριο πλεονέκτημα της αρχιτεκτονικής τριών επιπέδων είναι ότι επειδή κάθε επίπεδο λειτουργεί ανεξάρτητα, κάθε επίπεδο μπορεί να αναπτυχθεί ταυτόχρονα από μια ξεχωριστή ομάδα ανάπτυξης. Έτσι, μπορεί να ενημερωθεί ή και να κλιμακωθεί όπως απαιτείται χωρίς να επηρεάζονται τα άλλα επίπεδα. Για δεκαετίες η αρχιτεκτονική τριών επιπέδων ήταν η κυρίαρχη αρχιτεκτονική για τις εφαρμογές πελάτη-διακομιστή. Σήμερα, οι περισσότερες εφαρμογές τριών επιπέδων μεταβαίνουν σε πιο σύγχρονες τεχνολογίες εγγενείς στο υπολογιστικό νέφος (cloud), όπως δοχεία (containers) και μικροϋπηρεσίες (micro services) και για μετάβαση στο cloud.

Επίπεδο παρουσίασης – FrontEnd

Το επίπεδο παρουσίασης είναι η διεπαφή χρήστη και το επίπεδο επικοινωνίας της εφαρμογής, όπου ο τελικός χρήστης αλληλεπιδρά με την εφαρμογή. Ο κύριος σκοπός του είναι να εμφανίζει πληροφορίες και να συλλέγει πληροφορίες από τον χρήστη. Αυτό το επίπεδο ανώτατου επιπέδου μπορεί να εκτελεστεί σε ένα πρόγραμμα περιήγησης ιστού, ως εφαρμογή επιτραπέζιου υπολογιστή παρέχοντας μια γραφική διεπαφή χρήστη (GUI), για παράδειγμα. Τα επίπεδα παρουσίασης ιστού αναπτύσσονται με χρήση HTML, CSS και JavaScript.

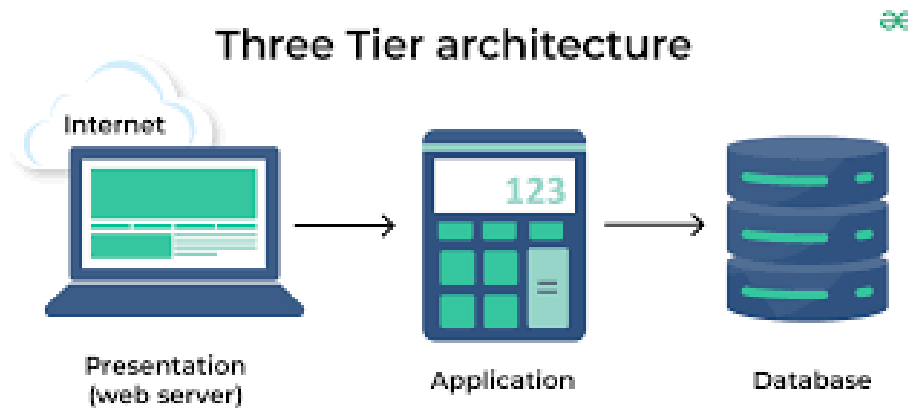
Επίπεδο εφαρμογής – Backend

Το επίπεδο εφαρμογής, γνωστό και ως λογικό επίπεδο ή μεσαίο επίπεδο, είναι η καρδιά της εφαρμογής. Σε αυτό το επίπεδο, οι πληροφορίες που συλλέγονται στο επίπεδο παρουσίασης υποβάλλονται σε επεξεργασία - μερικές φορές σε σχέση με άλλες πληροφορίες στο επίπεδο δεδομένων - χρησιμοποιώντας επιχειρηματική λογική, ένα συγκεκριμένο σύνολο επιχειρηματικών κανόνων. Το επίπεδο εφαρμογής μπορεί επίσης να προσθέσει, να διαγράψει ή να τροποποιήσει δεδομένα στο επίπεδο δεδομένων. Το επίπεδο εφαρμογής συνήθως αναπτύσσεται χρησιμοποιώντας Python, Java, Perl, PHP ή Ruby και επικοινωνεί με το επίπεδο δεδομένων χρησιμοποιώντας κλήσεις API.

Επίπεδο δεδομένων

Το επίπεδο δεδομένων, που μερικές φορές ονομάζεται επίπεδο βάσης δεδομένων, επίπεδο πρόσβασης δεδομένων, είναι το σημείο όπου αποθηκεύονται οι πληροφορίες που υποβάλλονται σε επεξεργασία από την εφαρμογή. Αυτό μπορεί να είναι ένα σύστημα διαχείρισης σχεσιακής βάσης δεδομένων όπως PostgreSQL ή σε διακομιστή βάσης δεδομένων NoSQL .

Σε μια εφαρμογή τριών επιπέδων, όλη η επικοινωνία περνά μέσα από το επίπεδο εφαρμογής. Το επίπεδο παρουσίασης και το επίπεδο δεδομένων δεν μπορούν να επικοινωνήσουν απευθείας μεταξύ τους.



Εικόνα 4. Αρχιτεκτονική τριών επιπέδων
(από <https://studentsxstudents.com/creating-a-three-tier-architecture-in-aws-681a691bf36e>)

3.4. Σχεδίαση και Υλοποίηση της βάσης δεδομένων

Για την δημιουργία της βάσης δεδομένων χρησιμοποιήθηκε η εφαρμογή PostgreSQL στη έκδοση 15.3. Η βάση δεδομένων ονομάστηκε AnalyticaWeb. Επίσης, δημιουργήθηκαν τρεις πίνακες για την διαχείριση και την αποθήκευση χρηστών και δεδομένων (Εικόνα 5).

- Ο πίνακας `signup` περιέχει πληροφορίες εγγραφής των χρηστών (όνομα, επίθετο, email, κωδικός και ημερομηνία γέννησης).
- Ο πίνακας `mqtt_subscriptions` καταγράφει τις συνδρομές των χρηστών σε διάφορα θέματα, εξασφαλίζοντας τη μοναδικότητα του συνδυασμού `username`, `topic`, `sensor_name`.
- Τέλος, στον πίνακα `mqtt_topics` αποθηκεύονται όλες οι τιμές που λαμβάνει κάθε αισθητήρας.

Αυτή η δομή επιτρέπει μια οργανωμένη και αποτελεσματική διαχείριση των πληροφοριών. Στο παράρτημα του παρόντος εγγράφου υπάρχουν παραδείγματα κώδικα για τη δημιουργία των τριών πινάκων.



signup		mqtt_subscriptions		mqtt_topics	
firstname	character	id	integer	id	integer
lastname	character	username	character	username	character
email	character	topic	character	topic	character
username	character	sensor_name	character	value	text
user_id	integer	user_id	integer	sensor_name	character
password	character	description	text	description	text
birthdate	date	unit	character	unit	character
		hostname	character	hostname	character
		port	integer	created_at	timestamp
		password	character		
		created_at	timestamp		

Εικόνα 5. Βάση Δεδομένων

3.5. Σχεδίαση και Περιγραφή λειτουργιών της Διεπαφής Χρήστη του Ιστότοπου

Στην ανάπτυξη του frontend, η διεπαφή χρήστη σχεδιάστηκε με γνώμονα τη βελτιωμένη εμπειρία του χρήστη και την ευκολία πλοήγησης. Οι σελίδες της εφαρμογής περιλαμβάνουν:

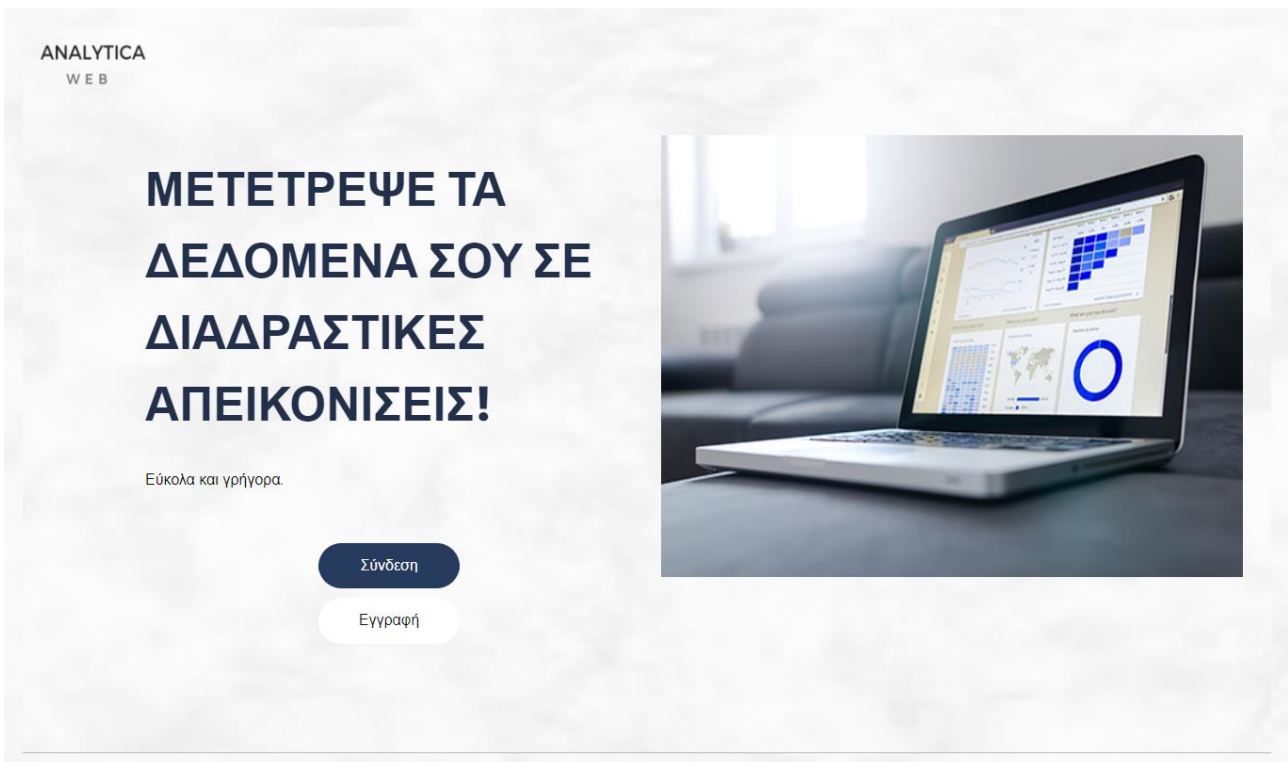
- την αρχική σελίδα,
- τις σελίδες εγγραφής, σύνδεσης και αλλαγής κωδικού, αρχική σελίδα χρήστη,
- τη σελίδα προβολής των εγγεγραμμένων αισθητήρων,
- τη σελίδα της φόρμας προσθήκης αισθητήρα,
- τη σελίδα πίνακα ελέγχου,
- τη σελίδα προβολής αρχείων χρήστη, τη σελίδα ανέβασμα αρχείων,
- τη σελίδα προβολής δεδομένων και στατιστικών κάθε αρχείου και τη σελίδα προβολής του τελικού διαγράμματος.

Κάθε σελίδα είναι σχεδιασμένη με HTML και CSS για να διασφαλίσει μια λειτουργική και καλαίσθητη διάταξη, ενώ η χρήση JavaScript επιτρέπει την αλληλεπίδραση με τον χρήστη και την δυναμική εμφάνιση δεδομένων. Με τη βοήθεια της βιβλιοθήκης Chart.js, δημιουργούνται διαγράμματα που απεικονίζουν τα δεδομένα των αισθητήρων σε πραγματικό χρόνο, διευκολύνοντας την κατανόηση και ανάλυση τους. Όλες οι σελίδες είναι σχεδιασμένες με responsive design, με σκοπό την άρτια οπτικοποίηση σε κάθε μορφής συσκευή. Η ανάπτυξη του FrontEnd διασφαλίζει την ομαλή αλληλεπίδραση των χρηστών με την εφαρμογή και την αποτελεσματική παρουσίαση των δεδομένων.



3.5.1. Αρχική Σελίδα

Όταν ο χρήστης εισέρχεται στον ιστότοπο, εμφανίζεται η αρχική σελίδα (Εικόνα 6), η οποία παρέχει μια συνοπτική επισκόπηση των βασικών λειτουργιών του συστήματος. Στην κορυφή της σελίδας, όπως και σε όλες τις άλλες σελίδες του ιστότοπου, βρίσκεται το λογότυπο, το οποίο λειτουργεί ως σημείο αναφοράς για την επωνυμία του ιστότοπου. Επίσης, στην αρχική σελίδα, υπάρχουν ευδιάκριτα κουμπιά για σύνδεση και εγγραφή, τα οποία επιτρέπουν στους χρήστες να αποκτήσουν πρόσβαση στις προσωπικές τους σελίδες ή να δημιουργήσουν νέους λογαριασμούς, αντίστοιχα. Η διαδικασία σύνδεσης χρήστη εξασφαλίζει ότι οι χρήστες της εφαρμογής μπορούν εύκολα να πλοηγηθούν και να επωφεληθούν από τις λειτουργίες που προσφέρει ο ιστότοπος.



Εικόνα 6. Αρχική σελίδα εφαρμογής

3.5.2. Σελίδες Εγγραφής, Σύνδεσης και Αλλαγής κωδικού

Κατά τη διαδικασία εγγραφής, ο χρήστης καλείται να συμπληρώσει όλα τα απαραίτητα πεδία για τη δημιουργία ενός νέου λογαριασμού (Εικόνα 8). Εάν το επιλεγμένο username είναι ήδη σε χρήση ή αν οι κωδικοί πρόσβασης δεν συμφωνούν με την επιβεβαίωση του κωδικού, το σύστημα εμφανίζει κατάλληλα μηνύματα σφάλματος, διευκολύνοντας την επίλυση των προβλημάτων (Εικόνα 9). Στην περίπτωση που το username είναι διαθέσιμο και οι κωδικοί ταιριάζουν, η εγγραφή προχωρά. Επιπλέον, παρέχεται η δυνατότητα ανακατεύθυνσης του χρήστη στη σελίδα σύνδεσης, αν το επιθυμεί.

Για την ασφάλεια των χρηστών, ο κωδικός πρόσβασης κρυπτογραφείται κατά την εγγραφή, εξασφαλίζοντας ότι οι ευαίσθητες πληροφορίες παραμένουν προστατευμένες (Εικόνα 7). Με την ολοκλήρωση της εγγραφής και εφόσον όλα τα στοιχεία έχουν εισαχθεί

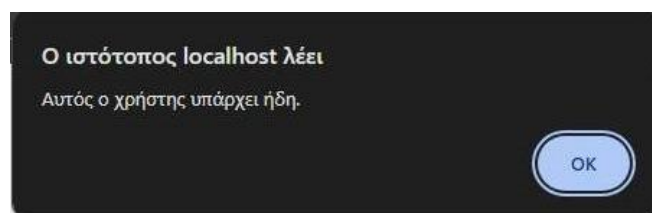


σωστά, ο χρήστης μεταφέρεται αυτόματα στη «Αρχική Σελίδα Χρήστη», όπου μπορεί να ξεκινήσει τη χρήση των επιπλέον λειτουργιών του ιστότοπου.

```
$hashed_password = password_hash($password, PASSWORD_DEFAULT);  
$hashed_confirm_password = password_hash($confirm_password, PASSWORD_DEFAULT);  
  
// Δημιουργία του φακέλου με το όνομα του χρήστη  
$uploadDir = "uploads/";  
$userDir = $uploadDir . $username;
```

Εικόνα 7. Παράδειγμα κώδικα (signup.php)

Εικόνα 8. Σελίδα εγγραφής



Εικόνα 9. Μήνυμα ειδοποίησης



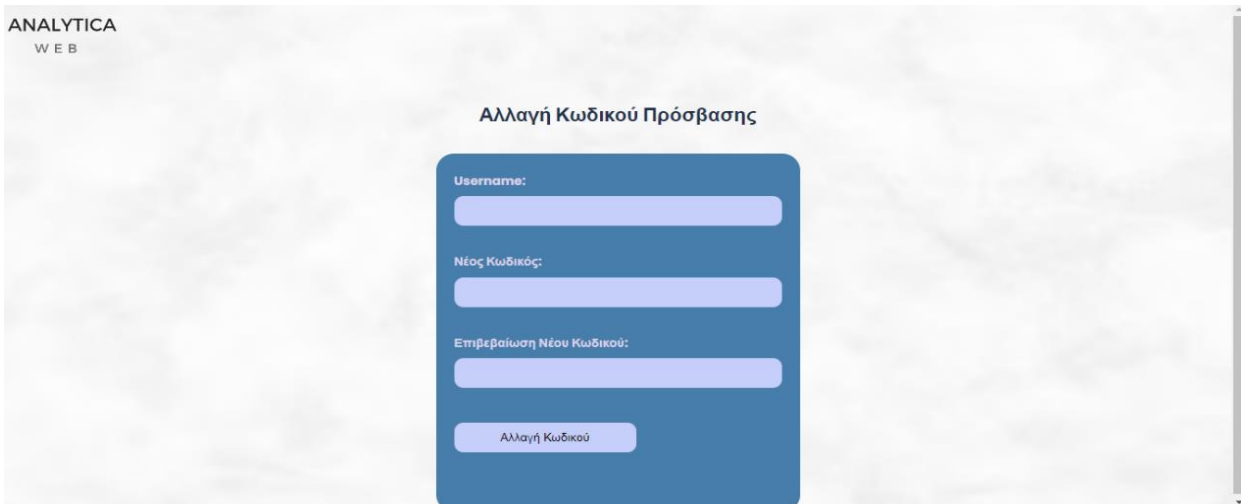
Κατά τη διαδικασία σύνδεσης, ο χρήστης πρέπει να εισαγάγει το όνομα χρήστη (username) και τον κωδικό πρόσβασης (Εικόνα 10). Εάν κάποιο από τα στοιχεία που εισάγει είναι εσφαλμένο, το σύστημα εμφανίζει ένα ενημερωτικό μήνυμα, ενημερώνοντάς τον για την ανάγκη διόρθωσης των δεδομένων.

Επιπλέον, παρέχεται η δυνατότητα ανακατεύθυνσης του χρήστη προς τη σελίδα εγγραφής για νέους χρήστες ή προς τη σελίδα αλλαγής κωδικού πρόσβασης σε περίπτωση που έχει ξεχάσει τον κωδικό του.

Μόλις ο χρήστης συμπληρώσει σωστά όλα τα απαιτούμενα πεδία και πατήσει το κουμπί «Σύνδεση», μεταφέρεται αυτόματα στη σελίδα «Αρχική Σελίδα Χρήστη», όπου μπορεί να ενημερωθεί για τις επόμενες ενέργειες του λογαριασμού του.

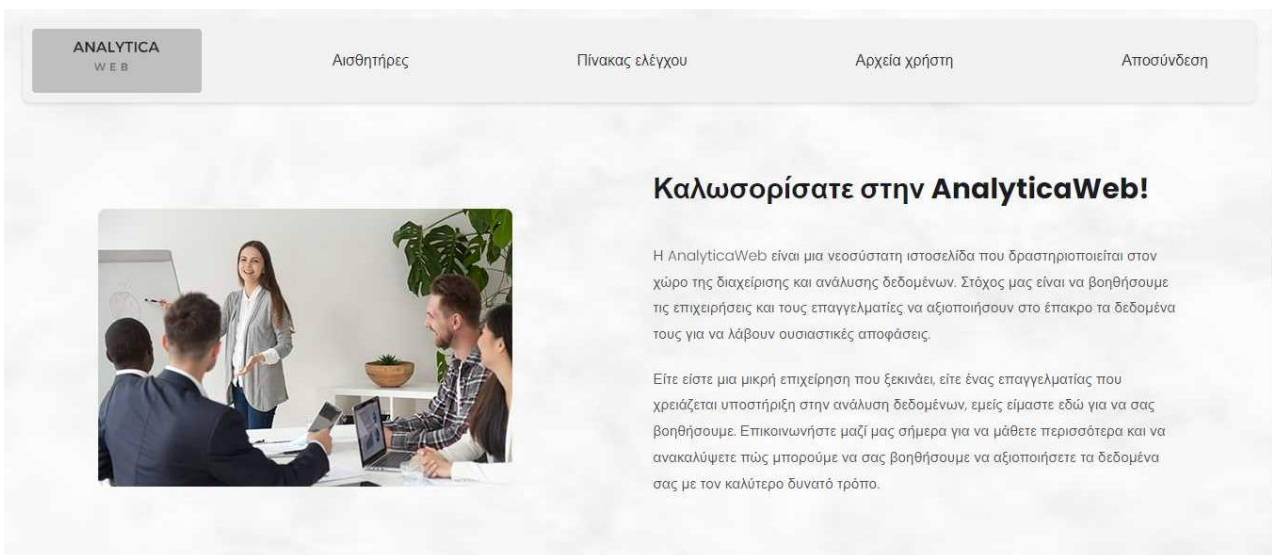
Εικόνα 10. Σελίδα Σύνδεσης

Στη σελίδα αλλαγής κωδικού πρόσβασης, ο χρήστης πρέπει να εισαγάγει το όνομα χρήστη (username) που χρησιμοποίησε κατά την εγγραφή του (Εικόνα 11). Εάν ο της κωδικός πρόσβασης δεν ταιριάζει με την επιβεβαίωση του κωδικού, το σύστημα εμφανίζει το αντίστοιχο μήνυμα σφάλματος, ενημερώνοντας τον χρήστη για την ανάγκη διορθώσεων. Με την ολοκλήρωση της διαδικασίας και αφού ο χρήστης εισάγει σωστά όλα τα απαιτούμενα στοιχεία, πατώντας το κουμπί «Αλλαγή Κωδικού», η αλλαγή κωδικού ολοκληρώνεται και ο χρήστης ανακατευθύνεται στη σελίδα «Σύνδεση». Εκεί μπορεί να εισέλθει με τα νέα του στοιχεία και να συνεχίσει τη χρήση του ιστότοπου.



Εικόνα 11. Σελίδα Αλλαγής Κωδικού Πρόσβασης

Μετά την επιτυχή ολοκλήρωση της εγγραφής ή της σύνδεσης, ο χρήστης ανακατευθύνεται στη «Αρχική Σελίδα Χρήστη» (Εικόνα 12). Σε αυτή τη σελίδα παρέχονται πληροφορίες σχετικά με το τις λειτουργίες του ιστοτόπου.



Εικόνα 12. Αρχική σελίδα χρήστη

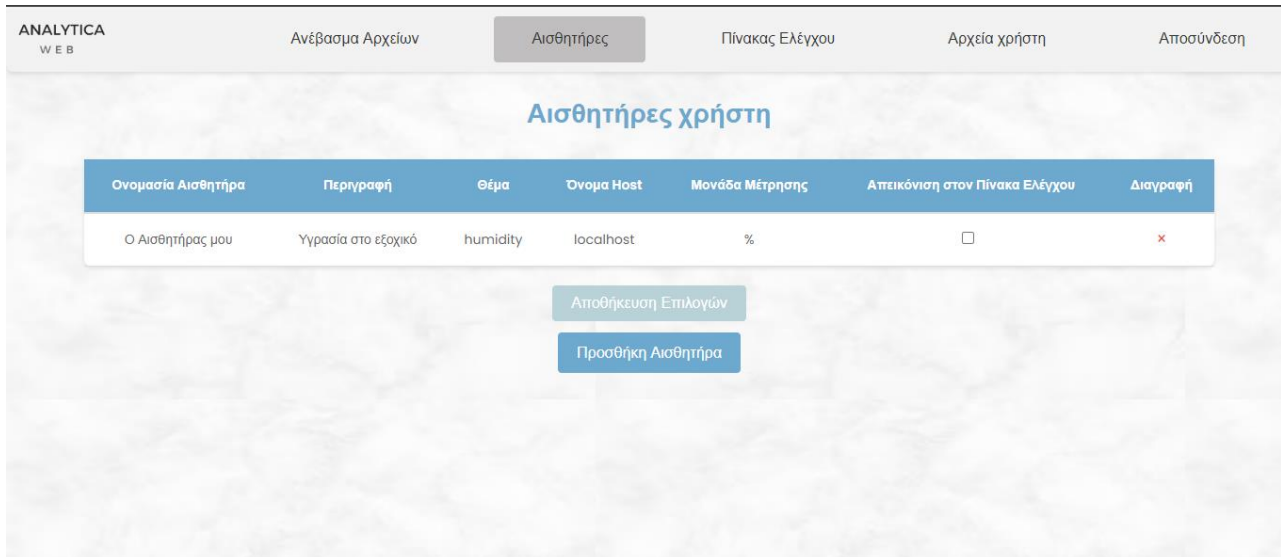
3.5.3. Σελίδα Προβολής των Εγγεγραμμένων Αισθητήρων

Στη σελίδα «Αισθητήρες», ο χρήστης βλέπει μια λίστα με όλους τους αισθητήρες που έχει δημιουργήσει (Εικόνα 13). Η σελίδα αυτή παρέχει λεπτομερείς πληροφορίες για κάθε αισθητήρα, όπως τα χαρακτηριστικά του.

Στη στήλη «Απεικόνιση στον Πίνακα Ελέγχου», υπάρχουν checkboxes που επιτρέπουν στον χρήστη να επιλέξει ποιους αισθητήρες επιθυμεί να εμφανίζονται στη σελίδα «Πίνακας Ελέγχου». Μετά την επιλογή των αισθητήρων, οι προτιμήσεις αποθηκεύονται με την επιλογή του κουμπιού «Αποθήκευση Επιλογών».



Για κάθε αισθητήρα, υπάρχει η δυνατότητα διαγραφής του μέσω του κουμπιού **x** που βρίσκεται στη τελευταία στήλη της λίστας. Αυτή η λειτουργία επιτρέπει την εύκολη απομάκρυνση των αισθητήρων που δεν χρειάζονται πια. Τέλος, το κουμπί «Προσθήκη Αισθητήρα» καθοδηγεί τον χρήστη στη σελίδα που επιτρέπει τη δημιουργία νέων αισθητήρων, ολοκληρώνοντας έτσι τη διαδικασία διαχείρισης και προσθήκης αισθητήρων.



Εικόνα 13. Σελίδα προβολής αισθητήρων χρήστη

3.5.4. Σελίδα Φόρμας Προσθήκης Αισθητήρα

Στη σελίδα δημιουργίας αισθητήρα, ο χρήστης έχει τη δυνατότητα να εισαγάγει τα επιθυμητά στοιχεία για τη δημιουργία ενός νέου αισθητήρα (Εικόνα 14). Η διαδικασία περιλαμβάνει τα εξής κουμπιά:

- «Προσθήκη»: Αυτό το κουμπί ολοκληρώνει τη διαδικασία δημιουργίας του νέου αισθητήρα, προσθέτοντάς τον στη λίστα των αισθητήρων του χρήστη με τα στοιχεία που έχουν καταχωρηθεί και οδηγώντας τον χρήστη στην σελίδα «Αισθητήρες».
- «Καθαρισμός»: Το κουμπί αυτό διαγράφει όλα τα δεδομένα που έχουν εισαχθεί στα πεδία της φόρμας, επιτρέποντας στον χρήστη να ξεκινήσει ξανά την εισαγωγή στοιχείων, αν χρειάζεται.
- «Ακύρωση»: Με την επιλογή αυτού του κουμπιού, ο χρήστης επιστρέφει στη σελίδα προβολής των αισθητήρων, χωρίς να κάνει αποθήκευση ή προσθήκη νέου αισθητήρα.

Αυτές οι επιλογές παρέχουν στον χρήστη ευελιξία κατά τη διαδικασία δημιουργίας και διαχείρισης αισθητήρων.



ANALYTICA WEB

Αισθητήρες Πίνακας ελέγχου Αρχεία χρήστη Αποσύνδεση

Προσθήκη Στοιχείων Αισθητήρα Χρήστη

Όνομασία Αισθητήρα:	<input type="text" value="Εισαγάγετε το όνομα του αισθητήρα"/>
Περιγραφή:	<input type="text" value="Εισαγάγετε περιγραφή"/>
Όνομα Host/Διεύθυνση IP:	<input type="text" value="test.mosquitto.org"/>
Αριθμός Θύρας:	<input type="text" value="8081"/>
Όνομα Χρήστη:	<input type="text" value="Εισαγάγετε το όνομα χρήστη"/>
Κωδικός Πρόσβασης:	<input type="text" value="Εισαγάγετε τον κωδικό πρόσβασης"/>
Θέμα:	<input type="text" value="Εισαγάγετε το θέμα"/>
Μονάδα Μέτρησης:	<input type="text" value="Εισαγάγετε τη μονάδα μέτρησης"/>

Εικόνα 14. Σελίδα προσθήκης αισθητήρα

3.5.5. Σελίδα Πίνακας Ελέγχου

Στη σελίδα «Πίνακας Ελέγχου», τα δεδομένα των αισθητήρων των συνδέσεων που έχει επιλέξει ο χρήστης μέσω των check boxes στη σελίδα «Αισθητήρες» απεικονίζονται διαγραμματικά (Εικόνα 15). Κάθε διάγραμμα περιλαμβάνει τις ακόλουθες πληροφορίες:

- Στην κορυφή του διαγράμματος, εμφανίζεται η ονομασία του αισθητήρα σε κόκκινο χρώμα.
- Κάτω από την ονομασία, εμφανίζεται σε γαλάζιο χρώμα το θέμα του διαγράμματος.
- Η περιγραφή του διαγράμματος εμφανίζεται σε λιλά χρώμα, παρέχοντας επιπλέον πληροφορίες και εξηγήσεις για τα δεδομένα που απεικονίζονται.
- Δίπλα από τις πληροφορίες εμφανίζονται η ημερομηνία κατά την οποία έχουν ληφθεί οι τιμές, καθώς επίσης και η τρέχουσα τιμή του αισθητήρα.
- Τέλος, υπάρχει το κουμπί «Εξαγωγή Δεδομένων», το οποίο επιτρέπει στον χρήστη να αποθηκεύσει τα δεδομένα του συγκεκριμένου διαγράμματος σε ένα αρχείο με κατάληξη .xlsx στον τοπικό δίσκο της συσκευής του (Εικόνα 16).



Εικόνα 15. Πίνακας ελέγχου



	A	B
1	Τιμή	Ημερομηνία και Ώρα
2	14	2024-09-29 22:28:08
3	14	2024-09-29 22:30:08
4	32	2024-09-29 22:32:08
5	29	2024-09-29 22:34:08
6	30	2024-09-29 22:34:47
7	12	2024-09-29 22:34:49
8	31	2024-09-29 22:34:50
9	24	2024-09-29 22:37:10
10	5	2024-09-29 22:37:15
11	19	2024-09-29 22:37:17
12	4	2024-09-29 22:39:17
13	35	2024-09-29 22:39:59
14	8	2024-09-29 22:40:01
15	2	2024-09-29 22:40:06
16	37	2024-09-29 22:40:26
17	37	2024-09-29 22:40:28
18	19	2024-09-29 22:40:31
19	30	2024-09-29 22:40:33
20	41	2024-09-29 22:41:02
21	13	2024-09-29 22:41:07
22	12	2024-09-29 22:43:07
23	42	2024-09-29 22:44:24
24	36	2024-09-29 22:46:25
25	11	2024-09-29 22:48:25
26	39	2024-09-29 22:50:26
27	7	2024-09-29 22:52:51
28	30	2024-09-29 22:54:25
29	33	2024-09-29 22:56:55
30	28	2024-09-30 12:43:51
31		

Εικόνα 16. Παράδειγμα αρχείου εξαγωγής

3.5.6. Σελίδα Προβολής Αρχείων Χρήστη και η Σελίδα Ανέβασμα Αρχείων

Στη σελίδα «Αρχεία Χρήστη» υπάρχει ένα κουμπί «Προσθήκη Αρχείου» το οποίο οδηγεί στη σελίδα «Ανέβασμα Αρχείων» την οποία θα δούμε στη συνέχεια (Εικόνα 17).

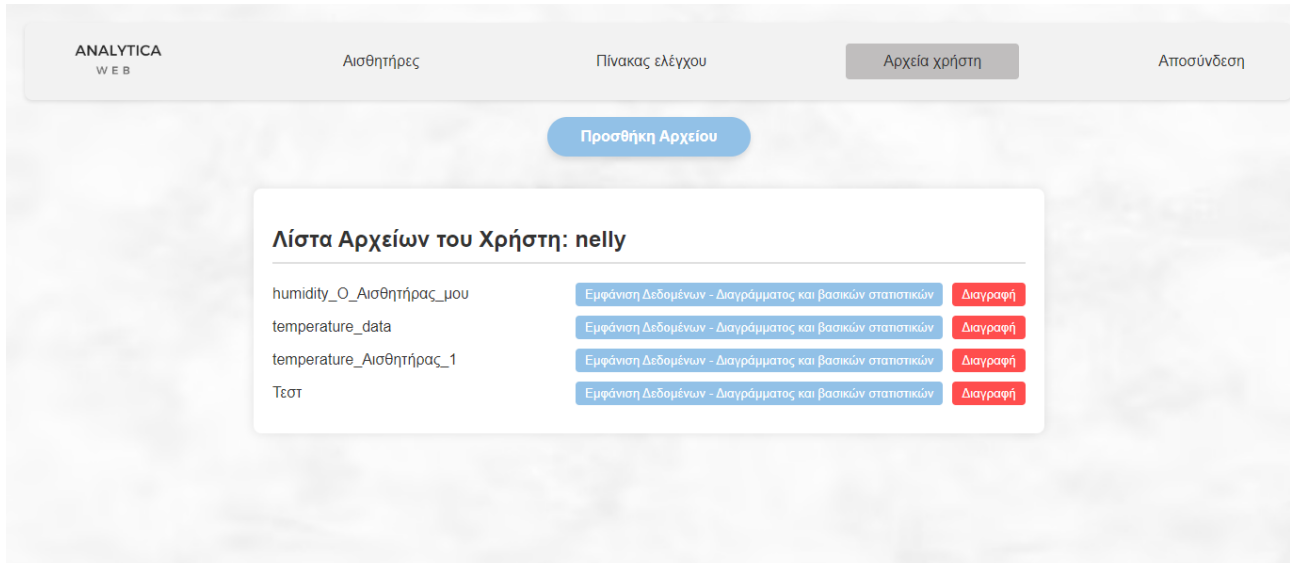
Εμφανίζονται, ακόμη, όλα τα αρχεία που έχει ανεβάσει ο χρήστης μέσω της σελίδας «Ανέβασμα Αρχείων». Για κάθε αρχείο στη λίστα, υπάρχουν δύο κουμπιά με τις εξής λειτουργίες:

- Το πρώτο κουμπί οδηγεί σε μια νέα σελίδα που εμφανίζει τα δεδομένα του επιλεγμένου αρχείου, επιτρέποντας στον χρήστη να δει λεπτομερώς το περιεχόμενο και τις σχετικές πληροφορίες.
- Το δεύτερο κουμπί είναι το κουμπί διαγραφής. Με την επιλογή αυτού του κουμπιού, το επιλεγμένο αρχείο διαγράφεται από τη λίστα και παύει να είναι διαθέσιμο.

Αυτές οι λειτουργίες εξασφαλίζουν ότι ο χρήστης μπορεί εύκολα να διαχειριστεί τα αρχεία



του, είτε προβάλλοντας τα δεδομένα τους είτε απομακρύνοντάς τα όταν δεν είναι πλέον απαραίτητα.

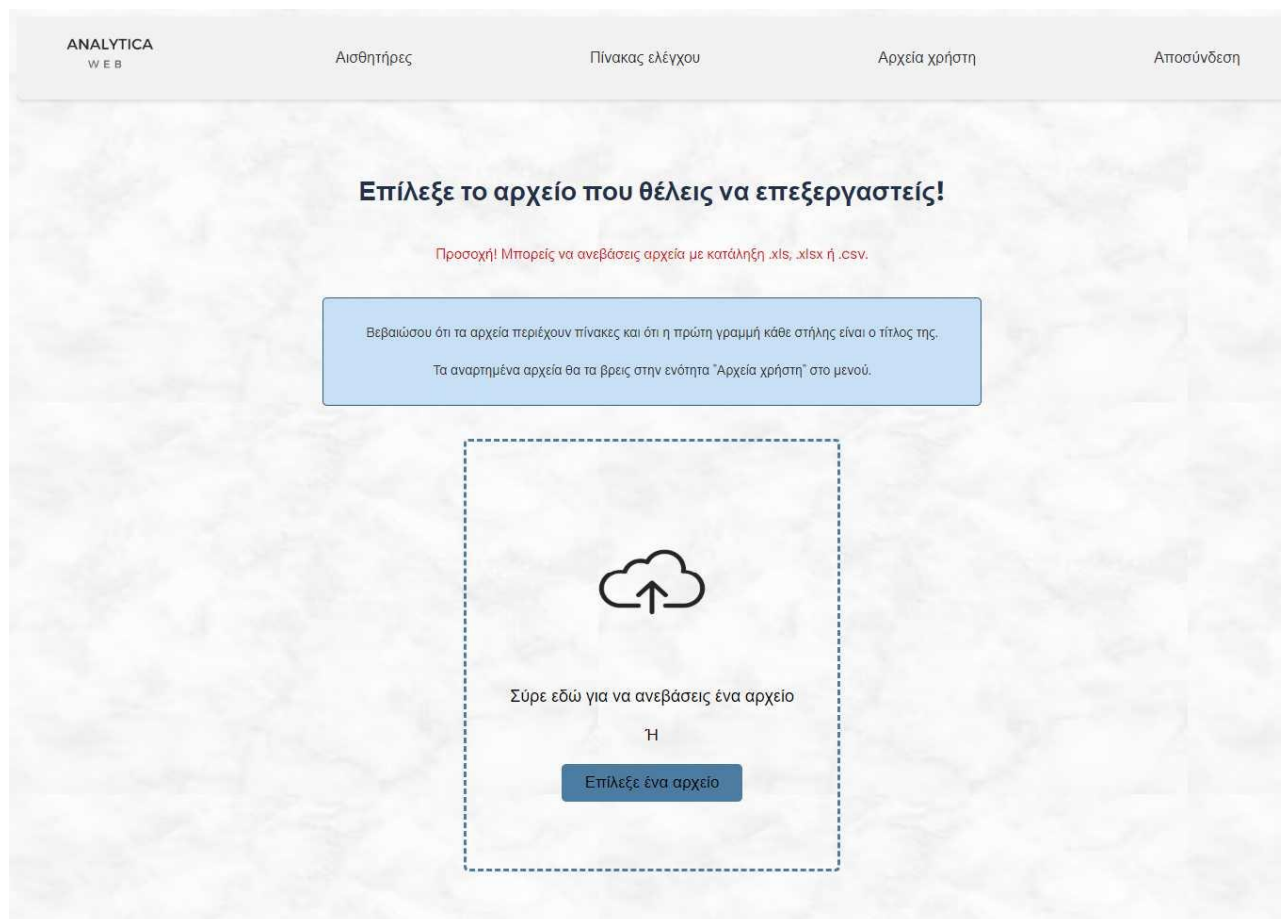


Εικόνα 17. Λίστα αρχείων χρήστη

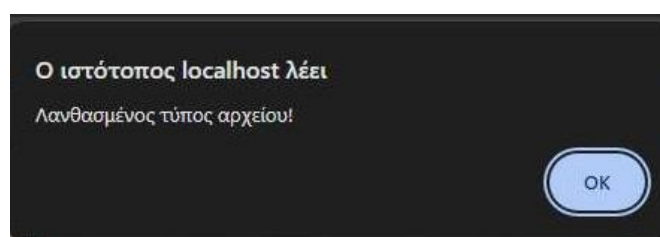
Πατώντας το κουμπί «Προσθήκη Αρχείου», εμφανίζεται η σελίδα «Ανέβασμα Αρχείων» στην οποία ο χρήστης έχει τη δυνατότητα να ανεβάσει αρχεία με καταλήξεις .xls, .xlsx ή .csv (Εικόνα 18). Υπάρχουν δύο επιλογές για το ανέβασμα των αρχείων:

- **Επιλογή Αρχείου:** Ο χρήστης μπορεί να επιλέξει ένα αρχείο από οποιονδήποτε φάκελο στη συσκευή του.
- **Drag & Drop:** Εναλλακτικά, ο χρήστης μπορεί να σύρει και να αποθέσει τα αρχεία στην καθορισμένη περιοχή της σελίδας για γρήγορη και εύκολη ανέβασμα.

Όλα τα αρχεία που ανεβάζει ο χρήστης αποθηκεύονται σε έναν ξεχωριστό φάκελο που δημιουργείται αυτόματα κατά την εγγραφή του χρήστη, με όνομα και το username του (Εικόνα 7). Αυτός ο φάκελος διασφαλίζει ότι τα αρχεία του κάθε χρήστη οργανώνονται και αποθηκεύονται με ασφάλεια, επιτρέποντας την εύκολη πρόσβαση και διαχείριση των αρχείων στο μέλλον. Είναι σημαντικό να σημειωθεί ότι, οποιαδήποτε από τις αποδεκτές μορφές και να έχει το αρχείο που αναρτά ο χρήστης, μετατρέπεται σε αρχείο .csv κατά την αποθήκευσή του στον φάκελο, για την καλύτερη επεξεργασία του. Σε περίπτωση επιλογής μη αποδεκτού τύπου αρχείου, εμφανίζεται ενημερωτικό μήνυμα.



Εικόνα 18. Σελίδα Ανέβασμα Αρχείων



Εικόνα 19. Μήνυμα ειδοποίησης

3.5.7. Σελίδα Προβολής Δεδομένων και Στατιστικών Αρχείου Χρήστη & Σελίδα Προβολής Διαγράμματος Αρχείου

Το κουμπί «Εμφάνιση Δεδομένων - Διαγράμματος και Βασικών Στατιστικών» της σελίδας με τη λίστα αρχείων οδηγεί τον χρήστη σε μια σελίδα όπου παρουσιάζεται μια αναλυτική απεικόνιση των δεδομένων του επιλεγμένου αρχείου (Εικόνα 20). Σε αυτή τη σελίδα, ο χρήστης θα δει:



- Έναν πρώτο πίνακα στον οποίο εμφανίζονται τα δεδομένα του επιλεγμένου αρχείου, παρέχοντας μια αναλυτική προβολή των καταχωρισμένων στοιχείων.
- Έναν δεύτερο πίνακα ο οποίος παρέχει βασικά στατιστικά στοιχεία για κάθε αριθμητική στήλη του αρχείου, όπως μέση, μέγιστη, ελάχιστη τιμή, εύρος και απόκλιση.

Ο χρήστης έχει τη δυνατότητα να:

- Επιλέξει ποια στήλη από τον πίνακα δεδομένων θα χρησιμοποιηθεί για τον άξονα X και ποια για τον άξονα Y του διαγράμματος.
- Επιλέξει τον τύπο του διαγράμματος που επιθυμεί, με διαθέσιμες επιλογές όπως γραμμικό, ράβδων, και διασποράς.

Επιπλέον, ο σύνδεσμος «Πίσω στη Λίστα Αρχείων» επιστρέφει τον χρήστη στη σελίδα «Αρχεία Χρήστη», επιτρέποντάς του να συνεχίσει τη διαχείριση των αρχείων του. Όταν ο χρήστης ολοκληρώσει τις επιλογές του και επιλέξει το κουμπί «Δημιουργία Διαγράμματος», μεταφέρεται στη σελίδα που θα παρουσιάσει το τελικό διάγραμμα, σύμφωνα με τις ρυθμίσεις και τις επιλογές που έχει κάνει.

The screenshot shows the ANALYTICA WEB interface. At the top, there are navigation tabs: Αισθητήρες, Πίνακας ελέγχου, Αρχεία χρήστη, and Αποσύνδεση. The main content area is divided into two panels: 'Δεδομένα Αρχείου' (Data File) and 'Στατιστικά Στοιχεία' (Statistical Data). The 'Δεδομένα Αρχείου' panel contains a table with columns 'time' and 'value'. The 'Στατιστικά Στοιχεία' panel contains a table with columns 'value' and 'value'. Below the tables, there are three dropdown menus for selecting the X-axis, Y-axis, and chart type. A 'Δημιουργία Διαγράμματος' button is located at the bottom.

time	value
11:33:43 π.μ.	23.8
11:33:51 π.μ.	29
11:34:01 π.μ.	24
11:34:11 π.μ.	23

value	value
Μέση τιμή	24.95
Μέγιστη τιμή	29
Ελάχιστη τιμή	23
Εύρος	6
Απόκλιση	2.368

Επιλέξτε στήλη για άξονα X: time

Επιλέξτε στήλη για άξονα Y: time

Επιλέξτε τύπο διαγράμματος: Γραμμικό διάγραμμα

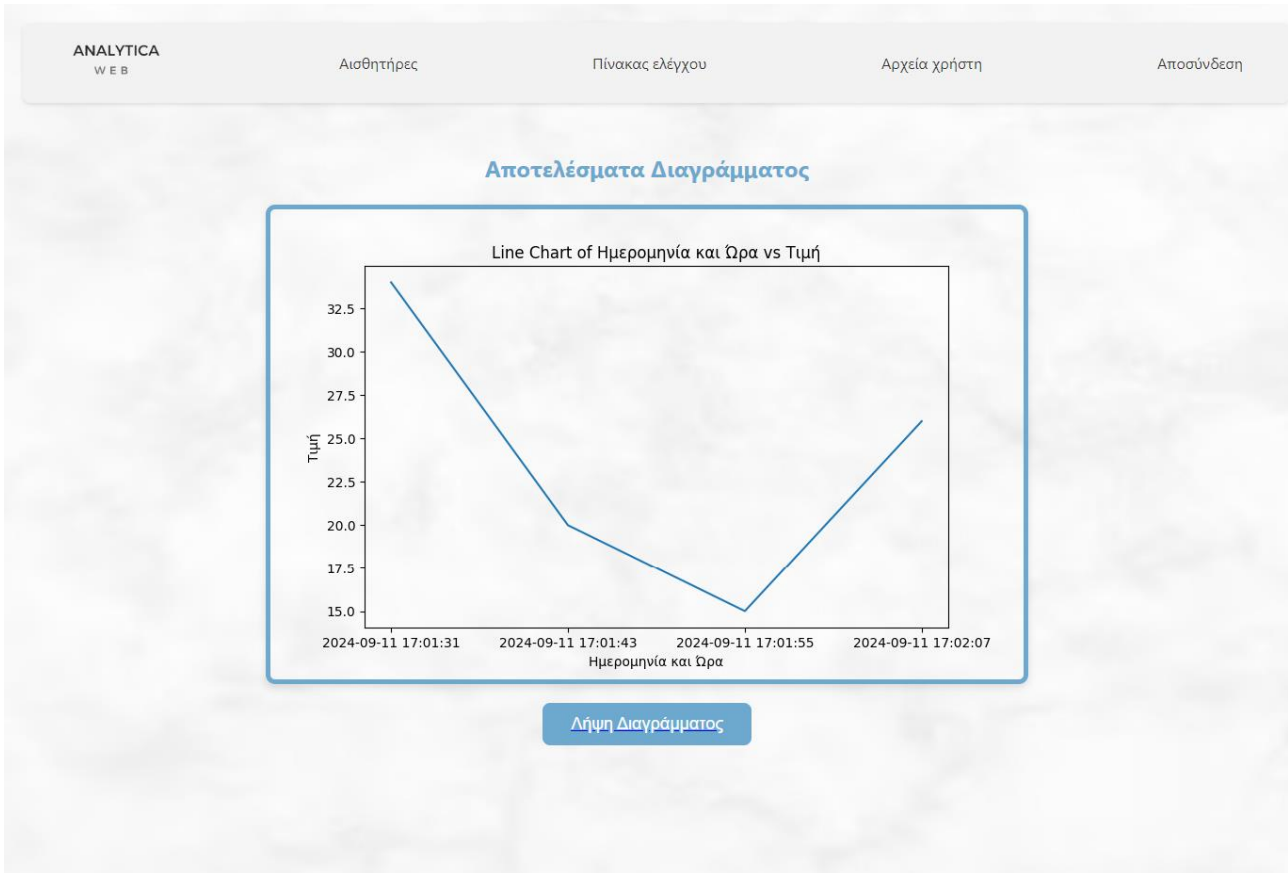
Δημιουργία Διαγράμματος

Εικόνα 20. Σελίδα δεδομένων και στατιστικών

Στη σελίδα αυτή, ο χρήστης βλέπει το διάγραμμα που έχει δημιουργηθεί (Εικόνα 21). Το διάγραμμα απεικονίζει τα δεδομένα του αρχείου με βάση τις ρυθμίσεις που έχει καθορίσει ο χρήστης. Στο κάτω μέρος της σελίδας, υπάρχει το κουμπί «Λήψη Διαγράμματος». Με την



επιλογή αυτού του κουμπιού, ο χρήστης μπορεί να αποθηκεύσει το διάγραμμα στον τοπικό δίσκο της συσκευής του σε μορφή εικόνας. Αυτό επιτρέπει την αποθήκευση και την αποδοχή του διαγράμματος για μελλοντική αναφορά ή χρήση. Εδώ τα διαγράμματα δημιουργούνται με τη βιβλιοθήκη matplotlib της Python.



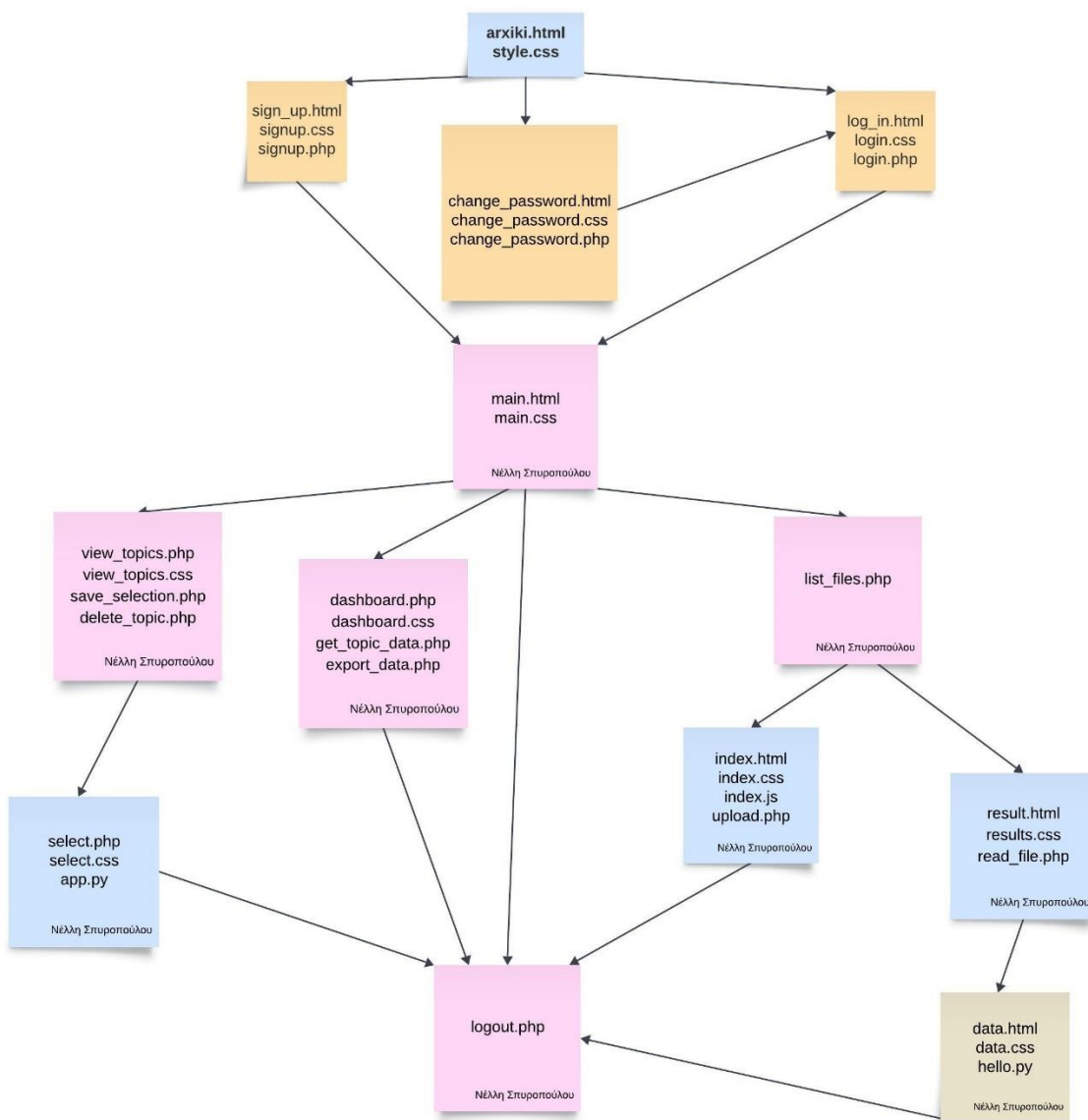
Εικόνα 21. Σελίδα προβολής διαγράμματος

Σε κάθε σελίδα, στο μενού που βρίσκεται στο πάνω μέρος της σελίδας, υπάρχει η επιλογή της αποσύνδεσης. Τέλος, αξίζει να σημειωθεί ότι όλες οι σελίδες που παρουσιάστηκαν, διαθέτουν responsive design για την καλύτερη εμφάνιση σε κάθε συσκευή και μέγεθος οθόνης.



3.6. Υλοποίηση των Λειτουργιών του Ισοτόπου

Για την καλύτερη κατανόηση της δομής και των λειτουργιών της εφαρμογής, σχεδιάστηκε το διάγραμμα (Εικόνα 22), το οποίο απεικονίζει τη σύνδεση μεταξύ των αρχείων. Αυτό το διάγραμμα δείχνει πως οι σελίδες και οι λειτουργίες αλληλεπιδρούν μεταξύ τους, καθιστώντας ομαλή την ροή των λειτουργιών και δεδομένων.



Εικόνα 22. Δενδροδιάγραμμα αρχείων εφαρμογής



Στον παρακάτω σύνδεσμο υπάρχει αναρτημένο το σύνολο του κώδικα της υλοποιημένης εφαρμογής:

<https://github.com/nelly-spyropoulou/analytica>

Η υλοποίηση των λειτουργιών της εφαρμογής περιλαμβάνει λειτουργίες, όπως η εγγραφή και η σύνδεση χρηστών, η προσθήκη και αφαίρεση αισθητήρων, η προβολή αισθητήρων σε πίνακα ελέγχου, η εξαγωγή δεδομένων από τους αισθητήρες, το ανέβασμα και η διαγραφή αρχείων και η δημιουργία γραφικών απεικονίσεων. Στην συνέχεια παρουσιάζονται ενδεικτικά αποσπάσματα κώδικα μαζί με μια περιγραφή των λειτουργιών που υλοποιούν.

Αρχικά, ο χρήστης έχει τη δυνατότητα να εγγραφεί στην πλατφόρμα (sign_up.html) δημιουργώντας έναν προσωπικό λογαριασμό, με τον οποίο συνδέεται για να αξιοποιήσει τις δυνατότητες της εφαρμογής. Με την εγγραφή του, δημιουργείται αυτόματα ένας ξεχωριστός φάκελος για τον κάθε χρήστη, στον οποίο αποθηκεύονται όλα τα αρχεία που ανεβάζει, αφού πρώτα μετατραπούν σε csv αρχεία (Εικόνα 23) , επιτρέποντάς του να οργανώνει και να διαχειρίζεται τα δεδομένα του με ευκολία.

```
// Μετατροπή σε CSV
$csvFile = 'uploads/' . $username . '/' . pathinfo(path: $filename, flags: PATHINFO_FILENAME) . '.csv';

if ($fileExt == 'xls' || $fileExt == 'xlsx') {
    $spreadsheet = IOFactory::load(filename: $fileDestination);
    $writer = new Csv(spreadsheet: $spreadsheet);
    $writer->save(filename: $csvFile);
} else {
    // Το αρχείο είναι ήδη CSV
    rename(from: $fileDestination, to: $csvFile);
}

// Διαγραφή του αρχικού αρχείου
if (file_exists(filename: $fileDestination)) {
    unlink(filename: $fileDestination);
}
```

Εικόνα 23. Παράδειγμα κώδικα μετατροπής αρχείου σε CSV (upload.php)

Στη σελίδα προσθήκης αισθητήρων ο χρήστης έχει τη δυνατότητα να συνδεθεί σε οποιοδήποτε θέμα της επιλογής του (select.php). Η σύνδεση με τον MQTT Broker και η προσθήκη αισθητήρα πραγματοποιούνται στο αρχείο app.py (Εικόνα 24). Η αποθήκευση δεδομένων από την φόρμα (frontend) στο backend γίνεται με τη χρήση AJAX και της τεχνολογίας fetch(), με τη μέθοδο POST. Τα δεδομένα αποστέλλονται στο κατάλληλο URL, όπου βρίσκεται ο Python Server, δηλαδή το flask.



```
def mqtt_subscribe(hostname, port, topic, username, password, sensor_name, unit, description, user_id):
    client = mqtt.Client(userdata={
        'user_id': user_id,
        'username': username,
        'sensor_name': sensor_name,
        'unit': unit,
        'description': description,
        'hostname': hostname
    })

    if username and password:
        client.username_pw_set(username, password)

    client.on_message = on_message

    try:
        client.connect(hostname, port, 60)
        client.subscribe(topic)
        client.loop_start()

        return {'status': 'success', 'message': 'Συνδεθήκαμε και εγγραφήκαμε στο θέμα επιτυχώς.'}

    except Exception as e:
        print(f"Σφάλμα κατά τη σύνδεση στον MQTT broker: {e}")
        return {'status': 'error', 'message': str(e)}
```

Εικόνα 24. Παράδειγμα κώδικα σύνδεσης στον MQTT broker (app.py)

Μέσα από μία σελίδα προβολής και διαχείρισης (view_topics.php), ο χρήστης μπορεί να βλέπει όλους τους αισθητήρες που έχει προσθέσει με λεπτομέρειες που αφορούν στο θέμα, την περιγραφή, το όνομα Host και τη μονάδα μέτρησης, ενώ του δίνεται η δυνατότητα να διαγράψει όσους αισθητήρες δεν χρειάζεται πια (Εικόνα 25).

```
// Ανάκτηση των δεδομένων από το POST
$data = json_decode(json: file_get_contents(filename: 'php://input'), associative: true);

if ($data === null) {
    error_log(message: 'Μη έγκυρα δεδομένα JSON: ' . json_last_error_msg());
    die(json_encode(value: ['status' => 'error', 'message' => 'Μη έγκυρα δεδομένα JSON.']));
}

$topic = filter_var(value: $data['topic'], filter: FILTER_SANITIZE_STRING);
$sensorName = filter_var(value: $data['sensor_name'], filter: FILTER_SANITIZE_STRING);
$username = $_SESSION['username'];

$query = "DELETE FROM mqtt_topics WHERE topic = $1 AND sensor_name = $2 AND username = $3";
$result = pg_query_params($conn, $query, [$topic, $sensorName, $username]);

if ($result) {
    echo json_encode(value: ['status' => 'success', 'message' => 'Η διαγραφή ολοκληρώθηκε με επιτυχία.']);
} else {
    error_log(message: 'Σφάλμα κατά τη διαγραφή: ' . pg_last_error($conn));
    echo json_encode(value: ['status' => 'error', 'message' => 'Σφάλμα κατά τη διαγραφή.']);
}
```

Εικόνα 25. Παράδειγμα κώδικα διαγραφής αισθητήρα (delete_topic.php)



Στην ίδια σελίδα ο χρήστης μπορεί να επιλέξει ποιοι από τους αισθητήρες τους θα εμφανιστούν στον Πίνακα ελέγχου. Η επιλογή αυτή γίνεται με τη βοήθεια Checkboxes (Εικόνα 26).

```
// Λήψη των επιλεγμένων θεμάτων από την αίτηση
$data = json_decode(json: file_get_contents(filename: 'php://input'), associative: true);
$selected_topics = isset($data['selected_topics']) ? $data['selected_topics'] : [];

// Αποθήκευση των επιλεγμένων θεμάτων στη συνεδρία
$_SESSION['selected_topics'] = $selected_topics;
```

Εικόνα 26. Παράδειγμα κώδικα αποθήκευσης επιλογών χρήστη (save_selection.php)

Μια άλλη σημαντική λειτουργία της εφαρμογής που σχετίζεται με τους αισθητήρες είναι ο πίνακας ελέγχου (Εικόνα 27), όπου ο χρήστης μπορεί να παρακολουθεί τα θέματα MQTT των αισθητήρων του μέσα από γραφήματα που απεικονίζουν τις καταγραφές. Υπάρχει, ακόμη, η δυνατότητα εξαγωγής δεδομένων σε ένα αρχείο Excel για κάθε αισθητήρα ξεχωριστά, το οποίο αποθηκεύεται στη συσκευή του για περαιτέρω χρήση (Εικόνα 28).

```
$query = "SELECT topic, sensor_name, description, value, created_at, unit
FROM mqtt_topics
WHERE username = $1 AND topic IN ($placeholders_str)
ORDER BY created_at DESC
LIMIT 10";
```

Εικόνα 27. Παράδειγμα κώδικα δημιουργίας νέου query για την εμφάνιση των 10 πιο πρόσφατων τιμών στο διάγραμμα (dashboard.php)



```
$spreadsheet = new Spreadsheet();
$sheet = $spreadsheet->getActiveSheet();
$sheet->setTitle(title: 'Data');

$sheet->setCellValue(coordinate: 'A1', value: 'Τιμή');
$sheet->setCellValue(coordinate: 'B1', value: 'Ημερομηνία και Ώρα');

$row = 2;
while ($data = pg_fetch_assoc($result)) {
    $formatted_date = (new DateTime(datetime: $data['created_at']))->format(format: 'Y-m-d H:i:s');
    $sheet->setCellValue(coordinate: 'A' . $row, value: $data['value']);
    $sheet->setCellValue(coordinate: 'B' . $row, value: $formatted_date);
    $row++;
}

pg_close($conn);

$filename = $filename_topic . '_' . $filename_sensor . '.xlsx';

header(header: 'Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
header(header: 'Content-Disposition: attachment;filename="' . $filename . '"');
header(header: 'Cache-Control: max-age=0');

$writer = new Xlsx(spreadsheet: $spreadsheet);
$writer->save(filename: 'php://output');
exit;
```

Εικόνα 28. Παράδειγμα κώδικα δημιουργίας αρχείου Excel (export_data.php)

Ακόμη, το ανέβασμα των αρχείων Excel πραγματοποιείται μέσω της ειδικά σχεδιασμένης σελίδας (Εικόνα 29) στον προσωπικό του φάκελο, να τα ανακτήσει ή να τα διαγράψει μέσα από μια σελίδα, όπου εμφανίζονται όλα τα αποθηκευμένα του αρχεία σε λίστα (Εικόνα 30). Η δυνατότητα προβολής και ανάλυσης των δεδομένων αυτών είναι διαθέσιμη μέσω πινάκων που παρουσιάζουν αναλυτικά τα στατιστικά και τα δεδομένα του κάθε αρχείου (Εικόνα 31).

```
function showFile() {
    let fileType = file.type;
    let validExtensions = ["application/vnd.ms-excel", "text/csv", "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"];
    if (validExtensions.includes(fileType)) {
        let fileName = file.name; // Πάιρνω το όνομα του αρχείου
        let fileInfoDiv = document.getElementById('fileInfo');
        fileInfoDiv.textContent = fileName;
        dropArea.appendChild(fileInfoDiv);
    }

    // Προσθήκη κουμπιού στο dropArea
    dropArea.appendChild(uploadButton);
} else {
    alert("Άγνωσμένος τύπος αρχείου!");
    dropArea.classList.remove("active");
}
}

function upload_file(e) {
    e.preventDefault();
    fileobj = e.dataTransfer.files[0];
    js_file_upload(fileobj);
}
```

Εικόνα 29. Παράδειγμα κώδικα ανεβάσματος αρχείου (index.js)



Επιπλέον, ο χρήστης, επιλέγοντας οποιοδήποτε από τα ανεβασμένα αρχεία του επιθυμεί, με τη βοήθεια κατάλληλων εργαλείων, μπορεί να δει τα δεδομένα του και ορισμένα βασικά στατιστικά για κάθε στήλη (Εικόνα 31). Η ανάγνωση των αρχείων γίνεται μέσα από το αρχείο `read_file.php`. (Εικόνα 30).

```
1 reference
function readCsvFile($filePath): array {
    $data = [];
    if (($handle = fopen(filename: $filePath, mode: 'r')) !== false) {
        while (($row = fgetcsv(stream: $handle, length: 1000, separator: ',')) !== false) {
            $data[] = $row;
        }
        fclose(stream: $handle);
    }
    return $data;
}
```

Εικόνα 30. Παράδειγμα κώδικα ανάγνωσης αρχείου (`read_file.php`)

```
if ($is_numeric_column && !empty($columnValues)) {
    $mean = round(num: array_sum(array: $columnValues) / count(value: $columnValues), precision: 3);
    $maxValue = max($columnValues);
    $minValue = min($columnValues);
    $range = $maxValue - $minValue;
    $stdDev = round(num: stat_standard_deviation(arr: $columnValues), precision: 3);

    $columnStatistics[$columnName] = [
        'mean' => $mean,
        'max' => $maxValue,
        'min' => $minValue,
        'range' => $range,
        'std_dev' => $stdDev
    ];
}
```

Εικόνα 31. Παράδειγμα κώδικα στατιστικών στοιχείων (`result.html`)

Για την ευκολότερη προσαρμογή των γραφημάτων η εφαρμογή παρέχει στο χρήστη τη δυνατότητα να καθορίσει τις στήλες των δεδομένων που θα χρησιμοποιηθούν για τους άξονες του διαγράμματος, όπως και να επιλέξει τον τύπο διαγράμματος που επιθυμεί, δημιουργώντας έτσι διαδραστικά διαγράμματα βάσει των αναγκών του (`data.html`). Τα διαγράμματα αυτά δημιουργούνται με τη βοήθεια του αρχείου `hello.py` (Εικόνα 32) και μπορούν, στη συνέχεια, να αποθηκευτούν ως εικόνες στη συσκευή του χρήστη, προσφέροντας τη δυνατότητα περαιτέρω χρήσης (Εικόνα 33).



```
@app.route('/data', methods=['POST'])
def get_data():
    data = request.json
    x_column = data.get('xColumn')
    y_column = data.get('yColumn')
    chart_type = data.get('chartType')
    file_path = data.get('filePath')

    try:
        df = pd.read_csv(file_path)
    except Exception as e:
        return jsonify(status="error", message=str(e)), 500

    plt.figure(figsize=(8, 5))

    if (chart_type == 'line'):
        plt.plot(df[x_column], df[y_column])
    elif (chart_type == 'bar'):
        plt.bar(df[x_column], df[y_column])
    elif (chart_type == 'scatter'):
        plt.scatter(df[x_column], df[y_column])
    else:
        return jsonify(status="error", message="Invalid chart type"), 400
```

Εικόνα 32. Παράδειγμα κώδικα λήψης επιλογών χρήστη (hello.py)

```
plt.xlabel(x_column)
plt.ylabel(y_column)
plt.title(f'{chart_type.capitalize()} Chart of {x_column} vs {y_column}')

image_path = 'static/chart.png'
```

Εικόνα 33. Παράδειγμα κώδικα δημιουργίας εικόνας διαγράμματος (hello.py)

Τέλος, η πλατφόρμα προσφέρει ευελιξία στην αλληλεπίδραση του χρήστη με το σύστημα, επιτρέποντας την αποσύνδεση από τον λογαριασμό του οποιαδήποτε στιγμή το επιθυμεί, διασφαλίζοντας έτσι τη διαχείριση της πρόσβασής του με ασφάλεια (Εικόνα 34). Με αυτό τον τρόπο, η εφαρμογή παρέχει ένα πλήρες και φιλικό περιβάλλον προς τον χρήστη για τη διαχείριση και ανάλυση δεδομένων, την παρακολούθηση αισθητήρων και τη δημιουργία διαγραμμάτων, καλύπτοντας με αποτελεσματικότητα και ευκολία τις ανάγκες κάθε χρήστη.



```
<?php
session_start();
// Εκκαθάριση όλων των δεδομένων συνεδρίας
$_SESSION = array();
// Αποστολή cookies στον περιηγητή με παρεμπόδιση στην αποθήκευση συνεδριακών cookies
if (ini_get(option: "session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(name: session_name(), value: '', expires_or_options: time() - 42000,
        path: $params["path"], domain: $params["domain"],
        secure: $params["secure"], httponly: $params["httponly"]
    );
}
session_destroy();
header(header: "Location: ../arxiki.html");
exit();
?>
```

Εικόνα 34. Παράδειγμα κώδικα δημιουργίας εικόνας διαγράμματος (logout.php)

3.7. Λειτουργίες του Backend

Η ανάπτυξη του backend πραγματοποιήθηκε με τη χρήση PHP και Python.

Διαχείριση Δεδομένων στο Backend

1. Επεξεργασία και Αποθήκευση Δεδομένων σύνδεσης σε MQTT Broker:

- Η αποθήκευση δεδομένων γίνεται μέσω της συνάρτησης `save_data_to_database()`, η οποία αποθηκεύει τα στοιχεία που έχουν σταλθεί μέσω της POST αίτησης στη βάση δεδομένων PostgreSQL (Εικόνα 35).

```
def save_data_to_database(user_id, username, topic, sensor_name, value, unit, description, hostname):
    connection = None
    cursor = None

    cursor.execute("""
        INSERT INTO mqtt_topics (user_id, username, topic, sensor_name, value, unit, description, hostname)
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s)
    """, (user_id, username, topic, sensor_name, value, unit, description, hostname))
```

Εικόνα 35. Παράδειγμα κώδικα (app.py)

2. Διαχείριση Σφαλμάτων (Εικόνα 36)

```
except Exception as e:
    print(f"Σφάλμα κατά τη σύνδεση στον MQTT broker: {e}")
    return {'status': 'error', 'message': str(e)}
```



```
except Exception as e:
    print("Σφάλμα αποθήκευσης στη βάση δεδομένων:", e)
    return {'status': 'error', 'message': str(e)}
```

Εικόνα 36. Παράδειγμα κώδικα (app.py)

3. Αποθήκευση δεδομένων εγγραφής και αρχείων

- Η αποθήκευση δεδομένων εγγραφής των χρηστών, καθώς επίσης και η αποθήκευση των αρχείων που ανεβάζει ο χρήστης γίνεται με τη χρήση PHP.

4. Ανάγνωση αρχείων

- Για την ανάγνωση των αρχείων που ανεβάζει ο χρήστης χρησιμοποιείται η συναρτήσεις `readCsvFile` και `readExcelFile` οι οποίες είναι γραμμένες σε PHP (Εικόνα 37).

```
1 reference
function readCsvFile($filePath): array {
    $data = [];
    if (($handle = fopen(filename: $filePath, mode: 'r')) !== false) {
        while (($row = fgetcsv(stream: $handle, length: 1000, separator: ',')) !== false) {
            $data[] = $row;
        }
        fclose(stream: $handle);
    }
    return $data;
}

1 reference
function readExcelFile($filePath): array {
    require 'vendor/autoload.php';
    $reader = \PhpOffice\PhpSpreadsheet\IOFactory::createReaderForFile(filename: $filePath);
    $spreadsheet = $reader->load(filename: $filePath);
    $data = $spreadsheet->getActiveSheet()->toArray(nullValue: null, calculateFormulas: true, formatData: true, r...true);
    return $data;
}
```

Εικόνα 37. Παράδειγμα κώδικα (read_file.php)

Ρύθμιση του Διακομιστή

1. Ενεργοποίηση CORS (Cross-Origin Resource Sharing):

- Ο εξυπηρετητής ρυθμίστηκε ώστε να επιτρέπει αιτήματα από διαφορετικά πεδία. Αυτό γίνεται με τη χρήση της βιβλιοθήκης `flask-cors`, η οποία επιτρέπει την επικοινωνία μεταξύ frontend και backend χωρίς περιορισμούς (Εικόνα 38).
- Έτσι, οι εφαρμογές που τρέχουν σε διαφορετικά domains μπορούν να στέλνουν αιτήματα στον server χωρίς να μπλοκάρονται από τους κανόνες ασφαλείας του προγράμματος περιήγησης.



```
from flask import Flask, request, jsonify, send_file
from flask_cors import CORS

app = Flask(__name__)
CORS(app, resources={r"/*": {"origins": "*"}})
```

Εικόνα 38. Παράδειγμα κώδικα (hello.py)

2. Δημιουργία Διαδρομής (Route):

- Δημιουργήθηκε μια διαδρομή που δέχεται αιτήματα POST (Εικόνα 39). Αυτή η διαδρομή επιτρέπει στον χρήστη να στέλνει δεδομένα σε μορφή JSON για να παραχθούν διαγράμματα.
- Το Flask παρέχει τον μηχανισμό για την παραλαβή των δεδομένων μέσω της βιβλιοθήκης request.

```
@app.route('/data', methods=['POST'])
def get_data():
    data = request.json
    x_column = data.get('xColumn')
    y_column = data.get('yColumn')
    chart_type = data.get('chartType')
    file_path = data.get('filePath')
//Υπόλοιπος κώδικας
```

Εικόνα 39. Παράδειγμα κώδικα (hello.py)

Διαχείριση Δεδομένων

1. Ανάγνωση Δεδομένων με Pandas:

- Η βιβλιοθήκη pandas χρησιμοποιείται για την ανάγνωση και επεξεργασία δεδομένων από αρχεία (στην προκειμένη περίπτωση αρχεία CSV) (Εικόνα 40).
- Ο διακομιστής διαβάζει το αρχείο που επιλέγεται από τον χρήστη και το μετατρέπει σε DataFrame για εύκολη ανάλυση και οπτικοποίηση.

```
import pandas as pd
df = pd.read_csv(file_path)
```

Εικόνα 40. Παράδειγμα κώδικα (hello.py)

2. Δημιουργία Διαγράμματος με Matplotlib:

- Χρησιμοποιώντας τη βιβλιοθήκη matplotlib, δημιουργείται το ζητούμενο



διάγραμμα βάσει των δεδομένων που διαβάστηκαν (Εικόνα 41).

- Ο χρήστης μπορεί να ζητήσει διάφορους τύπους γραφικών, όπως γραφήματα γραμμών, ράβδων ή διασποράς.

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 5))

if (chart_type == 'line'):
    plt.plot(df[x_column], df[y_column])
elif (chart_type == 'bar'):
    plt.bar(df[x_column], df[y_column])
elif (chart_type == 'scatter'):
    plt.scatter(df[x_column], df[y_column])
```

Εικόνα 41. Παράδειγμα κώδικα (hello.py)

Διαχείριση MQTT Συνδέσεων με Python

Στο backend, η διαχείριση των MQTT συνδέσεων γίνεται μέσω της επικοινωνίας με MQTT brokers. Αυτή η διαδικασία επιτρέπει στους χρήστες να αλληλεπιδρούν με τους αισθητήρες και να αποθηκεύουν δεδομένα.

Για τον έλεγχο της εφαρμογής κατά την διάρκεια της υλοποίησής της έγινε εγκατάσταση και ρύθμιση ενός τοπικού MQTT Broker

1. Εγκατάσταση και Ρύθμιση του MQTT Broker:
 - Πρώτα, πρέπει να εγκατασταθεί ένας MQTT broker, στην προκειμένη περίπτωση ο Mosquitto, για να διαχειρίζεται τις επικοινωνίες μεταξύ clients.
 - Ο broker θα διαχειρίζεται τις συνδέσεις και θα διανέμει τα μηνύματα στους subscribers που είναι εγγεγραμμένοι στα κατάλληλα θέματα (topics).

Επικοινωνία με τον MQTT Broker

1. Σύνδεση από το Frontend:
 - Με την υποβολή της φόρμας προσθήκης αισθητήρα από τον χρήστη, το frontend (JavaScript) συνδέεται με τον MQTT broker μέσω του API endpoint που ορίζεται στην εφαρμογή Flask (Εικόνα 42).
 - Με τη βοήθεια του client γίνεται και η αποστολή μηνυμάτων (Εικόνα 43).



```
@app.route('/subscribe', methods=['POST'])
def subscribe():
    data = request.get_json()

    hostname = data.get('hostname')
    port = int(data.get('port'))
    username = data.get('username')
    password = data.get('password')
    topic = data.get('topic')
    sensor_name = data.get('sensorName')
    unit = data.get('unit')
    description = data.get('description')
    user_id = data.get('user_id')
```

Εικόνα 42. Παράδειγμα κώδικα (app.py)

```
def mqtt_subscribe(hostname, port, topic, username, password, sensor_name, unit, description, user_id):
    client = mqtt.Client(userdata={
        'user_id': user_id,
        'username': username,
        'sensor_name': sensor_name,
        'unit': unit,
        'description': description,
        'hostname': hostname
    })
```

Εικόνα 43. Παράδειγμα κώδικα (app.py)

4. Έλεγχος Εφαρμογής

Στην ενότητα των πειραματικών αποτελεσμάτων, πραγματοποιήθηκαν σχολαστικοί έλεγχοι σε κάθε λειτουργία της εφαρμογής για την αξιολόγηση της σωστής λειτουργίας και της απόδοσης της. Η διαδικασία των δοκιμών ξεκίνησε με τις βασικές λειτουργίες σύνδεσης και εγγραφής των χρηστών. Αρχικά, δοκιμάστηκε η ροή της εγγραφής νέων χρηστών, από την οποία οι χρήστες εισάγουν τα στοιχεία τους, τα οποία καταχωρούνταν σωστά στη βάση δεδομένων. Επίσης, πραγματοποιήθηκαν δοκιμές στη λειτουργία σύνδεσης, όπου επιβεβαιώθηκε ότι οι εγγεγραμμένοι χρήστες μπορούσαν να συνδεθούν επιτυχώς στο σύστημα. Ο έλεγχος αυτός επαλήθευσε ότι η διαδικασία διαχείρισης των συνεδριών λειτουργούσε χωρίς προβλήματα, ενώ οι χρήστες προστατεύονταν από τυχόν σφάλματα κατά την αυθεντικοποίηση.

Στη συνέχεια, δόθηκε βάση στις λειτουργίες ανεβάσματος αρχείων. Οι χρήστες είχαν τη δυνατότητα να ανεβάζουν αρχεία δεδομένων στην εφαρμογή μέσω ενός εύχρηστου περιβάλλοντος χρήστη. Τα αρχεία αποθηκεύονταν στον διακομιστή σε έναν προσωποποιημένο φάκελο για κάθε χρήστη και εμφανίζονταν σε λίστα, δίνοντας στον χρήστη τη δυνατότητα να δει όλα τα ανεβασμένα αρχεία. Η λίστα με τα αρχεία ενημερωνόταν δυναμικά, και οι χρήστες μπορούσαν να επιλέξουν οποιοδήποτε αρχείο για



περαιτέρω ανάλυση ή ακόμα και διαγραφή.

Ακόμη, αξιολογήθηκε η δυνατότητα προβολής δεδομένων και στατιστικών για κάθε ανεβασμένο αρχείο. Όταν ένας χρήστης επιλέξει ένα αρχείο, η εφαρμογή αναλύει τα δεδομένα και παράγει βασικά στατιστικά, όπως μέση, μέγιστη και ελάχιστη τιμή, τυπική απόκλιση και εύρος τιμών. Οι δοκιμές επιβεβαίωσαν ότι η ανάλυση των δεδομένων ήταν γρήγορη και ότι τα αποτελέσματα παρουσιάζονταν με σαφήνεια, επιτρέποντας στον χρήστη την άμεση κατανόηση των χαρακτηριστικών των δεδομένων του.

Μία από τις πιο σημαντικές λειτουργίες της εφαρμογής που υποβλήθηκε σε εξονυχιστική δοκιμή ήταν η δυνατότητα δημιουργίας διαγραμμάτων. Οι χρήστες μπορούσαν να επιλέξουν στήλες από το αρχείο τους για τον ορισμό των αξόνων X και Y και να δημιουργήσουν διαγράμματα με διάφορους τύπους (π.χ., γραμμικά, ραβδογράμματα, διαγράμματα διασποράς). Οι δοκιμές επιβεβαίωσαν ότι τα διαγράμματα δημιουργούνταν σχετικά γρήγορα και η εφαρμογή προσέφερε στον χρήστη τη δυνατότητα να αποθηκεύσει τα διαγράμματα ως εικόνες, παρέχοντας έτσι μια πρακτική λειτουργία για την παρουσίαση των αποτελεσμάτων εκτός της εφαρμογής. Τα διαγράμματα ανταποκρίνονταν στις επιλογές του χρήστη και η διαδικασία ήταν φιλική και αποδοτική.

Η επόμενη δοκιμή αφορούσε στη λειτουργία σύνδεσης με έναν MQTT broker για τη διαχείριση και προβολή δεδομένων αισθητήρων. Στο πλαίσιο των δοκιμών, ελέγχθηκε η διαδικασία εγγραφής σε διάφορα θέματα (topics) των αισθητήρων, και επιβεβαιώθηκε ότι τα δεδομένα των αισθητήρων παρουσιάζονταν στον πίνακα ελέγχου της εφαρμογής σε πραγματικό χρόνο. Δοκιμάστηκε η προβολή όλων των διαθέσιμων αισθητήρων, όπου οι χρήστες μπορούσαν να δουν μια πλήρη λίστα των αισθητήρων που ήταν συνδεδεμένοι στο σύστημα. Η λίστα αυτή παρείχε πληροφορίες για κάθε αισθητήρα, όπως το όνομα και την περιγραφή, προσφέροντας μια συνολική εικόνα της τρέχουσας κατάστασης του συστήματος. Αυτό επέτρεπε στους χρήστες να παρακολουθούν ζωντανά τις τιμές των αισθητήρων. Η λειτουργία αυτή αποδείχθηκε ιδιαίτερα χρήσιμη για εφαρμογές που απαιτούν συνεχή παρακολούθηση δεδομένων.

Τέλος, πραγματοποιήθηκε έλεγχος της δυνατότητας εξαγωγής των δεδομένων σε αρχείο XLSX. Οι χρήστες είχαν την επιλογή να εξάγουν τα δεδομένα των αισθητήρων σε μορφή Excel (.xlsx), επιτρέποντας περαιτέρω ανάλυση εντός και εκτός του συστήματος. Οι δοκιμές έδειξαν ότι η εξαγωγή πραγματοποιούνταν γρήγορα και χωρίς σφάλματα, και τα δεδομένα εμφανίζονταν με σωστή μορφοποίηση στο αρχείο. Αυτή η δυνατότητα είναι εξαιρετικά χρήσιμη για χρήστες που θέλουν να χειριστούν ή να επεξεργαστούν τα δεδομένα τους με εργαλεία όπως το Microsoft Excel ή άλλες πλατφόρμες ανάλυσης δεδομένων.

Συνολικά, η εφαρμογή αποδείχθηκε σταθερή, φιλική προς τον χρήστη και πλήρως λειτουργική κατά τη διάρκεια των δοκιμών, με όλες τις λειτουργίες να εκτελούνται αποτελεσματικά και χωρίς σφάλματα.

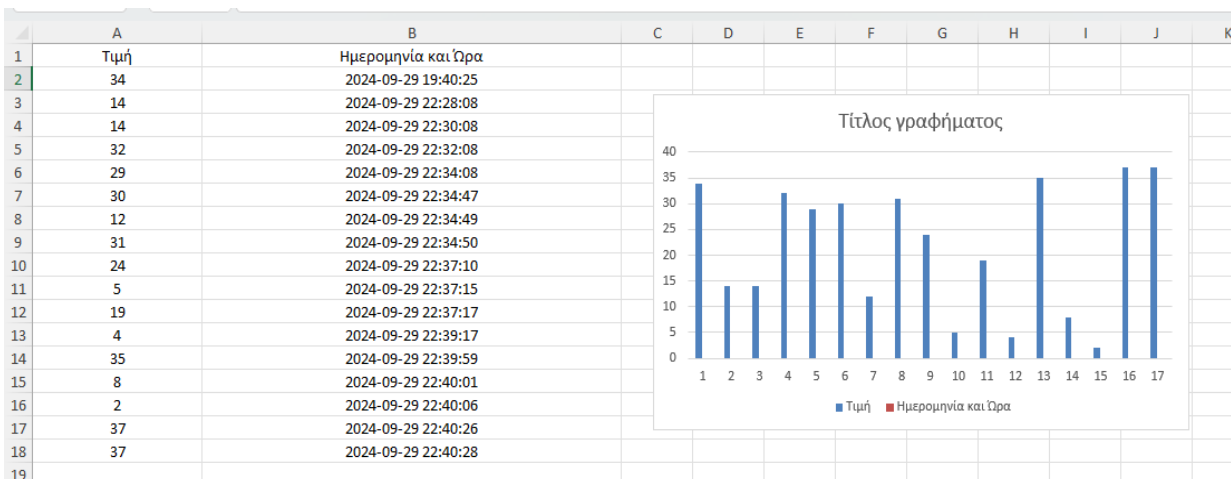


5. Συζήτηση

5.1. Σύγκριση με άλλες λύσεις

Η εφαρμογή που αναπτύχθηκε προσφέρει ένα μοναδικό συνδυασμό λειτουργιών. Πιο συγκεκριμένα, τη διαχείριση αρχείων, τη σύνδεση αισθητήρων, την ανάλυση δεδομένων και τη δημιουργία διαγραμμάτων σε μία πλατφόρμα. Αυτή η ενσωμάτωση των λειτουργιών διαφοροποιεί την εφαρμογή από πολλές άλλες λύσεις που κυκλοφορούν στην αγορά, καθώς παρέχει μια ολοκληρωμένη προσέγγιση για τη διαχείριση τόσο στατικών αρχείων (όπως αρχεία Excel) όσο και ζωντανών δεδομένων που προέρχονται από αισθητήρες μέσω του πρωτοκόλλου MQTT.

Συγκρίνοντας την εφαρμογή με άλλες πλατφόρμες διαχείρισης αρχείων και δεδομένων, όπως το Google Sheets και το Microsoft Excel (Εικόνα 44), βλέπουμε ότι αυτές οι εφαρμογές είναι πολύ ισχυρές στην επεξεργασία και ανάλυση δεδομένων, αλλά περιορίζονται όταν πρόκειται για ζωντανή ροή δεδομένων από εξωτερικές συσκευές ή αισθητήρες. Οι συγκεκριμένες εφαρμογές προσφέρουν σύνθετες δυνατότητες υπολογισμών και γραφημάτων, ωστόσο, η ενσωμάτωση με συσκευές IoT ή αισθητήρες συνήθως απαιτεί τη χρήση εξωτερικών εργαλείων και εφαρμογών για να επιτευχθεί. Στην παρούσα εφαρμογή, ο χρήστης έχει τη δυνατότητα να συνδέεται απευθείας με αισθητήρες, να λαμβάνει δεδομένα σε πραγματικό χρόνο και να τα αναλύει εντός της ίδιας πλατφόρμας, κάτι που εξοικονομεί χρόνο και διευκολύνει τη συνολική διαδικασία.

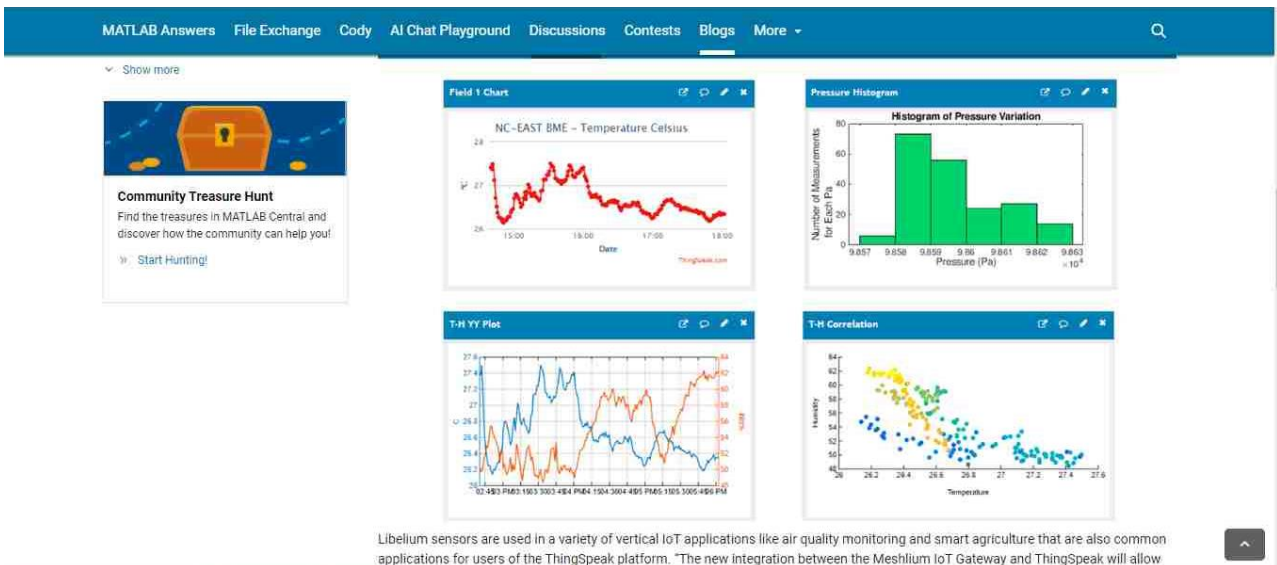


Εικόνα 44. Διάγραμμα σε αρχείο Excel

Επιπλέον, πλατφόρμες IoT όπως το ThingSpeak (Εικόνα 45) ή το Blynk (Εικόνα 46) εστιάζουν κυρίως στην παρακολούθηση και ανάλυση δεδομένων από αισθητήρες. Αν και αυτές οι λύσεις παρέχουν ισχυρές δυνατότητες παρακολούθησης δεδομένων σε πραγματικό χρόνο, δεν υποστηρίζουν εύκολα την επεξεργασία και ανάλυση δεδομένων από αρχεία όπως Excel, ενώ η λειτουργικότητά τους για δημιουργία διαγραμμάτων μπορεί να είναι πιο περιορισμένη ή να απαιτεί επιπλέον ρυθμίσεις. Στην εφαρμογή μας, οι χρήστες μπορούν να ανεβάσουν τα προσωπικά τους αρχεία Excel, να τα επεξεργαστούν και να



δημιουργήσουν προσαρμοσμένα γραφήματα από αυτά, τα οποία μπορούν να αποθηκευτούν και να κατεβούν ως εικόνες.



Εικόνα 45. ThingSpeak

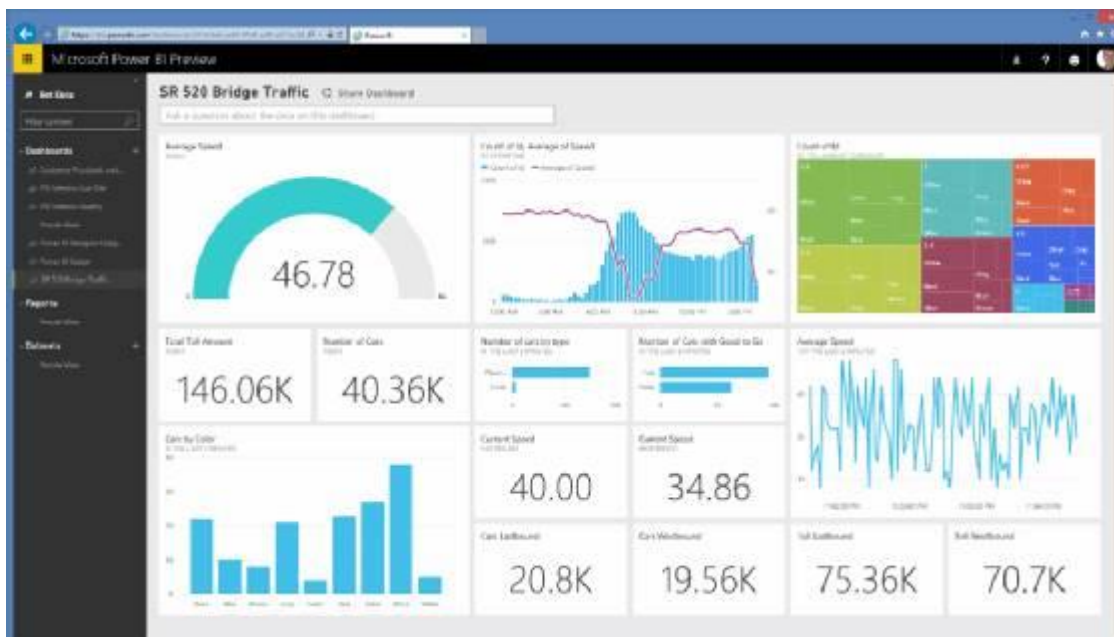


Εικόνα 46. Blynk

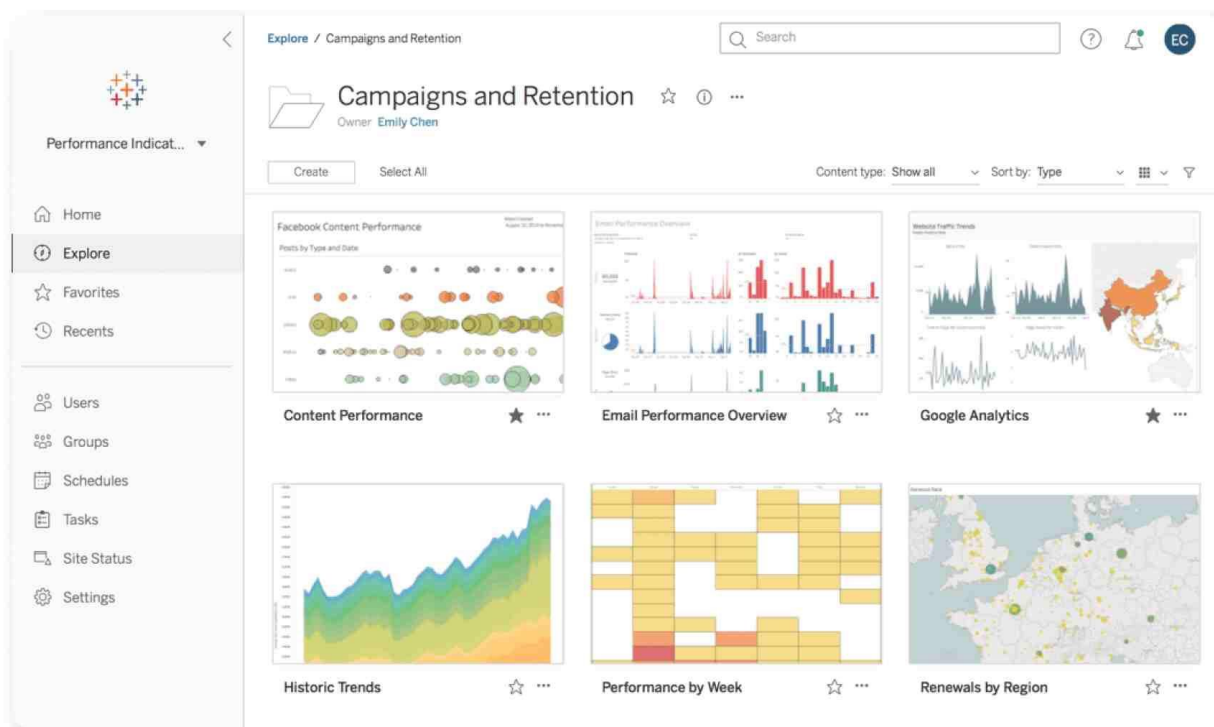
Επιπλέον, λύσεις BI (Business Intelligence) όπως το Power BI (Εικόνα 47) και το Tableau (Εικόνα 48) παρέχουν πιο σύνθετες δυνατότητες ανάλυσης και οπτικοποίησης δεδομένων, αλλά συνήθως επικεντρώνονται σε μεγάλης κλίμακας δεδομένα και απαιτούν βαθύτερη τεχνική γνώση για την πλήρη αξιοποίησή τους. Σε αντίθεση, η παρούσα εφαρμογή είναι σχεδιασμένη για πιο αρχάριους χρήστες, προσφέροντας έναν απλό και εύχρηστο τρόπο



για την οπτικοποίηση δεδομένων και τη δημιουργία διαγραμμάτων με βάση τα αρχεία που ανεβάζουν ή τα δεδομένα που λαμβάνουν από τους αισθητήρες τους, με ελάχιστες απαιτήσεις σε τεχνική γνώση.



Εικόνα 47. Power BI



Εικόνα 48. Tableau



Αυτό που διαφοροποιεί την εφαρμογή είναι η ευκολία χρήσης και η απλοποίηση των διαδικασιών που απαιτούνται για τη διαχείριση των δεδομένων και την οπτικοποίησή τους. Η δυνατότητα δημιουργίας διαγραμμάτων με την επιλογή επιθυμητών αξόνων και τύπου γραφήματος είναι μια χαρακτηριστική λειτουργία που προσφέρει σημαντική ευελιξία στον χρήστη.

5.2. Περιορισμοί της εφαρμογής

Παρά τις σημαντικές δυνατότητες της εφαρμογής, υπάρχουν ορισμένοι περιορισμοί που πρέπει να επισημανθούν. Καταρχάς, ενώ η εφαρμογή υποστηρίζει τη δημιουργία γραφημάτων και την ανάλυση δεδομένων, τα γραφήματα αυτά είναι αρκετά βασικά σε σύγκριση με τα σύνθετα γραφήματα που μπορούν να δημιουργηθούν σε επαγγελματικά εργαλεία BI όπως το Power BI ή το Tableau. Δεν υπάρχει προς το παρόν δυνατότητα για πιο σύνθετες αναλύσεις, όπως προβλέψεις δεδομένων, αναλύσεις συσχετίσεων, ή ενσωμάτωση αλγορίθμων μηχανικής μάθησης για την επεξεργασία των δεδομένων που συλλέγονται από αισθητήρες.

Επιπλέον, η εφαρμογή είναι σχεδιασμένη για μοναδική χρήση από κάθε χρήστη, με έμφαση στην ατομική διαχείριση των αρχείων και των αισθητήρων. Δεν υπάρχει υποστήριξη για συνεργατική εργασία ή τη δυνατότητα να διαχειριστούν πολλοί χρήστες τα ίδια δεδομένα ταυτόχρονα. Αυτό μπορεί να περιορίσει τη χρηστικότητα της εφαρμογής σε ομάδες εργασίας ή εταιρικά περιβάλλοντα, όπου η διαμοίραση και η κοινή διαχείριση δεδομένων είναι σημαντική. Μια μελλοντική βελτίωση θα μπορούσε να είναι η ενσωμάτωση της δυνατότητας για πολυπρόσωπη συνεργασία μέσα στην εφαρμογή, επιτρέποντας σε πολλούς χρήστες να συνεργάζονται πάνω σε κοινά δεδομένα, να μοιράζονται διαγράμματα και ανάλυση δεδομένων σε πραγματικό χρόνο. Αυτό θα ενίσχυε τη χρηστικότητα της εφαρμογής σε πιο επαγγελματικά περιβάλλοντα και θα επέκτεινε τις δυνατότητες της για πιο σύνθετες ανάγκες ανάλυσης.

Όσον αφορά τη διαχείριση των αισθητήρων, η εφαρμογή παρέχει βασικές δυνατότητες σύνδεσης με θέματα μέσω MQTT, αλλά δεν διαθέτει προηγμένες επιλογές διαχείρισης για πιο πολύπλοκα σενάρια, όπως η εκτέλεση αυτοματοποιημένων ενεργειών με βάση τα δεδομένα που συλλέγονται. Μια ενδεχόμενη βελτίωση θα μπορούσε να είναι η προσθήκη λειτουργιών για αυτοματοποιημένες αναλύσεις ή ειδοποιήσεις που βασίζονται σε δεδομένα αισθητήρων, όπως συμβαίνει σε πιο εξειδικευμένες IoT πλατφόρμες.

Επίσης, ενώ η δυνατότητα αποθήκευσης διαγραμμάτων ως εικόνες είναι χρήσιμη, θα μπορούσαν να προστεθούν επιπλέον μορφές εξαγωγής, όπως αρχεία PDF ή διαδραστικά διαγράμματα που θα μπορούσαν να προσαρμόζονται από τον χρήστη σε πραγματικό χρόνο. Επιπλέον, η δυνατότητα δημιουργίας και αποθήκευσης προσαρμοσμένων dashboards με ζωντανά δεδομένα από αισθητήρες θα προσέθετε αξία για τους χρήστες που χρειάζονται πιο σύνθετη οπτικοποίηση των δεδομένων τους.



Τέλος, η διεπαφή της εφαρμογής για την επιλογή τύπων γραφημάτων θα μπορούσε να γίνει πιο ευέλικτη, προσθέτοντας περισσότερες επιλογές για διαφορετικούς τύπους διαγραμμάτων, καθώς και δυνατότητες προσαρμογής του σχεδιασμού, όπως χρώματα, στυλ γραμμών και ετικέτες, που είναι χρήσιμα για πιο επαγγελματική οπτικοποίηση δεδομένων.

6. Συμπέρασμα

Συνοψίζοντας, η παρούσα εργασία πέτυχε να αναπτύξει έναν ολοκληρωμένο και λειτουργικό ιστότοπο για τη διαχείριση και ανάλυση δεδομένων από τους χρήστες, καλύπτοντας τόσο τις ανάγκες για εύκολη πρόσβαση και χρήση όσο και για ασφάλεια και αποδοτικότητα. Η εφαρμογή παρέχει ένα απλό αλλά ισχυρό σύστημα εγγραφής και σύνδεσης, το οποίο επιτρέπει στους χρήστες να δημιουργούν λογαριασμό και να συνδέονται γρήγορα και με ασφάλεια. Η χρηστικότητα της εφαρμογής ενισχύεται από τη δυνατότητα ανάρτησης αρχείων Excel, τα οποία αποθηκεύονται σε προσωπικούς φακέλους, προσφέροντας έτσι εύκολη πρόσβαση στα δεδομένα του κάθε χρήστη. Παράλληλα, η εφαρμογή παρέχει δυνατότητες διαχείρισης και προβολής αυτών των αρχείων μέσω μίας απλής διεπαφής.

Η δυνατότητα διαχείρισης αισθητήρων αποτελεί επίσης σημαντικό μέρος της εφαρμογής. Οι χρήστες μπορούν να συνδέονται με αισθητήρες, να διαχειρίζονται τις πληροφορίες που συλλέγουν και να παρακολουθούν ζωντανά τα δεδομένα μέσω ενός δυναμικού πίνακα ελέγχου. Αυτή η λειτουργία είναι ιδιαίτερα χρήσιμη για την παρακολούθηση των αισθητήρων σε πραγματικό χρόνο, δίνοντας στους χρήστες τη δυνατότητα να συλλέγουν και να αναλύουν δεδομένα από τους συνδεδεμένους αισθητήρες τους.

Κομβικό ρόλο στην εφαρμογή παίζει η δυνατότητα διαγραμματικής απεικόνισης των δεδομένων. Οι χρήστες μπορούν να δημιουργούν διαγράμματα επιλέγοντας τις επιθυμητές στήλες για τους άξονες X και Y, καθώς και τον τύπο του διαγράμματος (γραμμικό, ράβδων ή διασποράς). Επιπλέον, προσφέρεται η δυνατότητα αποθήκευσης αυτών των διαγραμμάτων ως εικόνες, γεγονός που ενισχύει την ευελιξία των χρηστών στην ανάλυση δεδομένων και τη δημιουργία αναφορών. Η αποθήκευση των γραφημάτων διευκολύνει τη μελλοντική χρήση και την παρουσίαση των αναλύσεων που έχουν πραγματοποιηθεί.

Όσον αφορά την ασφάλεια των δεδομένων, ενώ έχουν ληφθεί βασικά μέτρα, απαιτούνται πρόσθετες ενέργειες για την ενίσχυση της προστασίας της ιδιωτικότητας των χρηστών. Η ενσωμάτωση προηγμένων τεχνικών κρυπτογράφησης και άλλων μέτρων ασφάλειας είναι απαραίτητη για να διασφαλιστεί ότι τα δεδομένα των χρηστών παραμένουν προστατευμένα από πιθανές απειλές.

Για την περαιτέρω βελτίωση της εφαρμογής, προτείνονται μελλοντικές επεκτάσεις, όπως η προσθήκη εξελιγμένων επιλογών εξατομίκευσης των διαγραμμάτων, ώστε οι χρήστες να μπορούν να προσαρμόζουν τα γραφήματά τους με μεγαλύτερη ευελιξία και ακρίβεια. Επίσης, η βελτίωση της διαχείρισης αρχείων και η αναβάθμιση της υποδομής του συστήματος θα μπορούσαν να προσφέρουν ταχύτερη και πιο αποδοτική χρήση της εφαρμογής. Επιπλέον, οι ενέργειες για την ενίσχυση της ασφάλειας των δεδομένων μέσω



προηγμένων τεχνολογιών κρυπτογράφησης και προστασίας θα εξασφαλίσουν ότι η πλατφόρμα παραμένει αξιόπιστη και ασφαλής για τους χρήστες.

Παρά την επιτυχία της εφαρμογής στην παροχή αποτελεσματικών εργαλείων διαχείρισης και ανάλυσης δεδομένων, υπάρχουν πολλά πεδία που θα μπορούσαν να διερευνηθούν περαιτέρω. Μελλοντική έρευνα μπορεί να επικεντρωθεί στην ενσωμάτωση νέων τύπων συνδέσεων αισθητήρων, καθώς και πιο σύνθετων αλγορίθμων ανάλυσης δεδομένων, προσφέροντας στους χρήστες εξελιγμένες δυνατότητες πρόβλεψης και ανάλυσης τάσεων. Επίσης, θα μπορούσε να εξεταστεί η προσθήκη εργαλείων τεχνητής νοημοσύνης για την αυτόματη ανάλυση μεγάλων συνόλων δεδομένων, κάτι που θα καθιστούσε την εφαρμογή ακόμα πιο ισχυρή. Τέλος, η βελτίωση της ασφάλειας, με τη χρήση καινοτόμων τεχνολογιών κρυπτογράφησης και αυθεντικοποίησης, θα εξασφαλίσει ότι η εφαρμογή θα παραμείνει ασφαλής και αξιόπιστη για την προστασία των δεδομένων των χρηστών της.

Συνολικά, η εφαρμογή πέτυχε να προσφέρει μια ισχυρή βάση για τη διαχείριση και ανάλυση δεδομένων, επιτυγχάνοντας τους βασικούς στόχους της. Οι προτάσεις για μελλοντικές βελτιώσεις στοχεύουν στην ενίσχυση της απόδοσης, της ασφάλειας και της χρηστικότητας της πλατφόρμας, εξασφαλίζοντας ότι θα συνεχίσει να ανταποκρίνεται στις ανάγκες των χρηστών και να παραμένει ανταγωνιστική στον χώρο της διαχείρισης δεδομένων και αισθητήρων.



7. Βιβλιογραφία

[1] MQTT, τι είναι και που μπορούμε να το χρησιμοποιήσουμε;

<https://www.doomster.eu/thoughts/mqtt-ti-einai-kai-poy-xrisimopoiieitai/>

[2] HTML Βικιπαίδεια

<https://el.wikipedia.org/wiki/HTML>

[3] Εισαγωγή στο CSS

http://pages.cs.aueb.gr/courses/epl131/files/CSS_notes.pdf

[4] Τι Είναι η JavaScript και Πού Χρησιμοποιείται;

<https://bigblue.academy/gr/javascript>

[5] PHP Βικιπαίδεια

<https://el.wikipedia.org/wiki/PHP>

[6] Τι Είναι η Python; (Έννοια, Οφέλη & Τρόποι Εκμάθησης)

<https://bigblue.academy/gr/python-giati-prepei-na-xekiniseis-na-matheis-simera>

[7] Django vs Flask: Ποιο Είναι το Καλύτερο Python Web Framework

<https://bigblue.academy/gr/django-vs-flask>

[8] Flask-CORS

<https://flask-cors.readthedocs.io/en/v1.3.1/>

[9] Matplotlib: Visualization with Python

<https://matplotlib.org/>

[10] What Is Pandas Python Library?

<https://www.coursera.org/articles/what-is-pandas-python>

[11] What is NumPy?

<https://numpy.org/doc/stable/user/whatisnumpy.html>



[12] Seaborn: Everything you need to know about the Python data visualization tool
<https://datascientest.com/en/seaborn-everything-you-need-to-know-about-the-python-data-visualization-tool>

[13] Plotly Open Source Graphing Library for Python
<https://plotly.com/python/>

[14] Τι είναι η PostgreSQL και πως κάνουμε μία πλήρης εγκατάσταση
<https://www.dataplatform.gr/ti-einai-i-postgresql-kai-pos-kanoyme-mia-pliris/>

[15] Xampp Βικιπαίδεια
<https://en.wikipedia.org/wiki/XAMPP>

[16] What is three-tier architecture?
<https://www.ibm.com/topics/three-tier-architecture>



8. Παράρτημα

Δημιουργία βάσης δεδομένων

```
CREATE TABLE IF NOT EXISTS signup
```

```
(  
  firstname character varying (50),  
  lastname character varying (50),  
  email character varying (100),  
  username character varying (50),  
  password character varying (255),  
  user_id integer,  
  birthdate date,  
  CONSTRAINT signup_pkey PRIMARY KEY (username)  
)
```

```
CREATE TABLE IF NOT EXISTS mqtt_topics
```

```
(  
  id integer NOT NULL DEFAULT,  
  username character varying (255) NOT NULL,  
  topic character varying (255) NOT NULL,  
  value text NOT NULL,  
  sensor_name character varying (255),  
  description text,  
  unit character varying (50),  
  hostname character varying (255),  
  user_id integer,  
  created_at timestamp without time zone DEFAULT CURRENT_TIMESTAMP,  
  CONSTRAINT mqtt_topics_pkey PRIMARY KEY (id)  
)
```

```
CREATE TABLE IF NOT EXISTS mqtt_subscriptions
```

```
(  
  id integer NOT NULL DEFAULT,  
  username character varying (255) NOT NULL,  
  topic character varying (255),  
  sensor_name character varying (255),  
  description text,  
  user_id integer,  
  hostname character varying (255),  
  port integer,  
  unit character varying (50),  
  created_at timestamp without time zone DEFAULT CURRENT_TIMESTAMP,  
  CONSTRAINT mqtt_subscriptions_pkey PRIMARY KEY (username, sensor_name,  
topic)  
)
```