ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Ανάπτυξη γραφικού περιβάλλοντος εφαρμογής IOT

Χούχλιας Χρήστος

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Τζιάλλας Γρηγόριος
Καθηγητής Α Βαθμίδας

Λαμία Νοέμβριος 2022

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Ανάπτυξη γραφικού περιβάλλοντος εφαρμογής IOT

Χούχλιας Χρήστος

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Τζιάλλας Γρηγόριος
Καθηγητής Α Βαθμίδας

Λαμία Νοέμβριος 2022

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

UNIVERSITY OF THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

Development of a graphical user interface for an IOT application

Chouchlias Christos

FINAL THESIS

ADVISOR

Tziallas Grigorios
Professor

Lamia November 2022

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις [(1)], που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

*1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

*2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: .18../.10./2022..

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση
του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων
σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

ΠΕΡΙΛΗΨΗ

Στη παρούσα πτυχιακή εργασία, έγινε ανάπτυξη μιας διαδικτυακής εφαρμογής η οποία δίνει τη δυνατότητα στο χρήστη να διαχειρίζεται IOT συσκευές και παρουσιάζει σε αυτόν τα δεδομένα που συλλέγει σε μορφή διαγραμμάτων. Λόγω της συνεχής εξέλιξης των συστημάτων IOT και της εκτεταμένης χρήσης τους τα τελευταία χρόνια, μπορεί λοιπόν, να χρησιμοποιηθεί σε μια πληθώρα πεδίων εφαρμογής. Ένας από τους σημαντικότερους τομείς εφαρμογής , λόγω της επικείμενης ενεργειακής κρίσης, είναι αυτός της ενεργειακής διαχείρισης κτηρίων. Για το λόγο αυτό, και με βάση την αρχική εφαρμογή, αναπτύχθηκε ένα σύστημα ενεργειακής διαχείρισης για το Τμήμα Πληροφορικής με Εφαρμογές στη Βιοϊατρική του Πανεπιστημίου Θεσσαλίας για την παρακολούθηση της θερμοκρασίας και υγρασίας για κάθε χώρο του κτιρίου. Θα χρησιμοποιεί κατάλληλους αισθητήρες και θα ενημερώνει μεσω ειδοποιήσεων εάν οι τιμές που λαμβάνει δεν βρίσκονται εντός κάποιων προκαθορισμένων ορίων.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

ABSTRACT

The following thesis presents the development and implementation of a web application, which enables the user to manage IOT devices, and presents to him, the data they collect, through the form of diagrams. Due to the continuous development of IOT systems and their extensive use in recent years, it can therefore be used in a multitude of fields. Due to the impending energy crisis, one of the most important fields of application is the Building Energy Management Systems (BEMS). Based on the initial application, an energy management system was developed for the Department of Computer Science and Biomedical Informatics of the University of Thessaly to monitor the temperature and humidity for each room of the building. It will use appropriate sensors and inform through alerts if the values it receives are not within some predefined limits.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Table of Contents

ΠΕΡΙΛΗΨΗ I
ABSTRACT III

ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ 2

1.1 ΣΚΟΠΟΣ ΤΗΣ ΠΤΥΧΙΑΚΗΣ 2
1.2 ΔΟΜΗ ΕΡΓΑΣΙΑΣ 2

ΚΕΦΑΛΑΙΟ 2 ΕΝΕΡΓΕΙΑΚΗ ΚΡΙΣΗ 3

2.1 ΑΙΤΙΑ ΚΑΙ ΣΥΝΕΠΕΙΕΣ 3
2.2 ΤΡΟΠΟΙ ΑΝΤΙΜΕΤΩΠΙΣΗΣ 4
2.3 ΕΝΕΡΓΕΙΑΚΗ ΔΙΑΧΕΙΡΙΣΗ 5

ΚΕΦΑΛΑΙΟ 3 ΤΟ ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ 7

3.1 ΟΡΙΣΜΟΣ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ ΤΩΝ ΠΡΑΓΜΑΤΩΝ 7
3.2 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ ΤΩΝ ΠΡΑΓΜΑΤΩΝ 7
3.3 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ ΤΩΝ ΠΡΑΓΜΑΤΩΝ 8
3.4 ΠΡΩΤΟΚΟΛΛΑ 8
3.5 ΠΕΔΙΑ ΕΦΑΡΜΟΓΗΣ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ ΤΩΝ ΠΡΑΓΜΑΤΩΝ 9

ΚΕΦΑΛΑΙΟ 4 ΤΕΧΝΟΛΟΓΙΕΣ ΥΛΟΠΟΙΗΣΗΣ 12

4.1 HTML5 12
4.2 CSS 13
4.2.1 SASS 13
4.2.2 RESPONSIVE DESIGN 14
4.3 JAVASCRIPT 14
4.4 BOOTSTRAP 15
4.5 REACT 15
4.6 NODE JS 16
4.6.1 EXPRESS.JS 17
4.7 ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ 17
4.7.1 POSTGRESQL 19
4.8 API 19
4.9 ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ ΚΑΙ HASHING (PASSPORT ΚΑΙ BCRYPT) 19
4.10 GRAFANA 20

ΚΕΦΑΛΑΙΟ 5 ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ 22

5.1 ΠΕΡΙΓΡΑΦΗ 22

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

5.2 ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ BACK-END 22
5.2.1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ 23
5.2.2 SERVER 27
5.3 GRAFANA 35
5.4.1 ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ GRAFANA - ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΕΙΣ 35
5.4.2 ΣΥΝΔΕΣΗ ΤΟΥ GRAFANA ΜΕ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ POSTGRESQL 36
5.4.3 ΔΗΜΙΟΥΡΓΙΑ ΓΡΑΦΗΜΑΤΩΝ 37
5.4 ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ FRONT-END 42
5.4.1 ΔΟΜΗ ΑΡΧΕΙΩΝ 42
5.4.2 ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΕΠΕΞΗΓΗΣΗ ΚΩΔΙΚΑ 43

ΚΕΦΑΛΑΙΟ 6 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΠΙΛΟΤΙΚΗΣ ΕΦΑΡΜΟΓΗΣ 55

6.1 ΣΚΟΠΟΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ 55
6.2 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΠΙΛΟΤΙΚΗΣ ΕΦΑΡΜΟΓΗΣ 56
6.3 BACK-END ΠΙΛΟΤΙΚΗΣ ΕΦΑΡΜΟΓΗΣ 69

ΚΕΦΑΛΑΙΟ 7 ΣΥΜΠΕΡΑΣΜΑΤΑ 71

7.1 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ 71

ΒΙΒΛΙΟΓΡΑΦΙΑ 72

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

1.1 Σκοπός της πτυχιακής

Η τεχνολογία των συστημάτων IOT είναι ιδιαίτερα διαδομένη και σημαντική στη σημερινή εποχή λόγω της πληθώρας εφαρμογών που έχει. Το Διαδίκτυο των πραγμάτων (IOT) χρησιμοποιεί ένα σύνολο συσκευών και τεχνολογιών που ανιχνεύουν το περιβάλλον τους και ανταλλάσσουν δεδομένα μεταξύ τους μέσω του Διαδικτύου. Ο σκοπός της πτυχιακής εργασίας λοιπόν είναι η ανάπτυξη γραφικού περιβάλλοντος μιας διαδικτυακής εφαρμογής όπου θα διαχειρίζεται τέτοιου είδους IOT συσκευές και θα παρουσιάζει στον χρήστη τα δεδομένα που συλλέγουν σε μορφή διαγραμμάτων.

Τα τελευταία χρόνια παρατηρείται επίσης συνεχής αύξηση των παγκόσμιων ενεργειακών απαιτήσεων. Κρίνεται λοιπόν επιτακτική η ανάγκη εξοικονόμησης ενέργειας δηλαδή η προσπάθεια περιορισμού της σπατάλης των ενεργειακών αποθεμάτων στον κτιριακό τομέα όπου οι απαιτήσεις είναι ιδιαίτερα υψηλές. Σύμφωνα με τα τελευταία νέα, όπου συζητιούνται τρόποι αντιμετώπισης της ενεργειακής κρίσης με γνώμονα της χαμηλότερης κατανάλωσης θέρμανσης σε δημόσιες υπηρεσίες, δημιουργήθηκε μια διαδικτυακή εφαρμογή ως επέκταση της προηγούμενης για το Πανεπιστήμιο Θεσσαλίας, η οποία θα παρουσιάζει με γρήγορο, εύκολο και κατανοητό τρόπο τις τιμές θερμοκρασίας και υγρασίας των χώρων του κτιρίου, με τη χρήση κατάλληλων αισθητήρων και θα ενημερώνει τους χρήστες εάν οι τιμές αυτές δεν βρίσκονται εντός κάποιων προκαθορισμένων ορίων.

1.2 Δομή Εργασίας

Στο δεύτερο κεφάλαιο παρουσιάζονται η έννοια της ενεργειακής κρίσης, τα αίτια, οι συνέπειες και κάποιοι από τους τρόπους αντιμετώπισης της καθώς και η σημασία της ενεργειακής διαχείρισης στην εποχή μας.

Στο τρίτο κεφάλαιο ορίζεται η έννοια του διαδικτύου των πραγμάτων IOT, γίνεται μία σύντομη ιστορική αναδρομή και περιγράφονται τα χαρακτηριστικά τους , οι τρόποι λειτουργίας τους καθώς και τρία από τα πρωτοκόλλα που χρησιμοποιούν. Ακόμη αναλύονται τα πολλαπλά πεδία εφαρμογής τους.

Στο τέταρτο, αναλύονται οι τεχνολογίες με τις οποίες έγινε η ανάπτυξη της διαδικτυακής εφαρμογής όπως React, SASS, Node, Express, Grafana και PostgreSQL.

Στο πέμπτο, παρουσιάζεται η ανάπτυξη της διαδικτυακής εφαρμογής για τη διαχείριση IOT συσκευών και περιγράφεται ο τρόπος λειτουργίας της.

Στο έκτο κεφάλαιο γίνεται παρουσίαση όλων των σταδίων υλοποίησης της πιλοτικής εφαρμογής που αναπτύχθηκε για την ενεργειακή διαχείριση τοι Τμήματος Πληροφορικής με Εφαρμογές στη Βιοϊατρική του Πανεπιστημίου Θεσσαλίας.

Τέλος, στο έβδομο κεφάλαιο παρουσιάζονται τα συμπεράσματα και επίσης προτείνονται βελτιώσεις και μελλοντικές προεκτάσεις της εφαρμογής.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

ΚΕΦΑΛΑΙΟ 2 Ενεργειακή κρίση

Η έννοια της ενέργειας προσδιορίζεται από την ικανότητα ή την δυνατότητα να μπορεί κάποιος ή κάτι να βρίσκεται σε λειτουργία [1]. Η ενέργεια αποτέλεσε καθοριστικό παράγοντας για την ανάπτυξη και την εξέλιξη της ανθρωπότητας. Είναι ζωτικής σημασίας για όλες τις φυσικές διεργασίες, την ικανοποίηση των ανθρωπίνων αναγκών, της ποιότητας ζωής τους καθώς και της παγκόσμιας οικονομίας.

Η ηλεκτρική ενέργεια είναι μια δευτερεύουσα πηγή ενέργειας η οποία ρέει μέσω ηλεκτρικών γραμμών και άλλων υποδομών μεταφοράς και προέρχεται από τη μετατροπή πρωτογενών πηγών ενέργειας. Ως πρωτογενή αναφερόμαστε στην ενέργεια που βρίσκεται αυτούσια στη φύση και μπορεί να έχει πολλές μορφές συμπεριλαμβανομένης της πυρηνικής , της ορυκτής (πετρέλαιο , άνθρακας και φυσικό αέριο), αλλά και ανανεώσιμη όπως η ηλιακή , η αιολική, η γεωθερμία και η υδροηλεκτρική ενέργεια. Η ενέργεια δεν μπορεί να δημιουργηθεί ούτε να καταστραφεί αλλά μόνο να μετατραπεί από τη μια μορφή στην άλλη [2]. Εάν η ζήτηση ενέργειας δεν μπορεί να καλυφθεί με τους διαθέσιμους φυσικούς πόρους και τα αποθέματα τους, εμφανίζεται ενεργειακή κρίση.

2.1 Αίτια και συνέπειες

Στις περισσότερες πτυχές της ανθρώπινης ζωής χρησιμοποιείται ενέργεια που αντλείται στο μεγαλύτερο ποσοστό της από μη ανανεώσιμες πηγές ενέργειας. Η ενεργειακή κρίση λοιπόν δεν είναι ένα νέο φαινόμενο. Υπάρχουν πολλά ιστορικά παραδείγματα ενεργειακής κρίσης όπως η έλλειψη ξύλου στην Αγγλία του 18^{ου} αιώνα ή το εμπάργκο του πετρελαίου τη δεκαετία του ‘70. Διάφορα αίτια μπορούν να πυροδοτήσουν ενεργειακές κρίσεις. Κάποια από τα πιο σημαντικά είναι [3]:

- Υπερκατανάλωση των φυσικών πόρων
- Υπερπληθυσμός
- Ανεπάρκεια έργων υποδομής
- Μη αξιοποίηση ανανεώσιμων μορφών ενέργειας
- Σπατάλη ενέργειας
- Ανεπαρκή σύστημα διανομής
- Φυσικές καταστροφές
- Πόλεμοι
- Απρόβλεπτοι παράγοντες όπως ακραίες καιρικές συνθήκες λόγω υπερθέρμανσης του πλανήτη, αυξήσεις φόρων ή ραγδαίες πολιτικές αλλαγές

Μια τέτοια κατάσταση έχει εκτεταμένες συνέπειες και στην κοινωνική και οικονομική ζωή ενός τόπου όπως:

1. **Αύξηση της τιμής της ενέργειας**

   Μέρα με τη μέρα παρατηρείται αύξηση της ενεργειακής κατανάλωσης λόγω του υπερπληθυσμού και της βιομηχανικής και τεχνολογικής ανάπτυξης , η οποία επιφέρει τη ταχύτερη εξάντληση των ήδη περιορισμένων ανανεώσιμων πηγών ενέργειας. Συνεπώς εμφανίζεται δυσκολία στη κάλυψη των ενεργειακών απαιτήσεων και ως εκ τούτου την αύξηση της τιμής των καυσίμων και των λογαριασμών ενέργειας (π.χ. θέρμανση , υψηλότερο κόστος υπηρεσιών και νοσοκομειακής περίθαλψης)

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

2. **Συνέπειες στη βιομηχανία του τουρισμού**

   Η βιομηχανία του τουρισμού είναι άρρηκτα συνδεδεμένη με την άνοδο και πτώση των τιμών των καυσίμων κίνησης. Η δραματική αύξηση αυτών λοιπόν αποφέρει υψηλότερες τιμές μετακινήσεων καθιστώντας το συνολικό κόστος των ταξιδιών απαγορευτικό για πολύ μεγάλο ποσοστό του πληθυσμού. Ως αποτέλεσμα να πλήττονται οι οικονομίες χωρών όπου σημαντικό μέρος των εσόδων τους εξαρτάται από τουριστικές δραστηριότητες.

3. **Πολιτικές αναταραχές**

   Η αναζήτηση και αξιοποίηση των ελάχιστων ορυκτών καυσίμων προκαλεί πολύ συχνά εντάσεις μεταξύ των χωρών της υφηλίου. Επιπρόσθετα η συνεχής έλλειψη διαθέσιμων ενεργειακών πόρων κατάρρευση των αγορών ενέργειας με ιδιαίτερα σημαντικές απόρροιες για την κοινωνικοοικονομική και πολιτική κατάσταση τους.

Στη συνέχεια διευκρινίζεται η σημασία της ενεργειακής διαχείρισης και περιγράφονται οι τεχνολογίες που είναι κατάλληλες για την αντιμετώπισή της.

2.2 Τρόποι αντιμετώπισης

Κάποιοι από τος κύριους τρόπους αντιμετώπισης της ενεργειακής κρίσης είναι [4]:

- **Ανανεώσιμες πηγές ενέργειας:** Σε αντίθεση με τα ορυκτά καύσιμα, ορισμένες πηγές ενέργειας είναι πλήρως ανανεώσιμες και δεν εκπέμπουν αέρια θερμοκηπίου. Αυτές οι βιώσιμες εναλλακτικές λύσεις ενέργειας περιλαμβάνουν την ηλιακή ενέργεια, την υδροηλεκτρική ενέργεια, την αιολική ενέργεια, τη γεωθερμική ενέργεια και την ενέργεια από βιομάζα.

- **Ενεργειακή απόδοση και εξοικονόμηση ενέργειας:** Προκειμένου να αποφευχθεί μια ενεργειακή κρίση, είναι επίσης σημαντικό να καταναλώνουμε λιγότερη ενέργεια βελτιώνοντας και εκσυγχρονίζοντας τις ενεργειακές υποδομές, όπως λύσεις έξυπνων δικτύων και έξυπνες πόλεις. Είναι επίσης σημαντικό να αντικαταστήσουμε παλιές συσκευές με ενεργειακά αποδοτικές λύσεις, όπως η αντικατάσταση των παραδοσιακών λαμπτήρων με LED.

- **Μείωση της ζήτησης θέρμανσης στα σπίτια μέσω της ενεργειακής απόδοσης και της μετάβασης σε λύσεις ανανεώσιμης θερμότητας:**

  1. Θερμομόνωση και μετασκευή

  2. Δόμηση νέων κατασκευών με αντλίες θερμότητας, αντί για λέβητες αερίου ή λέβητες νέας τεχνολογίας, παθητικά ηλιακά συστήματα και ορισμένα ενεργητικά ηλιακά συστήματα

  3. Απενεργοποίηση των θερμοστατών, της θέρμανσης και του φωτισμού όταν βγαίνουμε από το δωμάτιο, παύσης υπέρμετρης χρήσης ηλεκτρικών συσκευών όπως μαγειρικές συσκευές, πλυντήρια πιάτων και ρούχων.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

- **Μείωση της ζήτησης βιομηχανικής ενέργειας (ιδιαίτερα για θέρμανση):** Η βιομηχανία έχει συνηθίσει τις χαμηλές τιμές του φυσικού αερίου τα τελευταία χρόνια, γεγονός που έχει επιβραδύνει την ανάπτυξη εναλλακτικών λύσεων. Οι υψηλές τιμές για το φυσικό αέριο σημαίνει ότι οι εταιρείες πρέπει να εξετάσουν λύσεις ανανεώσιμων πηγών θερμότητας, ιδιαίτερα ηλιακής θερμότητας, και να στραφούν στην ηλεκτροδότηση.

- **Μείωση της ζήτησης ηλεκτρικής ενέργειας, ιδιαίτερα για κλιματισμό:** Η χρήση φυσικού αερίου για ηλεκτρική ενέργεια έχει διπλασιαστεί τον τελευταίο χρόνο και σε πολλές χώρες παρέχει την περισσότερη ηλεκτρική ενέργεια. Η μείωση της ζήτησης ηλεκτρικής ενέργειας θα βοηθήσει επίσης. Αυτό που είναι πιο σημαντικό θα είναι να εξασφαλιστεί η ελάχιστη χρήση των κλιματιστικών. Ένας ανεμιστήρας οροφής καταναλώνει συνήθως 20 φορές λιγότερη ενέργεια και το σωστά σχεδιασμένο κτίριο δεν απαιτεί κλιματισμό. Καθώς οι χώρες θερμαίνονται λόγω της κλιματικής αλλαγής, δεν χρειάζεται να δούμε αύξηση στη χρήση του κλιματιστικού.

- **Μείωση της ζήτησης στις μεταφορές:** Πέρα από το φυσικό αέριο, αρκετές χώρες επιδιώκουν να μειώσουν τη ζήτηση πετρελαίου, με τις μεταφορές να αποτελούν πλέον τη μεγαλύτερη πηγή χρήσης ενέργειας σε πολλές χώρες. Βραχυπρόθεσμα, το να επιτρέπεται η εργασία από το σπίτι, η μείωση των ορίων ταχύτητας και η πραγματοποίηση εκστρατειών ενημέρωσης του κοινού για την ενθάρρυνση της στροφής στα δημόσια μέσα μεταφοράς και το ποδήλατο μπορεί να μειώσει άμεσα τη χρήση φυσικού αερίου. Μακροπρόθεσμα, η δημιουργία αυξημένων υποδομών ποδηλασίας και δημόσιων μεταφορών, η παροχή κινήτρων για μετάβαση σε ηλεκτρικά οχήματα και η προώθηση της χρήσης μικρότερων αυτοκινήτων, μπορεί επίσης να αυξήσει την ενεργειακή ασφάλεια.

2.3 Ενεργειακή Διαχείριση

Τα τελευταία χρόνια παρατηρείται συνεχής αύξηση των παγκόσμιων ενεργειακών απαιτήσεων, γεγονός που μπορεί μελλοντικά να προκαλέσει καταστροφικές συνέπειες στο περιβάλλον. Κρίνεται λοιπόν επιτακτική η ανάγκη εξοικονόμησης ενέργειας δηλαδή η προσπάθεια περιορισμού της σπατάλης των ενεργειακών αποθεμάτων στον κτιριακό τομέα όπου οι απαιτήσεις για ενέργεια είναι ιδιαίτερα υψηλές. Ενεργειακή Διαχείριση του κτιρίου, είναι μια συστηματική, οργανωμένη και συνεχής δραστηριότητα που αποτελείται από ένα προγραμματισμένο σύνολο διοικητικών, τεχνικών και οικονομικών δράσεων και στοχεύει στην εξασφάλιση συνθηκών και υπηρεσιών τέτοιων που να κάνουν την παραμονή των ενοίκων στα κτίρια ευχάριστη με την ελάχιστη δυνατή ενεργειακή κατανάλωση, και συνετή χρήση του ενεργειακού εξοπλισμού. Στην Ευρώπη ο κτιριακός τομέας καταναλώνει το 42% της συνολικής ενέργειας, το 50% των εκπομπών διοξειδίου του άνθρακα και το 35% των αερίων του θερμοκηπίου, έχοντας λοιπόν σημαντική επιρροή στο φαινομένου του θερμοκηπίου αλλά και στην οικονομική κατάσταση των νοικοκυριών. Η διαδικασία της ενεργειακής διαχείρισης αποτελείται από τέσσερα αλληλεξαρτώμενα στάδια, συγκεκριμένα τη σκέψη, το σχεδιασμό, την υλοποίηση και την καταμέτρηση. Κύρια εργαλεία για τη διαχείριση της ενέργειας αποτελούν η ενεργειακή επιθεώρηση και παρακολούθηση, ο καλύτερος σχεδιασμός συστημάτων ψύξης και θέρμανσης, η σωστή συντήρηση του εξοπλισμού, η παύση χρήσης παλαιών και ενεργοβόρων συσκευών και η λήψη μέτρων για εξοικονόμηση της ενέργειας που καταναλώνεται. Η Ευρωπαϊκή Ένωση έχει θεσπίσει κατάλληλο νομικό πλαίσιο για την καταπολέμηση του προβλήματος προσαρμοσμένο στις ανάγκες της κάθε χώρας.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Η Ελληνική κυβέρνηση ανακοίνωσε τον Κανονισμό Ενεργειακής Απόδοσης Κτιρίων (Κ.Εν.Α.Κ) ώστε να επιτευχθεί η αναβάθμιση των υπάρχων κτιρίων αλλά και η κατασκευή νέων κτιρίων με χαμηλή ενεργειακή κατανάλωση.
Σύμφωνα με την Υπουργική Απόφαση (ΚΥΑ - ΦΕΚ Β3424/2-7-2022) [5] λοιπόν θα πρέπει να επιβληθούν οι ακόλουθοι κανόνες:

1. Καθορισμός της εσωτερικής θερμοκρασίας (μέγιστης/ελάχιστης) τόσο σε θερμαινόμενα/ κλιματιζόμενα όσο και σε μη κτίρια βάσει των προδιαγραφών του προτύπου ΕΛΟΤ ΕΝ 15251:2007. Η εσωτερική θερμοκρασία των κτιρίων γραφείων του δημόσιου και του ευρύτερου δημοσίου τομέα, κατά τη θερινή περίοδο διατηρείται στους 27 ^{o} C και κατά τη χειμερινή στους 19 ^{o} C.

2. Απενεργοποίηση ψύξης/θέρμανσης σε χώρους και ώρες που δεν υπάρχουν εργαζόμενοι.

3. Απενεργοποίηση εξοπλισμού γραφείου σε χώρους και ώρες που δεν υπάρχουν εργαζόμενοι.

4. Συντήρηση συστημάτων θέρμανσης και ψύξης, συμπεριλαμβανομένων των κλιματιστικών μονάδων, τουλάχιστον μία φορά κατ’ έτος. Ειδικά για το τρέχον έτος, η προληπτική συντήρηση για τα συστήματα ψύξης πρέπει να έχει ολοκληρωθεί μέχρι τις 31.7.2022 και για τα συστήματα θέρμανσης μέχρι 31.10.2022.

5. Χρήση του νυχτερινού αερισμού των κτιρίων, όπου αυτό είναι δυνατό.

6. Σκίαση του κτιρίου προς βελτίωση της ενεργειακής του συμπεριφοράς, εφόσον αυτό δεν μειώνει σε μη ανεκτό επίπεδο τον φυσικό φωτισμό των εσωτερικών χώρων.

7. Μείωση της ενεργειακής κατανάλωσης κατά 10% τουλάχιστον, ενδεικτικά μέσω βελτιστοποίησης του χρονοπρογραμματισμού του φωτισμού των οδών και εξ ορθολογισμού του καλλωπιστικού/διακοσμητικού φωτισμού.

8. Δράσεις για τη βελτίωση της ενεργειακής αποδοτικότητας των αντλιοστασίων (π.χ. τοποθέτηση inverter).

Οι δράσεις αυτές έχουν ως κριτήρια :

- Την οικονομική αποδοτικότητα
- Τη βελτίωση της ασφάλειας, ποιότητας ζωής και παροχής υπηρεσιών στα κτίρια
- Την διατήρηση καλών περιβαλλοντικών συνθηκών
- Τον έλεγχο του συνολικού λειτουργικού ενεργειακού κόστους

Εκτιμάται ότι τα ενεργειακά οφέλη που μπορούν να προκύψουν με τη χρήση ενεργειακά αποδοτικών και οικονομικά βιώσιμων τεχνολογιών στα ευρωπαϊκά κτίρια, θα είναι μείωση της παρούσας χρήσης πετρελαίου κατά 10% και μείωση των εκπομπών ρύπων κατά 20%.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

ΚΕΦΑΛΑΙΟ 3 Το διαδίκτυο των πράγματων

Στο παρόν κεφάλαιο θα ασχοληθούμε με την έννοια του διαδικτύου των πραγμάτων IOT. Θα γίνει ιστορική αναδρομή και ορισμός των IOT και θα γίνει περιγραφή των χαρακτηριστικών τους , των τρόπων λειτουργίας τους καθώς και τριών από τα πρωτοκόλλα που χρησιμοποιούν. Τέλος θα γίνει αναφορά στη σπουδαιότητα στη σημερινή εποχή μέσω των πολλαπλών πεδίων εφαρμογής που έχουν.

3.1 Ορισμός του Διαδικτύου των πραγμάτων

Το Διαδίκτυο των πραγμάτων (IOT) είναι ένα δίκτυο φυσικών συσκευών με ενσωματωμένες υπολογιστικές δυνατότητες, ενεργοποιητών, λογισμικού και αισθητήρων που είναι σε θέση να ανιχνεύουν το περιβάλλον τους και να παράγουν, να χρησιμοποιούν και να ανταλλάσσουν δεδομένα μεταξύ τους μέσω του Διαδικτύου. Αυτές οι συσκευές συχνά αποκαλούμενες και ως «έξυπνα αντικείμενα», δίνουν τη δυνατότητα εξ αποστάσεως παρακολούθησης και ελέγχου φυσικών αντικειμένων σε πραγματικό χρόνο χρησιμοποιώντας έναν προσωπικό υπολογιστή ή μία κινητή συσκευή σε διαχειριστές δικτύου ή τελικούς χρήστες. Μπορεί να είναι τοποθετημένα σε κτήρια, οχήματα ή να είναι διάσπαρτα στο περιβάλλον. Η επικοινωνία μεταξύ αυτών των συσκευών πραγματοποιείται μέσω ενός Πρωτοκόλλου Διαδικτύου IP [6].

3.2 Ιστορική Αναδρομή του Διαδικτύου των πραγμάτων

Η ιδέα ενός δικτύου έξυπνων συσκευών εμφανίστηκε το 1982, όταν ορισμένοι φοιτητές αποφάσισαν να τροποποιήσουν ένα αυτόματο μηχάνημα Coca-Cola στο Πανεπιστήμιο Carnegie Mellon μετατρέποντας το στη πρώτη συνδεδεμένη στο ARPANET συσκευή, ικανή να αναφέρει το απόθεμά της και εάν τα ποτά που φορτώθηκαν πρόσφατα ήταν κρύα ή όχι, εξ αποστάσεως.

Η έννοια του «Διαδικτύου των πραγμάτων» εμφανίστηκε για πρώτη φορά σε μια ομιλία του Peter T. Lewis που δημοσιεύτηκε τον Σεπτέμβριο του 1985. Σύμφωνα με την οποία, «Το Διαδίκτυο των Πραγμάτων» είναι “η ενοποίηση ανθρώπων, διαδικασιών και τεχνολογίας με συνδεόμενες συσκευές και αισθητήρες για να καταστεί δυνατή η απομακρυσμένη παρακολούθηση, ο χειρισμός και η αξιολόγηση των τάσεων τέτοιων συσκευών”.

Ο όρος "Διαδίκτυο των πραγμάτων" επινοήθηκε το 1999 από τον Kevin Ashton της Procter & Gamble, ο οποίος πρότεινε την τοποθέτηση τσιπ αναγνώρισης ραδιοσυχνοτήτων (RFID) σε προϊόντα για την παρακολούθηση τους μέσω μιας αλυσίδας εφοδιασμού. Θεώρησε την αναγνώριση ραδιοσυχνοτήτων (RFID) ως απαραίτητη για το Διαδίκτυο των πραγμάτων ώστε οι υπολογιστές να διαχειρίζονται όλα τα διαφορετικά έξυπνα αντικείμενα.

Την επόμενη δεκαετία όλο και περισσότερες συνδεδεμένες συσκευές κυκλοφόρησαν στην αγορά εγείροντας το ενδιαφέρον του κοινού για την τεχνολογία IoT. Το 2000, η LG ανακοίνωσε το πρώτο έξυπνο ψυγείο, το 2007 κυκλοφόρησε το πρώτο iPhone και μέχρι το 2008, ο αριθμός των συνδεδεμένων συσκευών ξεπέρασε τον αριθμό των ανθρώπων στον πλανήτη. Το 2009 η τεχνολογία "Internet of Things" καθιερώθηκε από τον Tim Bray, που εκείνη τη χρονική στιγμή ήταν εργαζόμενος της Google, και η ίδια άρχισε τις δοκιμές αυτοκινήτων χωρίς οδηγό. Τέλος το 2011 εμφανίστηκε ο έξυπνος θερμοστάτης Nest της Google ο οποίος επέτρεπε τον απομακρυσμένο έλεγχο της κεντρικής θέρμανσης [7].

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

3.3 Χαρακτηριστικά του Διαδικτύου των πραγμάτων

Κάποια από τα θεμελιώδη χαρακτηριστικά του IoT είναι τα εξής [8]:

- **Διασυνδεσιμότητα:** Οποιαδήποτε συσκευή μπορεί να συνδεθεί με την παγκόσμια υποδομή πληροφόρησης και επικοινωνίας.

- **Ετερογένεια συσκευών:** Οι συσκευές που είναι συνδεδεμένες στο Διαδίκτυο είναι ετερογενείς, καθώς βασίζονται σε διαφορετικές πλατφόρμες και δίκτυα. Παρόλα αυτά μπορούν να αλληλοεπιδρούν με άλλες συσκευές ή πλατφόρμες υπηρεσιών, μέσω διαφορετικών δικτύων.

- **Δυναμικές αλλαγές:** Η κατάσταση των συσκευών μπορεί να αλλάξει δυναμικά π.χ. είναι ενεργές ή απενεργοποιημένες και επίσης η θέση και η ταχύτητά τους.

- **Μεγάλη κλίμακα χρηστών:** Το IOT εξυπηρετεί πολλούς και διαφορετικούς χρήστες. Υπάρχουν τρεις σημαντικές κατηγορίες χρηστών με διαφορετικές ανάγκες η καθεμία:
    1. Μεμονωμένοι πολίτες,
    2. Μια κοινότητα πολιτών (π.χ. πολίτες μιας πόλης, περιφέρειας, χώρας ή της κοινωνίας στο σύνολό της)
    3. Επιχειρήσεις

- **Ασφάλεια:** Δίνεται προτεραιότητα στον ασφαλή σχεδιασμό των συστημάτων IOT έτσι ώστε να εξασφαλίζεται η προστασία των προσωπικών δεδομένων και της ιδιωτικής ζωής των χρηστών.

- **Συνδεσιμότητα:** Η συνδεσιμότητα αφορά την προσβασιμότητα και τη συμβατότητα του δικτύου. Η προσβασιμότητα επιτρέπει τη εύκολη σύνδεση μιας συσκευής σε ένα δίκτυο, ενώ η συμβατότητα παρέχει την δυνατότητα δημιουργίας και χρήσης δεδομένων του δικτύου.

3.4 Πρωτοκολλά

Για τη σύνδεση των συσκευών του IOT είναι απαραίτητη η χρήση ενός πρωτοκόλλου. Υπάρχουν πολλά πρωτόκολλα συσκευών IOT όπως τα DDS (Data Distribution Service), CoAP (Constrained Application Protocol), XMPP (Extensive Messaging & Presence Protocol) κ.λπ. τα οποία μπορούν να υλοποιηθούν χρησιμοποιώντας διαφορετικές γλώσσες προγραμματισμού όπως Java ή Python. Οι περισσότερες συσκευές όμως χρησιμοποιούν το πρωτόκολλο HTTP το οποίο υποστηρίζεται ευρέως από όλα τα μεγάλα προγράμματα περιήγησης και εφαρμογές για κινητά. Το κύριο πλεονέκτημα της χρήσης αυτού του πρωτοκόλλου είναι ότι παρέχει ασφαλή σύνδεση μεταξύ διαφορετικών συσκευών και δικτύων [9].

1. Το πρώτο πρωτόκολλο που θα αναλυθεί είναι το Δίκτυο αισθητήρων κατανεμημένων συσκευών (DDS). Το DDS είναι ένα πρωτόκολλο για επικοινωνία σε πραγματικό χρόνο μεταξύ συσκευών και διακομιστών. Παρέχει χαμηλό κόστος και υψηλή απόδοση με μικρά πακέτα που προωθούνται εύκολα μέσω ασύρματων καναλιών. Χρησιμοποιεί το UDP (User Datagram Protocol) για την επικοινωνία με τον διακομιστή. Τα δεδομένα που αποστέλλονται από τη μια συσκευή στην άλλη ονομάζονται μήνυμα και μπορεί να είναι είτε

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

εντολή είτε ειδοποίηση συμβάντος ή οποιοσδήποτε άλλος τύπος δεδομένων που πρέπει να κοινοποιηθεί μεταξύ δύο συσκευών συνδεδεμένων μέσω πρωτοκόλλου DDS.

2. Άλλο ένα είναι το Περιορισμένο Πρωτόκολλο Εφαρμογής (CoAP). Αυτό αναπτύχθηκε για συστήματα δικτύωσης που βασίζονται σε πακέτα που βασίζονται σε TCP/IP μέσω του UDP. Το CoAP είναι επίσης γνωστό ως HTTP μέσω UDP (ή TCP). Αναπτύχθηκε από την Google, τη Microsoft και τη Nokia. Ο κύριος σκοπός πίσω από την ανάπτυξη αυτού του πρωτοκόλλου ήταν ότι επιτρέπει την αποστολή μηνυμάτων από υπηρεσίες ιστού χωρίς να είναι γνωστή η υποκείμενη τοπολογία δικτύου. Έτσι διευκολύνεται η πρόσβαση των χρηστών σε πληροφορίες μέσω κινητών τηλεφώνων ή άλλων συσκευών όπως φορητούς υπολογιστές/επιτραπέζιους υπολογιστές κ.λπ.

3. Το πρωτόκολλο XMPP είναι ένα σύστημα ανταλλαγής μηνυμάτων σε πραγματικό χρόνο. Χρησιμοποιεί έγγραφα XML για την ανταλλαγή μηνυμάτων μεταξύ δύο τελικών σημείων που συνδέονται μέσω ενός προγράμματος ανταλλαγής άμεσων μηνυμάτων, όπως το Jabber ή το Google Talk. Χρησιμοποιείται συχνά κατά την ανάπτυξη εφαρμογών που βασίζονται στην τεχνολογία RFID, επειδή οι περισσότερες ετικέτες RFID υποστηρίζουν αυτόν τον τύπο πρωτοκόλλου ανταλλαγής μηνυμάτων, αλλά δεν απαιτούν ειδική εγκατάσταση υλικού.

4. Το MQTT είναι ένα τυπικό πρωτόκολλο ανταλλαγής μηνυμάτων για το Διαδίκτυο των πραγμάτων (IOT). Έχει σχεδιαστεί ως ένα εξαιρετικά ελαφρύ μέσο μεταφοράς μηνυμάτων που είναι ιδανικό για τη σύνδεση απομακρυσμένων συσκευών με μικρό αποτύπωμα κώδικα και ελάχιστο εύρος ζώνης δικτύου. Το MQTT χρησιμοποιείται σήμερα σε μια μεγάλη ποικιλία βιομηχανιών, όπως η αυτοκινητοβιομηχανία, η μεταποίηση, οι τηλεπικοινωνίες, το πετρέλαιο και το φυσικό αέριο κ.λπ. [10].

5. Το AMQP είναι ένα ευρέως χρησιμοποιούμενο πρωτόκολλο ανταλλαγής μηνυμάτων που χρησιμοποιείται στη διαδικασία ανάπτυξης εφαρμογών ανοιχτού κώδικα. Επιτρέπει τη δια λειτουργικότητα των μηνυμάτων μεταξύ συστημάτων, ανεξάρτητα από το broker μηνυμάτων ή την πλατφόρμα που χρησιμοποιείται. Το AMQP είναι ένα πρωτόκολλο επιπέδου εφαρμογής που επιτρέπει στις client εφαρμογές να συνομιλούν με τον διακομιστή και να αλληλοεπιδρούν. Ορίζει τόσο το πρωτόκολλο του επιπέδου δικτύου όσο και μια αρχιτεκτονική υψηλού επιπέδου για τους brokers μηνυμάτων [11]. Οι κύριες λειτουργίες αυτού του πρωτοκόλλου IOT είναι οι εξής: Λήψη και τοποθέτηση μηνυμάτων σε ουρές, αποθήκευση μηνυμάτων και ρύθμιση σχέσης μεταξύ αυτών των στοιχείων.

3.5 Πεδία εφαρμογής του Διαδικτύου των πραγμάτων

Η τεχνολογία του IOT είναι ιδιαίτερα σημαντική στη σημερινή εποχή λόγω της πληθώρας εφαρμογών που έχει. Μπορεί να εφαρμοστεί σε τεράστιο φάσμα διαφορετικών τομέων όπως η υγειονομική περίθαλψη, η γεωργία, οι μεταφορές. οι επιχειρήσεις, και η βιομηχανία [12].

- **Κτιριακός και οικιακός αυτοματισμός:** Οι συσκευές IOT μπορούν να χρησιμοποιηθούν για την παρακολούθηση και τον έλεγχο των μηχανικών, ηλεκτρικών και ηλεκτρονικών συστημάτων που χρησιμοποιούνται σε διάφορους τύπους κτιρίων (π.χ. δημόσια και ιδιωτικά, βιομηχανικά, ιδρύματα ή κατοικίες) σε συστήματα οικιακού αυτοματισμού. Ο όρος "έξυπνο κτίριο" υπάρχει στο κόσμο της τεχνολογίας και της ενεργειακής εξοικονόμησης εδώ και αρκετό καιρό. Τα συστήματα IOT μπορούν να χρησιμοποιηθούν για τη δημιουργία έξυπνων σπιτιών όπου οι χρήστες μπορούν να ελέγχουν οποιαδήποτε οικιακή ηλεκτρική συσκευή όπως τη τηλεόραση, το πλυντήριο, το κλιματιστικό τους κτλ., να

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

ενεργοποιούν ή απενεργοποιούν τα συστήματα θέρμανσης ακόμη και να αλλάζουν τις συνθήκες φωτισμού του σπιτιού τους από οπουδήποτε χρησιμοποιώντας το κινητό τους τηλέφωνο ή το tablet τους. Αυτές οι συσκευές συνδέονται μέσω διαδικτύου ή πύργων κινητής τηλεφωνίας, ώστε να μπορούν να λαμβάνουν εντολές από απομακρυσμένες τοποθεσίες χωρίς να έχουν φυσικά σημεία πρόσβασης, όπως παραδείγματος χάρη διακόπτες. Τα συστήματα ενεργειακής διαχείρισης(Building Management Systems, BMS) ξεκίνησαν ως ένα σύνολο συστημάτων αυτοματισμού τα οποία διέθεταν διαφορετικά πρωτόκολλα επικοινωνίας. Με την ενσωμάτωση BMS κτιρίων μπορεί επίσης να επιτευχθεί η βελτίωση της απόδοσης της παραγωγής και της διανομής ηλεκτρικής ενέργειας και η δημιουργία ενεργειακά αποδοτικών και βασισμένων στο IOT «έξυπνων κτιρίων».

- **Ιατρική και υγειονομική περίθαλψη:** Το Internet of Medical Things (IoMT) που έχει αναφερθεί ως "Έξυπνη Υγεία" είναι μια εφαρμογή του IOT για τη δημιουργία ενός ψηφιοποιημένου συστήματος υγειονομικής περίθαλψης, που συνδέει τους διαθέσιμους ιατρικούς πόρους και τις υπηρεσίες υγειονομικής περίθαλψης. Οι συσκευές IOT μπορούν να χρησιμοποιηθούν για τη συλλογή και ανάλυση δεδομένων για έρευνα, την ενεργοποίηση συστημάτων απομακρυσμένης παρακολούθησης της υγείας όπως συσκευές παρακολούθησης της αρτηριακής πίεσης και του καρδιακού ρυθμού έως προηγμένες συσκευές ικανές να παρακολουθούν εξειδικευμένα εμφυτεύματα, όπως βηματοδότες ή προηγμένα ακουστικά βαρηκοΐας. Η εφαρμογή του IOT στην υγειονομική περίθαλψη διαδραματίζει θεμελιώδη ρόλο στη διαχείριση χρόνιων ασθενειών και στην πρόληψη και τον έλεγχο ασθενειών. Επιτρέπει στους επαγγελματίες υγείας να καταγράφουν τα δεδομένα του ασθενούς και να τα αναλύουν εφαρμόζοντας πολύπλοκους αλγόριθμους για να αποφασιστεί το πιο κατάλληλο πλάνο θεραπείας τους. Ορισμένα νοσοκομεία ακόμη έχουν εφαρμόσει την τεχνολογία των «έξυπνων κρεβατιών» που μπορούν να ανιχνεύουν τη διαθεσιμότητα τους ή τις κινήσεις του ασθενή. Επιπρόσθετα τα αρχεία ασθενών αποθηκεύονται σε μια βάση δεδομένων, επιτρέποντας στους γιατρούς και το υπόλοιπο ιατρικό προσωπικό να έχει πρόσβαση στις πληροφορίες των ασθενών. Τέλος πολλοί άνθρωποι έχουν ήδη υιοθετήσει wearable συσκευές για να παρακολουθούν την φυσική τους άσκηση, τον ύπνο ή άλλες συνήθειες τους, όπως ζυγαριές ή φορητές οθόνες καρδιάς.

- **Φροντίδα ηλικιωμένων:** Άλλη μία δυνατότητα που μας προσφέρει η εφαρμογή των συστημάτων υγείας IOT σε συνδυασμό με τις τεχνολογίες ενός έξυπνου σπιτιού είναι η παροχή βοήθειας σε ηλικιωμένα άτομα και σε άτομα με αναπηρίες. Αυτά τα οικιακά συστήματα χρησιμοποιούν υποβοηθητική τεχνολογία για την αντιμετώπιση των ειδικών αναγκών ενός ιδιοκτήτη. Ο φωνητικός έλεγχος μπορεί να βοηθήσει τους χρήστες με περιορισμούς όρασης και κινητικότητας, ενώ τα συστήματα προειδοποίησης μπορούν να συνδεθούν απευθείας με κοχλιακά εμφυτεύματα που φοριούνται από χρήστες με προβλήματα ακοής. Μπορούν επίσης να εξοπλιστούν με πρόσθετα χαρακτηριστικά ασφαλείας, συμπεριλαμβανομένων αισθητήρων που παρακολουθούν για ιατρικά επείγοντα περιστατικά όπως πτώσεις ή επιληπτικές κρίσεις. Η τεχνολογίες του έξυπνου σπιτιού με αυτόν τον τρόπο μπορούν να προσφέρουν στους χρήστες περισσότερη ελευθερία και υψηλότερη ποιότητα ζωής.

- **Περιβαλλοντική παρακολούθηση:** Οι εφαρμογές περιβαλλοντικής παρακολούθησης του IOT χρησιμοποιούν συνήθως αισθητήρες για να βοηθήσουν στην προστασία του περιβάλλοντος παρακολουθώντας την ποιότητα του αέρα ή του νερού, τις ατμοσφαιρικές ή εδαφικές συνθήκες, τις κινήσεις της άγριας ζωής και των οικοτόπων τους. Επίσης εφαρμογές όπως συστήματα έγκαιρης προειδοποίησης για τσουνάμι ή σεισμό μπορούν να χρησιμοποιηθούν από τις υπηρεσίες έκτακτης ανάγκης για την παροχή πιο αποτελεσματικής βοήθειας. Οι συσκευές IOT καλύπτουν συνήθως μια μεγάλη γεωγραφική

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

περιοχή και μπορούν επίσης να είναι κινητές. Τέλος η παρακολούθηση και ο έλεγχος των λειτουργιών βιώσιμων αστικών και αγροτικών υποδομών όπως γέφυρες, σιδηροδρομικές γραμμές και αιολικά πάρκα και τομείς όπως η διαχείριση απορριμμάτων μπορούν να ωφεληθούν από την αυτοματοποίηση που θα μπορούσε να φέρει το IOT.

- **Ωκεανός των πραγμάτων:** Το έργο Ocean of Things είναι ένα πρόγραμμα υπό την ηγεσία της DARPA που έχει σχεδιαστεί για τη δημιουργία ενός Διαδικτύου πραγμάτων σε μεγάλες ωκεάνιες περιοχές με σκοπό τη συλλογή, την παρακολούθηση και την ανάλυση δεδομένων περιβαλλοντικής δραστηριότητας και σκαφών. Το έργο συνεπάγεται την ανάπτυξη περίπου 50.000 πλωτήρων που στεγάζουν ένα σύνολο παθητικών αισθητήρων που ανιχνεύει και παρακολουθεί αυτόνομα στρατιωτικά και εμπορικά σκάφη ως μέρος ενός δικτύου που βασίζεται σε cloud.

- **Γεωργία:** Υπάρχουν πολυάριθμες εφαρμογές του IOT στη γεωργία, όπως η συλλογή δεδομένων για τη θερμοκρασία, τις βροχοπτώσεις, την υγρασία, την ταχύτητα του ανέμου, την προσβολή από παράσιτα και το περιεχόμενο του εδάφους. Μπορούν ακόμα να εφαρμόζουν τα δεδομένα που αποκτήθηκαν σε προγράμματα λίπανσης ακριβείας ώστε οι αγρότες να γνωρίζουν ακριβώς τι θρεπτικά συστατικά πρέπει να προστεθούν σε κάθε στάδιο του κύκλου ανάπτυξης των φυτών. Ο γενικός στόχος της χρήσης τέτοιων τεχνολογιών είναι να συνδυαστεί η γνώση και η διαίσθηση των αγροτών για τις εκτάσεις τους με τα δεδομένα που λαμβάνονται από τέτοιου είδους αισθητήρες, ώστε να επιτευχθεί η αυτοματοποίηση των γεωργικών τεχνικών, η ελαχιστοποίηση του κινδύνου, η μείωση της προσπάθειας που απαιτείται για τη διαχείριση των καλλιεργειών, η αύξηση της παραγωγικότητας του αγροκτήματος και επίσης η μείωση του κόστους.

- **Μεταφορές:** Το IOT μπορεί να βοηθήσει στην ενοποίηση των επικοινωνιών, του ελέγχου και της επεξεργασίας πληροφοριών σε διάφορα συστήματα μεταφορών. Η εφαρμογή του IoT επεκτείνεται σε όλες τις πτυχές των συστημάτων μεταφοράς (δηλαδή το όχημα, την υποδομή και τον οδηγό ή τον χρήστη). Η δυναμική αλληλεπίδραση μεταξύ αυτών των στοιχείων ενός συστήματος μεταφορών επιτρέπει την επικοινωνία μεταξύ οχημάτων αλλά και εντός του οχήματος, έξυπνο έλεγχο της κυκλοφορίας, έξυπνο πάρκινγκ, ηλεκτρονικά συστήματα είσπραξης διοδίων, ασφάλεια και οδική βοήθεια.

- **Βιομηχανικές εφαρμογές:** Το IOT μπορεί να συνδέσει διάφορες κατασκευαστικές συσκευές εξοπλισμένες με δυνατότητες ανίχνευσης, αναγνώρισης, επεξεργασίας, επικοινωνίας, ενεργοποίησης και δικτύωσης. Ο έλεγχος και η διαχείριση του δικτύου του εξοπλισμού κατασκευής και της διαδικασίας παραγωγής επιτρέπουν τη χρήση του IOT για βιομηχανικές εφαρμογές και έξυπνη κατασκευή. Τα ευφυή συστήματα IOT επιτρέπουν την ταχεία κατασκευή και βελτιστοποίηση νέων προϊόντων, καθώς και την ταχεία ανταπόκριση στις απαιτήσεις των προϊόντων.

- **Ψηφιοποίηση προϊόντων:** Υπάρχουν πολλές εφαρμογές έξυπνης ή ενεργής συσκευασίας στις οποίες ένας κωδικός QR ή μια ετικέτα NFC είναι τοποθετημένος στη συσκευασία ενός προϊόντος. Η ίδια η ετικέτα είναι παθητική, ωστόσο, περιέχει ένα μοναδικό αναγνωριστικό (συνήθως μια διεύθυνση URL) που επιτρέπει σε έναν χρήστη να έχει πρόσβαση σε ψηφιακό περιεχόμενο σχετικά με το προϊόν μέσω ενός smartphone. Ο έλεγχος ταυτότητας των μοναδικών αναγνωριστικών, και επομένως του ίδιου του προϊόντος, είναι δυνατός μέσω ενός ψηφιακού υδατογραφήματος ευαίσθητου σε αντιγραφή ή μοτίβου ανίχνευσης αντιγραφής για σάρωση κατά τη σάρωση ενός κωδικού QR. Ο όρος "Internet of Packaging" λοιπόν επινοήθηκε για να περιγράψει αυτές τις εφαρμογές στις οποίες χρησιμοποιούνται μοναδικά αναγνωριστικά, για την αυτοματοποίηση των αλυσίδων εφοδιασμού.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

ΚΕΦΑΛΑΙΟ 4 Τεχνολογίες υλοποίησης

Σε αυτήν την ενότητα παρουσιάζονται οι τεχνολογίες οι οποίες χρησιμοποιήθηκαν για την υλοποίηση της πτυχιακής εργασίας. Πιο συγκεκριμένα θα γίνει ανάλυση της , HTML5, CSS 3, SASS, JavaScript και της βιβλιοθήκης JavaScript React.JS , η node.js και η βιβλιοθήκη της express JS. Ως βάση δεδομένων χρησιμοποιείται η PostgreSQL για την αποθήκευση των χρηστών και των συσκευών αλλά και την ανάκτηση των δεδομένων από τους αισθητήρες για τη γραφική απεικόνιση. Και τέλος για τη γραφική απεικόνιση γίνεται χρήση του Grafana.

4.1 HTML5

Εικόνα 4.1

Η γλώσσα HTML (Hyper Text Markup Language , Γλώσσα Σήμανσης Υπερκειμένου) είναι ένα από τα πιο βασικά δομικά κομμάτια του διαδικτύου. Η HTML δεν είναι γλώσσα προγραμματισμού αλλά γλώσσα σήμανσης η οποία χρησιμοποιείται για κατασκευή ιστοσελίδας. Βασίζεται στη γλώσσα SGML (Standard Generalized Markup Language) οπού είναι ένα μεγαλύτερο σύστημα επεξεργασίας εγγράφων και βασική για τη δόμηση ιστοσελίδων. Με άλλα λόγια είναι υπεύθυνη για τη δομή τη τοποθέτηση και τη μορφή του περιεχομένου στη σελίδα. Αποτελείται από δομικά στοιχεία που χαρακτηρίζονται από ετικέτες (tags) ,όπου ανάμεσα τους μπορεί να τοποθετηθεί κείμενο , πίνακες , εικόνες , λίστες κλπ. Οι ετικέτες λειτουργούν ανά ζεύγη , μια ετικέτα έναρξης (π.χ. <h1>) και μια ετικέτα λήξης (π.χ. </h1) η οποία αναγνωρίζεται μέσω του χαρακτηριστικού συμβόλου “/”. Ο φυλλομετρητής Browser διαβάζει τα έγγραφα HTML και ερμηνεύει τα στοιχεία τους συνθέτοντας μια ιστοσελίδα και εμφανίζοντας τη στην οθόνη. Το περιεχόμενο των εγγράφων καθορίζεται από τα tags που περιέχονται.

Η HTML5 είναι η πέμπτη και η τρέχουσα έκδοση του προτύπου HTML η οποία κυκλοφόρησε επίσημα το 2014. Αποτελεί ένα σύνολο τεχνολογιών και όχι απλά μια νέα έκδοση της γλώσσας HTML. Ενσωματώνει όλα τα χαρακτηριστικά των προηγούμενων εκδόσεων προσθέτοντας και νέα χαρακτηριστικά δομής και σύνταξης όπως <nav> <article> <section> και <header> κλπ. Απλοποιείτε η προσθήκη στοιχείων πολυμέσων μέσω των ετικετών <audio> για εισαγωγή ήχου και <video> για εισαγωγή βίντεο. Ακόμα βοηθά τις μηχανές αναζήτησης να ανιχνεύουν τον ισότοπο πιο αποδοτικά. Τέλος η HTML5 αξιοποιείται και για τη δημιουργία εφαρμογών διαδικτύου για φορητές συσκευές όπως tablet και κινητά τηλέφωνα καθώς σε συνεργασία με τη τεχνολογία CSS οι εφαρμογές προσαρμόζονται στις διαστάσεις της κάθε οθόνης [13].

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

4.2 CSS

Εικόνα 4.2

Η CSS (Cascading Style Sheets) είναι μια γλώσσα που αποτελείται από διαδοχικά φύλλα ύφους που χρησιμοποιείται για να περιγράψει την παρουσίαση ενός εγγράφου γραμμένου σε γλώσσα σήμανσης HTML ή XML συμπεριλαμβανομένων και των SVG, MathML, XUL και XHTML [14]. Θεωρείται αναπόσπαστο κομμάτι των διαδικτύου για τη δημιουργία αισθητικά άρτιων και ανταποκρίσιμων σελίδων και εφαρμογών ακόμα και για αυτές που προορίζονται για κινητές συσκευές. Χρησιμοποιείται για τον έλεγχο της εμφάνισης των περιεχομένων μιας ιστοσελίδας όπως χρώματα , μέγεθος γραμματοσειρά, στοίχιση και φόντο.
Υπάρχουν τρεις τρόποι εισαγωγής ενός φύλλου στυλ για τη μορφοποίησή της HTML σελίδας : [15]

- **External CSS**
  Για την μορφοποίηση δημιουργούνται αρχεία με κατάληξη ".css" τα οποία είναι ξεχωριστά από τα αρχεία της HTML. Κάθε HTML σελίδα πρέπει να περιέχει μια αναφορά στο αρχείο CSS μέσα στην ετικέτα <head>. Έτσι δίνεται η δυνατότητα να γίνει η αλλαγή της εμφάνισης μιας ολόκληρης ιστοσελίδας τροποποιώντας μόνο ένα αρχείο.

- **Internal CSS**
  Χρησιμοποιείται για την αλλαγή του στυλ μόνο μιας σελίδας HTML. Δηλώνεται μέσα στην ετικέτα <style> που περικλείεται από την ετικέτα <head>.

- **Inline CSS**
  Εφαρμόζεται για την τροποποίηση της εμφάνισης ενός μόνο στοιχείου του κώδικα π.χ. This is a paragraph. που περιλαμβάνεται στην ετικέτα <body>.

4.2.1 SASS

Η Sass (Syntactically awesome style sheets) είναι μια γλώσσα ύφους που ερμηνεύεται ή μεταγλωττίζεται σε Cascading Style Sheetsσε (CSS). Επεκτείνει τη CSS βελτιώνοντας τη σύνταξη της, ώστε να της παρέχει επιπλέον δυνατότητες και βολικά εργαλεία. Τα Sass αρχεία αποθηκεύονται παραδοσιακά με επεκτάσεις .sass και .scss, αντίστοιχα. Μερικές από τις δυνατότητες που παρέχει είναι η χρήση μεταβλητών , μαθηματικών πράξεων , Mixin , βρόχων , συναρτήσεων, nesting κτλ.
Με τη χρήση των μεταβλητών και την επαναχρησιμοποίηση τους σε διάφορα τμήματα των αρχείων SASS δεν επαναλαμβάνονται τμήματα του κώδικα και επιτυγχάνεται η απλοποίηση του. Ακόμα με τη χρήση nesting δημιουργείται μια πιο φυσική και ευανάγνωστη σύνταξη του κώδικα και αποφεύγεται η επαναχρησιμοποίηση επιλογέων (Selectors) [16].

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

4.2.2 Responsive Design

Με τη πάροδο των χρόνων όλο και μεγαλύτερο μέρος του πληθυσμού χρησιμοποιεί στη καθημερινότητα του φορητές συσκευές για τη περιήγηση του στο διαδίκτυο. Δημιουργήθηκε έτσι η ανάγκη οι ιστοσελίδες να εμφανίζονται με το κατάλληλο τρόπο σε πληθώρα οθονών για τις διάφορες συσκευές όπως κινητά τηλέφωνα , tablets , κονσόλες παιχνιδιών , τηλεοράσεις και wearables. Μέσω του responsive design (ανταποκρινόμενος - προσαρμόσιμος σχεδιασμός) κάτι τέτοιο είναι εφικτό επειδή οι σελίδες προσαρμόζονται αυτόματα στο κινητό τηλέφωνο ή ταμπλέτα του χρήστη χωρίς να χρειαστεί αυτός να κάνει μεγέθυνση/σμίκρυνση η αλλαγή προσανατολισμού της οθόνης. Κάποια από τα πλεονεκτήματα είναι ότι προσφέρει ένα πιο ελκυστικό περιβάλλον περιήγησης καθώς προσαρμόζεται αυτόματα σε οποιαδήποτε ανάλυση. Τέλος με το responsive design δεν υπάρχει η ανάγκη δημιουργίας ξεχωριστών ιστοσελίδων και έτσι η σχεδίαση τους είναι οικονομικότερη. Στη συγκεκριμένη πτυχιακή χρησιμοποιήθηκαν CSS τεχνικές διάταξης όπως το flexbox και Media Queries [17].

4.3 JavaScript

Εικόνα 4.3

Η JavaScript είναι μια διερμηνευμένη γλώσσα προγραμματισμού υψηλού επιπέδου. Είναι μία από τις 3 κύριες τεχνολογίες του παγκόσμιου ιστού μαζί με την HTML και την CSS και είναι η πιο δημοφιλής scripting γλώσσα στον κόσμο. Είναι μια δομημένη γλώσσα αφότου είναι επηρεασμένη από τη C. Είναι δυναμική στη σύνταξη της και αντικειμενοστραφής με συναρτήσεις ως αντικείμενα πρώτης τάξης. Ο κώδικας της JavaScript μπορεί είτε να βρίσκεται εσωτερικά μαζί με τον κώδικα της HTML στο ίδιο αρχείο είτε εξωτερικά σε ξεχωριστό αρχείο. Τα αρχεία της JavaScript έχουν κατάληξη ".js". Αρχικά χρησιμοποιήθηκε ως client-side γλώσσα προγραμματισμού, δηλαδή για τη συγγραφή κώδικα από τη πλευρά του πελάτη. Στη συνέχεια χρησιμοποιήθηκε και για τη συγγραφή κώδικα από την πλευρά του διακομιστή , ειδικά μετά τη διάδοση του Node.js ενός μοντέλου προγραμματισμού βασισμένο στα events (γεγονότα) [18].

Κάποιες από τις πιο συνηθισμένες χρήσεις είναι :

1. Δίνουν τη δυνατότητα στους χρήστες να αλληλοεπιδρούν με τις σελίδες που έχουν δημιουργηθεί με HTML και CSS. Μερικά παραδείγματα είναι:
   - Drop-down μενού
   - Καρουζέλ εικόνων
   - Αναπαραγωγή εικόνας και βίντεο
   - Εμφάνιση γραφικών στοιχείων
   - Αναδυόμενα παράθυρα
   - Εμφάνιση χρονοδιακόπτη η αντίστροφης μέτρησης

2. Δημιουργία διαδικτυακών και εφαρμογών για κινητά.
   Μερικά παραδείγματα εφαρμογών αποτελούν τη PayPal, LinkedIn , Netflix.
3. Δημιουργία διακομιστών ιστού και ανάπτυξη εφαρμογών διακομιστή
4. Ανάπτυξη παιχνιδιών φυλλομετρητή.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

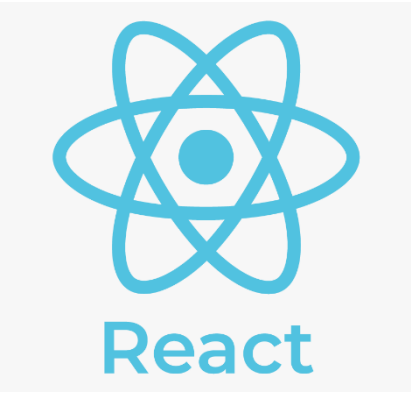
4.4 Bootstrap

Εικόνα 4.4

Το Bootstrap είναι ένα από τα διαδεδομένα Front-end πλαίσια για τη δημιουργία σελίδων και διαδικτυακών εφαρμογών. Περιλαμβάνει πρότυπα σχεδίασης που βασίζονται σε HTML και CSS για συνεπή τυπογραφία, διαδικτυακές φόρμες, κουμπιά πλοήγησης πίνακες καθώς και προαιρετικές επεκτάσεις JavaScript. Ακόμα ένα από τα βασικά χαρακτηριστικά του είναι ότι έχει σχεδιαστεί για να παρέχει τη δυνατότητα να δημιουργούνται ευκολά responsive ιστοσελίδες [19]. Στη πτυχιακή έχει χρησιμοποιηθεί μόνο για την οθόνη εισόδου (sign-in , sign-out) για ελαχιστοποίηση του χρόνου ανάπτυξης της σελίδας.

4.5 React

Αρχικά θα γίνει ορισμός των πλαισίων λογισμικού (frameworks) της JavaScript και στη συνέχεια θα γίνει αναφορά στις βιβλιοθήκες της JavaScript και τέλος ανάλυση της React. Τα πλαίσια λογισμικού (frameworks) είναι μια συλλογή από βιβλιοθήκες οι οποίες διευκολύνουν την ανάπτυξη ιστοτόπων και εφαρμογών με ευρείες δυνατότητες και λειτουργίες επεκτείνοντας τη JavaScript βελτιώνοντας τη σύνταξη της. Τα πιο δημοφιλή frameworks είναι τα Angular JS, Vue JS και άλλα. Μερικά από τα πλεονεκτήματα τους είναι η απλούστευση και καλύτερη διαχείριση του κώδικα και η επιλεκτική εμφάνιση (selective rendering) η οποία ανανεώνει μόνο τα στοιχεία της σελίδας που αλλάζουν κάθε φορά. Τα JavaScript Frameworks είναι συνήθως βασισμένα στο αρχιτεκτονικό πρότυπο MVC (Model-view-controller) αρχιτεκτονικής. Οι βιβλιοθήκες JavaScript είναι ήδη γραμμένος κώδικας JavaScript που επιτρέπει την ευκολότερη ανάπτυξη εφαρμογών που βασίζονται στη JavaScript. Ορισμένες βιβλιοθήκες όπως η Angular ταξινομούνται ως frameworks καθώς παρουσιάζουν δυνατότητες full-stack ενώ βιβλιοθήκες όπως η React ενώ θεωρείται ότι είναι framework από τους προγραμματιστές ουσιαστικά είναι μια Front-End βιβλιοθήκη. Μεγάλες εταιρίες όπως η Google ανάπτυξε το Framework Angular ενώ το Facebook και το Instagram δημιούργησαν τη βιβλιοθήκη React [20].



Εικόνα 4.5

Η React.js (γνωστό και ως ReactJS ή React) είναι μια βιβλιοθήκη JavaScript ανοιχτού κώδικα, front end. Δημιουργήθηκε το 2013 από τον Jordan Walke, ο οποίος εργάζεται στο Facebook ως μηχανικός λογισμικού. Σχεδιάστηκε για τη δημιουργία εφαρμογών μιας σελίδας (Single-Page Application) και παρέχει επαναχρησιμοποιήσιμα στοιχεία διεπαφής στο χρήστη. Ακόμα χρησιμεύει ως βάση κατά την ανάπτυξη εφαρμογών για κινητά. Παρέχει μια αποτελεσματική διεπαφή χρήστη κατά την ανάπτυξη δια δραστικών διαδικτυακών εφαρμογών. Τέλος κάνει πιο εύκολη την αποσφαλμάτωση (debugging) και τη δοκιμή (testing) [21].

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Μερικά από τα βασικά χαρακτηριστικά του βιβλιοθήκης React είναι ότι [22]:

- Υποστηρίζει JavaScript XML (JSX) που συνδυάζει JavaScript και HTML.
- Η React ξεπερνά την αναποτελεσματικότητα του DOM χρησιμοποιώντας το λεγόμενο Virtual DOM. Ακριβώς όπως το πραγματικό DOM, το Virtual DOM αντιπροσωπεύει όλα τα στοιχεία και τις ιδιότητες τους ως δέντρο κόμβου. Όταν κάτι αλλάζει, η React ενημερώνει το Virtual DOM και διακρίνοντας τις διαφορές από το πραγματικό DOM, το ενημερώνει μόνο με ό,τι έχει πραγματικά αλλάξει.
- Είναι φιλικό για την ανάπτυξη εφαρμογών προς κινητές συσκευές χωρίς πολλούς περιορισμούς.
- Είναι εύκολη στην εκμάθηση αφού είναι πολύ πιο φιλική προς το χρήστη από τα Vue.js, Angular.js ή Ember.js.
- Ένα από τα πλεονεκτήματα της χρήσης της React.js είναι η εξαιρετική υποστήριξη της κοινότητας με την πάροδο του χρόνου. Αυτό οφείλεται στο ότι είναι μια βιβλιοθήκη ανοιχτού κώδικα και λόγω ότι συνεχίζει να έχει την υποστήριξη και τους πόρους του Facebook.

4.6 Node JS

Το node js είναι ένα περιβάλλον εκτέλεσης (runtime environment) ανοιχτού κώδικα που “τρέχει” σε πληθώρα πλατφορμών (Windows, Linux, Unix, Mac OS X, κτλ.) και εκτελείται σε μια μηχανή JavaScript (δηλαδή μηχανή V8). Εκτελεί κώδικα JavaScript εκτός του φυλλομετρητή. Κάνοντας χρήση της node ανοίγεται η δυνατότητα της σχεδόν αποκλειστικής χρήσης της γλώσσας JavaScript σε όλο το «φάσμα» της εφαρμογής, δηλαδή από τη συσκευή του πελάτη μέχρι και τον διακομιστή.

Εικόνα 4.6

- Το Node.js μπορεί να δημιουργήσει δυναμικό περιεχόμενο σελίδας
- Το Node.js μπορεί να δημιουργεί, να ανοίγει, να διαβάζει, να γράφει, να διαγράφει και να κλείνει αρχεία στον διακομιστή
- Το Node.js μπορεί να συλλέγει δεδομένα φόρμας
- Το Node.js μπορεί να προσθέσει, να διαγράψει, να τροποποιήσει δεδομένα στη βάση δεδομένων σας
- Τα αρχεία Node.js έχουν επέκταση ".js".
- Το Node.js έχει μια αρχιτεκτονική που βασίζεται σε συμβάντα με δυνατότητα ασύγχρονης εισόδου/εξόδου. Αυτός ο σχεδιασμός έχει ως στόχο τη βέλτιστη απόδοση και επεκτασιμότητα σε εφαρμογές Ιστού με πολλές λειτουργίες εισόδου/εξόδου, καθώς και για εφαρμογές Ιστού σε πραγματικό χρόνο (π.χ. προγράμματα επικοινωνίας σε πραγματικό χρόνο και παιχνίδια προγράμματος περιήγησης) [23].

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

4.6.1 Express.js

Εικόνα 4.7

Το Express Js είναι ένα από τα πιο δημοφιλή framework της JavaScript και λειτουργεί ως ένα πρόσθετο (module) της Node.js και δημιουργήθηκε για την ανάπτυξη του back-end των εφαρμογών διαδικτύου και API. Η δημιουργία ενός back-end από την αρχή είναι μια απαιτητική διαδικασία αλλά χρησιμοποιώντας το Express.js, γίνεται εξοικονόμηση του χρόνο , με κώδικα μικρότερο σε μέγεθος και είναι εύκολο στην εκμάθησή. Το Express Js βασίζεται στο μοντέλο αρχιτεκτονικής MVC (Model View Controller).

Μερικά από τα βασικά χαρακτηριστικά του πλαισίου Express είναι ότι [24]:

- Επιτρέπει τη ρύθμιση ενδιάμεσου λογισμικού για να ανταποκρίνεται στα αιτήματα HTTP
- Καθορίζει ένα πίνακα δρομολόγησης για την εκτέλεση ενεργειών με βάση τη μέθοδο HPPT και τη διεύθυνση URL
- Επιτρέπει τη δυναμική απόδοση σελίδων HTML

4.7 Βάσεις Δεδομένων

Βάσεις δεδομένων [25] ονομάζουμε τις συλλογές δεδομένων που είναι οργανωμένες με τέτοιο τρόπο ώστε ένας υπολογιστής να μπορεί εύκολα να βρει τις επιθυμητές πληροφορίες. Χρησιμοποιούνται για να γίνεται η εκτέλεση των πράξεων αναζήτησης, ανάγνωσης, ενημέρωσης και διαγραφής τους. Στις βάσεις συναντώνται πολλές δομές δεδομένων όπως δέντρα, λίστες και πίνακες κατακερματισμού. Επίσης μπορούν να περιέχουν δεδομένα πολλών ειδών όπως ονόματα, ημερομηνίες και διάφορους τύπους δεδομένων (int, float, Boolean, char, sting, array, binary και αλλά).

Τα δεδομένα μιας βάσης δεδομένων αποθηκεύονται σε στοιχειώδεις μορφές:

- **Πεδίο (Field)**: είναι το μικρότερο κομμάτι δεδομένων στο οποίο μπορούμε να αναφερθούμε και περιέχει ένα μόνο χαρακτηριστικό ή ιδιότητα ενός στοιχείου της βάσης δεδομένων.

- **Εγγραφή (Record):** είναι ένα σύνολο από διαφορετικά πεδία που περιέχει όλες τις πληροφορίες για ένα στοιχείο της βάσης δεδομένων.

- **Αρχείο (File):** είναι ένα σύνολο από πολλά παρόμοια στοιχεία (εγγραφές) της βάσης δεδομένων.

- **Πρωτεύον Κλειδί (Primary Key):** είναι ένα πεδίο ή συνδυασμός πεδίων που χαρακτηρίζει μοναδικά μια εγγραφή.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

- **Κλειδί (Key):** είναι ένα πεδίο που δεν έχει κατ' ανάγκη μοναδική τιμή και που μπορούμε να το χρησιμοποιήσουμε για να κάνουμε αναζήτηση σ' ένα αρχείο.
- **Ξένο Κλειδί (Foreign Key):** είναι ένα πεδίο που έχει το ίδιο σύνολο τιμών με το πρωτεύον κλειδί ενός άλλου αρχείου.

Παράλληλα αναπτύχθηκαν και τα Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΣΔΒΔ, Relational Database Management Systems – RDBMS), δηλαδή συγκεκριμένα συστήματα που υλοποιούν σχεσιακές βάσεις δεδομένων. Παραδείγματα σχεσιακών συστημάτων βάσεων δεδομένων είναι η Oracle, ο SQL Server και η MySQL.

Οι αρμοδιότητές του είναι οι εξής :

- Η απόφαση για το είδος των πληροφοριών που πρέπει να αποθηκεύουν.
- Η απόφαση για τον τρόπο αποθήκευσης και πρόσβασης στις πληροφορίες αυτές.
- Η συνεργασία με τους τελικούς χρήστες.
- Η απόφαση για τον τρόπο εξασφάλισης των πληροφοριών.
- Η απόφαση για το κάθε πότε θα γίνονται αντίγραφα ασφαλείας (backup) των αρχείων.
- Η παρακολούθηση της σωστής λειτουργίας της βάσης δεδομένων και η απαιτούμενη προσαρμογή της.

Οι δυο κυριότερες κατηγορίες βάσεων τώρα είναι οι σχεσιακές και οι μη σχεσιακές βάσεις δεδομένων (SQL ή NoSQL):

- **Σχεσιακές βάσεις δεδομένων (SQL)** Οι σχεσιακές βάσεις δεδομένων είναι οργανωμένες σε πίνακες με ορισμένα κάθε φορά πεδία που ονομάζουμε στήλες. Για κάθε πίνακα, είναι αναγκαίο κάθε γραμμή (εγγραφή) να μπορεί να οριστεί μοναδικά για αυτό το λόγο γίνεται χρήση ενός πρωτεύοντος κλειδιού. Υφίσταται επίσης η έννοια του ξένου κλειδιού, η οποία προσδιορίζει τον τρόπο με τον οποίο συνδέονται οι πίνακες μεταξύ τους μέσω των σχέσεων. Ουσιαστικά, ο συνδυασμός τιμών των στηλών που ανήκουν σε ένα ξένο κλειδί σε έναν πίνακα, πρέπει να εντοπίζεται σε αντίστοιχες στήλες ενός άλλου πίνακα. Μία από τις πιο γνωστές βάσεις δεδομένων αποτελεί η PostgreSQL.

- **Μη σχεσιακές βάσεις δεδομένων (Not only SQL - NoSQL)** Οι μη σχεσιακές βάσεις δεδομένων μοιάζουν αρκετά στη δομή που αναφέρθηκε παραπάνω αφού περιέχουν δομές όμοιες με πίνακες με κύρια διαφορά το ότι κάθε καταχώρηση ενός «πίνακα» μπορεί να έχει τα δικά της ξεχωριστά πεδία. Έτσι δεν υπάρχουν περιορισμοί στην αποθήκευση δεδομένων. Η προσέγγιση NoSQL (NotOnlySQL) παρουσιάστηκε για πρώτη φορά το 2009 και θεωρήθηκε μια ομάδα συστημάτων βάσεων δεδομένων οι οποίες αναπτύσσονται με σκοπό τον αποδοτικό χειρισμό μεγάλων ποσοτήτων πληροφορίας και επεξεργασίας δεδομένων τα οποία είναι κατανεμημένα σε μεγάλο αριθμό servers. Σε συνδυασμό με τη γλώσσα SQL χρησιμοποιούν έναν άλλο τρόπο από τον σχεσιακό τρόπο αποθήκευσης μία αδόμητη ή ημιδομήμενη αναπαράσταση δεδομένων, πράγμα το οποίο σημαίνει ότι η δομή καθορίζεται ανάλογα με τις ανάγκες που προκύπτουν. Το μεγάλο πλεονέκτημα των συστημάτων αυτών είναι η προσαρμοστικότητα που διαθέτουν ο οποίος είναι ο βασικός λόγος που υιοθετήθηκαν από μεγάλες εταιρίες του χώρου του διαδικτύου όπως η Amazon, το Facebook και η Google. Επιπλέον, οι βάσεις δεδομένων τύπου NoSQL έχουν την ικανότητα να καλύψουν τις ανάγκες μεγάλου αριθμού χρηστών των εφαρμογών Internet, πετυχαίνοντας ταχύτατους χρόνους απόκρισης κάνοντας τη χρήση τους πολύ πιο ευχάριστη.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

4.7.1 PostgreSQL

Εικόνα 4.8

Η PostgreSQL είναι μια ανοικτού κώδικα σχεσιακή βάση δεδομένων βασισμένη στην SQL. Βασίζεται σε μια αποδεδειγμένα καλή αρχιτεκτονική η οποία συμβάλει στα υψηλά επίπεδα ανθεκτικότητας, ακεραιότητας δεδομένων και ορθότητας. Λειτουργεί σε όλα τα βασικά λειτουργικά συστήματα τα Windows , το UNIX (AIX, BSD, HP-UX, SGI, IRIX, MAC OS X, Solaris, 57 Tru64) και τα Linux. Χρησιμοποιείται ως κύρια αποθήκη δεδομένων για πολλές εφαρμογές ιστού, κινητών, γεωχωρικών και αναλυτικών στοιχείων και υποστηρίζει αποθήκευση μεγάλων δυαδικών αντικειμένων, όπως εικόνες, ήχοι ή βίντεο. Υποστηρίζει μεγάλο μέρος του προτύπου SQL και προσφέρει πολλά σύγχρονα χαρακτηριστικά όπως σύνθετα queries, ξένα κλειδιά, triggers, updatable views και ακεραιότητα συναλλαγών [26].

4.8 API

Το API (Application Programming Interface) είναι ένας μεσολαβητής λογισμικού που επιτρέπει τη μεταβίβαση δεδομένων μεταξύ δύο εφαρμογών έτσι ώστε να συνομιλούν μεταξύ τους. Κάθε φορά που χρησιμοποιείται μία εφαρμογή όπως το Facebook, στέλνεται ένα άμεσο μήνυμα ή ελέγχεται η πρόβλεψη του καιρό σε κάποια φορητή συσκευή, χρησιμοποιείται ένα API. Αν παραδείγματος χάρη μία εφαρμογή κινητού τηλεφώνου συνδεθεί στο Διαδίκτυο τότε στέλνει δεδομένα σε έναν διακομιστή. Στη συνέχεια, ο διακομιστής ανακτά αυτά τα δεδομένα, τα ερμηνεύει, εκτελεί τις απαραίτητες ενέργειες και τα στέλνει πίσω στο κινητό τηλέφωνο.

Στη συνέχεια, η εφαρμογή ερμηνεύει αυτά τα δεδομένα και τα παρουσιάζει με ευανάγνωστο τρόπο. Όλα αυτά συμβαίνουν μέσω ενός API. Τα σύγχρονα API συμμορφώνονται με πρότυπα HTTP και REST, τα οποία είναι ιδιαίτερα προσιτά για τον χρήστη. Όπως κάθε άλλο κομμάτι του παραγόμενου λογισμικού, το σύγχρονο API έχει τον δικό του κύκλο ζωής ανάπτυξης λογισμικού (SDLC) σχεδιασμού, δοκιμής, κατασκευής, διαχείρισης και έκδοσης [27]. Στη πιλοτική εφαρμογή θα γίνει σύνδεση με το API του OpenWeatherMap και θα λαμβάνεται η θερμοκρασία της Λαμίας για τη στιγμή που θα γίνεται το API request. (https://openweathermap.org/)

4.9 Αυθεντικοποίηση και Hashing (Passport και bcrypt)

Για ταυτοποίηση ή σύγκριση δεδομένων, όπως για παράδειγμα σε ένα σύστημα ‘login’, θα ήταν προτιμότερο να μην γίνεται αποστολή των ευαίσθητων δεδομένων (ειδικά κωδικών) αλλά μια κατακερματισμένη τιμή τους (hashed value), ή οποία, σε περίπτωση υποκλοπής δεν θα μπορεί να γίνει η επαναφορά τους στα αρχικά δεδομένα. Χρησιμοποιούνται διάφοροι αλγόριθμοι κατακερματισμού μετασχηματίζουν τα δεδομένα που λαμβάνουν ως είσοδο και δίνουν ως έξοδο μια καθορισμένου μεγέθους συμβολοσειρά (string) η οποία δεν μπορεί με αντιστροφή να παράγει την αρχική είσοδο. Η έξοδος αποκαλείται συνήθως "σύνοψη" (digest). Με την εξέλιξη

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

της τεχνολογίας τα υπολογιστικά συστήματα γίνονται όλο και πιο ισχυρά με αποτέλεσμα πολλοί από τους αλγορίθμους που χρησιμοποιούσαν για κρυπτογραφία στο παρελθόν όπως οι MD5, SHA0, SHA1 να έχουν εκτεθεί και πλέον δεν μπορούν να προστατέψουν με επιτυχία τα δεδομένα , ενώ άλλοι όπως οι αλγόριθμοι: SHA2, SHA3 χρησιμοποιούνται ακόμη και σήμερα.

Για το κατακερματισμό των κωδικών χρηστών στην εφαρμογή της πτυχιακής εργασίας χρησιμοποιήθηκε το bcrypt το οποίο είναι μια βιβλιοθήκη της node.js. Εκτός από το κατακερματισμό εισάγει και ένα salt στην είσοδο δεδομένων το οποίο αποτελεί άλλη μία δικλείδα ασφαλείας και προσθέτει τυχαία δεδομένα στους κωδικούς, έτσι ώστε να διαφοροποιήσει κοινούς κωδικούς πρόσβασης και να κάνει σχεδόν αδύνατη την αναγνώριση τους [28].

Εικόνα 4.9

Το Passport είναι ένα ενδιάμεσο λογισμικό (middleware) ελέγχου ταυτότητας για το Node.js το οποίο διευκολύνει τη Αυθεντικοποίηση (authentication) και εξουσιοδότηση (authorization) για τους χρήστες. Χρησιμοποιείται σε οποιαδήποτε διαδικτυακή εφαρμογή που βασίζεται στο framework της Express.js. Κάνει έλεγχο ταυτότητας των αιτημάτων μέσω διαφόρων μηχανισμών διαπιστευτηρίων , γνωστών ως strategies. Μια ακόμα δυνατότητα που παρέχει είναι ότι μπορεί να επιτρέψει στους χρήστες να εγγραφούν και συνδεθούν στη διαδικτυακή εφαρμογή μέσω των λογαριασμών τους στις υπηρεσίες Google , Facebook κ.α. μέσω της εφαρμογής ανοιχτού κώδικα OAuth.
Στη πτυχιακή έχει χρησιμοποιηθεί για Αυθεντικοποίηση (authentication) των χρηστών με όνομα και κωδικό μέσω του strategy “passport-local” [29].

4.10 Grafana

Εικόνα 4.10

Το Grafana είναι ένα ανοιχτού κώδικα dashboard εργαλείο αναλυτικών στοιχείων και δια δραστικής οπτικοποίησης πολλαπλών πλατφορμών. Παρέχει γραφήματα και ειδοποιήσεις για τον Ιστό όταν συνδέεται με υποστηριζόμενες πηγές δεδομένων. Λειτουργεί πλέον και με σχεσιακές βάσεις δεδομένων όπως η PostgreSQL. Είναι διαθέσιμο για εγκατάσταση σε ηλεκτρονικούς υπολογιστές είτε ως cloud υπηρεσία. Χωρίζεται σε front-end και back-end κομμάτι που είναι γραμμένα σε TypeScript και Go αντίστοιχα. Ακόμη μπορεί να γίνει και επεκτάσιμο μέσω ενός συστήματος plug-in [30].Τα Panel του Grafana μπορούν να εισαχθούν εύκολα σε οποιαδήποτε ιστοσελίδα αφού δίνει τη δυνατότητα να δημιουργήσει HTML κώδικα για την ενσωμάτωση ενός iframe για το οποιοδήποτε πάνελ.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Τα κύρια χαρακτηριστικά του Grafana είναι τα ακόλουθα:

- **Πάνελ**
  Το Grafana διαθέτει γρήγορες και ευέλικτες οπτικοποιήσεις. Δίνει τη δυνατότητα εισαγωγής δυναμικών και επαναχρησιμοποιήσιμών πάνελ από θερμικούς χάρτες και ιστογράμματα μέχρι γραφήματα σε γεωχάρτες.

- **Ειδοποιήσεις (Alerts)**
  Με το Grafana Alerting, μπορεί να γίνει δημιουργία, διαχείριση ή και σίγαση όλων των ειδοποιήσεων εύκολα μέσω της διεπαφής χρήστη.

- **Πρόσθετα (Plugins)**
  Τα πρόσθετα του Grafana είναι τα εξωτερικά στοιχεία λογισμικού που βελτιώνουν την εμπειρία του χρήστη με ένα ευρύ φάσμα νέων λειτουργιών, όπως βελτιωμένες απεικονίσεις, δυνατότητα εισαγωγής δεδομένων από άλλη πηγή δεδομένων και προσθήκη νέων λειτουργιών για αύξηση της αναγνωσιμότητας των στοιχείων που παρουσιάζονται.

- **Σχολιασμοί (Annotations)**
  Δίνει τη δυνατότητα να γίνει εισαγωγή σχολίων και tag μέσα στα γραφήματα με εύκολο τρόπο διατρέχοντας το δείκτη του ποντικιού πάνω στο γράφημα για συγκεκριμένη χρονική στιγμή.

- **Επεξεργαστής Panel**
  Το πρόγραμμα επεξεργασίας πίνακα διευκολύνει τη διαμόρφωση, την προσαρμογή και την εξερεύνηση όλων των πλαισίων μέσω της διεπαφής χρήστη για όλες τις απεικονίσεις.

Εικόνα 4.11

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

ΚΕΦΑΛΑΙΟ 5 Περιγραφή και ανάλυση της εφαρμογής

Ο σκοπός της πτυχιακής εργασίας είναι η ανάπτυξη μιας διαδικτυακής εφαρμογής όπου θα διαχειρίζεται IOT συσκευές και θα παρουσιάζει τα δεδομένα που συλλέγουν σε μορφή διαγραμμάτων. Θα γίνει περιγραφή του τρόπου λειτουργίας της εφαρμογής καθώς και τα στάδια δημιουργίας της.

5.1 Περιγραφή

Στα πλαίσια της πτυχιακής εργασίας δημιουργήθηκε μια διαδικτυακή εφαρμογή (web application), δηλαδή ένα πρόγραμμα που είναι διαθέσιμο μέσω του Διαδικτύου, αποθηκεύεται σε ένα απομακρυσμένο server και η χρήση του γίνεται μέσα από ένα φυλλομετρητή (web browser), για τη δημιουργία ενός γραφικού περιβάλλοντος για IOT συσκευές.

Τα μέρη που το αποτελούν είναι:

- Το κομμάτι του client side (front-end): Δηλαδή το περιεχόμενο που αποδίδει ο φυλλομετρητής και αποτελεί τη διεπαφή user interface με το χρήστη. Οι τεχνολογίες που χρησιμοποιούνται είναι η HTML , CSS, JavaScript και η βιβλιοθήκη της React. Μια από τις πλέον δημοφιλής αρχιτεκτονικές για την ανάπτυξη διαδικτυακών εφαρμογών είναι SPA (Single Page Application) με Web API από τη πλευρά του εξυπηρετητή (Server Side).

- Το κομμάτι του server side (back-end) και η βάση δεδομένων: Αποτελείται από το λογισμικό που εκτελείται στη πλευρά του εξυπηρετητή και επιστρέφει τις ανάλογες πληροφορίες όπου γίνονται τα request από το client side μέσω του πρωτοκόλλου HTTP. Η γλώσσα που χρησιμοποιούμε είναι Node.js.
  Περιλαμβάνει επίσης τη βάση δεδομένων μας και το σύστημα διαχείρισης της όπου γίνεται η χρήση της σχεσιακής βάσης δεδομένων PostgreSQL. Ακόμα , το ενδιάμεσο λογισμικό (middleware) για την παραγωγή και επεξεργασία δυναμικού περιεχομένου, γραμμένο σε Node.js.Πολλές φορές το ενδιάμεσο λογισμικό (middleware) ορίζεται ως τμήμα του back-end καθώς αποτελεί τη server-side της εφαρμογής σε αντίθεση με το front-end που αφορά τη client-side.

Για τη δημιουργία της εργασίας χρησιμοποιούμε μια Στοίβα Ιστού (Web stack) όπου είναι το σύνολο των τεχνολογιών δηλαδή ο συνδυασμός των γλωσσών προγραμματισμού , πλαισίων , βιβλιοθηκών , διακομιστών και διεπαφής χρήστη για τη δημιουργία της διαδικτυακής εφαρμογής. Η στοίβα που χρησιμοποιούμε είναι η PERN και αποτελείται από PostgreSQL, Express, React και Node.js. Συνδυάζοντας αυτές τις τεχνολογίες, δημιουργούμε μια Full-stack Web εφαρμογή με λειτουργίες CRUD.

5.2 Σχεδιασμός και Ανάπτυξη Back-end

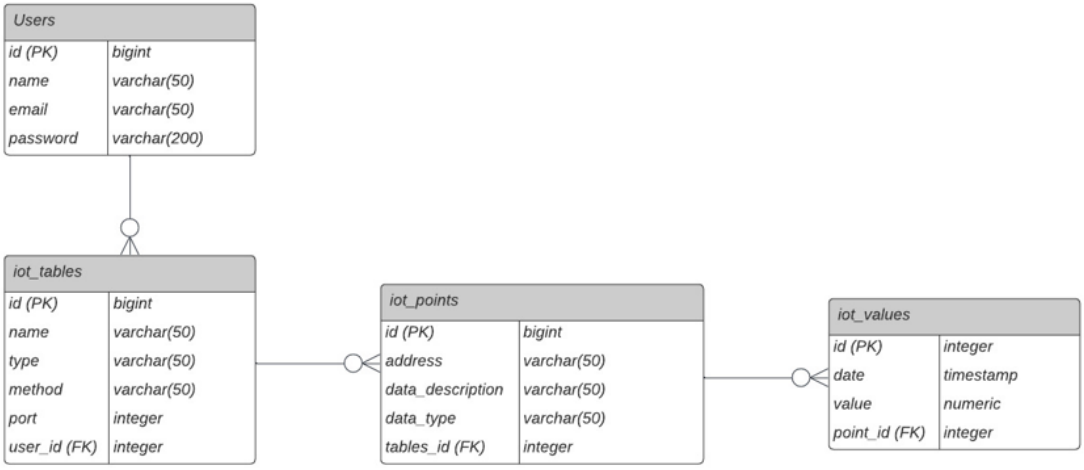
Στη συγκεκριμένη ενότητα θα γίνει ανάλυση του σχεδιασμού του back-end της εφαρμογής. Αναλύεται ο εξυπηρετητής της διαδικτυακής εφαρμογής (web application server) όπου έχει την ευθύνη της διαχείρισης της λογικής και ορθής απόδοσης των δεδομένων. Ακόμα ο εξυπηρετητής βάσης δεδομένων (database server) όπου είναι η βάση δεδομένων που αποθηκεύει και προμηθεύει τα δεόμενα που είναι σχετικά με τη λειτουργία της εφαρμογής μας.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

5.2.1 Βάση Δεδομένων

Σαν πρώτο βήμα για την υλοποίηση του back-end της εφαρμογής πρέπει να σχεδιαστεί η βάση δεδομένων.

Η δομή της βάση δεδομένων της εφαρμογής φαίνεται στο Διάγραμμα 5.1.



Διάγραμμα 5.1 Δομή της βάσης δεδομένων

Απαρτίζεται από 4 πίνακες (tables).

Αρχικά το table Users περιέχει τις πληροφορίες των χρηστών οι οποίες είναι : το μοναδικό αναγνωριστικό “id” που δημιουργείται αυτόματα με τη κάθε προσθήκη καινούργιου χρήστη , το όνομα “name” του χρήστη , η ηλεκτρονική διεύθυνση “email” στο οποίο υπάρχει και περιορισμός να είναι μοναδικό και τέλος ο κωδικός “password”.

Στη συνέχεια στο πίνακα iot_tables περιέχονται οι συσκευές όπου θα μπορεί να διαχειρίζεται ο χρήστης. Οι συσκευές αυτές έχουν ένα μοναδικό αναγνωριστικό “id” , όνομα “name” , τύπο “type” , μέθοδο “method” που χρησιμοποιούν , θύρα “port” με το οποίο συνδέονται και ως δευτερεύων κλειδί το id “user_id" του χρήστη στον οποίο ανήκουν. Ο χρήστης μπορεί να μην έχει καθόλου συσκευές και θα του δίνεται η δυνατότητα να προσθέσει όσες χρειάζεται.

Στο πίνακα iot_points περιέχονται επιπλέον πληροφορίες για τα devices. Κάποια συγκεκριμένη συσκευή πχ ένας αισθητήρας θερμοκρασίας μπορεί να χρησιμοποιείται σε διάφορες τοποθεσίες και να δέχεται διάφορα είδη δεδομένων. Στη πρώτη στήλη υπάρχει το μοναδικό αναγνωστικό “id” και στις υπόλοιπες υπάρχουν η διεύθυνση “address” , η περιγραφή των δεδομένων “data_description”, ο τύπος των δεδομένων “data_type”. Επίσης περιέχεται το δευτερεύων κλειδί “tables_id” του πίνακα “iot_tables” για την εξάρτηση του πίνακα Points από το tables.

Τέλος στο πίνακα IoT_values περιέχονται τα δεδομένα τα οποία αποθηκεύουν αυτές οι συσκευές. Υπάρχει πάλι μοναδικό αναγνωριστικό id , η τιμή την οποία μετράει η συσκευή “value” και η ημερομηνία και ώρα όπου έγινε η μέτρηση “date”.

Για τη δημιουργία της βάσης δεδομένων χρησιμοποιείται η PostgreSQL. Παρακάτω παρουσιάζονται οι εντολές SQL οι οποίες χρησιμοποιήθηκαν για τη δημιουργία της βάσης και των πινάκων ώστε να είναι δυνατή η ανάκτηση των πληροφοριών αυτών μέσω του API της εφαρμογής.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

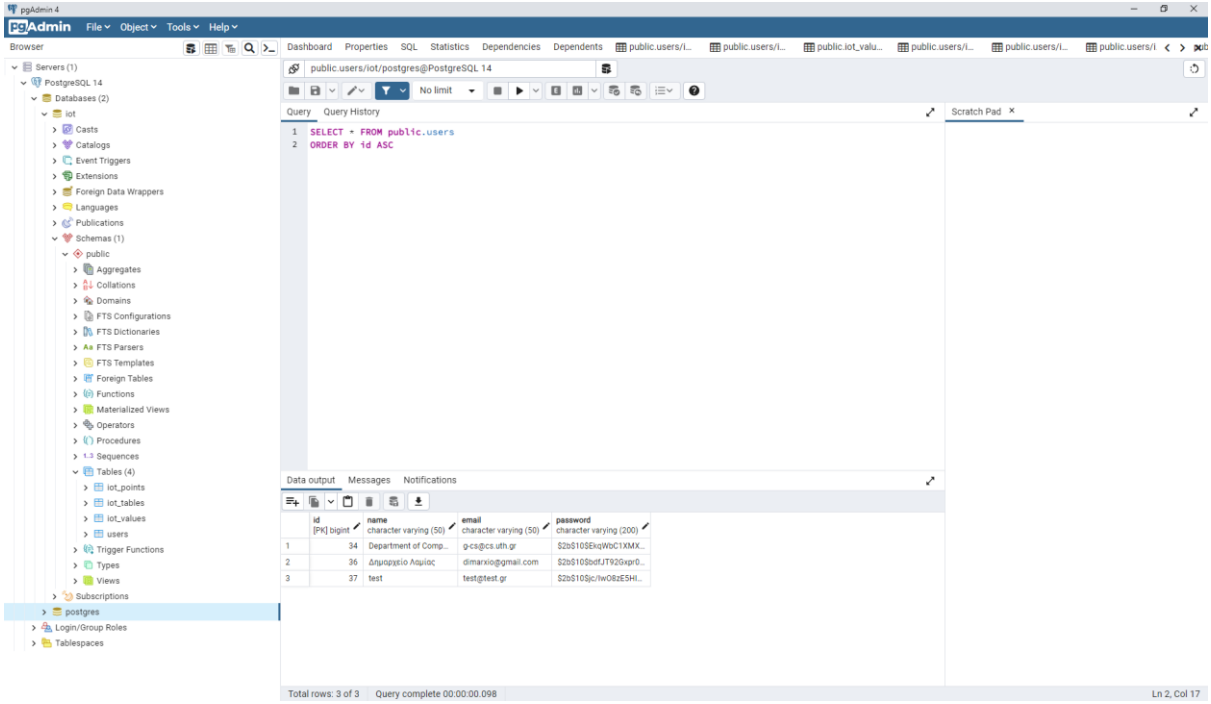
Αρχικά γίνεται δημιουργία της βάση δεδομένων με όνομα iot με την εντολή
```
CREATE DATABASE iot
```
και υστέρα η δημιουργία των tables με την εντολή CREATE TABLES (εικόνα 5.1)

```
1  CREATE TABLE users (
2    id BIGINT PRIMARY KEY NOT NULL,
3    name VARCHAR(50) NOT NULL,
4    email VARCHAR(50) UNIQUE NOT NULL,
5    password VARCHAR(200) NOT NULL,
6
7    UNIQUE (email)
8  );
9
10 CREATE TABLE iot_tables (
11   id BIGINT PRIMARY KEY NOT NULL,
12   name VARCHAR(50) NOT NULL,
13   type VARCHAR (50) NOT NULL,
14   method VARCHAR(50) NOT NULL,
15   port INTEGER NOT NULL,
16   user_id INTEGER,
17
18   FOREIGN KEY (user_id) REFERENCES users(id)
19 );
20
21 CREATE TABLE iot_points (
22   id BIGINT PRIMARY KEY NOT NULL,
23   address VARCHAR(50) NOT NULL,
24   data_description VARCHAR(50) NOT NULL,
25   data_type VARCHAR(50) NOT NULL,
26   tables_id INTEGER,
27
28   FOREIGN KEY (tables_id) REFERENCES iot_tables(id)
29 );
30
31 CREATE TABLE iot_values (
32   id BIGINT PRIMARY KEY NOT NULL,
33   date TIMESTAMP,
34   value NUMERIC,
35   point_id INTEGER,
36
37   FOREIGN KEY (points_id) REFERENCES iot_points(id)
38 );
```

Εικόνα 5.1 Εντολές για τη δημιουργία tables βάσης δεδομένων

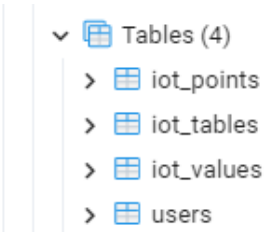
Η πρόσβαση στη βάση δεδομένων μπορεί να γίνει είτε μέσω του SQL Shell (psql), όπου είναι μια διεπαφή σε τερματικό για την PostgreSQL το οποίο επιτρέπει την εισαγωγή, επεξεργασία και εκτέλεση εντολών SQL, είτε μέσω του pgAdmin όπου είναι μια εφαρμογή διαχείρισης σε γραφικό περιβάλλον που χρησιμοποιείτε για την επικοινωνία με PostgreSQL (Εικόνα 5.2).

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212



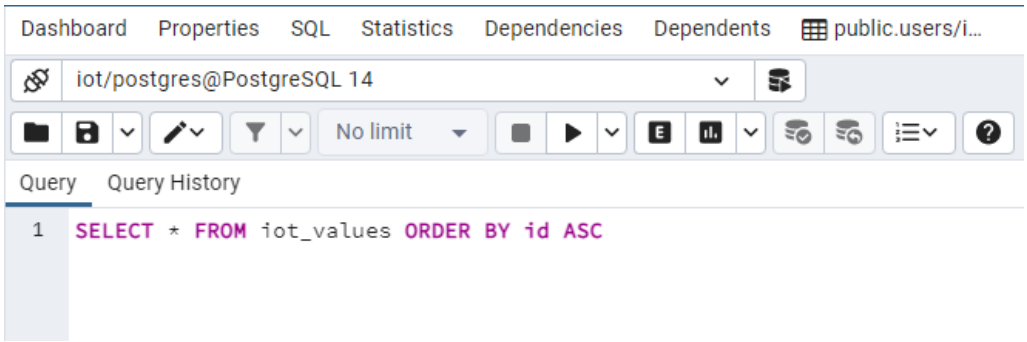
Εικόνα 5.2 Περιβάλλον εφαρμογής Postgres

Με αυτό το τρόπο γίνεται η εμφάνιση των πινάκων που έχουν δημιουργηθεί (εικόνα 5.3).



Εικόνα 5.3 Δομή πινάκων (tables)

Ακόμα δίνεται η δυνατότητα εκτέλεσης εντολών SQL για εισαγωγή , επεξεργασία , διαγραφή και παρουσίαση αυτών των δεδομένων (εικόνα 5.4)



Εικόνα 5.4 Εκτέλεση εντολών SQL

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Και σαν αποτέλεσμα εμφανίζει τα δεδομένα (εικόνα 5.5)

	id [PK] bigint	date timestamp without time zone	value numeric	point_id integer
1	61997	2022-09-25 15:37:10.19288	26.92320614	26
2	61998	2022-09-25 15:37:40.19288	26.46017647	26
3	61999	2022-09-25 15:38:10.19288	25.56013731	26
4	62000	2022-09-25 15:38:40.19288	28.19953053	26
5	62001	2022-09-25 15:39:10.19288	27.73245779	26
6	62002	2022-09-25 15:39:40.19288	28.04585114	26
7	62003	2022-09-25 15:40:10.19288	25.99810074	26
8	62004	2022-09-25 15:40:40.19288	27.69185318	26
9	62005	2022-09-25 15:41:10.19288	27.58250962	26

Εικόνα 5.5 Αποτελέσματα της εντολής SELECT

Η σύνδεση του node server με τη βάση δεδομένων γίνεται στο αρχείο με το όνομα db.js. (εικόνα 5.6). Αυτό γίνεται με τη χρήση ενός connection pool, όπου είναι μια μέθοδος με την οποία δημιουργείται ένα pool συνδέσεων ώστε οι συνδέσεις να μπορούν να επαναχρησιμοποιηθούν.

Αρχικά γίνεται εισαγωγή του module pg όπου χρησιμοποιείται για την επικοινωνία μεταξύ της εφαρμογής με τη Node.js. Στη μεταβλητή pool αποθηκεύεται η δομή στην οποία διατηρούνται οι απαραίτητες πληροφορίες για το περιβάλλον στο οποίο τρέχει η βάση δεδομένων, όπως και των διαπιστευτηρίων που θα επιτρέψουν τη σύνδεση με αυτή. Οι πληροφορίες που χρειάζονται είναι: ο user, ο κωδικός και το όνομα της βάση δεδομένων στην οποία θα γίνει η σύνδεση ,το port που χρησιμοποιεί και τέλος ο host.

Λόγω ότι η postgres μπορεί να διαχειρίζεται ένα μόνο querie στη βάση δεδομένων χρησιμοποιείται το Pool ώστε να μπορούν να πραγματοποιούν πολλαπλά queries και έτσι να εξασφαλίζεται η σταθερή σύνδεση του API στη βάση δεδομένων.

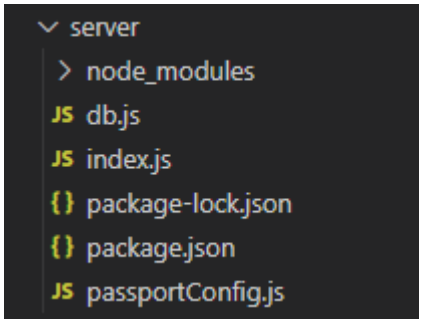
```
1   const {Pool} = require("pg");
2
3   const pool = new Pool({
4       user: "postgres",
5       password: ,
6       database:"iot",
7       port: 5432,
8       host:"localhost"
9   });
10
11  module.exports = pool;
```

Εικόνα 5.6 Κώδικας Connection Pool

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

5.2.2 Server

Για τη δημιουργία του server έχει γίνει χρήση της γλώσσας node.js με τη βιβλιοθήκη express για τη δημιουργία του API μας. Η βασική δομή των αρχείων είναι η εξής (εικόνα 5.7):



Εικόνα 5.7 Δομή αρχείων server

Ο σερβερ εκτελείται από το αρχείο index.js, όπου εκεί αρχικά εισάγονται όλα τα απαραίτητα module της Node.js όπως τα dependencies και το Middleware που θα χρησιμοποιηθούν.

Στα dependencies έχουμε :

- Την express για τη δημιουργία της back-end εφαρμογής και των API
- Το bcrypt για το hashing, δηλαδή τη κρυπτογράφηση του κωδικού
- Το cors όπου είναι ο μηχανισμός που επιτρέπει τη κοινή χρήση πόρων με τη react εφαρμογή όπου βρίσκεται σε διαφορετική διεύθυνση δικτύου
- Το pool για τη διαχείριση του pooling των συνδέσεων στη βάση δεδομένων όπου εισάγεται από το db.js αρχείο
- Το passport για την αυθεντικοποίηση του χρήστη (εικόνα 5.8).

```
1  //---------------------    IMPORTS    ---------------------- //
2
3  const express = require("express");
4  const cors = require("cors");
5  const pool = require("./db");
6  const bcrypt = require("bcrypt");
7  const session = require("express-session");
8  const passport = require("passport");
```

Εικόνα 5.8 Imports

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Για Middleware έχουμε (εικόνα 5.9):

```
// Middleware
const app = express();

app.use(cors());
app.use(express.json());
app.use(
  session({
    secret: "secretiot",
    resave: false,
    saveUninitialized: false,
  })
);

const port = process.env.SERVER_PORT || 5000;

app.use(passport.initialize());
app.use(passport.session());
require("./passportConfig")(passport);
```

Εικόνα 5.9 Middleware

Ακόμα δηλώνεται το port στο οποίο θα ακούει ο εξυπηρετητής και γίνεται προετοιμασία του περιβάλλοντος της εφαρμογής και της δρομολόγησης (εικόνα 5.10). Χρησιμοποιούμε το port 5000 (εικόνα 5.11).

```
26  const port = process.env.SERVER_PORT || 5000;
```

Εικόνα 5.10 Δήλωση μεταβλητής port

```
579  //---------------------- START SERVER ---------------------- //
580
581  app.listen(port, () => {
582    console.log("Server listening on port 5000");
583  });
584
```

Εικόνα 5.11 Εκκίνηση του Server

Στη συνέχεια θα γίνει ανάλυση των API που έχουν δημιουργηθεί.
Τα API τρέχουν με την Express και γίνεται η χρήση τους για:

- Τη δημιουργία ενός νέου χρήστη (User) στη βάση δεδομένων
- Την ανάκτηση του ονόματος του χρήστη
- Τη σύνδεση του στην εφαρμογή (μέσω του passport για authentication).
- Την επεξεργασία συσκευών (δημιουργία , ενημέρωση , εμφάνιση , διαγραφή)
- Τη δημιουργία τυχαίων δεδομένων στη βάση δεδομένων για τις συσκευές

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Για sign-up του χρήστη γίνεται το hash στο κωδικό και μετά από ένα έλεγχο αν υπάρχει ήδη κατοχυρωμένο το email στη βάση δεδομένων, τότε δημιουργείται ο χρήστης στη βάση με την INSERT INTO (εικόνα 5.12).

```
72  // Sign-up
73  app.post("/sign", async (req, res) => {
74    try {
75      const { name, email, password } = req.body;
76      let hashedPassword = await bcrypt.hash(password, 10);
77
78      pool.query(
79        "SELECT * FROM users WHERE email=$1",
80        [email],
81        (err, results) => {
82          if (err) {
83            console.log(err);
84          }
85          if (results.rows.length > 0) {
86            res.send("Email already registered");
87          } else {
88            pool.query(
89              "INSERT INTO users (name,email, password) VALUES($1,$2,$3)",
90              [name, email, hashedPassword],
91              (err, results) => {
92                if (err) {
93                  console.log(err);
94                } else {
95                  res.send("Success");
96                }
97              }
98            );
99          }
100       }
101     );
102   } catch (err) {
103     console.error(err.message);
104   }
105 });
```

Εικόνα 5.12 API για δημιουργία χρήστη

Για το login του user δημιουργούμε το API όπου τρέχουμε το local strategy του passport μέσω της εντολής passport.authenticate ώστε να γίνει το authentication (εικόνα 5.13):

```
58  // Sign in
59  app.post("/login", async (req, res, next) => {
60    passport.authenticate("local", (err, user, info) => {
61      if (err) console.log(err);
62      if (!user) res.send(info);
63      else {
64        req.logIn(user, (err) => {
65          if (err) console.log(err);
66          res.send({ id: user.id });
67        });
68      }
69    })(req, res, next);
```

Εικόνα 5.13 API για τη σύνδεση του χρήστη

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Το authentication γίνεται στο αρχείο passportConfig.js μέσω του local strategy και έπειτα γίνεται export το module, το οποίο στη συνέχεια εισάγεται στο index.js. Το αρχείο διαμορφώνεται ως εξής (εικόνα 5.14):

```
1  const LocalStrategy = require("passport-local").Strategy;const pool  = require("./db");
2  const bcrypt = require("bcrypt");
3
4
5  module.exports = function (passport) {
6
7    const authenticateUser = (email, password, done) => {
8      pool.query(
9        `SELECT * FROM users WHERE email = $1`,
10       [email],
11       (err, results) => {
12         if (err) {
13           throw err;
14         }
15
16         if (results.rows.length > 0) {
17           const user = results.rows[0];
18
19           bcrypt.compare(password, user.password, (err, isMatch) => {
20             if (err) {
21               console.log(err);
22             }
23             if (isMatch) {
24               return done(null, user);
25             } else {
26               return done(null, false,"Password is incorrect" );
27             }
28           });
29         } else {
30           return done(null, false,"No user with that email address");
31         }
32       }
33     );
34   };
35
36   passport.use(
37     new LocalStrategy(
38       { usernameField: "email", passwordField: "password" },
39       authenticateUser
40     )
41   );
42
43   passport.serializeUser((user, done) => done(null, user.id));
44
45   passport.deserializeUser((id, done) => {
46     pool.query(`SELECT * FROM users WHERE id = $1`, [id], (err, results) => {
47       if (err) {
48         return done(err);
49       }
50       return done(null, results.rows[0]);
51     });
52   });
53 };
```

Εικόνα 5.14 Authentication

Πιο συγκεκριμένα γίνεται αναζήτηση του χρήστη μέσω του email του και στη συνέχεια σύγκριση (bcrypt.compare) του hashed password της βάσης δεδομένων με το κωδικό που έχει εισάγει ο χρήστης στη φόρμα ,το οποίο και αυτό γίνεται hashed. Ακόμα δημιουργείται και ένα error handling όπου γίνεται επιστροφή μηνυμάτων σφάλματος εάν π.χ. δεν υπάρχει το email ή έχει εισαχθεί λανθασμένος κωδικός.

Ακόμα δίνεται η δυνατότητα δημιουργίας, ενημέρωσης , διαγραφής συσκευών (devices) καθώς και η ανάκτηση των πληροφοριών (name , type, method , port) για ένα device αλλά και για όλα τα devices που ανήκουν σε ένα χρήστη στο πίνακα (iot_tables) της βάσης δεδομένων.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Για επεξεργασία των συσκευών (update) έχουμε (εικόνα 5.15):

```
//Update a device

app.put("/devices/:deviceId", async (req, res) => {
  try {
    const deviceId = req.params.deviceId;
    const userId = req.query.id;

    const { name, type, method, port } = req.body;

    pool.query("SELECT * FROM users WHERE id=$1", [userId], (err, results) => {
      if (results.rows.length > 0) {
        pool.query(
          "UPDATE iot_tables SET name=$1,type=$2,method=$3,port=$4 WHERE user_id=$5 AND id=$6",
          [name, type, method, port, userId, deviceId],
          (err, results) => {
            if (err) {
              res.send("Error with the form");
              // console.log(err);
            } else {
              res.send("Success");
            }
          }
        );
      }
    });
  } catch (err) {
    console.error(err.message);
  }
});
```

Εικόνα 5.15 Κώδικας για ενημέρωση συσκευής

Με τον ίδιο περίπου τρόπο δημιουργούνται και τα άλλα API με διαφορές στις συνθήκες και στα queries. Με αυτό το τρόπο :

- Για ανάκτηση πληροφοριών μιας συσκευής έχουμε (εικόνα 5.16):

```
pool.query(
  "SELECT * FROM iot_tables WHERE user_id=$1 AND id=$2",
  [userId, deviceId],
```

Εικόνα 5.16

- Για ανάκτηση πληροφοριών όλων των συσκευών έχουμε (εικόνα 5.17):

```
pool.query(
  "SELECT * FROM iot_tables WHERE user_id=$1 ORDER BY name",
  [userId],
```

Εικόνα 5.17

- Για διαγραφή συσκευής έχουμε (εικόνα 5.18):

```
pool.query(
  "DELETE FROM iot_tables WHERE user_id=$1 AND id=$2",
  [userId, deviceId],
```

Εικόνα 5.18

- Εισαγωγή μιας συσκευής (εικόνα 5.19):

```
pool.query(
  "INSERT INTO iot_tables (name, type, method, port, user_id) VALUES($1,$2,$3,$4,$5)",
  [name, type, method, port, userId],
```

Εικόνα 5.19

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Στη συνέχεια με παρόμοιο τρόπο δίνεται η δυνατότητα δημιουργίας, ενημέρωσης , διαγραφής point καθώς και η ανάκτηση των πληροφοριών (address , data_description, data_type) για ένα point αλλά και για όλα τα points που ανήκουν σε ένα χρήστη στο πίνακα (iot_points) της βάσης δεδομένων. Με τον ίδιο τρόπο έχουμε queries Select , Delete , Insert και Update.

Τέλος για το πίνακα iot_values έχει γίνει δημιουργία API για την εισαγωγή τυχαίων δεδομένων (value) για συγκεκριμένο date (εικόνα 5.20).

```
527  // add Random data
528
529  app.post("/devices/:deviceId/points/:pointId/allData", async (req, res) => {
530    try {
531      const deviceId = req.params.deviceId;
532      const pointId = req.params.pointId;
533      const userId = req.query.id;
534
535      pool.query("SELECT * FROM users WHERE id=$1", [userId], (err, results) => {
536        if (results.rows.length > 0) {
537          pool.query(
538            "SELECT * FROM iot_tables WHERE id=$1",
539            [deviceId],
540            (err, results) => {
541              if (results.rows.length > 0) {
542                pool.query(
543                  "SELECT * FROM iot_points WHERE id=$1",
544                  [pointId],
545                  (err, results) => {
546                    if (results.rows.length > 0) {
547                      pool.query(
548                        "INSERT INTO iot_values ( date, value,point_id ) SELECT date, random()*100 AS value, $1 AS point_id
                         FROM generate_series(now() - interval '30 days', now(), interval '2 hour') AS g1(date)",
549                        [pointId],
550                        (err, results) => {
551                          if (err) {
552                            console.log(err);
553                          } else {
554                            res.send("Success");
555                          }
556                        }
557                      );
558                    } else {
559                      console.log(err);
560                    }
561                  }
562                );
563              }
564            }
565          );
566        } else {
567          console.log(err);
568        }
569      });
570    } catch (err) {
571      console.error(err.message);
572    }
573  });
```

Εικόνα 5.20 API για εισαγωγή τυχαίων δεδομένων

Η εκτέλεση του query γίνεται αφότου γίνουν όλοι οι έλεγχοι για το εάν υπάρχει ο user , η συσκευή που προσπαθεί να προσπελάσει και το point της συγκεκριμένης συσκευής .
Τα δεδομένα εισέρχονται στο πίνακα iot_values μέσω της INSERT INTO. Για τη στήλη date μέσω του generate_series δημιουργείτε ένα σύνολο δεδομένων ημερομηνίας και ώρας (timestamp) που ξεκινούν από 1 μήνα πριν μέχρι τη χρονική στιγμή που καλείτε το API, για κάθε 2 ώρες (interval).

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Ταυτόχρονα δημιουργούνται δεδομένα στη στήλη values μέσω της συνάρτησης random()*100 με τιμές από το 0 μέχρι το 100. Τέλος ως point_id εισέρχεται το id του point από το οποίο θα γίνει το POST. Οι τυχαίες τιμές που εισάγονται είναι ενδεικτικές ώστε να μπορεί να δημιουργηθεί αυτόματα διάγραμμα έχοντας παρελθοντικά δεδομένα.
Ένα παράδειγμα από τα δεδομένα που δημιουργούνται φαίνεται στην εικόνα 5.21

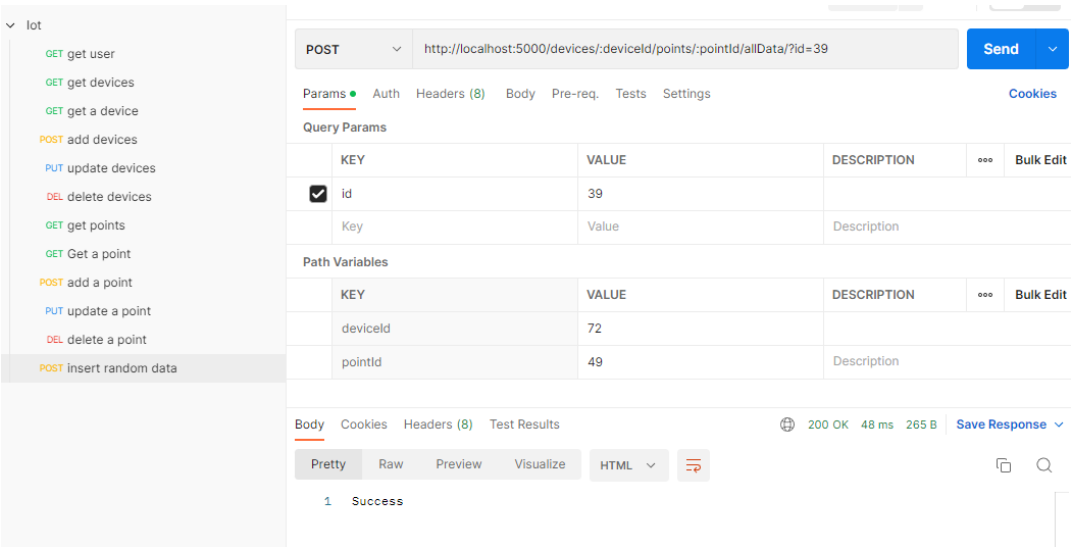
	id [PK] bigint	date timestamp without time zone	value numeric	point_id integer
1	71615	2022-09-08 18:30:50.23029	36.669594802331	49
2	71616	2022-09-08 20:30:50.23029	16.0978290791775	49
3	71617	2022-09-08 22:30:50.23029	3.74855503649592	49
4	71618	2022-09-09 00:30:50.23029	72.8309316508138	49
5	71619	2022-09-09 02:30:50.23029	60.8647621681985	49
6	71620	2022-09-09 04:30:50.23029	2.18274027857461	49
7	71621	2022-09-09 06:30:50.23029	0.0246819621278149	49
8	71622	2022-09-09 08:30:50.23029	3.53588625857171	49
9	71623	2022-09-09 10:30:50.23029	71.3262269421083	49
10	71624	2022-09-09 12:30:50.23029	7.39766127165637	49
11	71625	2022-09-09 14:30:50.23029	75.3275308825788	49
12	71626	2022-09-09 16:30:50.23029	9.07569614753996	49
13	71627	2022-09-09 18:30:50.23029	40.1070916461737	49
14	71628	2022-09-09 20:30:50.23029	48.5842793557694	49
15	71629	2022-09-09 22:30:50.23029	32.4123532595618	49
16	71630	2022-09-10 00:30:50.23029	63.7114069399068	49

Εικόνα 5.21

Για τον έλεγχο της σωστής λειτουργίας κάθε API έγινε η χρήση του Postman, το οποίο είναι μια πλατφόρμα για προγραμματιστές για τη σχεδίαση, δοκιμή και επανάληψη των API. Από εκεί γίνονται τα request δηλαδή GET , POST , UPDATE , DELETE στις διευθύνσεις των API και επιστρέφεται ως response, το response που έχει δηλωθεί στα API μας ,π.χ. “success” εάν έχει γίνει επιτυχημένα η εισαγωγή η ενημέρωση και η διαγραφή των στοιχείων της βάσης (post , put , delete), ή τα δεδομένα της βάσης όταν γίνεται get για το συγκεκριμένο χρήστη .

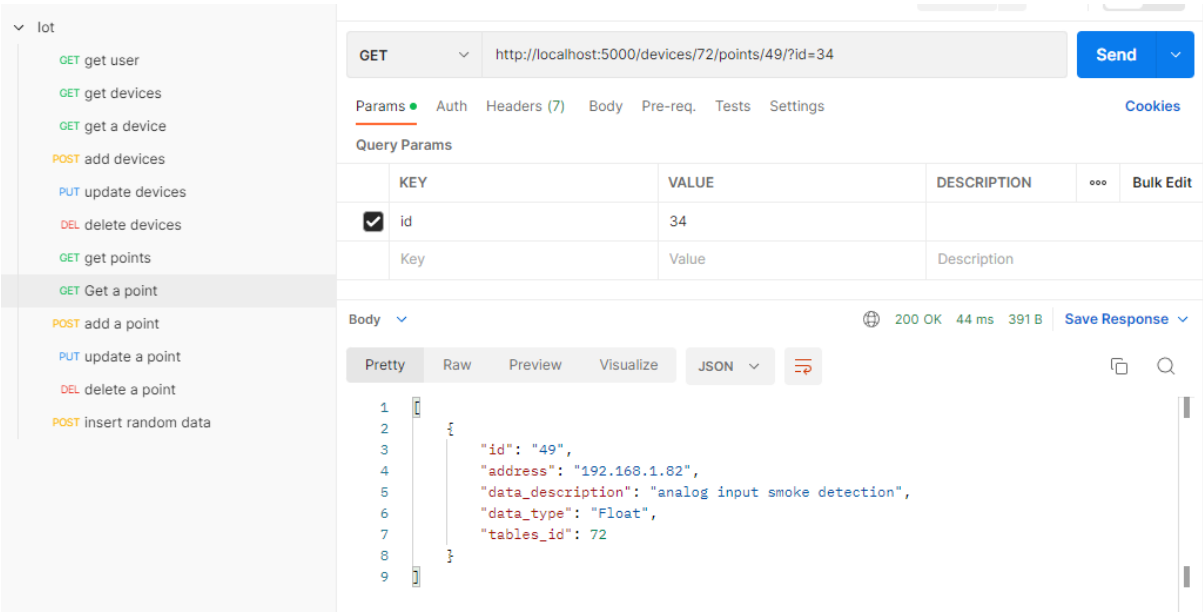
Έτσι για post στο “http://localhost:5000/devices/:deviceId/points/:pointId/allData/?id=39” δηλαδή για τη δημιουργία τυχαίων δεδομένων για device με id ίσο με 72 , id του Point ίσο με 49 και id του χρήστη ίσο με 39 όταν θα πραγματοποιείτε το POST request ως response έρχεται success εάν υπάρχουν ο χρήστης , η συσκευή και το point και γινεί επιταχυμένη η εισαγωγή στη βάση δεδομένων (εικόνα 5.22):

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212



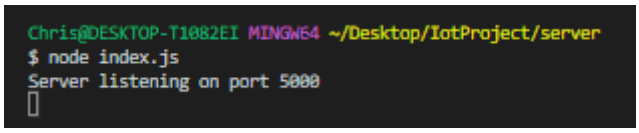
Εικόνα 5.22

Άλλο ένα παράδειγμα για Get ενός point με id 49 που ανήκει στη συσκευή με id 72 για user με id 34 ως response επιστρέφεται το JSON αρχείο με τα δεδομένα του συγκεκριμένου point δηλαδή το id , address, data_description, data_type και tables_id από τη βάση δεδομένων (εικόνα 5.23)



Εικόνα 5.23

Τέλος για να γίνει εκκίνηση του server εκτελείται στο terminal στη τοποθεσία όπου βρίσκεται το project για το φάκελο server η εντολή node index.js (εικόνα 5.24):



Εικόνα 5.24

Και έτσι ο server ακούει στο port 5000 όπου έχει οριστεί.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

5.3 Grafana

Μέσω του Grafana θα γίνει η οπτικοποίηση των δεδομένων (data visualization) σε μορφή διαγραμμάτων των συσκευών της βάσης δεδομένων. Στη συνέχεια θα γίνει αναφορά στο τρόπο εγκατάστασης του Grafana , στη διαδικασία για τη σύνδεση του με τη βάση δεδομένων και τέλος στο τρόπο με τον οποίο γίνεται η δημιουργία των γραφημάτων καθώς και πως αυτά θα μπορούν να εισαχθούν στην εφαρμογή.

5.4.1 Εγκατάσταση του Grafana - Παραμετροποιήσεις

Η εγκατάσταση του Grafana θα γίνει σε περιβάλλον Windows. Στην ιστοσελίδα υπάρχει documentation και ακολουθώντας τα βήματα μπορεί να γίνει επιτυχημένη εγκατάσταση με εύκολο τρόπο κατεβάζοντας το installer του Grafana της τελευταίας έκδοσης.

Η προεπιλεγμένη θύρα που χρησιμοποιεί είναι η 3000. Λόγω ότι η react εφαρμογή που θα δημιουργηθεί στο επόμενο κεφάλαιο χρησιμοποιεί και αυτή τη θύρα 3000 θα πρέπει να γίνουν κάποιες αλλαγές στο αρχείο ρυθμίσεων του Grafana. Έτσι πηγαίνοντας στο φάκελο εγκατάστασης του Grafana στο path “GrafanaLabs\grafana\conf” για αρχείο “sample.ini” θα γίνει αντιγραφή του και επικόλληση με αλλαγή ονόματος σε custom.ini (είναι λανθασμένη πρακτική οι αλλαγές να γίνονται στο sample.ini αφότου θα είναι δύσκολη η επαναφορά ρυθμίσεων). Στη συνέχεια ανοίγοντας το αρχείο custom.ini μπορούν να γίνουν οι αλλαγές στις ρυθμίσεις , αφαιρώντας τα comments “;” και αλλάζοντας τις ρυθμίσεις.

Οι αλλαγές που πρέπει να γίνουν είναι :

- Για την αλλαγή της προεπιλεγμένης θύρας από 3000 σε 3050 (εικόνα 5.25):

```
40  # The http port  to use
41  http_port = 3050
```

Εικόνα 5.25

- Για τη πρόσβαση των γραφημάτων στο web application μέσω iframe (εικόνα 5.26):

```
274  # set to true if you want to allow browsers to render Grafana in
275  # a <frame>, <iframe>, <embed> or <object>. default is false.
276  allow_embedding = true
```

Εικόνα 5.26

Και (εικόνα 5.27):

```
447  [auth.anonymous]
448  # enable anonymous access
449  enabled = true
450
451  # specify organization name that should be used for unauthenticated users
452  org_name = Main Org.
453
454  # specify role for unauthenticated users
455  org_role = Viewer
456
457  # mask the Grafana version number for unauthenticated users
458  hide_version = true
```

Εικόνα 5.27

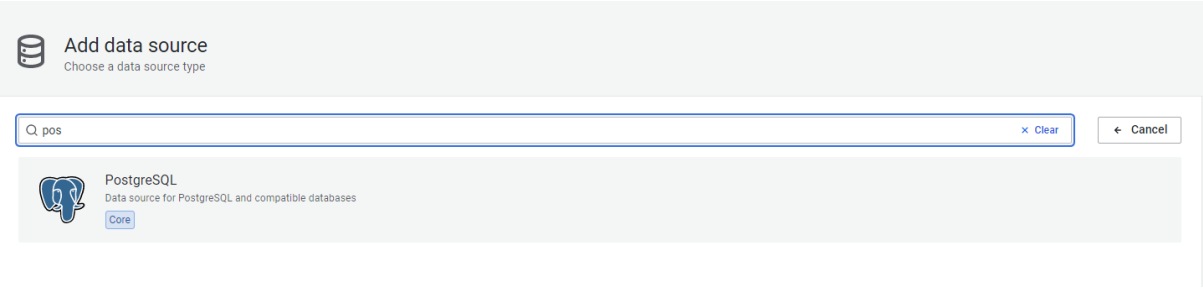
Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Η πρόσβαση στο Grafana γίνεται ανοίγοντας το φυλλομετρητή και γράφοντας ως διεύθυνση το “localhost:3050” όπου αρχικά θα ζητήσει όνομα και κωδικό για τη σύνδεση και βάζοντας “admin” και στα δυο πεδία δίνεται η πρόσβαση.

5.4.2 Σύνδεση του Grafana με τη βάση δεδομένων PostgreSQL

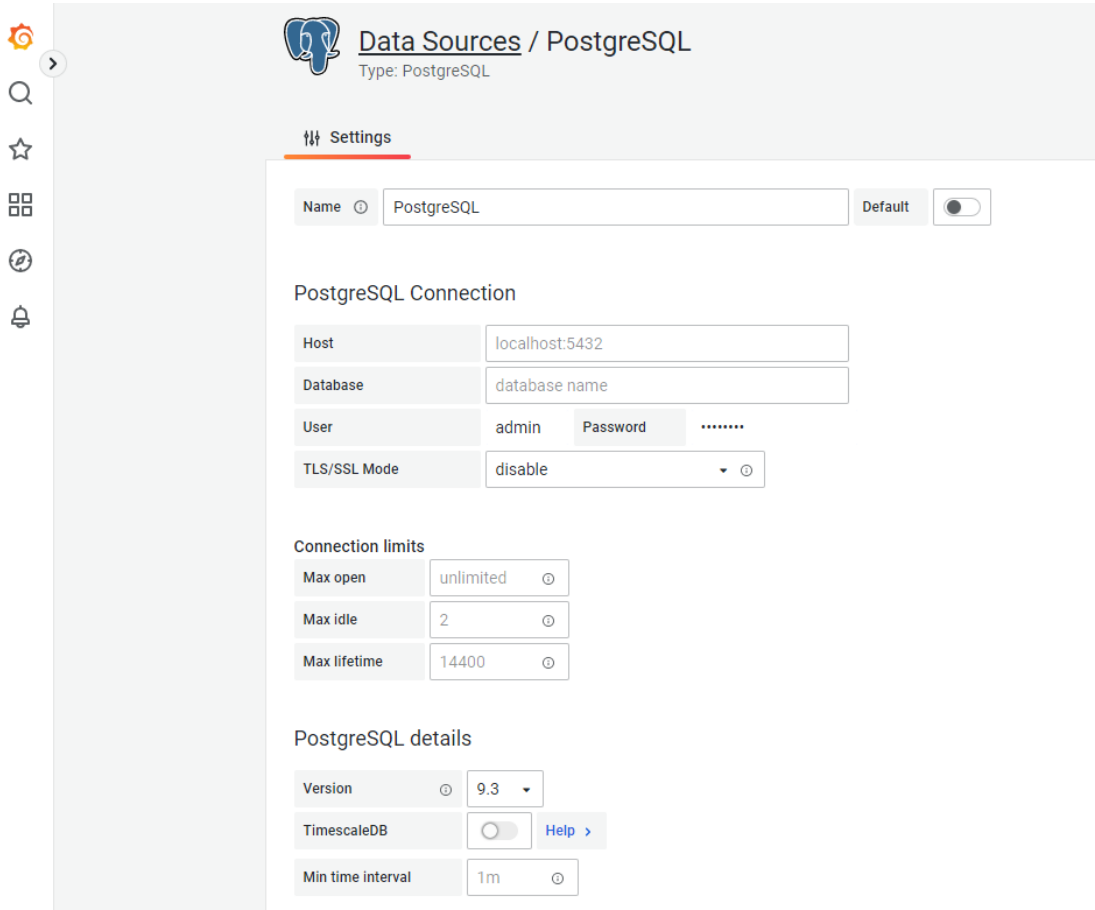
Αφότου έχει γίνει η πρόσβαση στο Grafana θα πρέπει να γίνει προσθήκη της πηγής δεδομένων στην οποία θα έχει πρόσβαση για τη δημιουργία των γραφημάτων.

Πατώντας στο configuration και διαλέγοντας add data source θα εμφανιστεί ένα μενού με τις διαθέσιμες βάσεις δεδομένων (εικόνα 5.28).



Εικόνα 5.28

Στη συγκεκριμένη εφαρμογή γίνεται η χρήση της PostgreSQL , οπότε επιλέγοντας τη ως data source θα μα εμφανίσει (εικόνα 5.29):



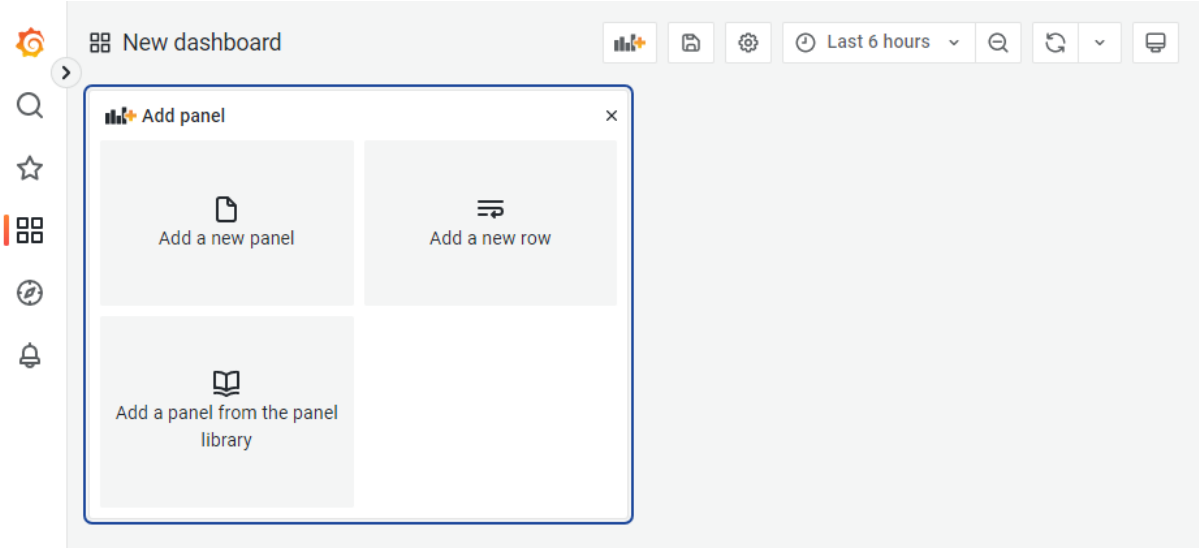
Εικόνα 5.29

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Εδώ γίνεται εισαγωγή των στοιχείων της βάσης δεδομένων (χρειάζεται προσοχή στην έκδοση της Postgre) και στη συνέχεια πατώντας στο save and test , εάν γίνει επιτυχημένα η σύνδεση θα εμφανίσει μήνυμα success data source is working.

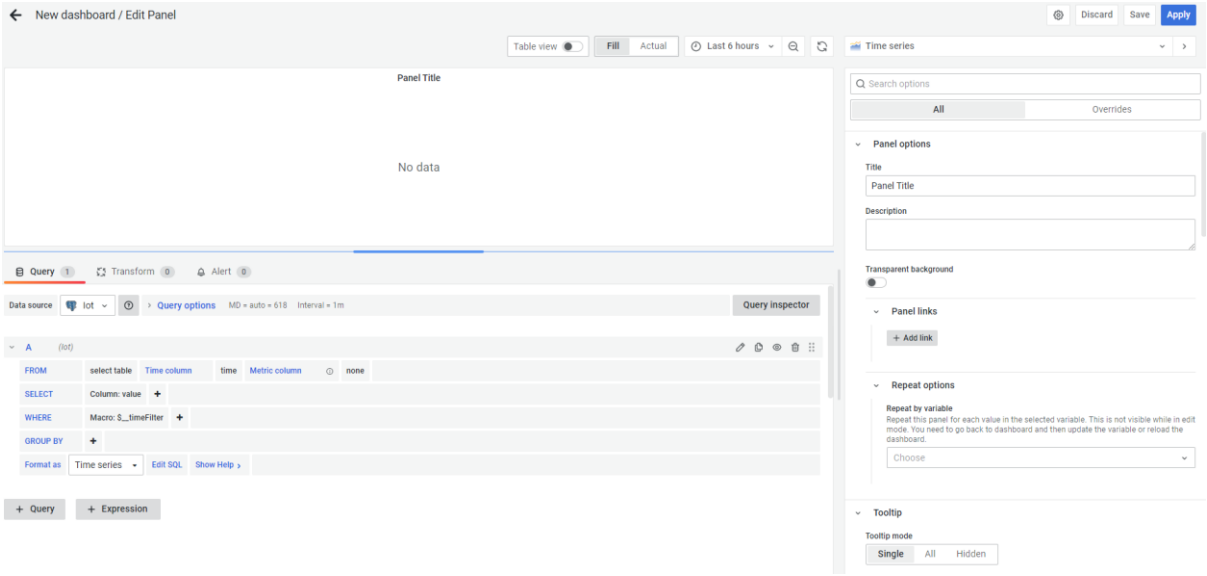
5.4.3 Δημιουργία γραφημάτων

Αφού γίνει η σύνδεση του Grafana με τα δεδομένα από τη PostgreSQL , για τη δημιουργία των γραφημάτων θα κατευθυνθούμε από το μενού στο Dashboards και μετά θα πατήσουμε στο “New dashboard” (εικόνα 5.30).



Εικόνα 5.30

Από εδώ πατώντας στο new panel κατευθυνόμαστε στο (εικόνα 5.31)

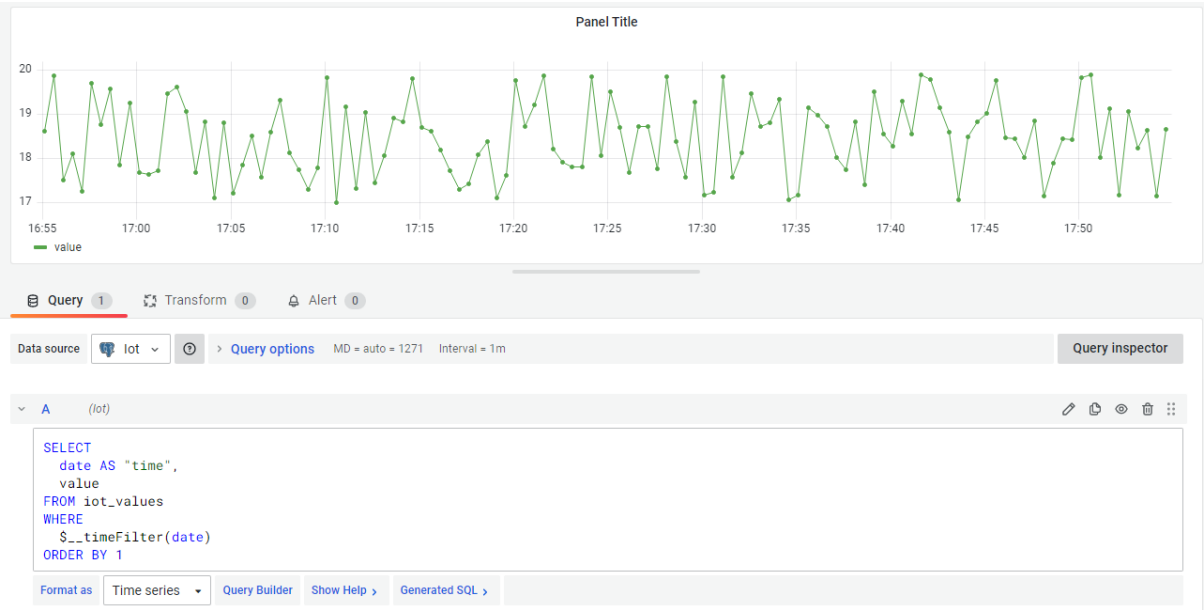


Εικόνα 5.31

όπου και μας δίνονται όλες οι επιλογές για το γράφημα που θέλουμε να φτιάξουμε, όπως τι visualization θέλουμε (time series, bar char, table , heatmap) , επιλογές για την εμφάνιση του γραφήματος (χρώμα , στυλ , στυλ των γραμμών) , όνομα του panel , επεξεργασία επιλογών για τα δεδομένα , κατώτατα όρια (threshold) και πολλά άλλα.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Το Grafana δίνει τη δυνατότητα να γράψουμε τα δικά μας SQL queries για τη δημιουργία των γραφημάτων πατώντας στο Edit SQL.
Το γράφημα που θα δημιουργήσουμε ένα ένα time series στο οποίο πρέπει να δηλωθεί ένα time column. Ως time column θα δηλώσουμε το column της βάσης δεδομένων που έχουμε στο οποίο υπάρχουν τα χρονικά δεδομένα , όπου για πίνακα iot_values θα είναι το date (όπου είναι τύπου timestamp). Έτσι για τη δημιουργία ενός γραφήματος το SQL query θα είναι (εικόνα 5.32):



Εικόνα 5.32

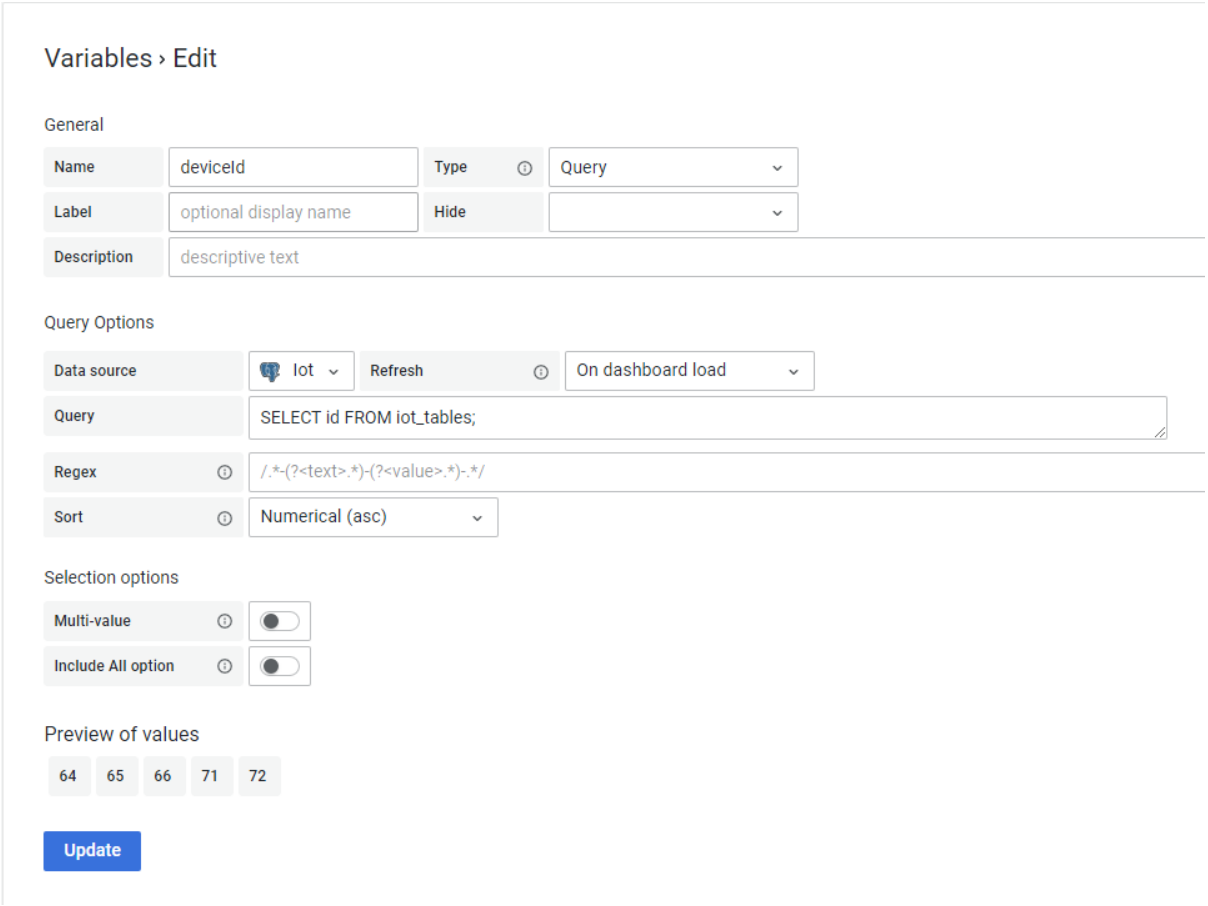
Μια ακόμα δυνατότητα που δίνει το Grafana είναι η δημιουργία μεταβλητών. Είναι αναγκαία η χρήση μεταβλητών έτσι ώστε να μπορεί να γίνει δυναμικά η δημιουργία ενός γραφήματος ανάλογα με τις συσκευές που θα έχουμε στη βάση δεδομένων.

Η βάση δεδομένων μας αποτελείται από devices και points και μέσω της επιλογής του συγκεκριμένου point (όπου ανήκει σε κάποιο device) θα γίνεται η δημιουργία του γραφήματος για τα δεδομένα που ανήκουν στο συγκεκριμένο Point.

Έτσι θέλουμε 4 μεταβλητές, μια deviceId όπου θα περιέχει τα devices της βάσης μας και μια pointsId για τα points μέσω των οποίων θα γίνεται ο διαχωρισμός των συσκευών που έχουμε, μια για το όνομα της συσκευής (name από iot_tables ώστε να ονομάζουμε δυναμικά το διάγραμμα ανάλογα με τη συσκευή που βρισκόμαστε , έτσι το όνομα θα είναι πχ “ temp_sensor”.

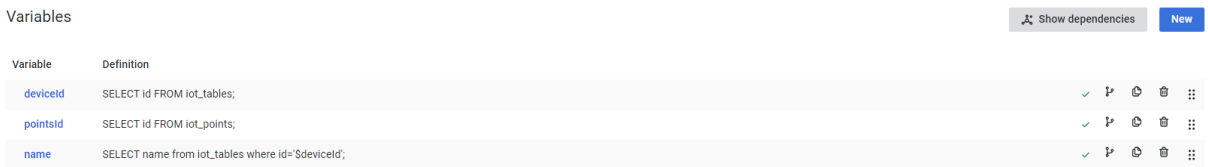
Πατώντας στα dashboard settings μας εμφανίζεται ένα μενού ρυθμίσεων και επιλέγοντας το variables μπορούμε να δημιουργήσουμε τις μεταβλητές μας.
Επιλέγοντας new κατευθυνόμαστε στη σελίδα όπου μπορούμε να δημιουργήσουμε ένα query από το οποίο θα επιλέγονται οι τιμές της μεταβλητής (εικόνα 5.33).

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212



Εικόνα 5.33

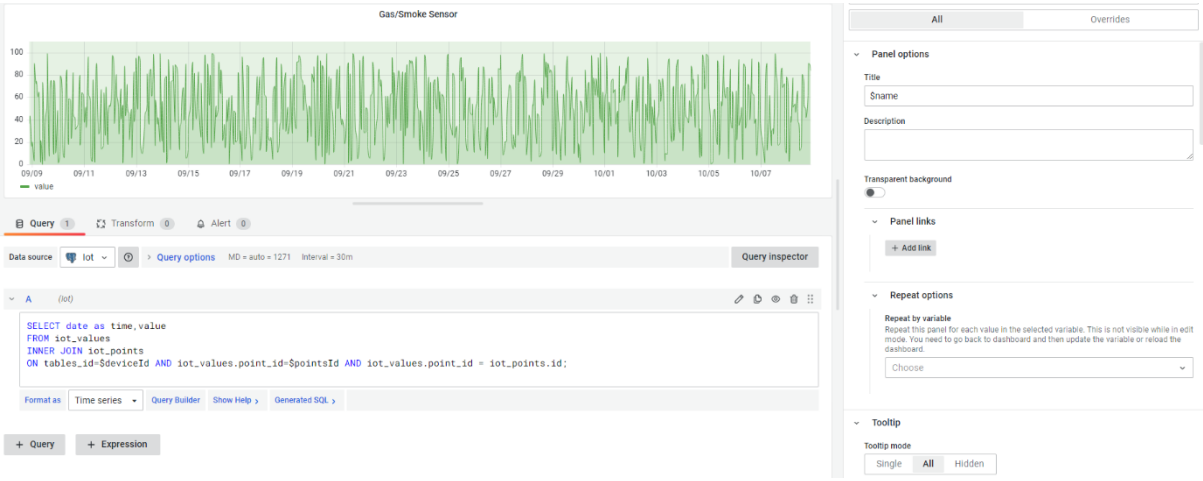
Εδώ δημιουργούμε τη μεταβλητή deviceId της οποίας οι τιμές είναι τα id των devices της βάσης δεδομένων. Αυτό επιτυγχάνεται με το query "SELECT id FROM iot_tables;" και από κάτω μπορούμε να δούμε ένα preview με τις τιμές (id) που θα πάρει η μεταβλητή. Πατώντας update δημιουργείται η μεταβλητή.
Παρακάτω φαίνονται όλες οι μεταβλητές που δημιουργούμε με τα queries τους (εικόνα 5.34):



Εικόνα 5.34

Αφότου έχουν δημιουργηθεί οι μεταβλητές πλέον μπορούν να δημιουργηθούν και τα γραφήματα που θα εφαρμοστούν στην εφαρμογή μας.
Έτσι πηγαίνοντας πάλι στο new panel θα δημιουργήσουμε το time series γράφημα όπου θα χρησιμοποιήσουμε στην εφαρμογή μας (εικόνα 5.35).

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212



Εικόνα 5.35

Πιο συγκεκριμένα :

1. Έχει γίνει επιλογή της βάσης δεδομένων Iot από τη PostgreSQL (εικόνα 5.36):

Εικόνα 5.36

2. Έχει γίνει αλλαγή του ονόματος (title) του panel σε (εικόνα 5.37):

Εικόνα 5.37

Με το “$” γίνεται η δήλωση της μεταβλητής που έχουμε δημιουργήσει στο προηγούμενο βήμα.

3. Το query που χρησιμοποιούμε είναι το εξής (εικόνα 5.38):

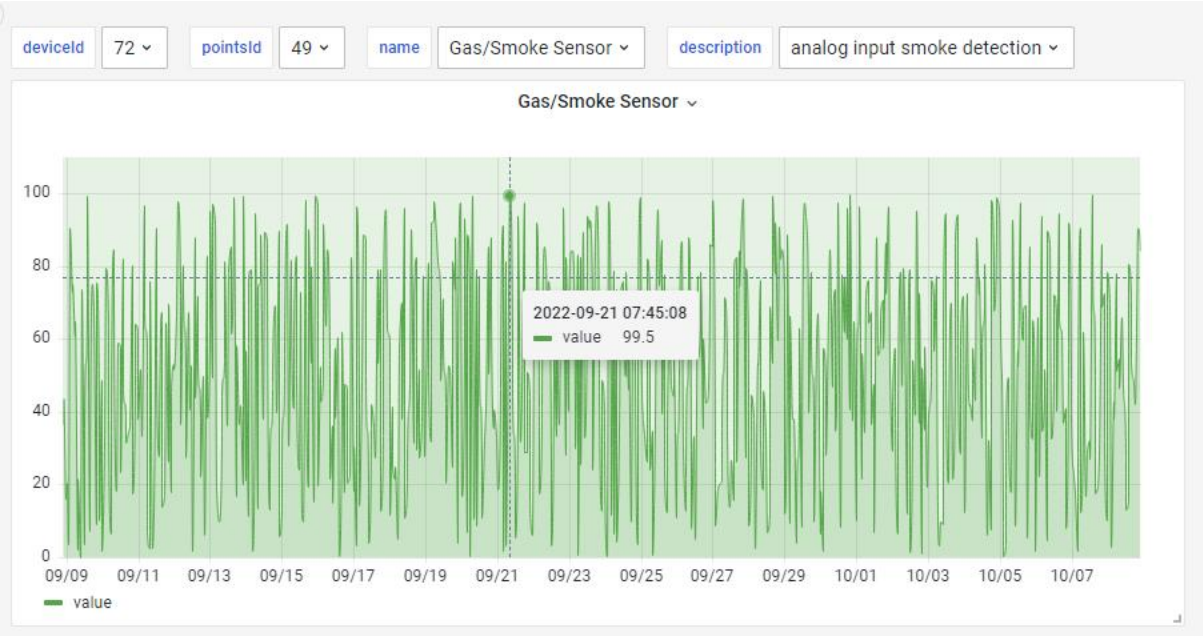
```
SELECT date as time,value
FROM iot_values
INNER JOIN iot_points
ON tables_id=$deviceId AND iot_values.point_id=$pointsId AND iot_values.point_id = iot_points.id;
```

Εικόνα 5.38

Επιλέγουμε από τη βάση δεδομένων Iot , τα columns date και value του πίνακα iot_values, και επιλέγουμε μόνο τις εγγραφές στις οποίες το point_id του iot_values είναι ίδιο με το id του iot_points, για id του πίνακα iot_points και id του πίνακα iot_tables να είναι ίσα με τις τιμές των μεταβλητών που έχουμε ορίσει.
Με αυτό το τρόπο όταν θα ζητάμε τα δεδομένα από ένα συγκεκριμένο point ενός device θα επιλέγονται τα δεδομένα τα οποία ζητήσαμε από τη βάση δεδομένων και θα χρησιμοποιούνται για τη δημιουργία του γραφήματος.

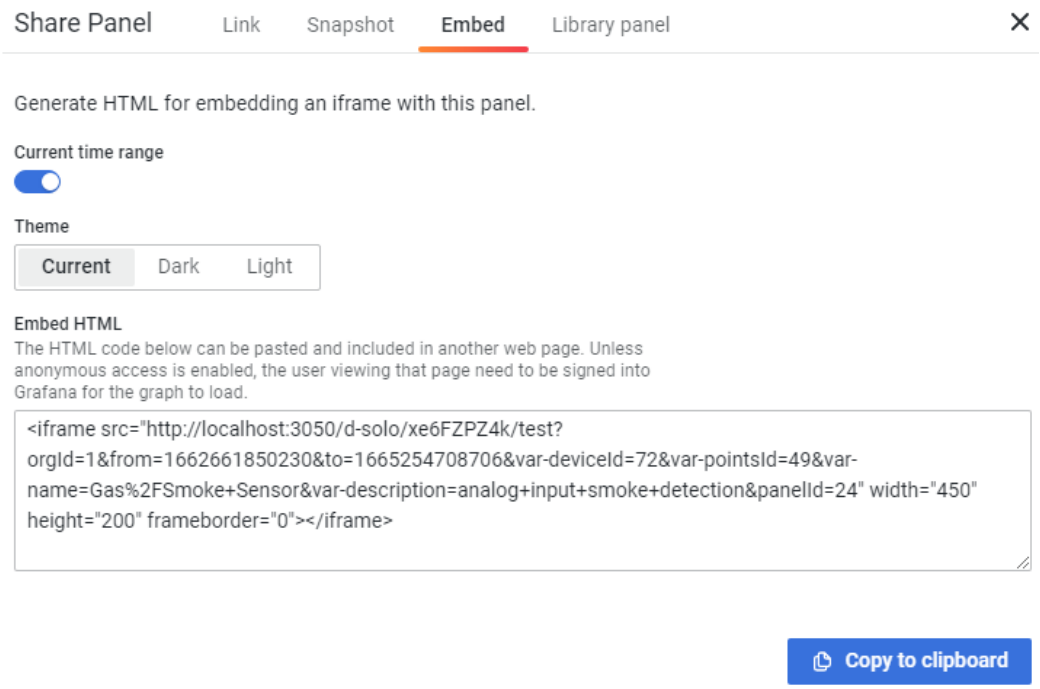
Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Αποθηκεύοντας τις αλλαγές , στο dashboard θα μπορούμε να δούμε το διάγραμμα , και αλλάζοντας τις τιμές των μεταβλητών αυτό αλλάζει δυναμικά (εικόνα 5.39).



Εικόνα 5.39

Τέλος αφότου δημιουργήσαμε το panel για να το χρησιμοποιήσουμε στην εφαρμογή μας, πατώντας στις επιλογές του και επιλέγοντας share και Embed θα μας δημιουργήσει ένα iframe το οποίο στη συνέχεια θα εισάγουμε στη σελίδα μας (εικόνα 5.40):



Εικόνα 5.40

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

5.4 Σχεδιασμός και ανάπτυξη Front-end

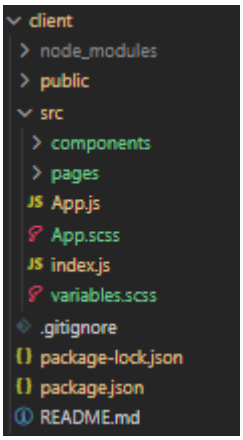
Το front-end της εφαρμογής αναπτύχθηκε πάνω στη βιβλιοθήκη JavaScript React. Στο project θα δημιουργήσουμε τη react εφαρμογή μας μέσω της εντολής " npx create-react-app client " η οποία θα δημιουργήσει τα απαραίτητα αρχεία για τη react εφαρμογή με το όνομα client.

5.4.1 Δομή αρχείων

Η δομή των αρχείων φαίνεται στην εικόνα 5.41:

Εκεί έχουμε :

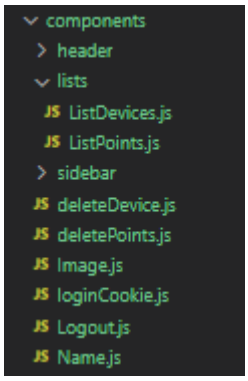
- App.js: Ο πυρήνας της λειτουργίας της εφαρμογής. Σε αυτό το αρχείο περιέχεται η βασική αρχικοποίηση της ιστοσελίδας. Αποτελεί το αρχείο για το App component που λειτουργεί ως container για όλα τα άλλα στοιχεία (component). Ακόμα από εκεί γίνεται η δρομολόγηση στις διάφορες σελίδες.
- index.js: Περιέχει τη λογική για να γίνει το render του App στοιχείου (component)
- variables.scss: Περιέχει δηλώσεις μεταβλητών για scss για την απλούστερη αλλαγή μεταβλητών για την εμφάνιση της σελίδας.
- Φάκελος public: Περιέχει στατικά αρχεία όπως το index.html , αρχεία βιβλιοθήκης JavaScript , εικόνες όπως το favicon για τη σελίδα και άλλα στοιχεία.
- Φάκελος components
- Φάκελος pages



Εικόνα 5.41

Τα components είναι τα βασικά δομικά στοιχεία κάθε react εφαρμογής. Ο φάκελος αποτελείται από μια συλλογή στοιχείων διεπαφής που μπορούν να χρησιμοποιηθούν στα διάφορα αρχεία.
Η δομή του φακέλου components είναι η εικόνα 5.42:

- Στο header και sidebar έχουμε τα components για τη κατασκευή του κύριου μενού στη πάνω μεριά της σελίδας και της πλαϊνής γραμμής μενού
- Το lists περιέχει τις λίστες για τα devices και points αντίστοιχα
- Στο Name.js επιστρέφετε το όνομα του χρήστη που έχει κάνει login
- Το Logout.js επιστρέφει κουμπί Logout όπου τρέχει το κώδικα για την αποσύνδεση του χρήστη
- Το Image.js επιστρέφει εικόνα με συγκεκριμένες ρυθμίσεις έτσι ώστε να μπορούμε να το επαναχρησιμοποιήσουμε σε διάφορα κομμάτια κώδικα
- Το loginCookie, deleteDevice και deletePoints είναι functions που περιέχουν κώδικα για να γίνει έλεγχος εάν είναι συνδεδεμένος ο χρήστης μέσω του cookie, για τη διαγραφή ενός device και point αντίστοιχα



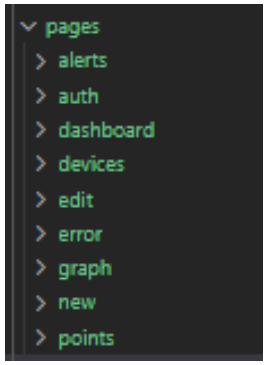
Εικόνα 5.42

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Φάκελος pages: Ο φάκελος περιέχει υποφακέλους για κάθε σελίδα της εφαρμογής ξεχωριστά. Μέσα σε κάθε υποφάκελο υπάρχει το αρχείο .js της κάθε σελίδας, το scss αρχείο της και οι εικόνες που υπάρχουν για κάθε σελίδα ξεχωριστά.

Η δομή του φακέλου pages είναι η εικόνα 5.43:

- Στο φάκελο auth έχουμε τη σελίδα για το login και register του χρήστη
- Στο φάκελο dashboard τη κεντρική σελίδα της εφαρμογής
- Στο devices τη σελίδα στην οποία εμφανίζονται τα devices
- Στο points τη σελίδα όπου εμφανίζονται τα points
- Στο edit τη σελίδα για το edit των devices και points αντίστοιχα
- Στο new τη σελίδα για εισαγωγή νέου device ή point αντίστοιχα
- Στο error τη σελίδα error 404 όταν ο χρήστης εισάγει URL σελίδα η οποία δεν υπάρχει
- Στο graph τη σελίδα στην οποία εμφανίζεται το γράφημα μαζί με όλες τις πληροφορίες του device και point στο οποίο ανήκει
- Το alert είναι μια κεντρική σελίδα για alert για τη πιλοτική εφαρμογή



Εικόνα 5.43

5.4.2 Παρουσίαση εφαρμογής και επεξήγηση κώδικα

Για το αρχείο App.js αρχικά θα μελετηθούν οι εισαγωγές (imports).

Εισάγουμε τη bootstrap η οποία θα χρησιμοποιηθεί για το login και register, το scss αρχείο για το App.js, το useState Hook της react ώστε να παρακολουθούμε τη κατάσταση της εφαρμογής όπως δεδομένα και ιδιότητες της, το useEffect Hook το οποίο το χρησιμοποιούμε για να τρέξουμε κώδικα(κλήση κάποιων functions) κάθε φορά που κάνουμε render κάποια σελίδα , το loginCookie για να γνωρίζουμε εάν ο χρήστης είναι συνδεδεμένος , το BrowserRouter , Routes και Route από το react-router-dom η οποία είναι μια βιβλιοθήκη για δρομολόγηση η οποία επιτρέπει τη πλοήγηση των σελίδων στη react και επιτρέπει την αλλαγή της διεύθυνσης URL και τέλος τις σελίδες που έχουμε δημιουργήσει Εικόνα 5.44.

```
1  import "bootstrap/dist/css/bootstrap.min.css";
2  import "./App.scss";
3  import React, { useState, useEffect } from "react";
4  import { BrowserRouter, Routes, Route } from "react-router-dom";
5  import loginCookie from "./components/loginCookie";
6  import Dashboard from "./pages/dashboard/Dashboard";
7  import Error from "./pages/error/Error";
8  import Auth from "./pages/auth/Auth";
9  import Devices from "./pages/devices/Devices";
10 import New from "./pages/new/New";
11 import Edit from "./pages/edit/Edit";
12 import Header from "./components/header/Header";
13 import Sidebar from "./components/sidebar/Sidebar";
14 import Points from "./pages/points/Points";
15 import EditPoints from "./pages/edit/EditPoints";
16 import NewPoint from "./pages/new/NewPoint";
17 import Graph from "./pages/graph/Graph";
18 import Alert from "./pages/alerts/Alert";
```

Εικόνα 5.44 Imports

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Στη συνέχεια για το αρχείο App.js έχουμε Εικόνα 5.45:

```
20  function App() {
21    const [isLogged, setIsLogged] = useState();
22
23    useEffect(() => {
24      loginCookie({ setIsLogged });
25    }, []);
26
27    return (
28      <div className="wrapper">
29        {!isLogged ? (
30          <Auth />
31        ) : (
32          <BrowserRouter>
33            <div className="home">
34              <Sidebar />
35              <div className="homeContainer">
36                <Header />
37                <div className="dashboard">
38                  <Routes>
39                    <Route path="/">
40                      <Route index element={<Dashboard />} />
41                      <Route path="dashboard" element={<Dashboard />} />
42                      <Route path="alerts" element={<Alert />} />
43                      <Route path="devices">
44                        <Route index element={<Devices />} />
45                        <Route path="new" element={<New />} />
46                        <Route path="edit/:id" element={<Edit id={window.location.pathname} />}/>
47                        <Route path=":deviceId" >
48
49                          <Route path="points">
50                            <Route index element={<Points />} />
51                            <Route path="new" element={<NewPoint />} />
52                            <Route path="edit/:id" element={<EditPoints/>}/>
53                            <Route path=":pointId" element={<Graph id={window.location.pathname} />}/>
54                          </Route>
55                        </Route>
56                      </Route>
57                      <Route path="*" element={<Error />} />
58                    </Route>
59                  </Routes>
60                </div>
61              </div>
62            </div>
63          </BrowserRouter>
64        )}
65      </div>
66    );
67  }
```

Εικόνα 5.45

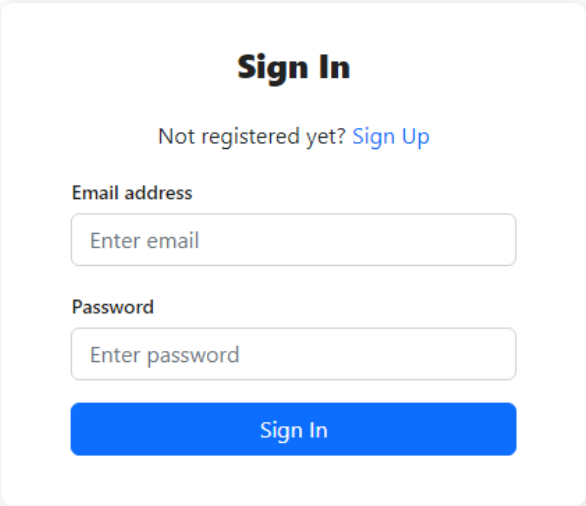
Πιο συγκεκριμένα στην αρχή αρχικοποιούμε τη μεταβλητή isLogged να είναι κενή και καλώντας την loginCookie από το αρχείο loginCookie.js, στο οποίο γίνεται ο έλεγχος για το αν είναι συνδεδεμένος ο χρήστης , λαμβάνει τιμή εάν ισχύει η συνθήκη. Στη συνέχεια κάνουμε έλεγχο εάν είναι κενή η μεταβλητή isLogged και εάν είναι τότε να εμφανίζεται η σελίδα για το login (Auth). Με αυτό το τρόπο περιορίζουμε τη πρόσβαση στην εφαρμογή μας αφότου οποιαδήποτε σελίδα και να ζητήσει ένας χρήστης δεν θα μπορεί να έχει πρόσβαση σε αυτή εάν δεν είναι συνδεδεμένος. Στη συνέχεια καλούμε το sidebar και το dashboard αφότου θα είναι σταθερά σε όλες τις σελίδες μας και μετά μέσω των routes θα καλούμε διαφορετική κύρια σελίδα στην εφαρμογή μας αναλόγως το path το οποίο θα υπάρχει στη URL του χρήστη. Έτσι για παράδειγμα με URL “http://localhost:3000/devices” εάν ο χρήστης είναι συνδεδεμένος θα μας εμφανίσει τη σελίδα η οποία θα αποτελείται από header , sidebar και ακόμα τη σελίδα Devices από το αρχείο Devices.js από το φάκελο pages. Ακόμα γίνεται χρήση παραμέτρων URL όπως στο path = “edit:id” όπου το χρησιμοποιούμε ώστε να γίνεται δήλωση του route δυναμικά και έτσι εάν ο χρήστης ζητήσει ένα συγκεκριμένο device με το id του

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

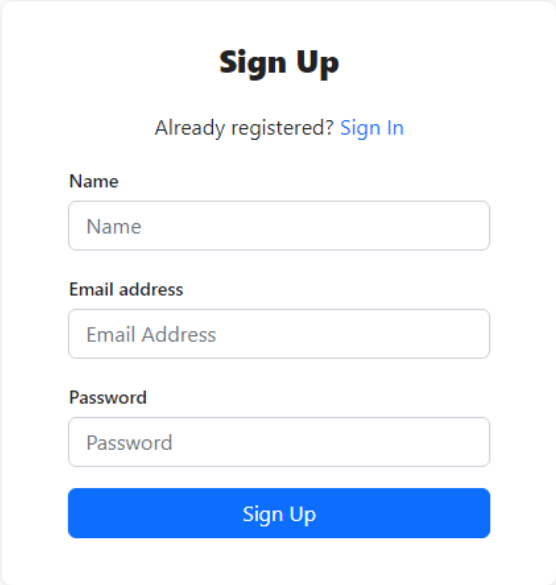
“http://localhost:3000/devices/edit/72” εάν υπάρχει το device , να του δείχνει τη σελίδα Edit δηλαδή τη σελίδα για την επεξεργασία της συγκεκριμένης συσκευής, με id ίσο με 72.

Στη συνέχεια θα παρουσιαστούν οι διάφορες σελίδες της εφαρμογής, με παράλληλη ανάλυση τμημάτων κώδικα.

Για την οθόνη εισόδου έχουμε εικόνα 5.45 και εικόνα 5.46 :

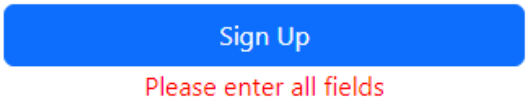


Εικόνα 5.45



Εικόνα 5.46

Κάθε φορά που ο χρήστης θα εισέρχεται στην εφαρμογή , εάν δεν έχει κάνει login θα του εμφανίζεται η σελίδα για σύνδεση δηλαδή η σελίδα από το αρχείο Auth.js. Από εκεί πατώντας sign up θα του εμφανίζεται η σελίδα για τη δημιουργία λογαριασμού. Για το sign up θα πρέπει να εισάγει το όνομα του , το email το οποίο δεν θα πρέπει να υπάρχει στη βάση δεδομένων και τέλος το password. Εάν κάνει κάποιο λάθος του εμφανίζεται κατάλληλο μήνυμα εικόνα 5.47 , ενώ εάν κάνει την εγγραφή επιτυχημένα θα εμφανίζει alert το οποίο θα τον κατευθύνει στη σελίδα εισόδου εικόνα 5.48. Με τον ίδιο τρόπο λειτουργεί και το login με το email και το κωδικό , όπου με επιτυχημένη είσοδο, κατευθύνει το χρήστη στη σελίδα την οποία ζήτησε.



Εικόνα 5.47

Εικόνα 5.48

Η συγκεκριμένη σελίδα δημιουργήθηκε με τη χρήση της bootstrap , η οποία παρέχει το στυλ της φόρμας και κάνει τη σελίδα responsive. Το sign-in και sign-up βρίσκονται στην ίδια σελίδα και μέσω use state hook όπου ως τιμή κρατάει login ή signup μπορεί να αλλάξει τη κατάσταση και να

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

εμφανίσει τη συγκεκριμένη φόρμα ανάλογα με τη τιμή που θα έχει. Για το sign-in και sign-up καλούνται συναρτήσεις οι οποίες κάνουν fetch στο σερβερ μας για την εισαγωγή των στοιχείων του χρήστη στη βάση δεδομένων (sign-up) η τον έλεγχο εάν είναι σωστά τα δεδομένα εισόδου (sign-in). Εάν γίνει επιτυχημένη είσοδος, θα αποθηκευτεί το id που έχει ως cookie στο browser του από όπου και θα έχει πρόσβαση στη σελίδα. Κάποια τμήματα κώδικα εμφανίζονται στην εικόνα 5.49 ,όπου δείχνει τη δημιουργία της φόρμας για login και στην εικόνα 5.50 όπου δείχνουμε τη συνάρτηση η οποία καλείται όταν γίνεται το sign up και γίνεται το fetch από το server. Για το login χρησιμοποιούνται κλάσεις της bootstrap , όταν γίνεται click του sign-up καλείται η changeAuthMode η οποία αλλάζει το state σε sign -up και τέλος όταν γίνεται submit καλείται η handleLoginSubmit η οποία κάνει και το fetch. Με τον ίδιο τρόπο λειτουργεί και για sign-up η οποία καλεί τη postSign όπου γίνεται το fetch μέσω του axios. Περισσότερες πληροφορίες για το axios θα δοθούν για επόμενη σελίδα που θα είναι πιο σημαντική για την εφαρμογή.

```
53  // Sign up
54  const postSign = async() =>{
55    try{
56      await Axios.post("http://localhost:5000/sign",{
57        name:name,
58        email:email,
59        password:password
60      })
61      .then((response) => {
62        if(response.data === "Success"){
63          alert("Succesfully registerd. Please sign in");
64          changeAuthMode();
65        }
66        else{
67          setErrors(response.data);
68        }
69      });
70      }catch(err){
71        console.error(err.message);
72      }
73  }
```

Εικόνα 5.49

```
98  // Login Form
99  if (authMode === "login") {
100   return (
101     <div className="Auth-form-container">
102       <form onSubmit={handleLoginSubmit} className="Auth-form">
103         <div className="Auth-form-content">
104           <h3 className="Auth-form-title">Sign In</h3>
105           <div className="text-center">
106             Not registered yet?{" "}
107             <span className="link-primary" onClick={changeAuthMode}>
108               Sign Up
109             </span>
110           </div>
111           <div className="form-group mt-3">
112             <label>Email address</label>
113             <input
114               onChange={(e) =>setEmail(e.target.value)}
115               name="email"
116               type="email"
117               className="form-control mt-1"
118               placeholder="Enter email"
119               value={email}
120             />
121           </div>
122           <div className="form-group mt-3">
123             <label>Password</label>
124             <input
125               onChange={(e) =>setPassword(e.target.value)}
126               name="password"
127               type="password"
128               className="form-control mt-1"
129               placeholder="Enter password"
130               value={password}
131             />
132           </div>
133           <div className="d-grid gap-2 mt-3">
134             <button type="submit" className="btn btn-primary">
135               Sign In
136             </button>
137           </div>
138           <p style={{color:"red", textAlign:"center"}}>{errors}</p>
139         </div>
140       </form>
141     </div>
142   )
143 }
```

Εικόνα 5.50

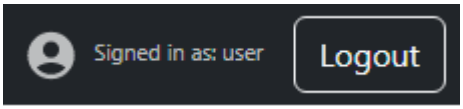
Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Η αρχική σελίδα της εφαρμογής δηλαδή η “http://localhost:3000/” όπως και η “ http://localhost:3000/dashboard” εμφανίζουν τη σελίδα εικόνα 5.51

Εικόνα 5.51

Όπως είχε αναφερθεί και προηγούμενος οποιαδήποτε σελίδα εκτός της auth όταν καλείται αποτελείται από το συνδυασμό του header , sidebar και της σελίδας την οποία ζητάει ο χρήστης.

Έτσι όπως φαίνεται παραπάνω στο πάνω μέρος της οθόνης υπάρχει το header.
Το Header.js αποτελείται από το component Name.js όπου εμφανίζει πάνω δεξιά το όνομα με το οποίο έχει εισέλθει ο χρήστης και το component Logout.js όπου είναι το κουμπί με το οποίο ο χρήστης μπορεί να κάνει το logout από την εφαρμογή εικόνας. Το logout πραγματοποιείται διαγράφοντας το cookie με το id του χρήστη που είναι συνδεδεμένος εκείνη τη στιγμή εικόνα 5.52 και εικόνα 5.53.



Εικόνα 5.52

```
4   export default function Logout() {
5
6     const handleOnClick = () =>{
7       Cookies.remove("id");
8       window.location.reload();
9     }
10
11    return <Button variant="outline-light" onClick={handleOnClick}>Logout</Button>;
12  }
```

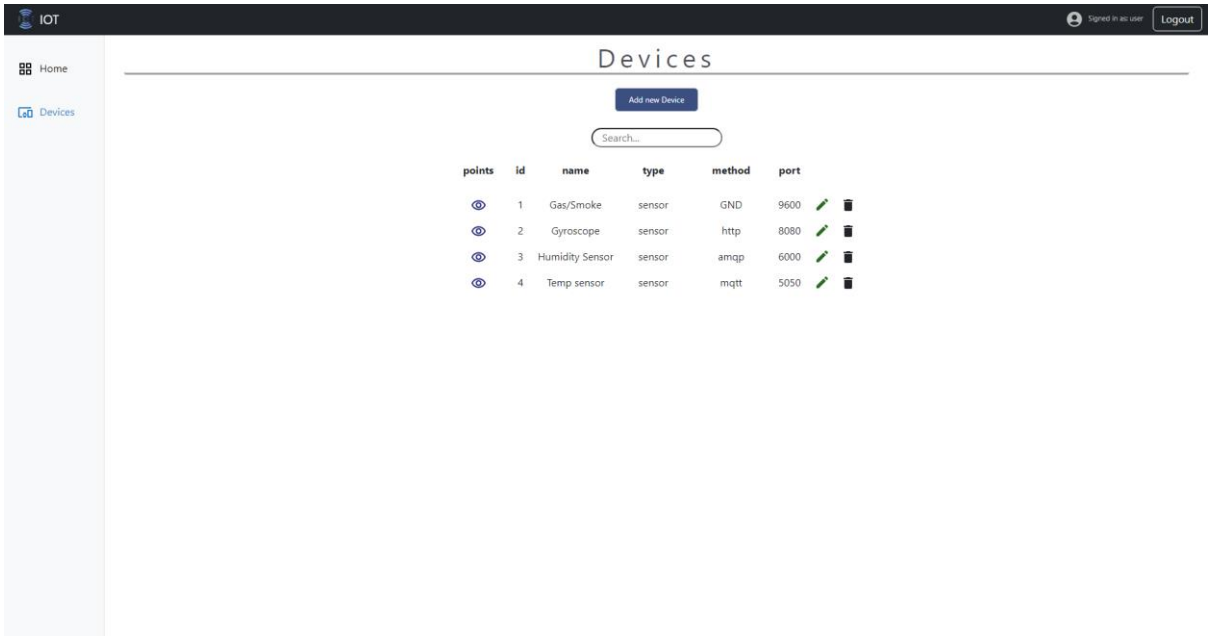
Εικόνα 5.53

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Το Sidebar.js αποτελείται από το logo στο πάνω αριστερά μέρος της οθόνης, το οποίο πατώντας το, κατευθύνει το χρήστη στην αρχική οθόνη. Ακόμα περιέχει συνδέσμους πλοήγησης για την αρχική και τα devices. Όταν ενεργοποιείται ένας τέτοιος σύνδεσμος αλλάζει το χρώμα του σε μπλε για να μας υποδείξει που βρισκόμαστε. Ακόμα για το responsive design για width κάτω των 600 πιξελ εμφανίζονται μόνο τα εικονίδια των συνδέσμων πλοήγησης ώστε να μικραίνει σε μέγεθος το sidebar.

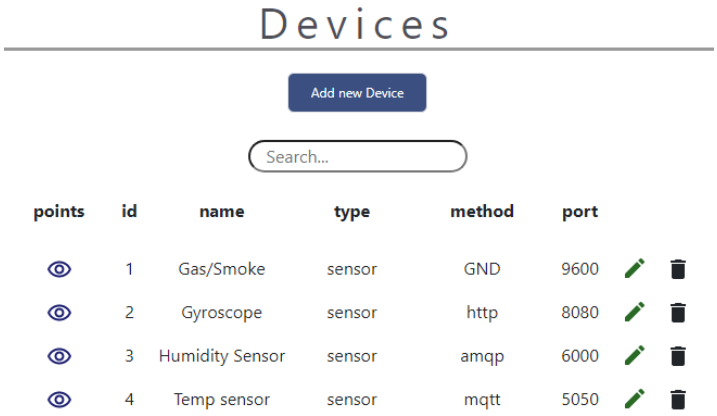
Αφότου έγινε ανάλυση των στατικών header και sidebar τώρα μπορεί να γίνει πλέον ανάλυση των σελίδων όπου έχουμε.
Η αρχική σελίδα όπως είδαμε παραπάνω περιέχει μια εικόνα iot συσκευών και ως τίτλο το όνομα της εφαρμογής “Γραφικό περιβάλλον εφαρμογής IOT” και το όνομα του χρήστη που έχει εισέλθει.

Στη συνέχεια πατώντας Devices στο sidebar γίνεται κατεύθυνση στη σελίδα ”http://localhost:3000/devices” όπου και περιέχονται οι συσκευές που έχει ο χρήστης (εικόνα 5.54).



Εικόνα 5.54

Πιο συγκεκριμένα ο χρήστης έχει τη δυνατότητα να εισάγει καινούργιες συσκευές , να επεξεργαστεί τις ήδη υπάρχων και ακόμα να τις διαγράψει , να κάνει αναζήτηση μέσω ονόματος , τύπου, μεθόδου ή port και τέλος πατώντας στο points της κάθε συσκευής να κατευθυνθεί στα points που ανήκουν στη συγκεκριμένη συσκευή.



Εικόνα 5.55

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Στη συνέχεια θα γίνει ανάλυση του κώδικα της σελίδας Devices.js.

Αρχικά θα γίνει ανάλυση των imports. Εδώ έχουμε το cookies από το js-cookie το οποίο το χρησιμοποιούμε για να μας βοηθήσει να αποκτήσουμε πρόσβαση στο cookie που είναι αποθηκευμένο στο browser μας. Αυτό το cookie αφότου δημιουργήθηκε από την επιτυχημένη είσοδο στην εφαρμογή περιέχει το id του χρήστη όπου αργότερα μέσω αυτού θα γίνει το σωστό API call ώστε να γίνει πρόσβαση μόνο στα αρχεία που ανήκουν στο συγκεκριμένο χρήστη. Με την εντολή “const userId = Cookies.get("id");” παίρνουμε τη τιμή του cookie και την αποθηκεύουμε σε μια μεταβλητή userId. Στη συνέχεια εισάγουμε τη βιβλιοθήκη axios η οποία είναι μια HTTP βιβλιοθήκη βασισμένη σε promises και δίνει τη δυνατότητα να γίνεται με εύκολο τρόπο η αλληλεπίδραση με μια υπηρεσία API. Έτσι μπορούμε να κάνουμε GET , POST , PUT και DELETE αιτήματα στο back-end που έχουμε δημιουργήσει εικόνα 5.56.

```
1  import React, { useState, useEffect } from "react";
2  import "./devices.scss";
3  import Axios from "axios";
4  import Cookies from "js-cookie";
```

Εικόνα 5.56

Τη στιγμή που φορτώνεται η σελίδα μέσω του useEffect καλείται η συνάρτηση getDevices εικόνα 5.57

```
38  useEffect(() => {
39    getDevices();
40  }, []);
```

Εικόνα 5.57

Η συνάρτηση getDevices φαίνεται στην εικόνα 5.58. Πραγματοποιείται ένα Get HTTP request στο API που έχουμε δημιουργήσει. Το request θα γίνει στο server στη σελίδα “http://localhost:5000/devices/all/?id=${userId}”. Από εκεί ζητάμε να μας επιστρέψει όλες τις συσκευές που ανήκουν στο συγκεκριμένο χρήστη, το id του χρήστη το περνάμε με τη χρήση query string στη URL παίρνοντας το ως παράμετρο id. Στη συνέχεια τα δεδομένα (response.data) μου μας επιστρέφει ο server τα περνάμε στη μεταβλητή devices που δηλώσαμε παραπάνω. Το query που τρέχει στο server για την ανάκτηση των δεδομένων από το server θυμίζουμε ότι είναι το “ "SELECT * FROM iot_tables WHERE user_id=$1 ORDER BY name",[userId], {res.send(results.rows);}”

```
14  const [devices, setDevices] = useState("");
15
16  const getDevices = async () => {
17    try {
18      await Axios.get(`http://localhost:5000/devices/all/?id=${userId}`).then(
19        (response) => {
20          setDevices(response.data);
21        }
22      );
23    } catch (err) {
24      console.log(err);
25    }
26  };
```

Εικόνα 5.58

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Για το search έχουμε δημιουργήσει ένα input tag εικόνα 5.59

```
54          <input
55            className="search"
56            type="text"
57            placeholder="Search..."
58            onChange={(e) => setQuery(e.target.value)}
59          />
```

Εικόνα 5.59

Όπου κάθε φορά που πραγματοποιείται αλλαγή , δηλαδή ο χρήστης εισάγει ένα χαρακτήρα στο πεδίο search , αλλάζει το state της μεταβλητής που query που έχουμε δηλώσει , και στην εμφάνιση της λίστας των συσκευών εικόνα 5.60

```
76                {search()?.map((device, index) => (
```

Εικόνα 5.60

καλείται κάθε φορά η συνάρτηση search (εικόνα 5.61) ,

```
28  const search = () => {
29    return Array.isArray(devices)
30      ? devices?.filter(
31          (device) =>
32            device.name.toLowerCase().includes(query) ||
33            device.type.toLowerCase().includes(query) ||
34            device.method.toLowerCase().includes(query) ||
35            device.port.toString().includes(query)
36        )
37      : null;
38  };
```

Εικόνα 5.61

η οποία φιλτράρει τις συσκευές και εμφανίζει μόνο αυτές που περιέχουν τους χαρακτήρες που έχει εισάγει ο χρήστης και περιέχονται μέσα στο όνομα , τύπο , μέθοδο ή port των συσκευών.

Όσον αφορά το κουμπί Add new device , μας κατευθύνει σε καινούργια σελίδα με URL “http://localhost:3000/devices/new” , οπου μέσω του router της App.js εμφανίζει το αρχείο New.js.

Η σελίδα εμφανίζεται μια φόρμα για τη δημιουργία καινούργιας συσκευής (εικόνα 5.62).

Εικόνα 5.62

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

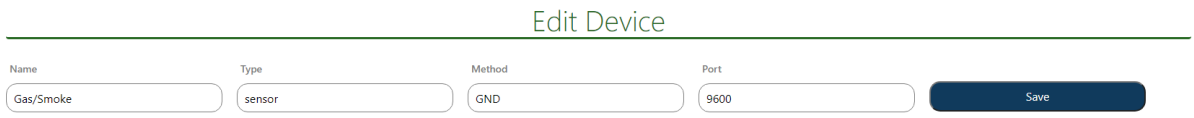
Όταν ο χρήστης εισάγει σωστά τα στοιχεία της φόρμας και πατήσει το Add δηλαδή κάνει submit τη φόρμα καλείται η συνάρτηση postDevice (εικόνα 5.62)

```
22  const postDevice = async() =>{
23    try{
24      await Axios.post(`http://localhost:5000/devices/?id=${userId}`,{
25        name:name,
26        type:type,
27        method:method,
28        port:port
29      })
30     .then((response) => {
31      if(response.data === "Success"){
32        setErrors("");
33        alert("Succesfully added the device");
34        window.location.assign(`http://localhost:3000/devices`);
35      }
36      else{
37        setErrors(response.data);
38      }
39    });
40    }catch(err){
41      console.error(err.message);
42    }
```

Εικόνα 5.63

Έτσι γίνεται Post στο API και στέλνονται οι πληροφορίες που έχει εισάγει στη φόρμα. Εάν έχει κάνει λάθος στα στοιχεία υπάρχει μηχανισμός για χειρισμό σφαλμάτων οπου εμφανίζει στο χρήστη το πρόβλημα που προέκυψε, αλλιώς εμφανίζει ένα alert το οποίο τον ενημερώνει ότι έγινε επιτυχημένη η προσθήκη και το κατευθύνει στη σελίδα των devices.

Το κουμπί edit δουλεύει με παρόμοιο τρόπο όπως το add new device και κατευθύνει το χρήστη στη σελίδα edit, οπου εμφανίζεται η φόρμα με τα δεδομένα που περιέχει το device και ο χρήστης μπορεί να τα επεξεργαστεί και να ανανεώσει τις πληροφορίες του εικόνα 5.64.



Εικόνα 5.64

Για το delete καλείται η συνάρτηση deleteDevice εικόνα 5.65, η οποία παίρνει σαν όρισμα το id της συσκευής από την οποία καλείται.

```
7   const deleteDevice = async (id) => {
8     try {
9       await Axios.delete(
10        `http://localhost:5000/devices/${id}/?id=${userId}`
11      ).then((response) => {
12        alert("Successfully Deleted");
13        window.location.reload();
14      });
15    } catch (err) {
16      console.log(err);
17    }
18  };
```

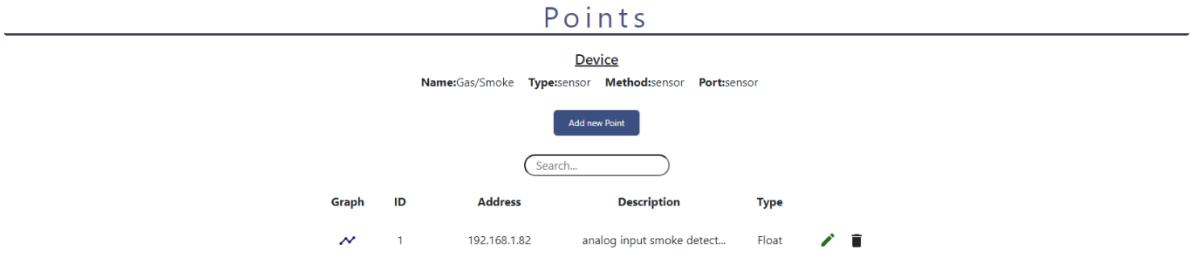
Εικόνα 5.65

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Τέλος όταν πατήσει το σύμβολο κάτω από την ετικέτα Points θα κατευθυνθεί στη σελίδα εικόνα 5.67 με URL που φαίνεται στην εικόνα 5.66 η οποία ως μεταβλητή device.id περιέχει το id του συγκεκριμένου device..

```
<a href={"http://localhost:3000/devices/"+device.id+"/points"}>
```

Εικόνα 5.66



Εικόνα 5.67

Η σελίδα έχει την ίδια μορφή με τη σελίδα για τα devices. Στο πάνω μέρος περιέχονται οι πληροφορίες της συσκευής στην οποία ανήκουν τα points και με ίδιο τρόπο παρέχονται οι ίδιες λειτουργίες για επεξεργασία, διαγραφή και εισαγωγή. Η πρόσβαση στο id του device ώστε να μπορεί να γίνει το HTTP request στο server γίνεται μέσα από το URL της σελίδας. Πιο συγκεκριμένα η δομή της σελίδας μας θα είναι πχ “http://localhost:3000/devices/72/points” από εδώ θα αποθηκεύεται η τιμή 72 σε μια μεταβλητή deviceId μεσω της εντολής εικόνα 5.68

```
14  const deviceId = window.location.href
15    .split("/devices/")[1]
16    .split("/points")[0];
```

Εικόνα 5.68

Στη συνέχεια θα πραγματοποιείται το Get request για την εμφάνιση των συσκευών, εικόνα 5.69, οπου θα ζητάει τα points για το συγκεκριμένο device του χρήστη οπου είναι συνδεδεμένος. Με τον ίδιο τρόπο γίνονται όλες οι άλλες λειτουργίες.

```
18  const getDevice = async () => {
19    try {
20      await Axios.get(
21        `http://localhost:5000/devices/${deviceId}/?id=${userId}
22      ).then((response) => {
23        setDevice(response.data[0]);
24      });
25    } catch (err) {
26      console.log(err);
27    }
28  };
```

Εικόνα 5.69

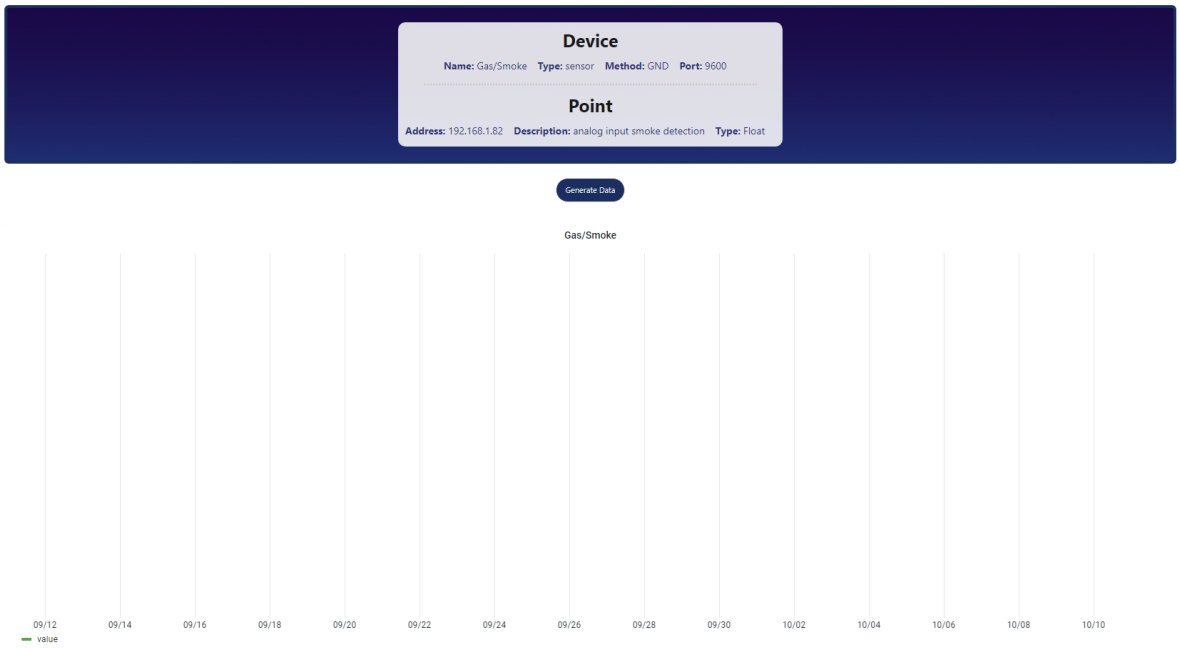
Τέλος πατώντας στο κουμπί Graph (εικόνα γραφήματος) για κάποιο point κατευθυνόμαστε στη τελικά σελίδα μας εικόνα 5.70, τη σελίδα του γραφήματος από το Graph.js για device το id του device και point.id το id του point.

```
a href={`http://localhost:3000/devices/${device}/points/${point.id}`}>
```

Εικόνα 5.70

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Η σελίδα του γραφήματος είναι η εξής (εικόνα 5.71):



Εικόνα 5.71

Οπου στο πάνω μέρος φαίνονται τα δεδομένα του point και του αισθητήρα οπου ανήκει. Το γράφημα αρχικά είναι κενό αφότου δεν υπάρχουν δεδομένα μιας και θα είναι καινούργιος ο αισθητήρας που θα έχει δημιουργηθεί.
Για τη δημιουργία των δεδομένων έχει δημιουργηθεί κουμπί Generate data το οποίο καλεί το API που είχαμε δημιουργήσει για τη δημιουργία τυχαίων δεδομένων από το 0 μέχρι το 100, για χρόνους του τελευταίου μηνά εικόνα 5.72.

```
119  const insertAllData = async () => {
120    try {
121      await Axios.post(
122        `http://localhost:5000/devices/${deviceId}/points/${pointId}/allData/?id=${userId}`,
123        {
124          deviceId: deviceId,
125          pointId: pointId,
126        }
127      ).then((response) => {
128        if (response.data === "Success") {
129          setErrors("");
130          alert("Succesfully added all data");
131          window.location.reload();
132        } else {
133          setErrors(response.data);
134        }
135      });
136    } catch (err) {
137      console.error(err.message);
138    }
139  };
```

Εικόνα 5.72

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Στη συνέχεια αφότου γίνει επιτυχημένα η εισαγωγή δεδομένων , η σελίδα ανανεώνεται αυτόματα και πλέον έχουμε το γράφημα με τα δεδομένα της συσκευής εικόνα 5.73.

Εικόνα 5.73

Η εισαγωγή του γραφήματος στη σελίδα γίνεται μεσω του Grafana και του iframe tag. Έτσι έχουμε εικόνα 5.74

```
<iframe
  className="frameGraph"
  src={`http://localhost:3050/d-solo/xe6FZPZ4k/test?orgId=1&from=${time - 2592000000}&to=${time}&var-deviceId=${device.id}
  &var-pointsId=${pointId}&var-name=${device.name}&var-description=%CE%91%CE%AF%CE%B8%CE%BF%CF%85%CF%83%CE%B1+1&panelId=24`}
  width="100%"
  height="600"
  frameBorder="0"
  title="graph"
></iframe>
```

Εικόνα 5.74

Στο URL ορίζουμε ότι θέλουμε αρχικά να μας εμφανίσει τα δεδομένα του τελευταίου μήνα (time – 259200000) οπού time η ώρα οπου έγινε η πρόσβαση στη σελίδα και ακόμα δηλώνουμε το id του device και το id του point που βρισκόμαστε ώστε να δημιουργηθεί στο Grafana μεσω των μεταβλητών δυναμικά το διάγραμμα για τη συγκεκριμένη συσκευή που βρισκόμαστε. Τέλος περνάμε ως όρισμα και το device.name οπου θα είναι το όνομα που εμφανίζεται στο διάγραμμα μας το οποίο και αυτό το παίρνουμε μεσω του API μας από το όνομα της συσκευής στη βάση δεδομένων.

Τέλος η εκκίνηση της React εφαρμογής γίνεται μεσω της εκτέλεσης της εντολής εικόνα 5.75 στο terminal στη τοποθεσία όπου βρίσκεται το project για το φάκελο client:

```
Chris@DESKTOP-T1082EI MINGW64 ~/Desktop/IotProject/client (master)
$ npm start
```

Εικόνα 5.75

Και έτσι ο server ακούει στο προεπιλεγμένο port της React το 3000.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

ΚΕΦΑΛΑΙΟ 6 Παρουσίαση της πιλοτικής εφαρμογής

Παρακάτω θα γίνει ανάλυση της δημιουργίας και των επιμέρους λειτουργιών της πιλοτικής εφαρμογής που αναπτύχθηκε ως επέκταση του γραφικού περιβάλλοντος της εφαρμογής IOT. Παρουσιάζεται το Front end τμήμα της εφαρμογής και το οποίο βασίστηκε στο Back end τμήμα της πτυχιακής εργασίας "Ανάπτυξη εφαρμογής IOT για την συλλογή και αποθήκευση δεδομένων" του Χελιώτη Κίμων Κωνσταντίνου. Συνδυάζοντας αυτά τα δυο μοντέλα ,έχει γίνει δημιουργία μιας ολοκληρωμένης εφαρμογής ενεργειακής διαχείρισης για το Τμήμα Πληροφορικής με Εφαρμογές στη Βιοϊατρική του Πανεπιστημίου Θεσσαλίας με πολλές δυνατότητες αλλά και μελλοντικές επεκτάσεις.

6.1 Σκοπός της εφαρμογής

Η πιλοτική εφαρμογή δημιουργήθηκε για τη καλύτερη ενεργειακή διαχείριση του Τμήματος Πληροφορικής με Εφαρμογές στη Βιοϊατρική του Πανεπιστημίου Θεσσαλίας ώστε να επιτευχθεί η μείωση της ενεργειακής του κατανάλωσης, σύμφωνα με τη νέα Υπουργική Απόφαση (ΚΥΑ - ΦΕΚ Β3424/2-7-2022) η οποία έχει ως στόχο τη ελαχιστοποίηση της ενεργειακής κατανάλωσης κατά 10% περίπου για την αντιμετώπιση της ενεργειακής κρίσης.

Πιο συγκεκριμένα θα ασχοληθούμε με ένα από τα μετρά του σχεδίου, που αναφέρεται στο καθορισμό εσωτερικής θερμοκρασίας (μέγιστης/ ελάχιστης) τόσο σε θερμαινόμενα/κλιματιζόμενα όσο και σε μη κτίρια βάσει των προδιαγραφών του προτύπου ΕΛΟΤ ΕΝ 15251:2007. Η εσωτερική θερμοκρασία των κτιρίων γραφείων του δημόσιου και του ευρύτερου δημοσίου τομέα, κατά την θερινή περίοδο θα πρέπει να τηρείται στους 27 ° C και κατά την χειμερινή στους 19 ° C [31].

Για την υλοποίηση του παραπάνω στόχου αρχικά θα γίνεται συλλογή των δεδομένων θερμοκρασίας και υγρασίας από τους διάφορους χώρους του Πανεπιστημίου μέσω αισθητήρων. Οι αισθητήρες με τη χρήση διαφόρων πρωτοκόλλων θα αποθηκεύουν τις πληροφορίες αυτές σε μία βάση δεδομένων.

Στη συνέχεια έχει δημιουργηθεί μία διαδικτυακή εφαρμογή για την παρακολούθηση της θερμοκρασίας και υγρασίας σε κάθε χώρο του κτιρίου. Θα δίνει τη δυνατότητα διαχείρισης των αισθητήρων που αναφέραμε και θα προβάλλει τις τιμές που έχουν συλλέξει μέσω γραφημάτων.

Ακόμα θα ενημερώνει μέσω ειδοποιήσεων τους χρήστες, εάν κάποιος από τους χώρους δεν τηρεί τις προκαθορισμένες θερμοκρασίες που ορίζει η νέα Υπουργική Απόφαση.

Η πιλοτική εφαρμογή χρησιμοποιεί τη δομή της εφαρμογής ανάπτυξης γραφικού περιβάλλοντος εφαρμογής IOT του κεφαλαίου 5, με κάποιες τροποποιήσεις και ακόμα τη προσθήκη λειτουργίας ενημερώσεων (alerts).

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

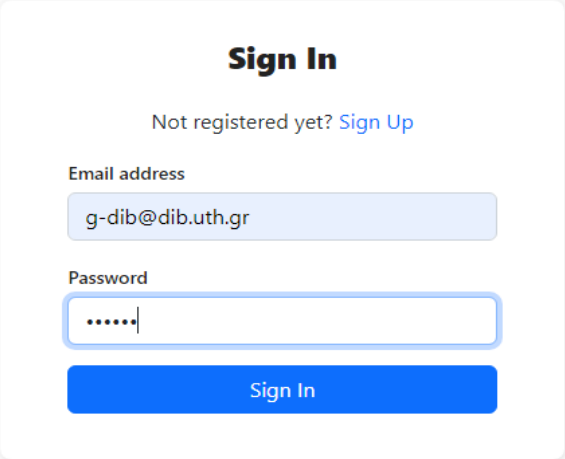
6.2 Παρουσίαση της πιλοτικής εφαρμογής

Για το τμήμα Πληροφορικής με Εφαρμογές στη Βιοϊατρική έχει δημιουργηθεί ένας καινούργιος χρήστης (user) στη βάση δεδομένων.(εικόνα 6.1)

	id [PK] bigint	name character varying (50)	email character varying (50)	password character varying (200)
1	34	Department of Computer Science	g-dib@dib.uth.gr	$2b$10$EkqWbC1XMX...

Εικόνα 6.1

Με το συγκεκριμένο λογαριασμό όταν εισέρχεται κάποιος από το Πανεπιστήμιο ως χρήστης, μέσω του id του θα αναγνωρίζεται ότι έγινε είσοδος από το Πανεπιστήμιο και θα εμφανίζεται η πιλοτική εφαρμογή (εικόνα 6.2) αντί η εφαρμογή που αναλύσαμε στο προηγούμενο κεφάλαιο.



Εικόνα 6.2 Sign-in σελίδα

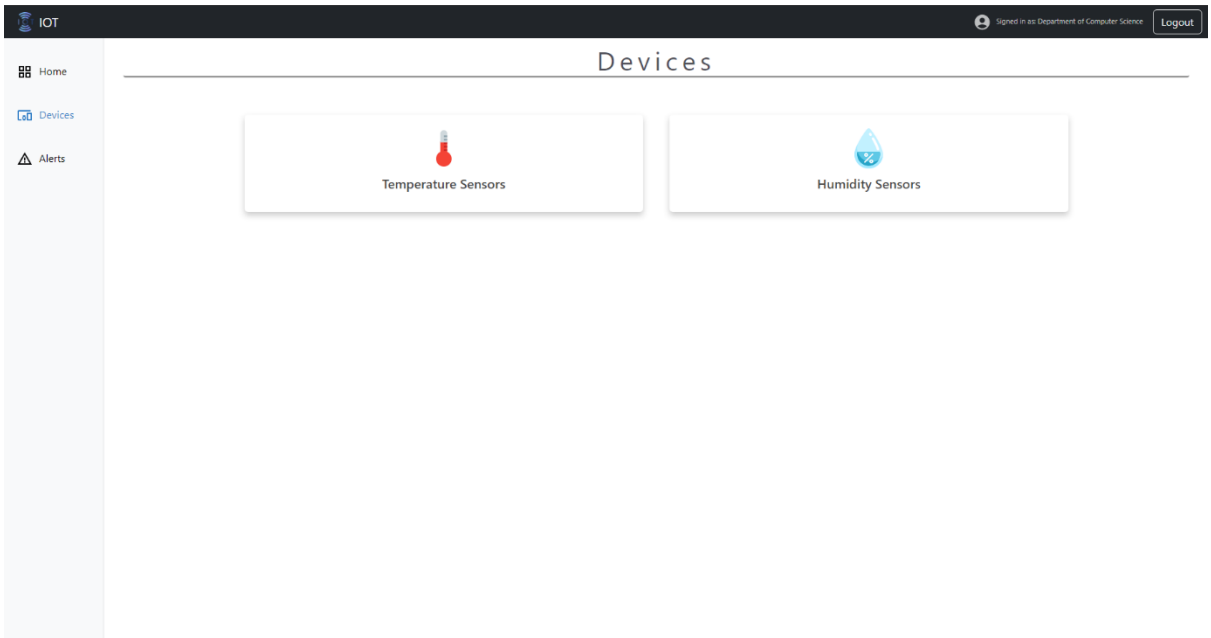
Κατά επέκταση με τον ίδιο τρόπο θα μπορεί να δημιουργηθεί αντίστοιχη εφαρμογή και για άλλες δημοσιές υπηρεσίες οι οποίες θα εισέρχονται σε αυτή με το δικό τους λογαριασμό.Με την επιτυχή είσοδο στην εφαρμογή, παρουσιάζεται η αρχική οθόνη η οποία αποτελείται από το τίτλο “Σύστημα

Εικόνα 6.3

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

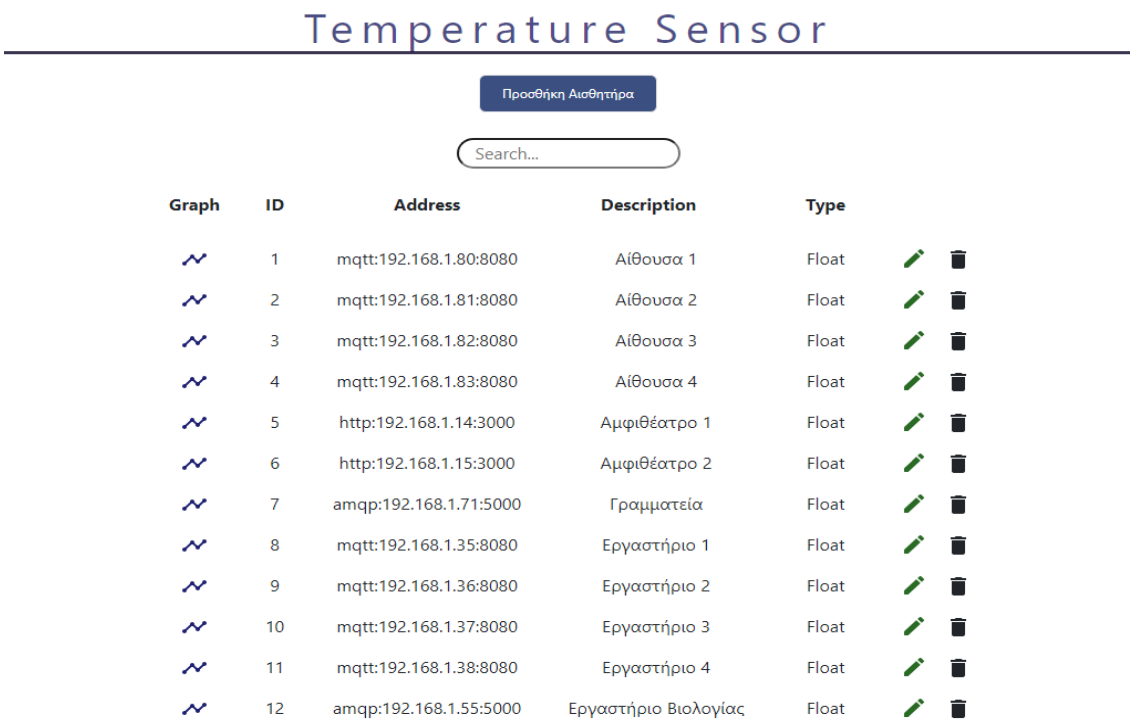
ενεργειακής διαχείρισης” , από κάτω το όνομα του πανεπιστήμιου και μια φωτογραφία του (εικόνα 6.3).
Από εδώ οι επιλογές που μπορεί να κάνει είναι να αποσυνδεθεί πατώντας Logout στο πάνω δεξιά μέρος της οθόνης ή να πατήσει τις επιλογές του sidebar Devices ή Alerts.

Πατώντας Devices κατευθύνεται στη σελίδα των συσκευών (εικόνα 6.4). Για τη πιλοτική εφαρμογή έχουμε μόνο 2 είδη συσκευών τους αισθητήρες θερμοκρασίας και υγρασίας που θα μπορούν να εγκατασταθούν στις διάφορες αίθουσες.



Εικόνα 6.4

Από το μενού των συσκευών εάν ο χρήστης πατήσει temperature sensor κατευθύνεται σε σελίδα με τις αίθουσες του πανεπιστήμιου για αισθητήρες θερμοκρασίας. (εικόνα 6.5)



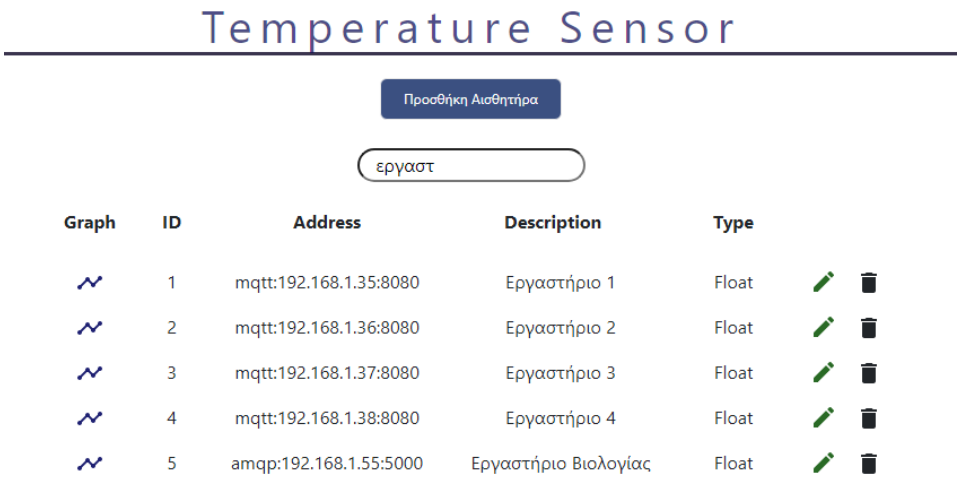
Graph	ID	Address	Description	Type
	1	mqtt:192.168.1.80:8080	Αίθουσα 1	Float
	2	mqtt:192.168.1.81:8080	Αίθουσα 2	Float
	3	mqtt:192.168.1.82:8080	Αίθουσα 3	Float
	4	mqtt:192.168.1.83:8080	Αίθουσα 4	Float
	5	http:192.168.1.14:3000	Αμφιθέατρο 1	Float
	6	http:192.168.1.15:3000	Αμφιθέατρο 2	Float
	7	amqp:192.168.1.71:5000	Γραμματεία	Float
	8	mqtt:192.168.1.35:8080	Εργαστήριο 1	Float
	9	mqtt:192.168.1.36:8080	Εργαστήριο 2	Float
	10	mqtt:192.168.1.37:8080	Εργαστήριο 3	Float
	11	mqtt:192.168.1.38:8080	Εργαστήριο 4	Float
	12	amqp:192.168.1.55:5000	Εργαστήριο Βιολογίας	Float

Εικόνα 6.5

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Εμφανίζονται οι διάφορες αίθουσες του πανεπιστήμιου , όπως η γραμματεία, τα εργαστήρια. Έχει γίνει ήδη εισαγωγή των αιθουσών του Πανεπιστημίου. Στο address εμφανίζεται το πρωτόκολλο , η διεύθυνση και το port που χρησιμοποιούν οι αισθητήρες και στο description παρουσιάζεται η αίθουσα στην οποία βρίσκονται. Από εδώ μπορεί να γίνει διαγραφή της αίθουσας (π.χ. εάν κάποια αίθουσα του πανεπιστήμιου έχει κλείσει) , να γίνει επεξεργασία των στοιχείων (π.χ. εάν άλλαξε το πρωτόκολλο σε κάποια αίθουσα , το address του η το όνομα της αίθουσας) και τέλος να γίνει εισαγωγή καινούργιο αισθητήρα σε αίθουσα (εάν ανοίξει καινούργια αίθουσα στο Πανεπιστήμιο).

Ακόμα δίνεται η δυνατότητα αναζήτησης του αισθητήρα μέσω του πρωτοκόλλου του, της διεύθυνσης του port ή της αίθουσας που βρίσκεται π.χ. για να εμφανιστούν μόνο τα εργαστήρια του κτιρίου γράφουμε: (εικόνα 6.6)



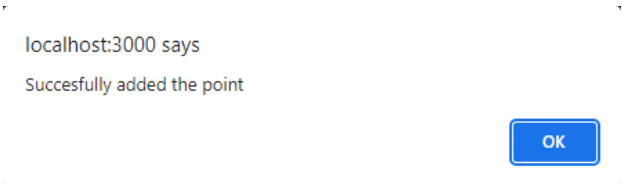
Εικόνα 6.6

Από το κουμπί Προσθήκη Αισθητήρα μπορεί να γίνει ανακατεύθυνση στη σελίδα εικόνα οπου γίνεται εισαγωγή των στοιχείων του αισθητήρα. (εικόνα 6.7)

Εικόνα 6.7

Εάν υπάρχει κάποιο πρόβλημα στην εισαγωγή θα εμφανίσει κατάλληλο μήνυμα (εικόνα 6.8) ενώ εάν γίνει επιτυχημένα η εισαγωγή ενημερώνει το χρήστη μεσω alert στη σελίδα και το κατευθύνει πάλι στην οθόνη αισθητήρων. (εικόνα 6.9)

Εικόνα 6.8

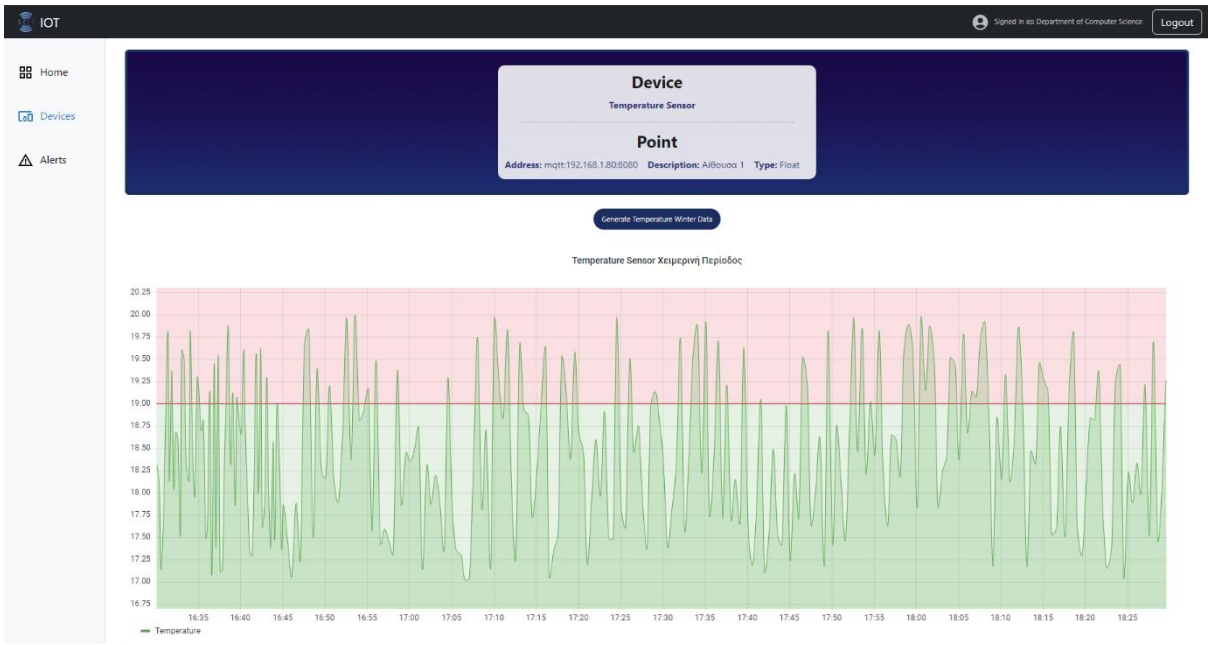


Εικόνα 6.9

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Εάν ο χρήστης πατήσει στο σύμβολο κάτω από την ετικέτα Graph για κάποιο αισθητήρα θα κατευθυνθεί στη σελίδα για τη συγκεκριμένη αίθουσα (εικόνα 6.10). Στο πάνω μέρος εμφανίζονται οι πληροφορίες του αισθητήρα, αν δηλαδή βρισκόμαστε σε αισθητήρα θερμοκρασίας η υγρασίας και από κάτω οι πληροφορίες για το πρωτόκολλο, τη διεύθυνση και το port που χρησιμοποιεί καθώς και την αίθουσα στην οποία βρίσκεται.

Από κάτω βλέπουμε το γράφημα με τις τιμές τις οποίες περιέχει ο αισθητήρας.
Στη βάση δεδομένων αποθηκεύονται πληροφορίες από τους αισθητήρες και έτσι διαμορφώνονται οι τιμές στο γράφημα.



Εικόνα 6.10

Μέχρι αυτό το σημείο η πιλοτική εφαρμογή λειτουργεί με τον ίδιο περίπου τρόπο όπως η εφαρμογή της ενότητας 5 με κάποιες διαφοροποιήσεις. Τώρα θα δείξουμε τις λειτουργίες που έχουμε προσθέσει για τη πιλοτική εφαρμογή για το σύστημα διαχείρισης ηλεκτρικής ενέργειας.

Αρχικά σύμφωνα με τα καινούργια μέτρα η εσωτερική θερμοκρασία πρέπει να διατηρείται, κατά την θερινή περίοδο στους 27 ° C και κατά την χειμερινή στους 19 ° C. Οπότε το πρόγραμμα πρέπει να έχει τη δυνατότητα να προσαρμόζεται ανάλογα με την εποχή. Αυτό επιτυγχάνεται με τη συνάρτησης Date() οπου βρίσκει τη τρέχουσα ημερομηνία και στη συνέχεια μεσω της getMonth εξάγετε ο μήνας. Εάν ο μήνας είναι αναμεσά στον Απρίλιο και στη τελευταία μέρα του Σεπτέμβριου βρισκόμαστε στη θερινή περίοδο αλλιώς βρισκόμαστε στη χειμερινή. (εικόνα 6.11)
Στη συνέχεια στην εφαρμογή θα γίνεται έλεγχος της περιόδου και θα εμφανίζει καταλληλά το διάγραμμα , το threshold και τα alerts για χειμερινή και θερινή περίοδο ξεχωριστά.

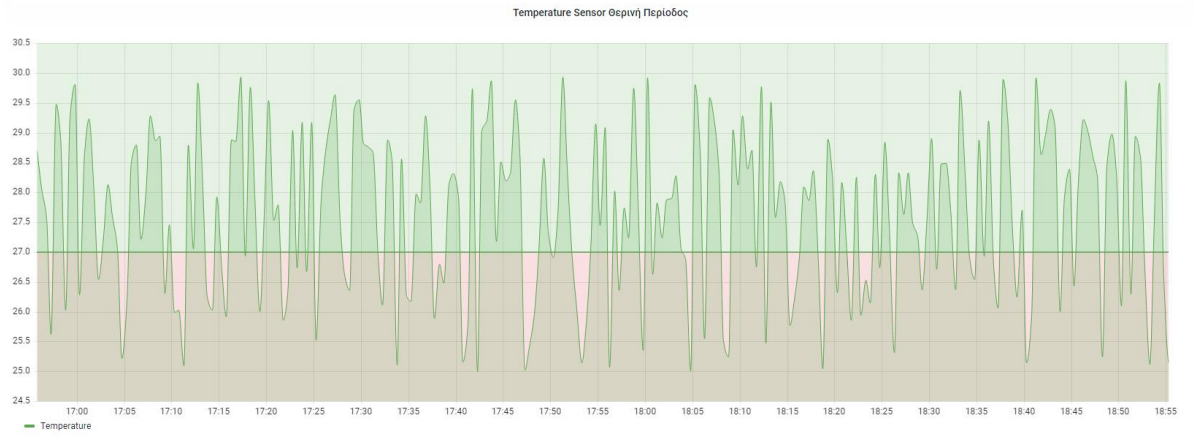
```
const month = new Date();
const current_month = month.getMonth()+1;

const findSeason = () => {
if (current_month >=4 && current_month<=9 ){
  setThreshold("summer")
}
else if ((current_month >=10 && current_month<=12) || (current_month >=1 && current_month<=3) ){
  setThreshold("winter");
}
};
```

Εικόνα 6.11

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Ακόμα στο γράφημα έχει εισαχθεί ένα όριο (threshold) (εικόνα 6.12) για θερμοκρασία 27 ° C τη θερινή περίοδο οπου κάτω από αυτό βρίσκονται οι μη επιτρεπτές τιμές θερμοκρασίας (δηλαδή κάτω των 27 για θερινή) και πάνω από αυτό οι επιθυμητές. Αντίστοιχα για χειμερινή περίοδο το όριο αλλάζει στους 19 βαθμούς και οι επιτρεπτές είναι αυτές κάτω των 19 βαθμών.



Εικόνα 6.12

Στα πλαίσιο του Demo της εφαρμογής έχει δημιουργηθεί και ένα κουμπί generate temperature data με το οποίο εισάγονται τυχαίες τιμές στη βάση δεδομένων ώστε να προσομοιωθεί η εισαγωγή τιμών από τους αληθινούς αισθητήρες. Για χειμερινή περίοδο εισάγονται τιμές κοντά στους 19 βαθμούς και για θερινή κοντά στους 27 ώστε να γίνει η λειτουργία της εφαρμογής πιο κατανοητή.

Ακόμα για να προσομοιάσουμε τη λειτουργία των αισθητήρων και να μπορέσουν να λειτουργήσουν τα alerts θα γίνει εισαγωγή μελλοντικών δεδομένων αφότου οι αισθητήρες θα δίνουν live συνεχόμενες τις τιμές θερμοκρασίας στη βάση δεδομένων.

Ο κώδικας για τη κλήση του API φαίνεται στην εικόνα 6.13.

```
75  const insertWinterData = async () => {
76    try {
77      await Axios.post(
78        `http://localhost:5000/devices/${deviceId}/points/${pointId}/winterData/?id=${userId}`,
79        {
80          deviceId: deviceId,
81          pointId: pointId,
82        }
83      ).then((response) => {
84        if (response.data === "Success") {
85          setErrors("");
86          alert("Succesfully added current data");
87          window.location.reload();
88        } else {
89          setErrors(response.data);
90        }
91      });
92    } catch (err) {
93      console.error(err.message);
94    }
95  };
```

Εικόνα 6.13

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Στην εικόνα 6.14 φαίνεται το query οπου χρησιμοποιούμε στο index.js αρχείο του σερβερ για να περνάει τυχαίες τιμές ανάμεσα στο 17 και 20 για χρονική περίοδο 2 ώρες πριν τη τωρινή ώρα όπου γίνεται το πάτημα του κουμπιού μέχρι 3 ώρες μετά με περίοδο 30 δευτερολέπτων. Για τη θερινή περίοδο θα προστίθενται τιμές ανάμεσα στο 25 έως 30.

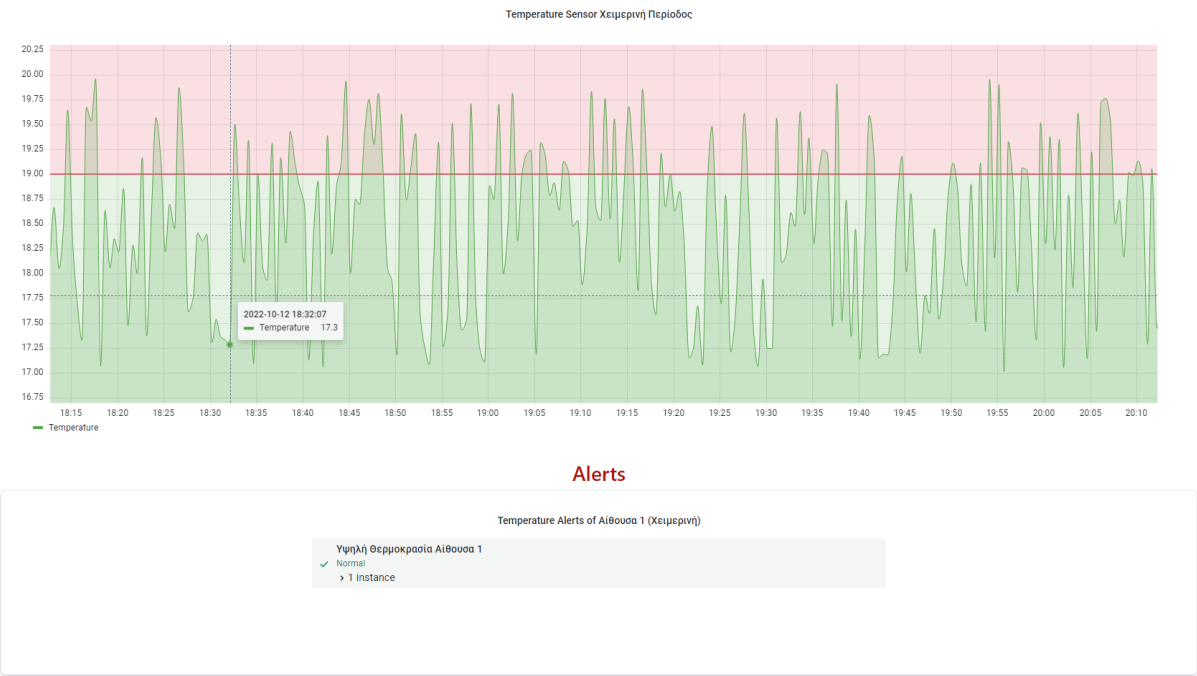
```
pool.query(
  "INSERT INTO iot_values ( date, value,point_id ) SELECT date, random()*(20-17)+17 AS value, $1 AS point_id FROM generate_series(now()
  - interval '5 hour', now() , interval '30 seconds') AS g1(date)",
  [pointId],
```

Εικόνα 6.14

Υπενθυμίζουμε ότι το κουμπί αυτό δεν θα υπάρχει στη πιλοτική εφαρμογή και έχει δημιουργηθεί μόνο για λόγους testing και παρουσίασης της εφαρμογής. Ακόμα έχει γίνει σε συνεργασία με τη πτυχιακή του συνάδελφου.

Για να εισάγουμε το γράφημα στην εφαρμογή όπως και στο κεφάλαιο 5 εισάγουμε το iframe από το Grafana όπου δηλώνουμε να μας εμφανίζει το διάγραμμα για 2 ώρες πριν, μέχρι τη τωρινή ώρα , ώστε να γίνεται live ενημέρωση του διαγράμματος με τις μελλοντικές τιμές και να προσομοιώνεται η κανονική λειτουργία ενός αισθητήρα.

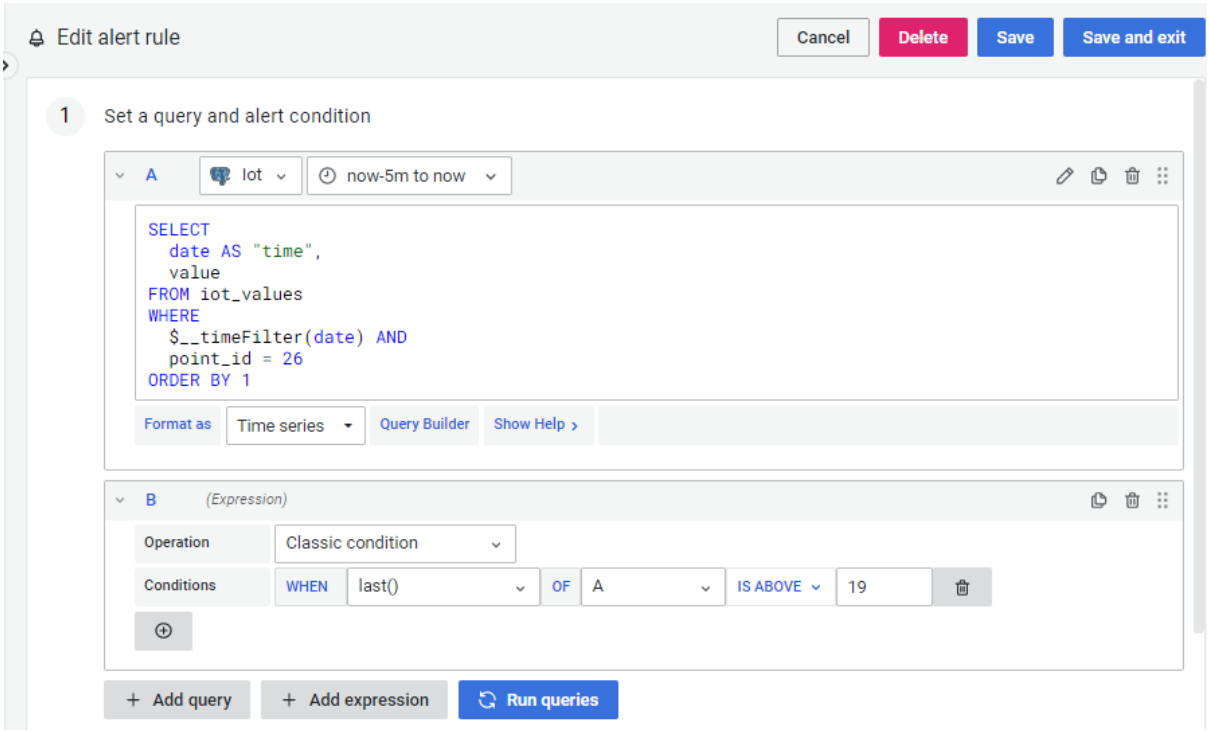
Ακόμα στο τέλος της σελίδας του αισθητήρα κάτω από το γράφημα έχει γίνει εισαγωγή panel για ενημέρωση (alert) του χρήστη εάν η θερμοκρασία στη συγκεκριμένη αίθουσα βρίσκεται εκτός επιθυμητών ορίων (εικόνα 6.15).



Εικόνα 6.15

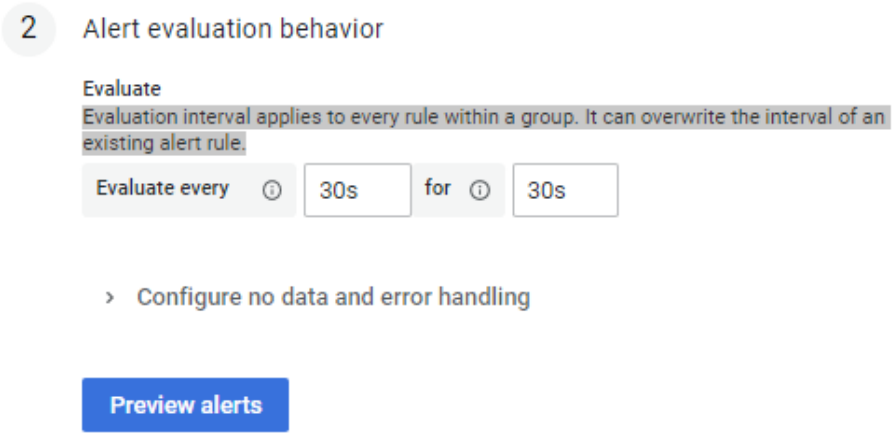
Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Τα alert δημιουργήθηκαν μέσω Grafana όπου πηγαίνοντας στη σελίδα alerting δίνεται η δυνατότητα καινούργιου alert rule. Από εκεί αρχικά δηλώνουμε τη βάση από την οποία θα παίρνει τα δεδομένα και τη συνθήκη στην οποία θα γίνεται το alert . Παραδείγματος χάρη για την αίθουσα 1 δημιουργούμε το SQL query για να δείξουμε ότι θέλουμε να επιλέξουμε τις τιμές για την αίθουσα 1 και πάνω σε αυτές τρέχει η συνθήκη η οποία αναφέρει ότι εάν η τελευταία τιμή είναι μεγαλύτερη των 19 βαθμών για χειμερινή περίοδο τότε κάνε το alert. (εικόνα 6.16)



Εικόνα 6.16

Ακόμα μπορούμε να δηλώσουμε το διάστημα αξιολόγησης του κανόνα , οπου για χάρη του simulation έχουμε δηλώσει να γίνεται κάθε 30 δευτερόλεπτα (εικόνα 6.17).

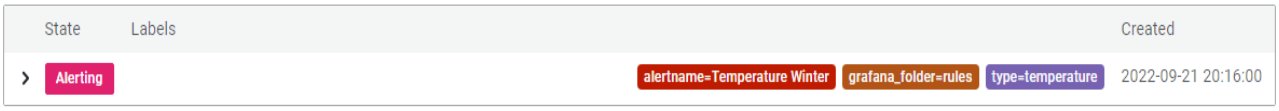


Εικόνα 6.17

Τα alert που δημιουργούνται έχουν τρεις καταστάσεις normal , pending και firing , οπου εάν η θερμοκρασία βρίσκεται σε φυσιολογικά όρια δηλαδή κάτω των 19 για χειμερινή περίοδο βρίσκεται

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

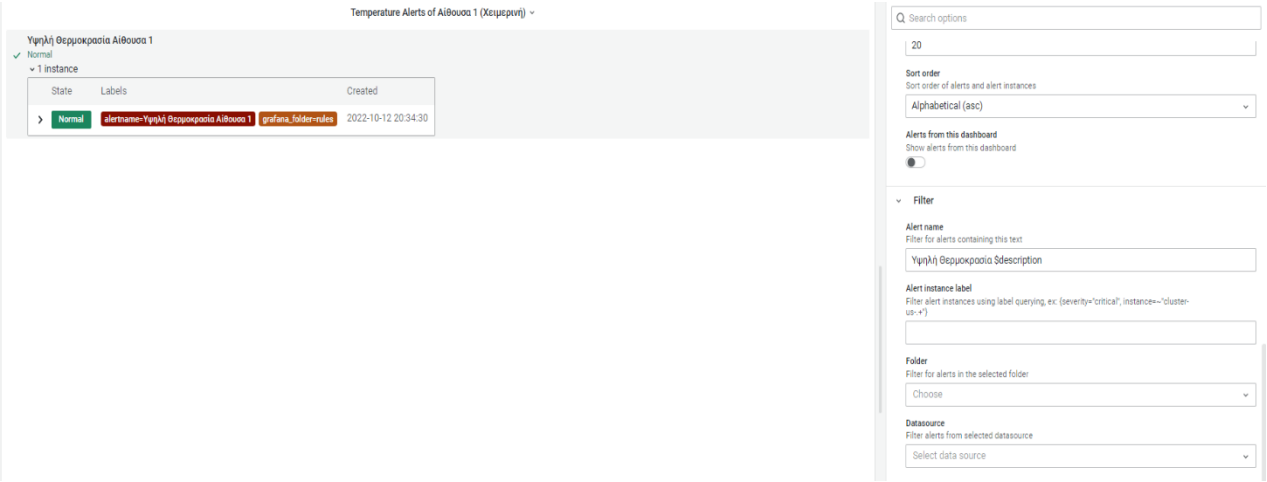
στο normal , εάν τη πρώτη φορά ανεβεί άνω των 19 μπαίνει σε κατάσταση pending και εάν συνεχίσει μετά από κάποιο διάστημα (στην εφαρμογή έχει δηλωθεί διάστημα 30 δευτερολέπτων) να βρίσκεται εκτός της επιτρεπόμενης τιμής τότε μπαίνει σε κατάσταση firing οπου γίνεται και το alert (εικόνα 6.18).



Εικόνα 6.18

Αφότου δημιουργούνται οι κανόνες alert, δημιουργούμε ένα καινούργιο panel στο dashboard του Grafana στο οποίο θα μας δείχνει το συγκεκριμένο alert που θέλουμε και μεσω iframe θα το εισάγουμε στη σελίδα.

Ο τρόπος με τον οποίο ξεχωρίζουμε πιο alert θέλουμε για κάθε αίθουσα ξεχωριστά είναι μεσω των μεταβλητών του Grafana. Φιλτράρουμε τα alert μέσω του ονόματος της αίθουσας δηλαδή τη τιμή του description του πίνακα iot_points της βάσης δεδομένων (εικόνα 6.19).



Εικόνα 6.19

Ένα πρόβλημα της εφαρμογής είναι το ότι και κατά τη χειμερινή και κατά τη θερινή περίοδο, υπάρχουν μέρες όπου η θερμοκρασία του περιβάλλοντος είναι άνω των 19 βαθμών για χειμερινή και λιγότερο των 27 για θερινή περίοδο. Αυτό θα έχει ως αποτέλεσμα να μας χτυπάει alerts η εφαρμογή όλες αυτές τις μέρες όπου δεν θα έπρεπε.

Ένας τρόπος λύσης αυτού είναι να κάνουμε fetch δεδομένα από ένα API καιρού και για τα alerts η συνθήκη να γίνει η εξής.
Εάν (πχ για χειμερινή περίοδο) η θερμοκρασία από τον αισθητήρα είναι μεγαλύτερη των 19 βαθμών και η θερμοκρασία του περιβάλλοντος είναι μικρότερη των 19 βαθμών τότε χτυπά alert.

Έτσι εάν έχουμε περισσότερο των 19 βαθμών στην αίθουσα αλλά και η εξωτερική θερμοκρασία είναι άνω των 19 τότε δεν θα εμφανίζεται alert μιας και η θερμοκρασία εξαρτάται από την εξωτερική θερμοκρασία και όχι από θέρμανση.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Το API που θα χρησιμοποιήσουμε είναι από το openweathermap και μέσω αυτού θα παίρνουμε τις θερμοκρασίες από τη Λαμία τη συγκεκριμένη στιγμή που θα πραγματοποιείται το API call. Ως απάντηση θα μας επιστρέφει ένα json αρχείο το οποίο φαίνεται στην εικόνα 6.20.


```
{
  "coord": {
      "lon": 22.4333,
      "lat": 38.9
  },
  "weather": [
    {
        "id": 802,
        "main": "Clouds",
        "description": "scattered clouds",
        "icon": "03d"
    }
  ],
  "base": "stations",
  "main": {
      "temp": 20.6,
      "feels_like": 19.81,
      "temp_min": 20.6,
      "temp_max": 21.46,
      "pressure": 1016,
      "humidity": 42,
      "sea_level": 1016,
      "grnd_level": 1004
  },
  "visibility": 10000,
  "wind": {
      "speed": 2.04,
      "deg": 111,
      "gust": 1.85
  },
  "clouds": {
      "all": 40
  },
  "dt": 1663857011,
  "sys": {
      "type": 2,
      "id": 2006713,
      "country": "GR",
      "sunrise": 1663820260,
      "sunset": 1663864113
  },
  "timezone": 10800,
  "id": 258620,
  "name": "Lamia",
  "cod": 200
}
```


Εικόνα 6.20

Από το json αρχείο μας ενδιαφέρει η μεταβλητή temp, η οποία δείχνει τη θερμοκρασία της Λαμίας.

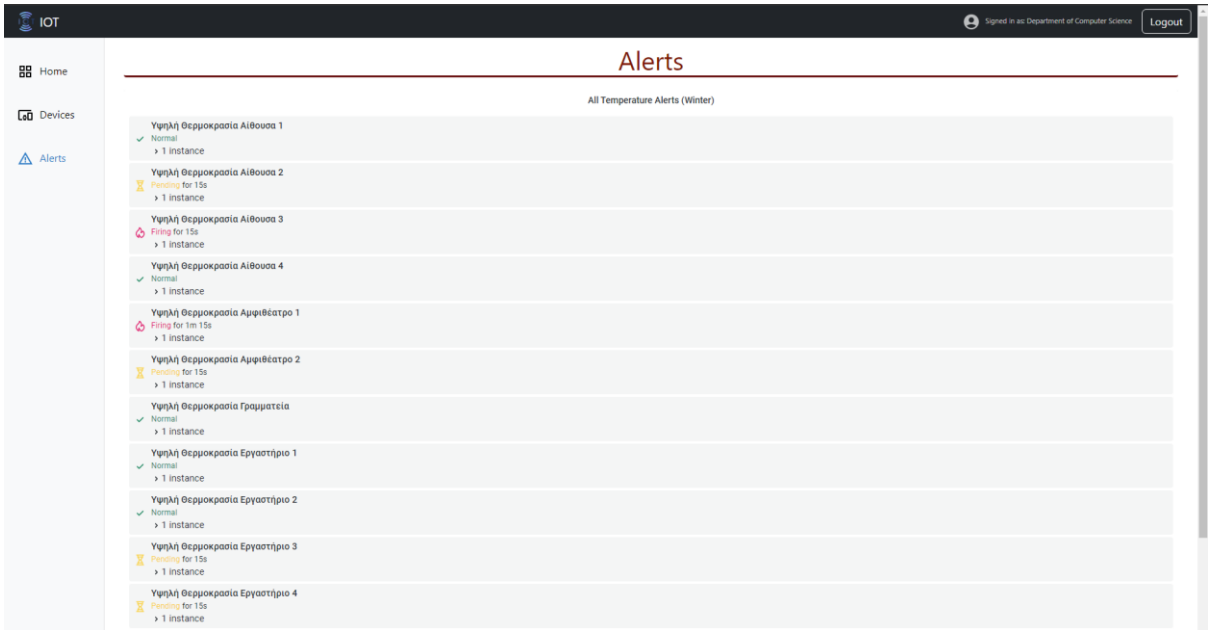
Το API το οποίο καλούμαι είναι το :
«https://api.openweathermap.org/data/2.5/weather?id=258620&units=metric&appid=id
Όπου ως id βάζουμε τη τιμή του api key

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Έτσι εάν η θερμοκρασία στη Λαμία είναι άνω των 19 (χειμερινή) και κάτω των 27 (θερινή) δεν θα εμφανίζει τα alerts και αντί αυτού θα εμφανίζει μήνυμα με το οποίο θα γίνεται ενημέρωση ότι δεν υπάρχουν alerts λόγω εξωτερικής θερμοκρασίας εικόνα 6.21.

Εικόνα 6.21

Τέλος έχει δημιουργηθεί μια κεντρική σελίδα για όλα τα alerts τα οποία περιέχονται σε όλες οι αίθουσες του πανεπιστήμιου. Η πρόσβαση σε αυτή τη σελίδα γίνεται μεσω του sidebar μεσω του μενού alerts (εικόνα 6.22).



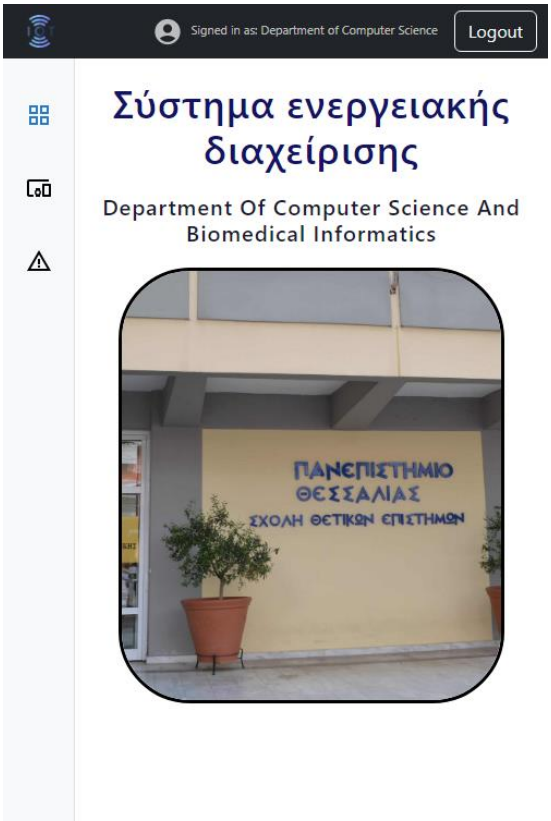
Εικόνα 6.22

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Ακόμα για τους αισθητήρες υγρασίας το διάγραμμα λειτουργεί με τον ίδιο τρόπο όπως οι αισθητήρες θερμοκρασίας, με τη διαφορά ότι οι τυχαίες τιμές που εισάγονται, βρίσκονται ανάμεσα στο 0 και το 100 και δεν υπάρχουν τα alerts εικόνα (6.23).

Εικόνα 6.23

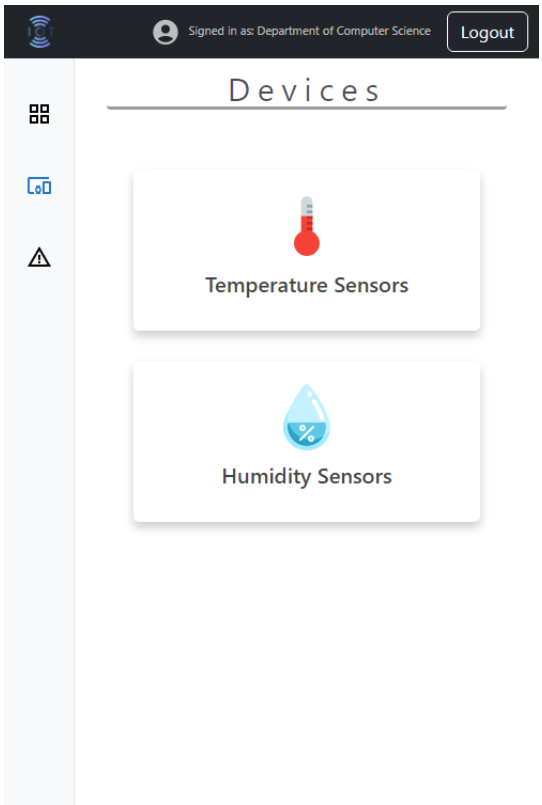
Τέλος, η εφαρμογή έχει δημιουργηθεί με γνώμονα το responsive design, ώστε να λειτουργεί για όλα τα μεγέθη συσκευών , καθιστώντας πιο εύκολη τη χρήση σε κινητές συσκευές. Μερικά παραδείγματα φαίνονται στις παρακάτω εικόνες.
Για την αρχική οθόνη έχουμε εικόνα 6.



Εικόνα 6.24

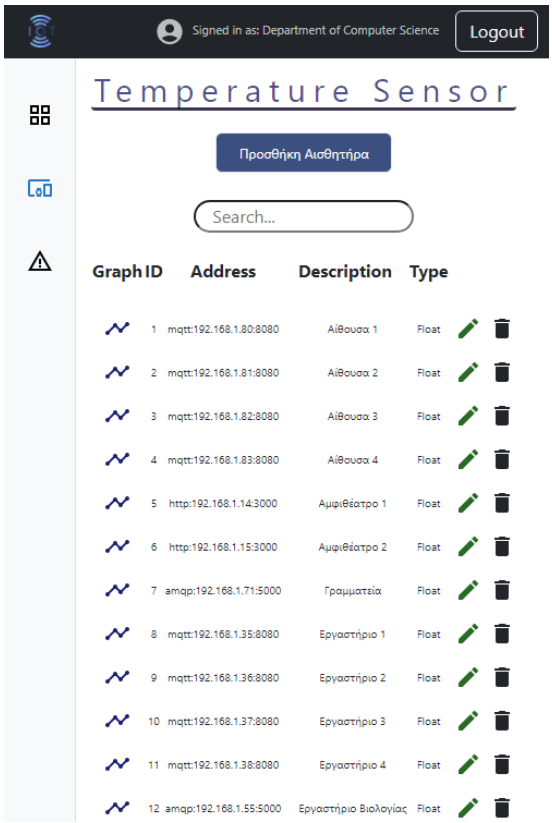
Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Για τη σελίδα των συσκευών έχουμε εικόνα 6.25.



Εικόνα 6.25

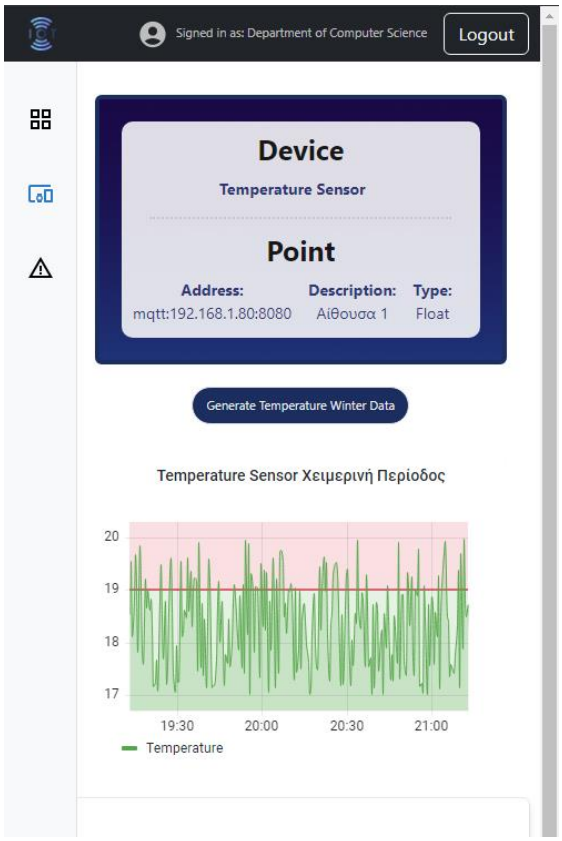
Για τη σελίδα των αισθητήρων έχουμε εικόνα 6.26.



Εικόνα 6.26

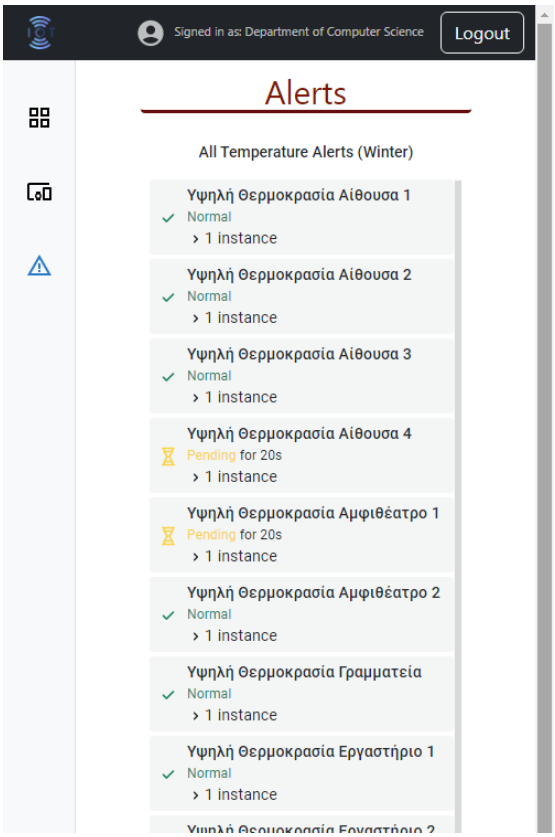
Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Για τη σελίδα του γραφήματος έχουμε εικόνα 6.27



Εικόνα 6.27

και για τη σελίδα των alerts εικόνα 6.28



Εικόνα 6.28

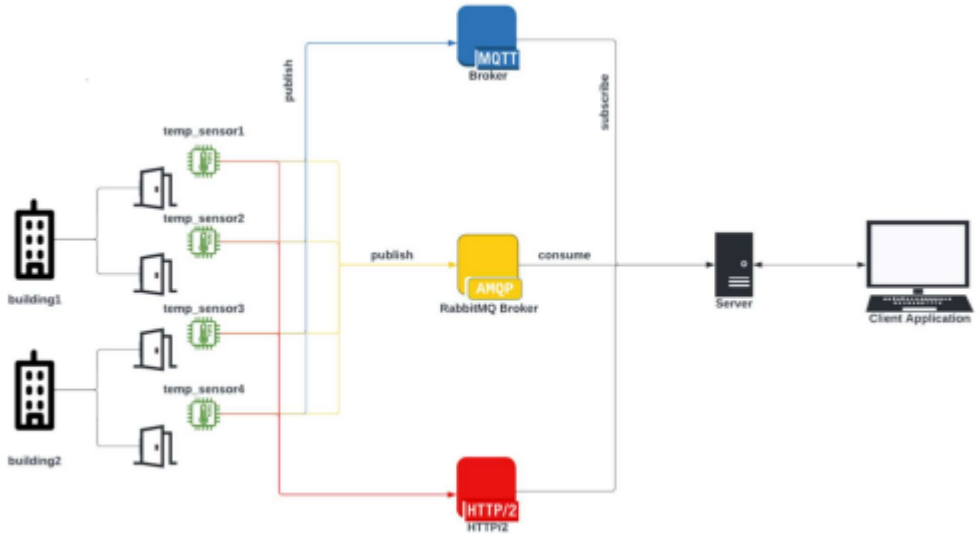
Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

6.3 Back-end πιλοτικής εφαρμογής

Στα πλαίσια της πτυχιακής εργασίας με όνομα “Ανάπτυξη εφαρμογής IOT για την συλλογή και αποθήκευση δεδομένων” του Χελιώτη Κίμων Κωνσταντίνου αναπτύχθηκε μια βάση δεδομένων για αποθήκευση δεδομένων από αισθητήρες θερμοκρασίας και υγρασίας οι οποίες θα βρίσκονται στις αίθουσες του Πανεπιστημίου Θεσσαλίας.

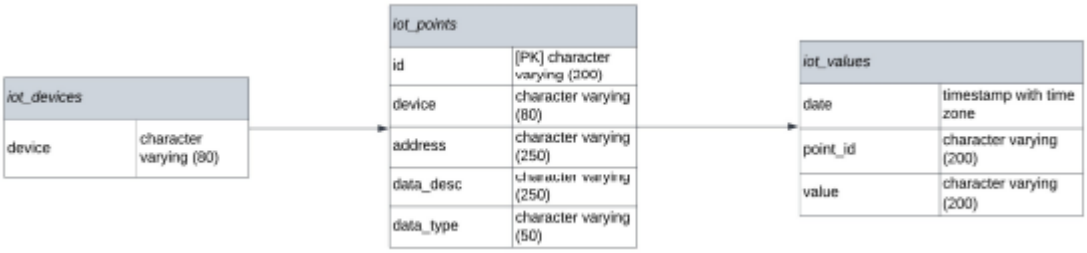
Χρησιμοποιώντας αυτή τη βάση δεδομένων αναπτύχθηκε το front-end κομμάτι της πιλοτικής εφαρμογής που αναλύσαμε παραπάνω. Με αυτό το τρόπο έχει υλοποιηθεί μια ολοκληρωμένη εφαρμογή για το σύστημα ενεργειακής διαχείρισης του τμήματος Πληροφορικής με Εφαρμογές στη Βιοϊατρική του Πανεπιστημίου Θεσσαλίας.

Η λειτουργία του συστήματος, γίνεται με την επικοινωνία του server με τον κάθε αισθητήρα που έχει εγκατασταθεί σε συγκεκριμένους χώρους του Πανεπιστημίου. Ξεκινάει η επικοινωνία μεταξύ αισθητήρα και server, και ανάλογα με την επιλογή του πρωτοκόλλου, στέλνονται τα δεδομένα και δρομολογούνται από τον server στη βάση δεδομένων. Παρακάτω φαίνεται η λειτουργία του συστήματος το οποίο καταλήγει στο Client Application που αποτελεί το αντικείμενο αυτής της πτυχιακής (εικόνα 6.29).



Εικόνα 6.29

Παρακάτω δίνεται η δομή της βάσης δεδομένων (εικόνα 6.30).
Η συλλογή και αποθήκευση των δεδομένων γίνεται με τη χρήση των τριών πρωτοκόλλων επικοινωνίας: MQTT, HTTP και AMQP. Περιέχονται τρία κομμάτια back-end τα οποία χωρίζονται ανάλογα με τα πρωτόκολλα.



Εικόνα 6.30

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

Για τα δεδομένα του κτιρίου ,έχει δημιουργηθεί ο ανάλογος πίνακας, ο οποίος δείχνει τους αισθητήρες θερμοκρασίας που υπάρχουν σε κάθε χώρο του κτιρίου. Στον πίνακα iot_points της εφαρμογής, έχουν δημιουργηθεί δεδομένα, τα οποία δείχνουν τον κάθε αισθητήρα, τον χώρο στον οποίο βρίσκεται καθώς και το id για να είναι δυνατή η αναγνώριση του αισθητήρα, από τον οποίο γίνεται η λήψη των δεδομένων, όπως φαίνεται στην εικόνα 6.31

	id [PK] character varying (200)	device character varying (80)	address character varying (250)	data_desc character varying (250)	data_type character varying (50)
1	1	temp_sensor	mqtt:192.168.1.31:8080	temperature_hall1	float
2	10	temp_sensor	http:192.168.1.32:8080	temperature_control_r...	float
3	11	temp_sensor	mqtt:192.168.1.33:8080	temperature_auditoriu...	float
4	12	temp_sensor	mqtt:192.168.1.34:8080	temperature_auditoriu...	float
5	13	temp_sensor	mqtt:192.168.1.35:8080	temperature_computer...	float
6	14	temp_sensor	mqtt:192.168.1.36:8080	temperature_computer...	float
7	15	temp_sensor	mqtt:192.168.1.37:8080	temperature_computer...	float
8	16	temp_sensor	mqtt:192.168.1.38:8080	temperature_computer...	float
9	17	temp_sensor	amqp:192.168.1.55:50...	temperature_biology_lab	float
10	18	humidity_sensor	http:192.168.1.11:3000	humidity_building1	float
11	19	humidity_sensor	http:192.168.1.13:3000	humidity_building2	float
12	2	temp_sensor	amqp:192.168.1.70:50...	temperature_office1	float
13	20	humidity_sensor	http:192.168.1.14:3000	humidity_auditorium1	float
14	21	humidity_sensor	http:192.168.1.15:3000	humidity_auditorium2	float
15	3	temp_sensor	amqp:192.168.1.72:50...	temperature_office2	float
16	4	temp_sensor	amqp:192.168.1.71:50...	temperature_secretary...	float
17	6	temp_sensor	mqtt:192.168.1.80:8080	temperature_classroo...	float
18	7	temp_sensor	mqtt:192.168.1.81:8080	temperature_classroo...	float
19	8	temp_sensor	mqtt:192.168.1.82:8080	temperature_classroo...	float
20	9	temp_sensor	mqtt:192.168.1.83:8080	temperature_classroo...	float

Εικόνα 6.31

Στη συνέχεια, αφού γίνει η λήψη των δεδομένων από τους αισθητήρες, αυτά αποθηκεύονται στη βάση δεδομένων, στον πίνακα iot_values, όπως φαίνεται στην εικόνα 6.32.

	date timestamp with time zone	point_id character varying (200)	value character varying (200)
1	2022-09-25 02:38:38.389...	14	23.87
2	2022-09-25 02:38:38.607...	6	15.91
3	2022-09-25 02:38:38.872...	21	19.46
4	2022-09-25 02:38:39.171...	9	26.84
5	2022-09-25 02:38:39.448...	6	27.22
6	2022-09-25 02:38:39.676...	12	21.80
7	2022-09-25 02:38:39.971...	9	21.74
8	2022-09-25 02:38:40.212...	14	18.68
9	2022-09-25 02:38:40.467...	6	23.62
10	2022-09-25 02:38:40.734...	5	24.25

Εικόνα 6.32

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

ΚΕΦΑΛΑΙΟ 7 Συμπεράσματα

Με την ολοκλήρωση της πτυχιακής εργασίας και τη δημιουργία της εφαρμογής IOT καθώς και της πιλοτικής εφαρμογής για την ενεργειακή διαχείριση, επιτεύχθηκε η καλύτερη κατανόηση των τεχνολογιών IOT καθώς και των πολλών πεδίων εφαρμογής τους, και της σπουδαιότητας της ενεργειακής διαχείρισης, ιδιαίτερα στον κτιριακό τομέα, για τη μείωση της ενεργειακής κατανάλωσης. Μέσω της ανάπτυξης της διαδικτυακής εφαρμογής, πραγματοποιήθηκε εκμάθηση πολλών εργαλείων και τεχνολογιών που είναι απαραίτητα για τη κατασκευή και ανάπτυξη ιστοσελίδων.

7.1 Μελλοντικές επεκτάσεις

Πληθώρα επεκτάσεων μπορούν να πραγματοποιηθούν στο μέλλον τόσο για την εφαρμογή ανάπτυξης γραφικού περιβάλλοντος IOT όσο και για τη πιλοτική εφαρμογή.

Όσον αφορά την εφαρμογή IOT μπορεί να γίνει επέκταση της και εφαρμογή της σε ένα φάσμα διαφορετικών τομέων. Πιο συγκεκριμένα με την σύνδεση πιο εξειδικευμένων αισθητήρων ανάλογα με τον τομέα που θέλουμε να την χρησιμοποιήσουμε και βελτίωση του συστήματος ειδοποιήσεων και γραφημάτων μπορεί να αξιοποιηθεί, παραδείγματος χάρη, στην υγειονομική περίθαλψη με σύστημα παρακολούθησης των βιομετρικών στοιχείων και συνεπώς της υγεία των ασθενών και εμφάνιση alert για επείγουσες καταστάσεις στο ιατρικό προσωπικό ή στη γεωργία με ειδικούς αισθητήρες που θα συλλέγουν δεδομένα για τις ατμοσφαιρικές ή εδαφικές συνθήκες και θα βοηθούν τους αγρότες για τη σωστή διαχείριση των καλλιεργειών τους.

Για τη πιλοτική εφαρμογή θα μπορούσε να ενσωματωθεί ένα σύστημα αυτοματισμού το οποίο θα ελέγχει τους θερμοστάτες για να ρυθμίζει αυτόματα τη θερμοκρασία των χώρων, ώστε να τηρείται σε προκαθορισμένες τιμές. Η επέκταση αυτού μπορεί να λύσει σημαντικά προβλήματα στην κατανάλωση ενέργειας συνεπώς και της οικονομίας.

Τέλος το ότι η εφαρμογή είναι παραμετροποιήσιμη, σημαίνει ότι μπορεί να χρησιμοποιηθεί είτε για ιδιωτική χρήση είτε σε μεγάλες επιχειρήσεις του δημοσίου, ή και του ιδιωτικού τομέα.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Works Cited the Editors of Encyclopedia Britannica. “Energy | Definition, Types, & Examples.” *Encyclopedia Britannica*, 2 Jan. 2019.

[2] Onyeka, M. (n.d.). *ENERGY CRISIS AND ITS EFFECTS ON NATIONAL DEVELOPMENT: THE NEED FOR ENVIRONMENTAL EDUCATION IN NIGERIA*.

[3] *Attention Required! | Cloudflare*. (n.d.). Retrieved from https://www.conserve-energy-future.com/causes-and-solutions-to-the-global-energy-crisis.php

[4] Energy Efficiency: An overlooked solution to the energy crisis. (n.d.). WWF. Retrieved from https://wwf.panda.org/wwf_news/?5538966/energy-efficiency-solution

[5] *Κοινή Υπουργική Απόφαση ΥΠΕΝ/ΔΕΠΕΑ/68315/502/2022 - ΦΕΚ 3424/Β/2-7-2022 (Κωδικοποιημένη)*. (2022, July 2). e-nomothesia.gr | Τράπεζα Πληροφοριών Νομοθεσίας. Retrieved from https://www.e-nomothesia.gr/kat-demosia-dioikese/koine-upourgike-apophase-upen-depea-68315-502-2022.html

[6] Clark, J. (2020, August 28). What is the Internet of Things, and how does it work? IBM Business Operations Blog. Retrieved from https://www.ibm.com/blogs/internet-of-things/what-is-the-iot/

[7] Castle, A. (2022, April 22). IoT Technologies Explained: History, Examples, Risks & Future. Vision of Humanity. Retrieved from https://www.visionofhumanity.org/what-is-the-internet-of-things/

[8] Characteristics of Internet of Things (IoT). (2022, July 31). RF Page. Retrieved from https://www.rfpage.com/characteristics-of-internet-of-things/

[9] Gregersen, C. (2021, February 5). A Complete Guide to IoT Protocols & Standards In 2021. Nabto. Retrieved from https://www.nabto.com/guide-iot-protocols-standards/

[10] MQTT - The Standard for IoT Messaging. (n.d.). Retrieved from https://mqtt.org/

[11] FAQ: What is AMQP and why is it used in RabbitMQ? (2019, November 21). CloudAMQP. Retrieved from https://www.cloudamqp.com/blog/what-is-amqp-and-why-is-it-used-in-rabbitmq.html

[12] Internet of Things (IOT). (n.d.). Retrieved from https://www.sas.com/el_gr/insights/big-data/internet-of-things.html

[13] 11. Εισαγωγή στην HTML. (n.d.). Retrieved from http://ebooks.edu.gr/ebooks/v/html/8547/2714/Pliroforiki_A-Lykeiou_html-empl/index3_11.html

[14] CSS: Cascading Style Sheets | MDN. Retrieved from https://developer.mozilla.org/en-US/docs/Web/CSS

[15] How to add CSS. (n.d.). Retrieved from https://www.w3schools.com/css/css_howto.asp

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212

[16] (7 Benefits of Using SASS Over Conventional CSS | Mugo Web | eZ Platform Specialists in Canada and the USA, n.d.)

[17] Wikipedia contributors. Responsive design. Βικιπαίδεια. Retrieved from https://el.wikipedia.org/wiki/Responsive_design

[18] About JavaScript - JavaScript | MDN. (2022, September 12). Retrieved from https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript

[19] Υποενότητα 2.1: Front-end frameworks: Bootstrap Framework. (n.d.). Retrieved from https://elearn.ellak.gr/mod/book/view.php?id=4993&chapterid=2855

[20] *What's a JS library? (article)*. (n.d.). Khan Academy. Retrieved from https://www.khanacademy.org/computing/computer-programming/html-css-js/using-js-libraries-in-your-webpage/a/whats-a-js-library

[21] Acharya, D. P. (2022, September 30). *The 40 Best JavaScript Libraries and Frameworks for 2022*. Kinsta®. Retrieved from https://kinsta.com/blog/javascript-libraries/

[22] *Top 10 Benefits of React.js for Application Development*. (n.d.). NCube. Retrieved October 4, 2022, from https://ncube.com/blog/top-10-benefits-of-react-js-for-application-development

[23] Node.js Introduction. (n.d.). Retrieved from https://www.w3schools.com/nodejs/nodejs_intro.asp

[24] Codecademy. (n.d.). What is Express.js? Retrieved from https://www.codecademy.com/article/what-is-express-js

[25] Εισαγωγή στις Βάσεις Δεδομένων (DataBases) Retrieved from http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-DataBases.html

[26] 1. What Is PostgreSQL? PostgreSQL Documentation. Retrieved from https://www.postgresql.org/docs/15/intro-whatis.html

[27] What is an API? (Application Programming Interface). (n.d.). MuleSoft. Retrieved from https://www.mulesoft.com/resources/api/what-is-an-api

[28] *npm: bcrypt*. (2021, February 26). Npm. Retrieved from https://www.npmjs.com/package/bcrypt

[29] *Documentation*. (n.d.). Passport.js. Retrieved from https://www.passportjs.org/docs/

[30] Grafana® Features. (n.d.). Grafana Labs. Retrieved from https://grafana.com/grafana/

[31] "Κοινή Υπουργική Απόφαση ΥΠΕΝ/ΔΕΠΕΑ/68315/502/2022 - ΦΕΚ 3424/Β/2-7-2022 (Κωδικοποιημένη)." E-Nomothesia.gr | Τράπεζα Πληροφοριών Νομοθεσίας, www.e-nomothesia.gr/kat-demosia-dioikese/koine-upourgike-apophase-upen-depea-68315-502-2022.html.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:32:16 EEST - 167.114.118.212