



“Ανάπτυξη Λογισμικού Για Εφαρμογές Σεισμικής Επικινδυνότητας”

Πτυχιακή Εργασία

Γκόγκας Γεώργιος

Επιβλέπων: Αυλωνίτης Μάρκος

Κέρκυρα, Σεπτέμβριος 2020

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ.....	3
1 - ΕΙΣΑΓΩΓΗ.....	4
1.1 Φάσεις Διαχείρισης Δεδομένων.....	7
1.2 Διαδικτυακές Εφαρμογές.....	8
1.3 Δεδομένα.....	10
1.3.1 Δεδομένα και Σεισμοί.....	10
1.3.2 Αποθήκευση Δεδομένων.....	12
1.4 Σκοπός της Εργασίας.....	13
1.4.1 Στάδια που υλοποιήθηκαν....	14
2 – Γενική Περιγραφή.....	15
2.1 Τεχνολογίες που Χρησιμοποιήθηκαν.....	16
2.1.1 Wamp Server.....	16
2.1.2 Bootstrap 4	18
2.1.3 Arduino IDE	19
2.2 Υλικό που χρησιμοποιήθηκε	22
2.2.1 Πλακέτα Ανάπτυξης.....	22
2.2.2 Αισθητήρες.....	27
3 – Περιγραφή Συστήματος.....	31
3.1 Συνδεσμολογία Υλικού.....	31
3.2 Σχήμα Βάσης Δεδομένων.....	33
3.3 Περιγραφή κομματιών λογισμικού.....	35
4 – Παρουσίαση Συστήματος.....	37
5 – Συμπεράσματα και Μελλοντικές κατευθύνσεις.....	43
6 – Παράρτημα.....	46
7 – Βιβλιογραφία	100

1 - ΕΙΣΑΓΩΓΗ

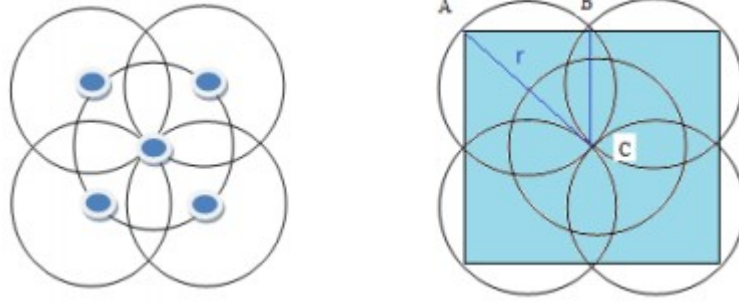
Η εποχή την οποία διανύουμε συχνά χαρακτηρίζεται και ως εποχή της πληροφορίας. Αυτό συμβαίνει όχι γιατί πρώτα δεν ήταν χρήσιμη η πληροφορία, αλλά επειδή σήμερα είναι πιο εύκολο από ποτέ να την αντλήσουμε. Όπως είναι γνωστό η πληροφορία πηγάζει από τα δεδομένα και με την εξέλιξη της τεχνολογίας πλέον μπορούμε να αποθηκεύουμε μεγάλο όγκο και παντός τύπου δεδομένα τα οποία στη συνέχεια μπορούμε να τα επεξεργαστούμε και να αντλήσουμε σημαντικές πληροφορίες.

Η συλλογή των δεδομένων μπορεί να γίνει με πολλούς και διάφορους τρόπους και συνήθως εξαρτάται από το είδος των δεδομένων που μας ενδιαφέρουν, τα οποία μπορεί να είναι χρηματοοικονομικής φύσεως, εμπορικής, επιστημονικής φύσεως, περιβαλλοντικής, δημόσιας υγείας και πολλά άλλα. Τα τελευταία χρόνια ωστόσο έχει αναπτυχθεί μια ολόκληρη οικονομία γύρω από εφαρμογές που κάνουν χρήση δεδομένων που συλλέγονται μέσω αισθητήρων. Ανάλογα με τον τύπο αισθητήρων και τον τρόπο με τον οποίο τους χρησιμοποιούμε, μπορούμε να χωρίσουμε τη διαδικασία συλλογής σε δύο κατηγορίες/ τεχνικές οι οποίες είναι και οι πιο διαδεδομένες. Αυτές είναι η τοποθέτηση αισθητήρων στον χώρο (Sensor Deployment) και η χρήση αισθητήρων πλήθους (Crowd Sensing). Παρακάτω αναλύουμε μερικά από τα χαρακτηριστικά τους και τις διαφορές τους.

Sensor Deployment

Η συγκεκριμένη διαδικασία αφορά την δημιουργία ενός ή περισσότερων ασύρματων δικτύων αισθητήρων (Wireless Sensor Network), τα οποία αποτελούνται από ιδικά σχεδιασμένα συστήματα αισθητήρων. Τα συστήματα αυτά είναι στατικά και τοποθετούνται σε μια περιοχή ενδιαφέροντος προκειμένου να συλλέξουν δεδομένα από το περιβάλλον στο οποίο βρίσκονται.

Το πιο διαδεδομένο μοντέλο δικτύου ασύρματων αισθητήρων είναι αυτό του πελάτη-εξυπηρετητή (client - server), όπου το δίκτυο αποτελείται από έναν αριθμό κόμβων πελατών και έναν κόμβο εξυπηρετητή. Με αυτόν τον τρόπο οι κόμβοι πελάτες είναι υπεύθυνοι για την συλλογή δεδομένων από το περιβάλλον, μέσω των αισθητήρων που είναι προσαρμοσμένοι επάνω τους, και την αποστολή τους στον κόμβο εξυπηρετητή ο οποίος με τη σειρά του συλλέγει τα δεδομένα από όλους τους κόμβους πελάτες τα επεξεργάζεται και τα στέλνει σε κάποια απομακρυσμένη βάση δεδομένων. Έτσι δεν απαιτείται μεγάλη επεξεργαστική ισχύ μνήμη και πολυπλοκότητα των κόμβων πελατών κερδίζοντας έτσι σε κόστος το οποίο είναι σημαντικός παράγοντας σε τέτοιου είδους συστήματα.



Σχήμα 1

Περιγραφή: Ωφέλιμη κάλυψη με συνδεδεμένους αισθητήρες

Σημαντικό ρόλο σε τέτοιου είδους συστήματα αποτελεί ο τρόπος με τον οποίο αυτά θα διαταχθούν στο χώρο. Ανάλογα με την περιοχή ενδιαφέροντος αλλάζει και η διάταξη των αισθητήρων, δηλαδή διαφορετική διάταξη θα χρησιμοποιηθεί εάν η περιοχή ενδιαφέροντος αποτελεί ένα τετράγωνο στο χώρο χωρίς φυσικά εμπόδια, διαφορετική ένα κύκλο με εμπόδια και διαφορετική ένα ακαθόριστο σχήμα. Όπως γίνεται κατανοητό τα σχήματα και οι διατάξεις είναι άπειρες και για τον λόγο αυτό έχουν αναπτυχθεί διάφοροι αλγόριθμοι που βοηθούν ώστε να γίνει αποδοτικά η εγκατάσταση των αισθητήρων, ωστόσο δεν θα επεκταθούμε περαιτέρω στην παρούσα εργασία.

Τέλος το συγκεκριμένο μοντέλο παρέχει μεγάλη ακρίβεια και ποικιλία σε δεδομένα αισθητήρων, ωστόσο για να γίνει εφικτό αυτό θα πρέπει να δαπανηθεί ένα αρκετά μεγάλο χρηματικό ποσό για την αγορά, την εγκατάσταση και την συντήρηση αυτών των συστημάτων, γίνεται αμέσως κατανοητό ότι η ποιότητα των δεδομένων είναι ανάλογη του κόστους γεγονός που αποτελεί και το κυριότερο ελάττωμα του συγκεκριμένου μοντέλου.

Crowd Sensing

Με την ραγδαία εξέλιξη της τεχνολογίας τα τελευταία χρόνια μπορούμε να έχουμε στην τσέπη μας αντί για ένα απλό κινητό τηλέφωνο, έναν αρκετά ισχυρό υπολογιστή στον οποίο είναι προσαρμοσμένοι ένας αριθμός χρήσιμων αισθητήρων. Το μοντέλο του crowd sensing λοιπόν εκμεταλλεύεται αυτά τα έξυπνα τηλέφωνα (smartphones) και τις δυνατότητες τους προκειμένου να συλλέξει χρήσιμα δεδομένα για την εκάστοτε εφαρμογή που τα χρησιμοποιεί, οι οποίες κατά κύριο λόγο έχουν ανθρωποκεντρικό και περιβαλλοντολογικό χαρακτήρα. Το συγκεκριμένο μοντέλο χωρίζεται σε δύο κατηγορίες, το participatory (συμμετοχικό) και το opportunistic (ευκαιριακό) sensing.



Εικόνα 1

Περιγραφή: Έξυπνα τηλέφωνα και δυνατότητες

Με τον όρο *participatory sensing* (Wang et al., 2015) εννοούμε τη συλλογή δεδομένων από τα έξυπνα τηλέφωνα, όπου ο χρήστης των οποίων λαμβάνει μέρος ενεργά στην διαδικασία συλλογής και καλείται σε ορισμένες περιπτώσεις να εκτελέσει σημαντικές ενέργειες προκειμένου να γίνει εφικτή η συλλογή των δεδομένων, όπως η ενεργοποίηση/ απενεργοποίηση των αισθητήρων του κινητού ή η εκτέλεση κάποιας εφαρμογής χειροκίνητα. Συνήθως παρέχονται στο χρήστη κάποια κίνητρα, κυρίως χρηματικά προκειμένου να συμμετάσχει σε ένα τέτοιο πείραμα.

Αντίθετα με τον όρο *opportunistic sensing* εννοούμε τη συλλογή δεδομένων από τα έξυπνα τηλέφωνα, όπου ο κάτοχος των οποίων δεν συμμετέχει ενεργά στην συλλογή δεδομένων και πολλές φορές δεν γνωρίζει καν πότε γίνεται η συλλογή. Ωστόσο αυτό για να γίνει εφικτό θα πρέπει πρώτα να συναινέσει ο χρήστης και να θέσει κάποιους όρους για το πότε και ποιοι αισθητήρες θα μπορούν να χρησιμοποιηθούν.

Τέλος, όσο αναφορά το μοντέλο του *crowd sensing* γενικότερα, γίνεται κατανοητό ότι έχει ένα προβάδισμα σε σχέση με το *sensor deployment* ως προς το κόστος εφαρμογής. Ωστόσο ο περιορισμένος αριθμός αισθητήρων των κινητών τηλεφώνων σε συνδυασμό με την απόδοσή τους καθιστά δυνατή τη χρήση αυτού του μοντέλου μόνο σε ένα περιορισμένο αριθμό εφαρμογών, γεγονός που είναι και το σημαντικότερο μειονέκτημα του.

1.1 Φάσεις διαχείρισης δεδομένων

Στην προηγούμενη ενότητα της εισαγωγής κάναμε λόγο για μερικούς από τους τρόπους με τους οποίους μπορεί να γίνει η συλλογή των δεδομένων των αισθητήρων. Ωστόσο

αυτό αποτελεί μόνο την μία και αρχική φάση της διαχείρισης των δεδομένων, αυτή της συλλογής των δεδομένων. Τα στάδια ή φάσεις όμως είναι επτά και είναι τα παρακάτω.

- Συλλογή Δεδομένων
- Μετάδοση Δεδομένων
- Καθαρισμός Δεδομένων
- Αποθήκευση Δεδομένων
- Συμπίεση Δεδομένων
- Επεξεργασία / Ανάλυση Δεδομένων
- Παρουσίαση Δεδομένων

Συλλογή : Όπως αναφέραμε και στην προηγούμενη ενότητα, αυτό το στάδιο αναφέρεται στους τρόπους με τους οποίους επιτυγχάνεται η συλλογή των δεδομένων καθώς και στις τεχνολογίες που χρησιμοποιούνται για να γίνει αυτή εφικτή.

Μετάδοση : Αφορά τις τεχνολογίες και τα πρωτόκολλα που χρησιμοποιούνται για την αποστολή των δεδομένων από το σύστημα αισθητήρων (ή οποιοδήποτε άλλο σύστημα συλλογής δεδομένων) προς τη δομή η οποία είναι υπεύθυνη για την αποθήκευση (π.χ. βάση δεδομένων σε server).

Καθαρισμός : Είναι η διαδικασία κατά την οποία τα δεδομένα που συλλέγονται, ελέγχονται για την ορθότητά τους ώστε τυχόν σφάλματα κατά τη διαδικασία συλλογής να μην συμπεριληφθούν στην αποθήκευση και επηρεάσουν το τελικό αποτέλεσμα.

Αποθήκευση : Είναι το στάδιο στο οποίο τα δεδομένα που έχουν συλλεχθεί, αποθηκεύονται σε μία βάση δεδομένων για πιθανών περαιτέρω επεξεργασία.

Συμπίεση : Στο στάδιο αυτό τα δεδομένα που έχουν αποθηκευτεί στη βάση υπόκεινται σε κάποιου είδους συμπίεση η μορφή της οποίας εξαρτάται από τον εκάστοτε αλγόριθμο που θα χρησιμοποιηθεί. Το στάδιο αυτό είναι απαραίτητο σε εφαρμογές που αποθηκεύουν μεγάλο όγκο δεδομένων ανά το χρόνο.

Επεξεργασία : Είναι το στάδιο στο οποίο τα αποθηκευμένα δεδομένα επεξεργάζονται έτσι ώστε να παραχθεί η απαιτούμενη πληροφορία.

Παρουσίαση : Στο συγκεκριμένο στάδιο επιλέγουμε ποιες τεχνολογίες θα χρησιμοποιηθούν για να προβληθούν τα δεδομένα που επεξεργαστήκαν στον εκάστοτε χρήστη της εφαρμογής και αφορά κυρίως το γραφικό περιβάλλον και πως αυτό παρουσιάζεται στον χρήστη.

1.2 Διαδικτυακές Εφαρμογές

Ως διαδικτυακή εφαρμογή (Web Application) χαρακτηρίζεται οποιαδήποτε εφαρμογή χρησιμοποιεί το διαδίκτυο προκειμένου να γίνει προσβάσιμη από τους χρήστες, απαραίτητη προϋπόθεση ωστόσο για τη χρήση της είναι η χρησιμοποίηση ενός περιηγητή (Browser). Οι εφαρμογές αυτές κατά κύριο λόγο εκτελούνται απομακρυσμένα σε κάποιο ισχυρό μηχάνημα που λειτουργεί ως σταθμός εξυπηρέτησης (Server), γεγονός που καθιστά εφικτή την παροχή των υπηρεσιών σε περισσότερους από έναν χρήστη ταυτόχρονα χωρίς καθυστέρηση.

Εκτός από τις διαδικτυακές εφαρμογές, οι ψηφιακές εφαρμογές μπορεί να είναι και τοπικές εφαρμογές ή κινητές εφαρμογές (Mobile Applications). Οι τοπικές εφαρμογές είναι αυτές που δημιουργήθηκαν πρώτες καθώς χρησιμοποιούν τοπικά υπολογιστικά συστήματα για τη λειτουργία τους όπως Laptop και Desktop, οι συγκεκριμένες εφαρμογές είναι και οι πιο διαδεδομένες μέχρι και σήμερα, αυτό όμως τείνει να αλλάξει τα τελευταία χρόνια. Σε αυτό έχει συμβάλει σημαντικά η ανάπτυξη της τεχνολογίας και η εξέλιξη των «έξυπνων» τηλεφώνων τα οποία χρησιμοποιούν οι κινητές εφαρμογές για την λειτουργία τους. Η ραγδαία ανάπτυξη της υπολογιστικής δύναμης αυτών των συσκευών καθιστά δυνατή την εκτέλεση πολύπλοκων και απαιτητικών εφαρμογών που πριν από λίγα χρόνια έμοιαζε αδύνατο.

Πλεονεκτήματα

Όπως κάθε προϊόν έτσι και οι διαδικτυακές εφαρμογές έχουν τα πλεονεκτήματα και τα μειονεκτήματά τους. Ως ένα από τα σημαντικότερα πλεονεκτήματα μπορεί να χαρακτηριστεί η ανεξαρτησία από το είδος συσκευής. Ο χρήστης έχει άμεση πρόσβαση σε οποιαδήποτε εφαρμογή από οποιαδήποτε συσκευή είτε αυτή είναι υπολογιστής laptop ή smartphone, χωρίς να χρειάζεται εγκατάσταση κάποιου επιπρόσθετου λογισμικού. Η μόνη απαραίτητη εφαρμογή είναι ο περιηγητής ο οποίος είναι προεγκατεστημένος στα περισσότερα λειτουργικά συστήματα, γεγονός που δίνει τη δυνατότητα σε μεγάλες επιχειρήσεις και οργανισμούς να γλιτώσουν σημαντικούς πόρους χρόνο και χρήμα από την εγκατάσταση σε μεγάλο αριθμό υπολογιστικών συστημάτων, καθώς εκτελούνται απομακρυσμένα σε κάποιο server. Αντίθετα οι άλλες δύο κατηγορίες απαιτούν από το χρήστη την εγκατάστασή τους τοπικά κάτι που δεσμεύει χώρο και χρόνο.

Επίσης ένα άλλο πλεονέκτημα των διαδικτυακών εφαρμογών είναι η γρήγορη αναβάθμισή τους. Το γεγονός ότι είναι εγκατεστημένες σε ένα server έχει σαν αποτέλεσμα την άμεση εγκατάσταση αναβαθμίσεων και ενημερώσεων για όλους τους χρήστες χωρίς την παρέμβαση αυτών και χωρίς την δέσμευση περεταίρω χώρου, σε αντίθεση με τις άλλες δύο κατηγορίες που απαιτούν την παρέμβαση του χρήστη για την ενημέρωση για την αναβάθμιση, μία διαδικασία που πολλές φορές μπορεί να είναι πολύ χρονοβόρα ειδικά όταν πρόκειται για πολλά συστήματα και να απαιτεί αρκετή μνήμη.

Τέλος, οι διαδικτυακές εφαρμογές από κατασκευαστική και προγραμματιστική άποψη είναι πιο εύκολο να δημιουργηθούν καθώς είναι συμβατές με όλα τα λειτουργικά και υπολογιστικά συστήματα, έτσι η εφαρμογή είναι μία και ίδια για όλους τους χρήστες. Από την άλλη οι κινητές εφαρμογές και οι τοπικές γενικότερα πρέπει να δημιουργηθούν από την αρχή και εξ ολοκλήρου για κάθε ένα λειτουργικό σύστημα και πολλές φορές το

αποτέλεσμα της εφαρμογής είναι διαφορετικό για κάθε ένα από αυτά προκαλώντας σύγχυση στους χρήστες.

Μειονεκτήματα

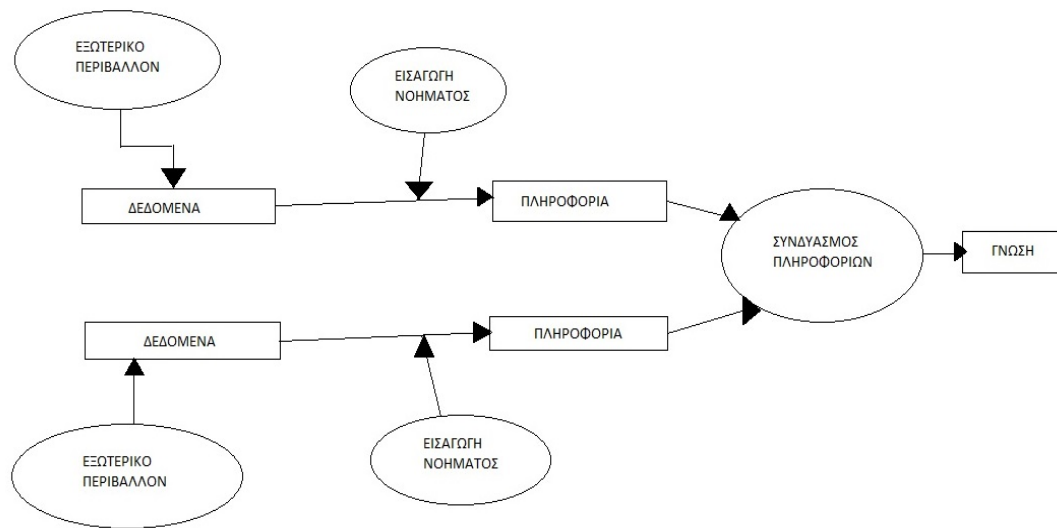
Ένα από τα πιο χαρακτηριστικά μειονεκτήματα των διαδικτυακών εφαρμογών είναι η αδυναμία λειτουργίας τους όταν υπάρχει έλλειψη πρόσβασης στο διαδίκτυο, καθώς η λειτουργία τους βασίζεται σε ένα απομακρυσμένο υπολογιστικό σύστημα που είναι προσβάσιμο μόνο με τη χρήση του διαδικτύου. Ωστόσο το συγκεκριμένο μειονέκτημα τα τελευταία χρόνια με την εξέλιξη των τεχνολογιών των ασύρματων δικτύων (wifi, 3G-5G) που γίνονται όλο και πιο γρήγορα και προσβάσιμα σε μεγαλύτερο αριθμό περιοχών, τείνει να πάψει να υφίσταται.

Επίσης, ένα μειονέκτημα που αφορά το κατασκευαστικό κομμάτι είναι η έλλειψη καθολικής συμβατότητας με όλους τους περιηγητές αυτό συμβαίνει διότι δεν είναι όλοι οι περιηγητές πλήρως συμβατοί με τις νέες τεχνολογίες όπως είναι η HTML5. Ως αποτέλεσμα είναι ότι αρκετές εφαρμογές μπορεί να αντιμετωπίσουν προβλήματα συμβατότητας με κάποια από τα προγράμματα περιήγησης και να μην μπορούν να λειτουργήσουν αποτελεσματικά ή και καθόλου. Η μόνη λύση σε αυτό το πρόβλημα είναι η επικοινωνία κατασκευαστή με πελάτη ώστε να προβούν στην επιλογή ορισμένων από τους περιηγητές του εμπορίου ως βασικούς με τη χρήση των οποίων η λειτουργία της εφαρμογής θα είναι ομαλή. Το σημαντικότερο ίσως πρόβλημα από την εξάρτηση από τους περιηγητές είναι η πιθανή μη συμβατότητα της εφαρμογής μετά από μία αναβάθμιση τις έκδοσης ενός περιηγητή και αυτό διότι πρέπει να ληφθούν έγκαιρα δραστικά μέτρα ώστε να μην επηρεαστούν οι χρήστες.

Τέλος, ένα μειονέκτημα που όπως είδαμε και πιο πάνω είναι επίσης και πλεονέκτημα είναι η άμεση αναβάθμιση και ενημέρωση, καθώς δεν δίνεται στο χρήστη η δυνατότητα επιλογής αν θα προβεί στην εγκατάσταση τις ενημερώσεις ή όχι. Έτσι μετά την αναβάθμιση μπορεί να αλλάξουν αρκετά πράγματα στην εφαρμογή τα οποία ενδέχεται να προκαλέσουν σύγχυση στο χρήστη που έχει συνηθίσει την προηγούμενη έκδοση της εφαρμογής.

1.3 Δεδομένα

Ως δεδομένα μπορούμε να ορίσουμε τα ακατέργαστα στοιχεία τα οποία τα χρησιμοποιούμε για επεξεργασία προκειμένου να αντλήσουμε από αυτά πληροφορία. Μέσα από την ανακάλυψη, την κατανόηση και τη σωστή επιλογή και κατανομή των πληροφοριών μπορούμε να εξορύξουμε γνώση η οποία με τη σειρά της θα μας βοηθήσει να λάβουμε σημαντικές αποφάσεις για συσχετιζόμενα προβλήματα σε διάφορους τομείς της ζωής. Δεν είναι τυχαίο άλλωστε ότι η εποχή μας αναφέρεται και ως εποχή της πληροφορίας. Πλέον έχει γίνει γνωστό ότι η πληροφορία είναι δύναμη για όποιον την κατέχει. Με την ραγδαία ανάπτυξη της τεχνολογίας τα τελευταία χρόνια καθίσταται δυνατή η ανάπτυξη βάσεων δεδομένων και αποθηκών δεδομένων που μπορούν να συλλέξουν μεγάλο όγκο δεδομένων και κατ'επέκταση πληροφορίας.



Σχήμα 2

Περιγραφή: μια γενική όψη της μετατροπής δεδομένων σε γνώση

1.3.1 Δεδομένα και Σεισμοί

Όπως αναφέραμε και παραπάνω τα δεδομένα οποιουδήποτε τύπου είναι πολύ σημαντικά για την κατανόηση του περιβάλλοντος, ωστόσο για μερικά δεδομένα κρίνεται απαραίτητη η συλλογή τους και είναι αυτά που προκύπτουν από επικίνδυνα για τον άνθρωπο φαινόμενα όπως είναι αυτά των σεισμών με τα οποία και θα ασχοληθούμε στην παρούσα εργασία.

Οι σεισμοί είναι ένα φαινόμενο το οποίο εκδηλώνεται κατά κύριο λόγο χωρίς σαφή προειδοποίηση, είναι αδύνατον να αποτραπεί και παρά τη μικρή χρονική διάρκεια τους κάθε χρόνο προκαλούν μεγάλες υλικές ζημιές σε όλο το πλάτος του πλανήτη, αλλά πολλές φορές κοστίζουν και ανθρώπινες ζωές, γεγονός που από πολύ νωρίς οδήγησε τους επιστήμονες να ασχοληθούν με αυτό το φαινόμενο.

Στην ουσία πρόκειται για την ξαφνική κίνηση / μετάθεση της επιφάνειας της γης που ωστόσο πυροδοτείται από μετατοπίσεις που γίνονται πιο βαθιά στον φλοιό της γης όπου βρίσκεται το επίκεντρο. Το μέγεθος ενός σεισμού είναι μοναδικό και ορίζει την ποσότητα της ενέργειας που απελευθερώνεται, υπολογίζεται με συγκεκριμένους μαθηματικούς τύπους και μετράται με βάση την κλίμακα Ρίχτερ.

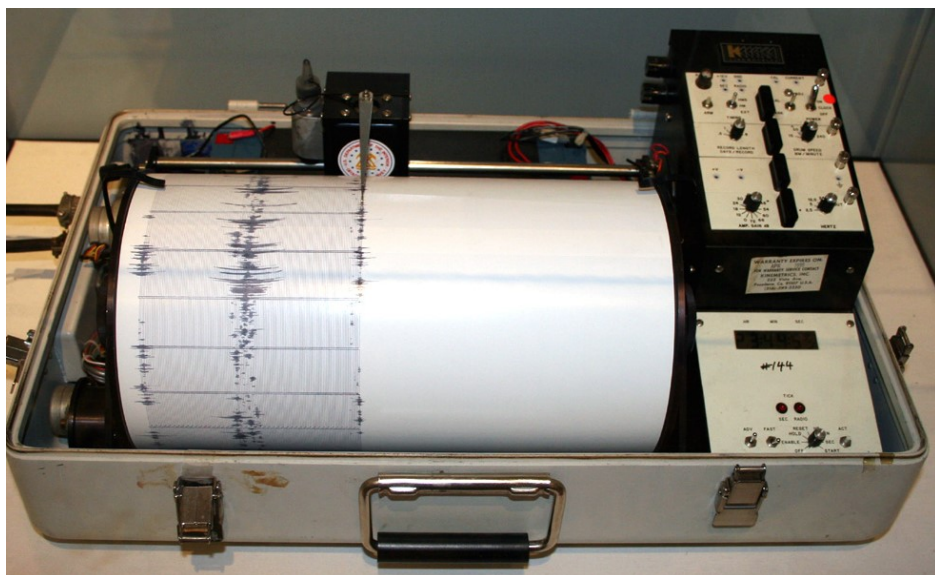
Η καταγραφή αυτού του φαινομένου αρχικά περιοριζόταν μόνο στον χρόνο άφιξης ενός σεισμού και γινόταν με την χρήση ενός **σεισμοσκοπίου**, το οποίο ήταν ένα απλό όργανο αποτελούμενο από μία μάζα που βρισκόταν σε ασταθή ισορροπία και η ανατροπή της από το σεισμό έβαζε σε λειτουργία κάποιο χρονόμετρο ή σταματούσε κάποιο ρολόι, παρέχοντας την ακριβή χρονική στιγμή άφιξης του σεισμού.



Εικόνα 2

Περιγραφή: Σεισμοσκόπιο

Αργότερα δημιουργήθηκαν οι **σεισμογράφοι** με την βοήθεια των οποίων πέρα από τον χρόνο άφιξης ενός σεισμού ήταν πλέον δυνατή και η καταγραφή όλων των τύπων των σεισμικών κυμάτων, ώστε να γίνεται κατ' επέκταση δυνατός ο προσδιορισμός της δύναμης και της κατεύθυνσης ενός σεισμού. Οι πρώτοι σεισμογράφοι ήταν αναλογικοί και αποτελούνταν από ένα εκκρεμές, ένα σύστημα ενίσχυσης και ένα σύστημα αναγραφής, με τα κύματα να καταγράφονται σε χάρτινο κύλινδρο.



Εικόνα 3

Περιγραφή: Σεισμογράφος φορητός

Τέλος, τα τελευταία χρόνια με την εμφάνιση των υπολογιστών και των μετατροπών αναλογικού σήματος σε ψηφιακό η λειτουργικότητα και οι δυνατότητες του εξοπλισμού σεισμομέτρησης έχουν αυξηθεί σημαντικά. Ενώ παράλληλα υπάρχει η τάση εύρεσης νέων πιο προσιτών και αποτελεσματικών τρόπων καταγραφής των σεισμικών φαινομένων με την επικρατέστερη να είναι η χρήση ψηφιακών επιταχυνσιόμετρων, από υψηλού κόστους μεγάλης ακρίβειας αισθητήρες μέχρι αισθητήρες που βρίσκονται στα γνωστά σε όλους μας έξυπνα κινητά τηλέφωνα. Γεγονός που θα μπορεί να μας δώσει ακόμα πιο πολλά χρήσιμα δεδομένα σχετικά με το συγκεκριμένο αυτό φαινόμενο. Κάτι τέτοιο μπορεί να γίνει εφικτό χάρη στις ιδιότητες ενός τέτοιου αισθητήρα που μπορεί να καταγράψει την επιτάχυνση ενός σώματος στο οποίο είναι προσαρμοσμένος (του εδάφους στην περίπτωση μας) στους τρεις άξονες, η οποία κατ' επέκταση μπορεί να μας δώσει πληροφορίες για τις δονήσεις, η ακρίβεια των οποίων συνήθως είναι ανάλογη του κόστους ενός τέτοιου αισθητήρα.

1.3.2 Αποθήκευση Δεδομένων

Τα χρήσιμα αυτά δεδομένα λοιπόν πρέπει με κάποιον τρόπο να αποθηκευτούν προκειμένου να παράξουμε γνώση. Οι βασικότερες μορφές αποθήκευσης είναι σε αρχεία, σε βάσεις δεδομένων (DataBases) και σε αποθήκες δεδομένων (Data Warehouses).

Η τεχνολογία των βάσεων δεδομένων αναπτύχθηκε λόγω των περιορισμένων δυνατοτήτων που παρείχαν τα συστήματα αρχείων αλλά και λόγω της ανάγκης για έναν ενιαίο τρόπο αποθήκευσης που δεν θα χρειαζόταν η δημιουργία και αποθήκευση αρχείων δεδομένων για κάθε εφαρμογή ξεχωριστά, επιτυγχάνοντας την αποτελεσματική οργάνωση συλλογών δεδομένων. Η καθιέρωση της τεχνολογίας αυτής οδήγησε στην συστηματοποίηση της διαδικασίας και παράλληλα με την ανάπτυξή της στη δυνατότητα αποθήκευσης μεγάλου όγκου δεδομένων. Ως αποτέλεσμα αυτής της ανάπτυξης μπορεί να θεωρηθεί η δημιουργία των αποθηκών δεδομένων. Μια αποθήκη δεδομένων είναι μια ολοκληρωμένη (intergrated) βάση δεδομένων, η οποία περιλαμβάνει δεδομένα που συλλέχθηκαν και ομογενοποιήθηκαν από πολλές και ετερογενείς βάσεις δεδομένων. Τέλος η διαχείριση των βάσεων δεδομένων γίνεται με τη χρήση κατάλληλων λογισμικών διαχείρισης βάσεων που σε συνδυασμό με τη χρήση γλωσσών διαχείρισης δεδομένων παρέχουν στο χρήστη τη δυνατότητα επεξεργασίας, ανάκλησης, τροποποίησης και διαγραφής δεδομένων, μία από τις πιο γνωστές και ευρέως διαδεδομένες γλώσσες διαχείρισης δεδομένων είναι η SQL (Structured Query Language). Παρακάτω (πίνακας 1) βλέπουμε έναν χαρακτηριστικό πίνακα με τις τεχνολογίες αποθήκευσης, τους τύπους ερωτημάτων που δέχονται και το είδος των δεδομένων που εξάγουν.

Τεχνολογία	Ερωτήματα	Χαρακτηριστικά
Αρχεία	«Ποιο το σύνολο πωλήσεων τα τελευταία 5 έτη;»	Ανάκτηση στατικών δεδομένων
Σχεσιακές ΒΔ	«Δώσε το σύνολο πωλήσεων στη Μακεδονία φέτος»	Ανάκτηση δυναμικών δεδομένων σε επίπεδο εγγραφής

Αποθήκες Δεδομένων	«Δώσε το σύνολο πωλήσεων στη Μακεδονία φέτος. Ανάλυσε κατά νομό και μήνα»	Ανάκτηση δυναμικών, ετερογενών δεδομένων, σε πολλαπλά επίπεδα
--------------------	---	---

Πίνακας 1: Σύγκριση τεχνολογιών δεδομένων

1.4 Σκοπός της Εργασίας

Σκοπός της παρούσας πτυχιακής εργασίας είναι η μελέτη και η υλοποίηση ενός συστήματος συλλογής δεδομένων από άκρη σε άκρη. Πιο αναλυτικά, μιας διαδικτυακής εφαρμογής μέσα από την οποία μπορεί ο κάθε χρήστης που έχει πρόσβαση στο διαδίκτυο και μία οποιαδήποτε συσκευή (Desktop, Laptop, Tablet, Smartphone), να δει ανά πάσα στιγμή εικονοποιημένα δεδομένα αισθητήρων θερμοκρασίας αέρα και επιτάχυνσης υπό την μορφή διαγραμμάτων σε σχέση με τον χρόνο καθώς και διάφορα στατιστικά. Τα δεδομένα για την δημιουργία των διαγραμμάτων συλλέγονται από ψηφιακούς αισθητήρες οι οποίοι είναι προσαρμοσμένοι επάνω σε συμβατούς μικροεπεξεργαστές τύπου Arduino όπου με τη βοήθεια τους τα δεδομένα αυτά αποστέλλονται στον server. Επίσης παρέχεται και η δυνατότητα προσθήκης σταθμών αισθητήρων από το χρήστη στο συνολικό σύστημα.

Μέσα από την συγκεκριμένη υλοποίηση στο τέλος της παρούσας εργασίας θα είμαστε σε θέση να γνωρίζουμε αν ένα τέτοιο σύστημα που βασίζεται σε επεξεργαστές και αισθητήρες χαμηλού κόστους, μπορεί να αποτελέσει ή όχι μία βάση για συλλογή χρησίμων δεδομένων που μπορούν να συμβάλουν στην ανίχνευση σεισμών καθώς και βάση για μελλοντικές επεκτάσεις σε μεγαλύτερη κλίμακα.

1.4.1 Στάδια που υλοποιήθηκαν

Η παρούσα εργασία αποτελεί την υλοποίηση ενός συστήματος από άκρη σε άκρη. Ένα τέτοιο σύστημα με τη σειρά του αποτελείται από έναν αριθμό σταδίων τα οποία είναι απαραίτητα για τη λειτουργία του. Τα στάδια αυτά τα αναφέραμε παραπάνω και είναι αυτά της συλλογής, της μετάδοσης, του καθαρισμού, της αποθήκευσης, της συμπίεσης, της επεξεργασίας και της παρουσίασης. Ωστόσο στην υλοποίησή μας θα ασχοληθούμε μόνο με τα στάδια της συλλογής, της μετάδοσης, της αποθήκευσης, της επεξεργασίας και της παρουσίασης. Σε αυτή την ενότητα θα δούμε τι γίνεται ακριβώς σε αυτά τα στάδια, χωρίς όμως να μπορούμε σε λεπτομέρειες καθώς θα αναλυθούν περεταίρω σε επόμενα κεφάλαια.

Συλλογή

Η συλλογή των δεδομένων πραγματοποιήθηκε με την βοήθεια ενός μικροεπεξεργαστή – πλακέτας ανάπτυξης (Development board) τύπου Arduino και δύο αισθητήρων, ενός θερμοκρασίας και ενός επιτάχυνσης (Accelerometer), τα οποία θα αναλύσουμε αργότερα στην ενότητα του υλικού. Για να γίνει εφικτή η συλλογή προγραμματίσαμε κατάλληλα σε γλώσσα προγραμματισμού C++ την πλακέτα ώστε να αντλεί δεδομένα από τους αισθητήρες ανά τακτά χρονικά διαστήματα. Εφόσον μας ενδιέφερε να αναγνωρίζουμε έγκαιρα την οποιαδήποτε κίνηση στους άξονες X, Y, Z , θέσαμε η ανάγνωση από τον αισθητήρα επιτάχυνσης να γίνεται ανά 300ms. Ενώ η ανάγνωση του αισθητήρα θερμοκρασίας γίνεται ανά μία ώρα κάτι που είναι επαρκές για να χαρτογραφήσουμε τις αλλαγές της θερμοκρασίας.

Μετάδοση

Η μετάδοση των δεδομένων που συλλέγονται από τους αισθητήρες επιτυγχάνεται με τη χρήση του Wi-fi module το οποίο στην περίπτωσή μας είναι ενσωματωμένο στην πλακέτα ανάπτυξης με τη βοήθεια του οποίου συνδεόμαστε στο τοπικό μας δίκτυο ασύρματα και κατ' επέκταση στον εξυπηρετητή που φιλοξενεί τη βάση δεδομένων μας. Εν συνεχεία εκκινούμε μια HTTP (Hypertext Transfer Protocol) σύνδεση πελάτη και στέλνουμε ένα POST αίτημα στον εξυπηρετητή μας με τα δεδομένα που συλλέξαμε. Με τη βοήθεια της γλώσσας προγραμματισμού PHP (Hypertext Preprocessor) λαμβάνουμε το POST αίτημα από τη μεριά του εξυπηρετητή.

Αποθήκευση

Η αποθήκευση των δεδομένων γίνεται εφικτή εάν στον εξυπηρετητή ο οποίος λάβει το POST request έχουμε κάνει εγκατάσταση κάποιου προγράμματος διαχείρισης βάσεων δεδομένων, όπως στη δικιά μας περίπτωση της MySQL. Έτσι με τον κατάλληλο προγραμματισμό σε γλώσσα SQL (Structured Query Language) η οποία χρησιμοποιείται για την επικοινωνία με τις βάσεις δεδομένων, δημιουργούμε μια βάση δεδομένων, σε περίπτωση που δεν υπάρχει, με όλους τους πίνακες και τα αντικείμενα τα οποία χρειαζόμαστε. Στη συνέχεια εισάγουμε / αποθηκεύουμε τα δεδομένα που συλλέξαμε κατάλληλα στη βάση που δημιουργήσαμε. Σε αυτό το σημείο να τονίσουμε ότι προτού εισάγουμε τα δεδομένα στη βάση προηγείται έλεγχος για τυχόν μη έγκυρες τιμές, χαρακτήρες, κενά κ.α.

Επεξεργασία

Στο στάδιο αυτό κάνουμε την απαραίτητη επεξεργασία των δεδομένων, είτε αυτήν αφορά τα ήδη αποθηκευμένα δεδομένα (όπως για παράδειγμα την αφαίρεση τιμών που δεν μας ενδιαφέρουν από τη βάση) είτε τα δεδομένα που πρόκειται να εμφανιστούν στον χρήστη όπως υπολογισμός μέσης τιμής, μέγιστης και ελάχιστης τιμής δεδομένης χρονικής περιόδου κ.α.. Η επεξεργασία αυτή γίνεται κατά κύριο λόγο με τη χρήση και πάλι της γλώσσας προγραμματισμού SQL.

Παρουσίαση

Η παρουσίαση αφορά το κομμάτι αυτό που στην ουσία το αποτέλεσμα βλέπει ο χρήστης. Δηλαδή αφορά το γραφικό κομμάτι της διαδικτυακής εφαρμογής και το πώς αυτά τα δεδομένα που συλλέξαμε, αποθηκεύσαμε και επεξεργαστήκαμε εμφανίζονται στον χρήστη. Για να γίνει αυτό κάναμε χρήση των γλωσσών προγραμματισμού HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) και JavaScript. Επίσης για περεταίρω παραμετροποίηση κάναμε χρήση των βιβλιοθηκών Bootstrap και CanvasJs.

2 – ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ

Όπως αναφέραμε παραπάνω στο κεφάλαιο της εισαγωγής, ο σκοπός της παρούσας πτυχιακής εργασίας είναι η κατασκευή μιας διαδικτυακής εφαρμογής μέσα από την οποία ο χρήστης θα μπορεί να δει δεδομένα θερμοκρασίας και επιτάχυνσης σε μορφή διαγραμμάτων ως προς τον χρόνο, ενώ παράλληλα θα εμφανίζονται στο χρήστη διάφορα στατιστικά και θα έχει τη δυνατότητα να εισάγει και το δικό του σύστημα. Για να γίνουν όλα τα παραπάνω εφικτά, είναι αναγκαία η χρήση ορισμένων εργαλείων, τεχνολογιών και γλωσσών προγραμματισμού. Σε αυτό το κεφάλαιο λοιπόν θα δούμε τι από όλα αυτά χρησιμοποιήθηκε και θα αναλύσουμε ορισμένες από αυτές τις τεχνολογίες. Επίσης θα δούμε και το υλικό που χρησιμοποιήθηκε και τους λόγους για τους οποίους επιλέχθηκε.

2.1 Τεχνολογίες Που Χρησιμοποιήθηκαν

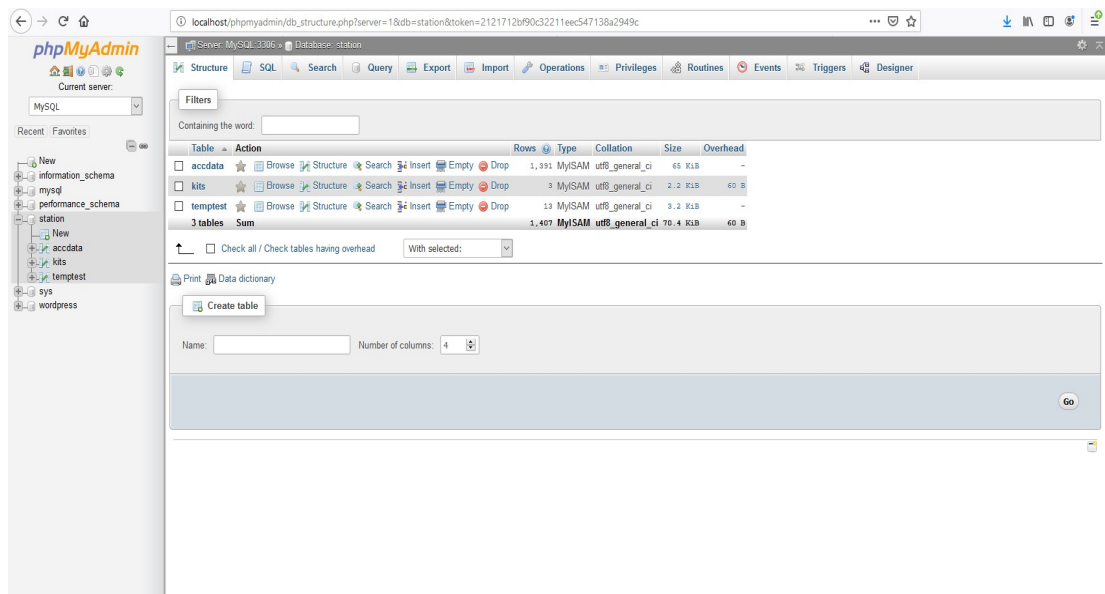
Για την κατασκευή της εφαρμογής μας αρχικά χρειάστηκε να στήσουμε έναν server, σε αυτό μας βοήθησε το Wamp Server λογισμικό. Στη συνέχεια για την κατασκευή του οπτικού κομματιού, αυτό που θα βλέπει ο χρήστης δηλαδή, χρησιμοποιήσαμε την βιβλιοθήκη Bootstrap 4 και για τις εικόνες και τα λογότυπα τα προγράμματα Adobe Photoshop και illustrator. Τέλος, για τον προγραμματισμό του μικροεπεξεργαστή χρειαστήκαμε το Arduino IDE. Παρακάτω θα δούμε μερικά από αυτά πιο αναλυτικά.

2.1.1 Wamp Server

Όπως αναφέραμε και στο προηγούμενο κεφάλαιο, απαραίτητη προϋπόθεση για την λειτουργία μιας διαδικτυακής εφαρμογής είναι η ύπαρξη ενός εξυπηρετητή (Server) στον οποίο θα συνδέεται και ο οποίος θα παρέχει τη βάση δεδομένων και την επεξεργαστική ισχύ. Στο κομμάτι αυτό της δημιουργίας ενός εξυπηρετητή μας βοήθησε το Wamp Server. Πρόκειται για μία πλατφόρμα η οποία μας δίνει τη δυνατότητα να κατασκευάσουμε δυναμικές διαδικτυακές εφαρμογές με τη χρήση Apache, PHP, MySQL ή MariaDB. Η εγκατάσταση του είναι πλήρης αυτοματοποιημένη και εγκαθιστά όλα τα απαραίτητα πακέτα. Μετά την εγκατάσταση δημιουργείται ένας Apache HTTP εξυπηρετητής που είναι υπεύθυνος για την επικοινωνία με το πρόγραμμα πλοήγησης χρησιμοποιώντας το πρωτόκολλο HTTP . Τέλος μας παρέχει το εργαλείο phpMyAdmin το οποίο μας δίνει τη δυνατότητα να διαχειριζόμαστε τη MySQL βάση δεδομένων μας μέσω του διαδικτύου, μερικές από τις λειτουργίες που μας παρέχει χωρίς τη χρήση κώδικα είναι:

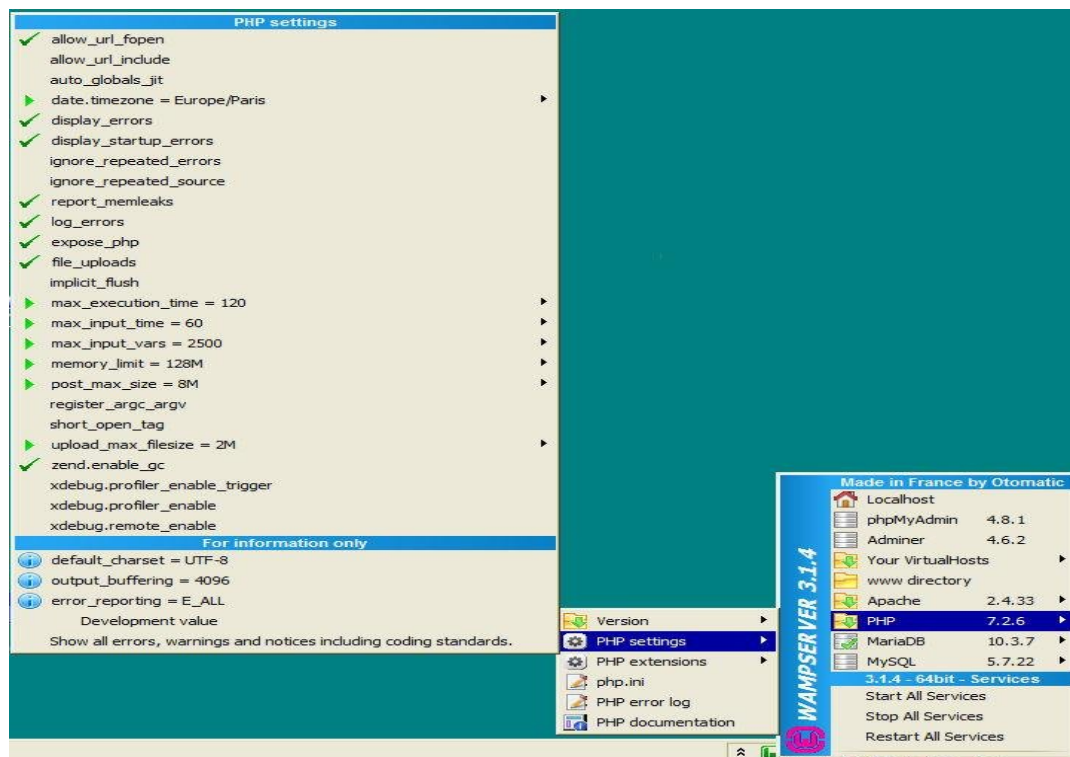
- Δημιουργία / διαγραφή βάσης δεδομένων
- Δημιουργία / διαγραφή / μετατροπή πίνακα δεδομένων
- Δημιουργία / διαγραφή / επεξεργασία στοιχείου πίνακα
- Εισαγωγή δεδομένων από CSV και SQL
- Εξαγωγή δεδομένων σε διάφορες μορφές (PDF,XML, LATEX κ.α.)
- Δυνατότητα διαχείρισης πολλών server
- Δημιουργία γραφημάτων της κατάστασης της βάσης δεδομένων
- Δημιουργία γραφημάτων συσχέτισης πινάκων

Αυτά είναι μόνο μερικές από τις δυνατότητες που έχει ο χρήστης του phpMyAdmin ως διαχειριστής. Τέλος, παρέχεται φυσικά η δυνατότητα εκτέλεσης SQL εντολών κώδικα. Στην εικόνα 1 βλέπουμε το περιβάλλον χρήσης του εργαλείου.



Εικόνα 4: περιβάλλον phpMyAdmin

Τέλος, η πλατφόρμα Wamp Server παρέχεται δωρεάν σε διάφορες εκδόσεις ανάλογα τον τύπο και τη έκδοση του λειτουργικού συστήματος που τρέχει το μηχάνημα μας ως εξυπηρετητής. Εμείς επιλέξαμε την έκδοση των 64bit για Windows 10 που συμβαδίζει με το σύστημά μας και την εγκαταστήσαμε τοπικά (localhost). Παρακάτω [Εικόνα 2] φαίνονται μερικές από τις δυνατότητες για ρυθμίσεις που μας δίνει το wamp.



Εικόνα 5: Wamp Server

2.1.2 Bootstrap 4

Πρόκειται για μια βιβλιοθήκη ή καλύτερα ένα εργαλείο για την κατασκευή ιστοσελίδων με χρήση των γλωσσών HTML, CSS, Javascript, κύριο χαρακτηριστικό του οποίου είναι η δημιουργία προσαρμοζόμενων αυτόματα εφαρμογών στο μέγεθος της οθόνης του κάθε χρήστη (responsive). Αυτό επιτυγχάνεται με τις διάφορες κλάσεις, μεταβλητές καθώς και στο σύστημα πλέγματος το οποίο διαθέτει (grid system). Απαραίτητη για την λειτουργία όλων αυτών είναι η χρήση της javascript και πιο συγκεκριμένα της JQuery την οποία πρέπει να συμπεριλάβουμε στο πρόγραμμά μας. Η εγκατάσταση είναι αρκετά εύκολη, αρκεί να κατεβάσουμε τα απαραίτητα αρχεία από το επίσημο site της bootstrap και να τα ενσωματώσουμε σε κάθε HTML αρχείο μας όπως φαίνεται στην [εικόνα 4, 5, 6].

Τι είναι όμως όλα αυτά; Αρχικά HTML (Hypertext Markup Language) είναι η γλώσσα με την οποία κατασκευάζουμε αρχεία τα οποία θέλουμε να τα εμφανίσουμε σε έναν περιηγητή (Web browser) και μπορεί να συνδυαστεί με τις γλώσσες CSS και Javascript. Με τις κατάλληλες εντολές δημιουργούμε λοιπόν την δομή μιας ιστοσελίδας, δηλαδή που θα εμφανίζονται εικόνες, κείμενο, επικεφαλίδες κ.α.. Με τη χρήση της CSS προσθέτουμε το πώς θέλουμε όλα τα παραπάνω να εμφανίζονται, δηλαδή μπορούμε να επιλέξουμε χρώματα γραμματοσειρών, φόντο, αποστάσεις μεταξύ στοιχείων και πολλά ακόμα δίνοντας την δυνατότητα να δημιουργήσουμε ένα πρωτότυπο και μοναδικό site. Τέλος η Javascript μας βοηθάει στο να θέσουμε ορισμένες συμπεριφορές που θα θέλαμε να εκτελεί ο ιστότοπος μας, ένα απτό παράδειγμα είναι η εμφάνιση ενός pop-up κουτιού μετά από το πάτημα ενός κουμπιού.

Παρακάτω βλέπουμε πώς να συμπεριλάβουμε τα απαραίτητα αρχεία. Τα αρχεία αυτά είναι το bootstrap.css και το bootstrap.js επίσης παρέχεται και CDN link για τα ίδια αρχεία.

Για να ενεργοποιήσουμε την προσαρμογή στις οθόνες(responsive) προσθέτουμε την παρακάτω γραμμή στο πεδίο <head> στα HTML αρχεία μας:

```
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

Εικόνα 6: bootstrap 4 responsive tag

Στο ίδιο πεδίο μετά προσθέτουμε το css αρχείο:

```
<!-- Bootstrap -->
<link rel="stylesheet" type="text/css" href="bootstrap.min.css">
```

Εικόνα 7: Εισαγωγή bootstrap css

Και τέλος, πριν κλείσουμε το πεδίο <body> στα HTML αρχεία μας προσθέτουμε τα παρακάτω:

```
<!-- =====>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
<script type="text/javascript" src="chartjs/js/tether.min.js"></script>
<script type="text/javascript" src="chartjs/js/bootstrap.min.js"></script>
```

Εικόνα 8: Εισαγωγή bootstrap js, JQuery

Στον ισότοπο του bootstrap θα βρούμε πληθώρα παραδειγμάτων και επεξηγήσεων που κάνουν πιο εύκολη τη χρήση από τους προγραμματιστές, επίσης υπάρχουν και έτοιμα θέματα τα οποία ωστόσο είναι επί πληρωμή. Παρακάτω [Πίνακας 2] βλέπουμε μια επεξήγηση του bootstrap grid-system που θα μας βοηθήσει να καταλάβουμε αργότερα τη δομή και τον κώδικα των HTML αρχείων που κατασκευάστηκαν στην παρούσα πτυχιακή εργασία.

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

Πίνακας 2: Bootstrap 4 grid system

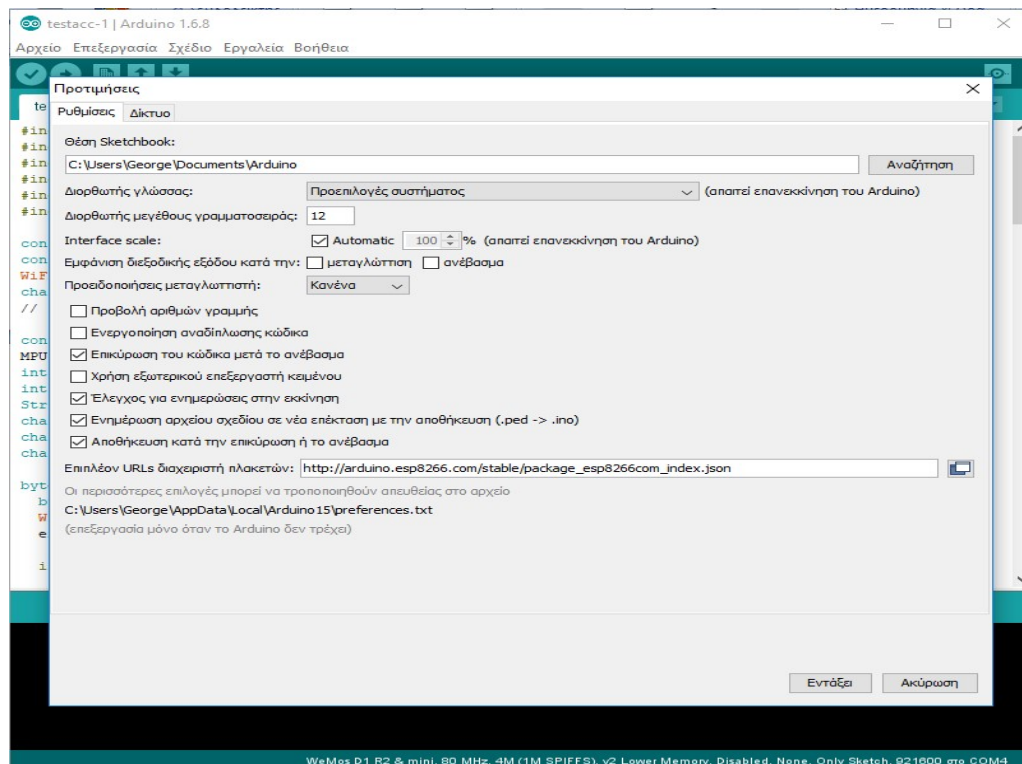
2.1.3 Arduino IDE

Εφόσον στην παρούσα πτυχιακή θα χρησιμοποιήσουμε και μικροεπεξεργαστή (development board) για τη συλλογή δεδομένων, θα πρέπει παράλληλα να χρησιμοποιήσουμε και ένα κατάλληλο πρόγραμμα που θα μας δώσει τη δυνατότητα να προγραμματίσουμε αυτό το μικροεπεξεργαστή. Αυτό το πρόγραμμα είναι το Arduino IDE (integrated development environment), πρόκειται για μία εφαρμογή που είναι γραμμένη σε Java και έχει εκδόσεις για όλα τα λειτουργικά συστήματα (Windows, Linux, macOS), και υποστηρίζει όλους τους μικροεπεξεργαστές τύπου Arduino αλλά και πολλούς άλλους με τη χρήση επιπλέον βιβλιοθηκών. Μέσα από αυτό το

περιβάλλον ο χρήστης μπορεί να ανεβάσει το δικό του πρόγραμμα στον δικό του μικροεπεξεργαστή χρησιμοποιώντας τις υποστηριζόμενες γλώσσες C/C++ καθώς ο κώδικας περνάει από τους compilers avr-gcc και avr-g++ ώστε να μετατραπεί σε γλώσσα μηχανής και να μπορεί να εκτελεστεί από τον μικροεπεξεργαστή.

Ωστόσο για να χρησιμοποιήσουμε εμείς το συγκεκριμένο λογισμικό χιάστηκε να κατεβάσουμε και να εγκαταστήσουμε κάποια επιπλέον πακέτα, καθώς το Wemos d1 r2 (development board) στο οποίο βασιστήκαμε δεν ήταν προ-εγκατεστημένο στο λογισμικό για άμεση χρήση. Έτσι ακολουθήσαμε τα παρακάτω βήματα.

- Εγκατάσταση της τελευταίας έκδοσης του Arduino IDE από την επίσημη ιστοσελίδα.
- Αφού εγκαταστήσουμε το λογισμικό, προσθέτουμε τη βιβλιοθήκη για το Wemos D1 R2. Πηγαίνοντας στο «Αρχείο» -> «Προτιμήσεις» και εισάγοντας στο πλαίσιο «Επιπλέον URLs διαχειριστή πλακετών», το url: http://arduino.esp8266.com/stable/package_esp8266com_index.json

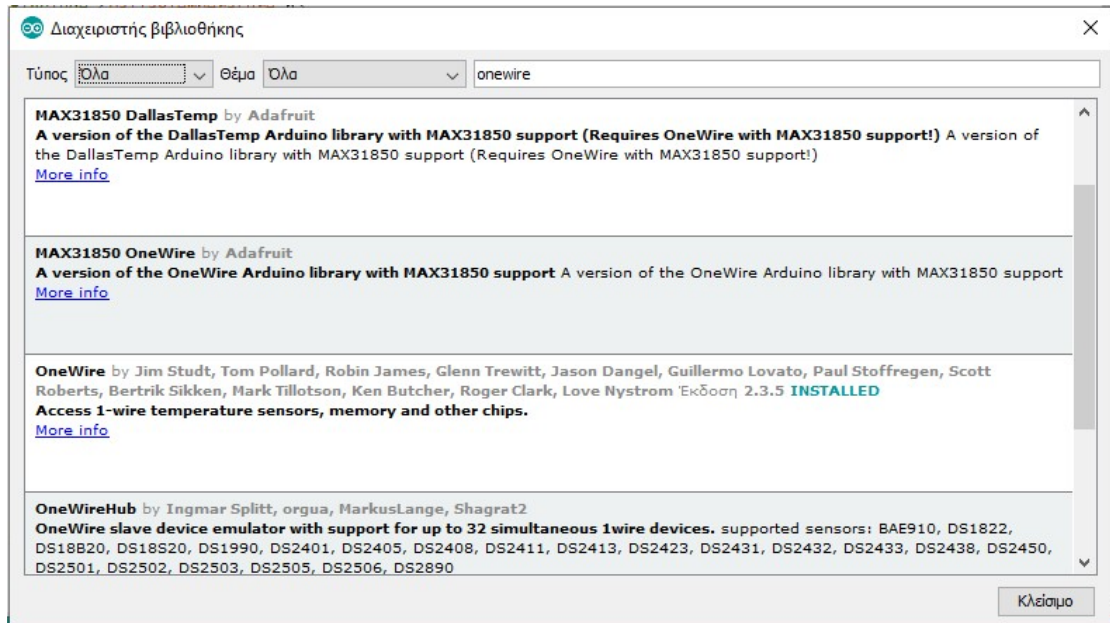


Εικόνα 9: Url βιβλιοθήκης

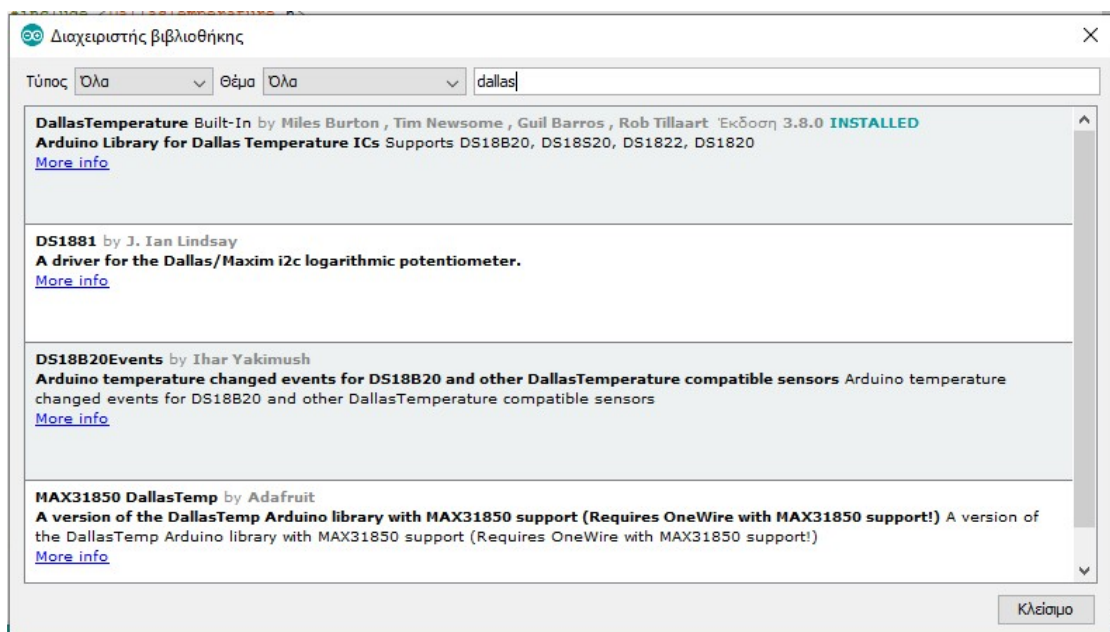
- Στη συνέχεια πάμε στο «Εργαλεία > Πλακέτα > Διαχειριστής Πλακετών» και στο παράθυρο που εμφανίζεται βρίσκουμε τη βιβλιοθήκη “esp8266” και πατάμε εγκατάσταση.
- Τέλος, επιλέγουμε την πλακέτα και είμαστε έτοιμοι να στη χρησιμοποιήσουμε «Εργαλεία > Πλακέτες > Wemos D1 R2». Στη συνέχεια για να χρησιμοποιήσουμε τον αισθητήρα θερμοκρασίας ο οποίος είναι ο DS18B20 θα

πρέπει να εγκαταστήσουμε τις απαραίτητες βιβλιοθήκες. Την βιβλιοθήκη OneWire by Jim Studt και την DallasTemperature by Miles Burton.

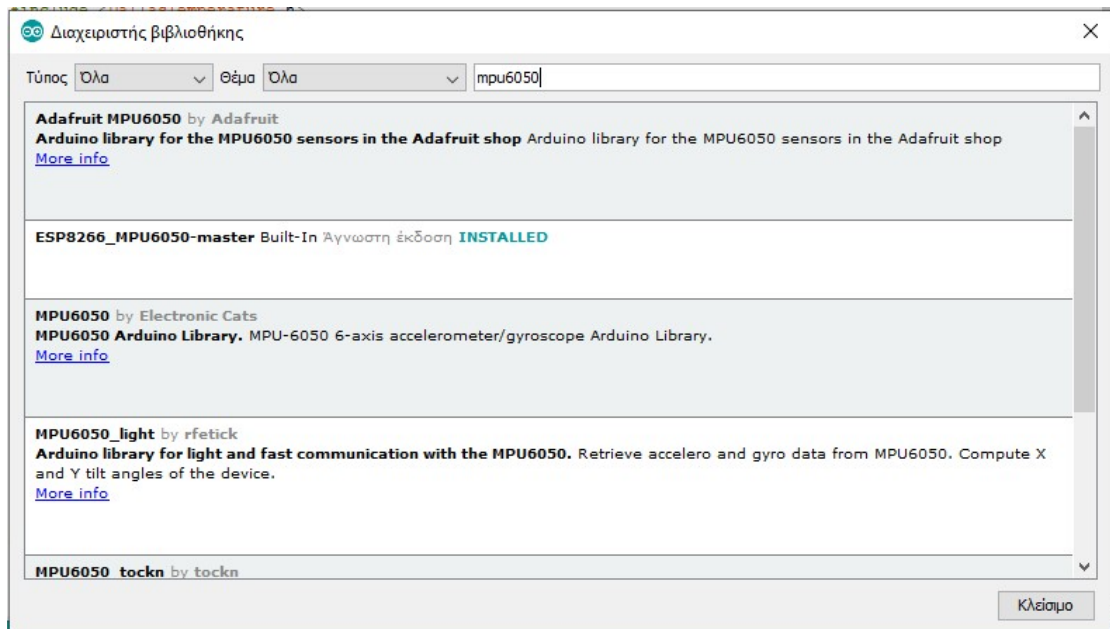
- Πηγαίνουμε «Σχέδιο > Συμπερίληψη Βιβλιοθήκης > Διαχείριση Βιβλιοθηκών» και πληκτρολογούμε στην αναζήτηση «onewire».
- Πατάμε εγκατάσταση στην βιβλιοθήκη “OneWire” by Jim Studt όπως φαίνεται στην παρακάτω εικόνα (η ένδειξη INSTALLED θα εμφανιστεί μετά την εγκατάσταση).



- Με τον ίδιο τρόπο πληκτρολογούμε στην αναζήτηση βιβλιοθήκης τη λέξη «dallas» και πατάμε εγκατάσταση στη βιβλιοθήκη “DallasTemperature” by Miles Burton, όπως φαίνεται στην παρακάτω εικόνα.



- Στη συνέχεια εγκαθιστούμε την βιβλιοθήκη για τον δεύτερο αισθητήρα που θα χρησιμοποιήσουμε, πληκτρολογώντας στην αναζήτηση βιβλιοθήκης “mpu6050” και πατάμε εγκατάσταση στην βιβλιοθήκη “ESP8299_MPU6050-master” όπως φαίνεται παρακάτω.



2.2 Υλικό που χρησιμοποιήθηκε

Όπως αναφέραμε και πιο πάνω, τα δεδομένα θερμοκρασίας και επιτάχυνσης που θα εμφανίζονται στην εφαρμογή μας, τα λαμβάνουμε με συγκεκριμένους αισθητήρες που είναι προσαρμοσμένοι στην πλακέτα μας (Wemos D1 R2). Αυτά λοιπόν αποτελούν το υλικό κομμάτι της πτυχιακής εργασίας και παρακάτω θα αναλύσουμε μερικά από τα χαρακτηριστικά τους.

2.2.1 Πλακέτα Ανάπτυξης (development board)

Οι πλακέτες ανάπτυξης δεν είναι τίποτα παραπάνω από μία απλοποιημένη μορφή ενός ηλεκτρονικού υπολογιστή, δηλαδή μία τυπωμένη πλακέτα κυκλωμάτων (μητρική) στην οποία είναι ενσωματωμένος ένας μικροεπεξεργαστής μαζί με τα απολύτως

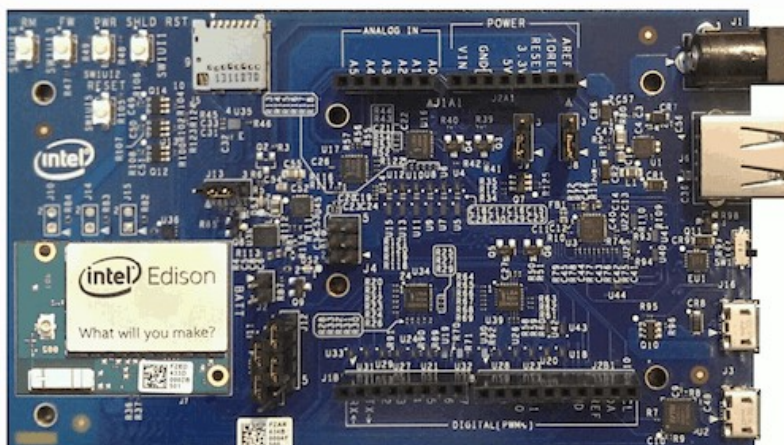
απαραίτητα για την λειτουργία του όπως μνήμη (RAM), υποδοχή παροχής ρεύματος, θύρα εισόδου / εξόδου δεδομένων (USB, microUSB κτλ.).

Ο λόγος ύπαρξής τους τα πρώτα χρόνια ήταν κατά κύριο λόγο για τον έλεγχο (testing) και την εκμάθηση ενός καινούριου μικροεπεξεργαστή από τους προγραμματιστές πριν τη μαζική παραγωγή αυτού και την ενσωμάτωσή του σε ολοκληρωμένα υπολογιστικά συστήματα. Αυτό γινόταν κυρίως για τη μείωση του κόστους καθώς και για εξοικονόμηση χρόνου. Ωστόσο καθώς η τεχνολογία έχει προχωρήσει ραγδαία από τότε, κάτι τέτοιο πλέον δεν έχει νόημα.

Όμως, χάρη στο χαμηλό τους κόστος και στην απλότητά τους έγιναν ευρέως γνωστά σε ερασιτέχνες καθώς προσέφεραν το κατάλληλο περιβάλλον για εκμάθηση των τεχνολογιών που διέπουν τον κλάδο. Έτσι πολλές εταιρίες μέχρι και σήμερα συνεχίζουν να κατασκευάζουν τέτοιου είδους πλακέτες, μερικές από τις πιο γνωστές θα παρουσιάσουμε παρακάτω.

Intel Edison

Πρόκειται για μία από τις πιο ισχυρές πλακέτες ανάπτυξης κατασκευασμένη από την Intel και είχε ως σκοπό να εξυπηρετήσει εφαρμογές υψηλού επιπέδου.



Αργότερα η εταιρία κατασκεύασε την πλακέτα ανάπτυξης που βλέπουμε παραπάνω η οποία είναι συμβατή με Arduino Uno πάνω στην οποία βρίσκεται προσαρμοσμένο το Intel Edison chip. Τα χαρακτηριστικά του οποίου είναι τα εξής:

- Διπύρηνος επεξεργαστής Intel Atom (Z34XX) στα 500MHz.
- 1 GB DDR RAM.
- 4 GB eMMC αποθηκευτικό χώρο.
- Wi-Fi IEEE 802.11

- Bluetooth 4.0
- 14 ψηφιακές εισόδους / εξόδους.
- 6 αναλογικές εισόδους.
- 2 micro USB θύρες.
- 1 USB θύρα.
- Micro SD Card θύρα.
- Υποδοχή ρεύματος DC (7- 15 VDC).

Raspberry Pi

Αναμφίβολα μία από τις πιο γνωστές και πιο ισχυρές πλατφόρμες για αναπτυξιακές εφαρμογές είναι αυτή του Raspberry Pi. Πρόκειται για μία σειρά υπολογιστών μίας πλακέτας (single board computer), καθώς διαθέτουν και το δικό τους λειτουργικό σύστημα, εννέα διαφορετικών εκδοχών. Οι συγκεκριμένες πλακέτες κατασκευάζονται από την Raspberry Pi Foundation και έχει ως κύριο στόχο στην εκμάθηση της επιστήμης των υπολογιστών στα σχολεία. Σε αυτή τη σειρά ξεχωρίζουν το πιο ισχυρό Raspberry Pi 4 (Εικόνα 8) και το πιο μικρό και οικονομικό Raspberry Pi Zero W (Εικόνα 9), τα χαρακτηριστικά των οποίων αναφέρουμε παρακάτω.



Εικόνα 10: Raspberry Pi 4

- Broadcom BCM2711 Quad core Cortex-A72 64-bit 1.5GHz.
- Από 2 έως 8 GB LPDDR4-3200 SDRAM.
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE.
- Gigabit Ethernet.
- 2 USB 3.0 θήρες και 2 USB 2.0 θήρες.

- 40 pin GPIO header.
- 2 micro-HDMI θύρες.
- 2-lane MIPI DSI display θύρα.
- 2-lane MIPI CSI camera θύρα.
- Θύρα audio.
- Micro-SD card θύρα.
- 5V DC από τη θύρα USB-C.



Εικόνα 11: Raspberry Pi Zero W

- 802.11 b/g/n wireless LAN.
- Bluetooth 4.1
- 1GHz, single-core CPU.
- 512MB RAM.
- Mini HDMI.
- 2 Micro USB / power θύρες.
- Micro SD card θύρα.
- 40-pin header.
- Θύρα CSI κάμερας.

Arduino Uno

Είναι ίσως μία από τις πιο διαδεδομένες πλακέτες ανάπτυξης. Κατασκευάζεται από την Arduino.cc και αποτελεί ένα open-source (ανοικτού κώδικα), τόσο υλικού όσο και λογισμικού, μικροελεγκτή βασισμένο στον μικροεπεξεργαστή ATmega328P και είναι ίσως από τα πρώτα συστήματα που χρησιμοποιήθηκαν για το σκοπό της εκμάθησης της επιστήμης των υπολογιστών. Διαθέτει τεράστια υποστήριξη σε βιβλιοθήκες και λογισμικό γενικότερα αλλά και σε υλικό κάτι που είναι ένα από τα μεγάλα πλεονεκτήματά του έναντι στις άλλες πλακέτες που αναφέραμε.



- ATmega328P microchip 8-bit 16MHz.
- 5V Operating Voltage.
- Ρεύμα εισόδου 7 – 20V.
- 14 pins ψηφιακές εισόδους / εξόδους.
- 6 pins αναλογικές εισόδους / εξόδους.
- 32 KB αποθηκευτικό χώρο (Flash memory).
- 2 KB SRAM.
- 1 KB EEPROM.

Wemos D1 R2

Όπως αναφέραμε και παραπάνω ο μικροελεγκτής Arduino είναι open source γεγονός που δίνει τη δυνατότητα δε άλλες εταιρίες να δημιουργήσουν τις δικές τους πλακέτες βασιζόμενες στην ήδη υπάρχουσα αρχιτεκτονική του Arduino Uno. Ένα τέτοιο παράδειγμα είναι και το Wemos D1 R2 το οποίο είναι κινεζικής κατασκευής και βασισμένο στο Arduino Uno και είναι πλήρως συμβατό με αυτό τόσο σε επίπεδο υλικού όσο και λογισμικού, ενώ παράλληλα αποτελεί και μια αναβάθμιση σε σύγκριση με το Uno προσφέροντας κάποια πολύ σημαντικά επιπλέον χαρακτηριστικά, τα οποία αναφέρουμε παρακάτω.



- Μικροεπεξεργαστής ESP-8266EX 80MHz/160MHz.
- Wi-Fi IEEE 802.11
- Ρεύμα λειτουργίας 3.3V
 - 11 pins ψηφιακές εισόδους / εξόδους.
- 1 pin αναλογικές εισόδους / εξόδους.
- Μνήμη 4 MB.
- 1 Micro USB θύρα.
- Σύνδεση ρεύματος 9-24V.
- Συμβατό με Arduino IDE, nodeMCU.

Στην παρούσα εργασία εμείς επιλέξαμε να ασχοληθούμε με την συγκεκριμένη πλακέτα τόσο για την συμβατότητα της με το Arduino Uno όσο και για την τιμή της η οποία είναι κατά πολύ χαμηλότερη από τις υπόλοιπες πλακέτες ανάπτυξης, με σχεδόν το ένα δέκατο της τιμής ενός Arduino, χωρίς ωστόσο να αποτελεί πρόβλημα η διαθεσιμότητα του προϊόντος καθώς μπορούμε να το βρούμε σε αρκετά καταστήματα.

2.2.2 Αισθητήρες

Στη προσπάθεια του ανθρώπου να κατανοήσει, να μελετήσει και να αντλήσει πληροφορίες σχετικά με τον περιβάλλον χώρο, τις ιδιότητες που τον διέπει καθώς και στην εξέλιξη νέων τεχνολογιών σημαντικό ρόλο έπαιξε η δημιουργία αισθητήρων. Οι αισθητήρες είναι συσκευές οι οποίες παράγουν σαν έξοδο ρεύμα ή κάποιο άλλο σήμα εξόδου με βάση τις αλλαγές στο περιβάλλον τις οποίες καλούνται να μετρήσουν. Ο τύπος και η ποσότητα του σήματος εξόδου ενός αισθητήρα καθορίζει και τον τύπο του αισθητήρα. Σε γενικές γραμμές οι αισθητήρες χωρίζονται σε δύο κατηγορίες, στους παθητικούς (Passive sensors) και στους ενεργητικούς (Active sensors).

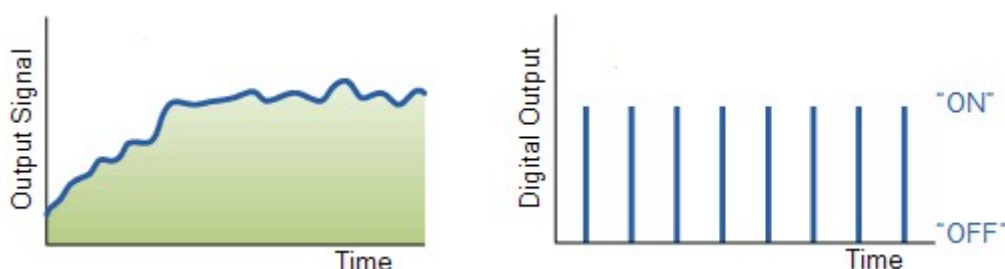
Ως ενεργητικούς αισθητήρες χαρακτηρίζουμε αυτούς οι οποίοι χρειάζονται/προσθέτουν ένα πρόσθετο εξωτερικό σήμα για να λειτουργήσουν και να παράγουν έξοδο το οποίο σήμα δημιουργείται από τους ίδιους. Ένα χαρακτηριστικό παράδειγμα τέτοιου αισθητήρα είναι ένα ραντάρ ή ένα sonar τα οποία για να μετρήσουν την απόσταση από τα αντικείμενα στο χώρο εκπέμπουν ραδιοκύματα ή ηχητικά κύματα και χρησιμοποιούν την αντανάκλαση τους για να προσδιορίσουν την απόσταση.

Αντίθετα οι παθητικοί αισθητήρες δεν χρειάζονται κάποια πρόσθετη πηγή ή ενέργεια για να παράξουν αποτέλεσμα, καθώς η αλλαγή της τιμής την οποία μετράνε σημαίνει και αυτόματη αλλαγή των χαρακτηριστικών τους που κατ'επέκταση οδηγεί στην αλλαγή της τιμής εξόδου. Ένα χαρακτηριστικό παράδειγμα είναι το θερμοστοιχείο (thermocouple) το οποίο παράγει ρεύμα όταν εκτίθεται σε θερμότητα.

Ωστόσο υπάρχει και ακόμα μια κατηγορία με βάση την οποία μπορούν να διαχωριστούν οι αισθητήρες και αυτή είναι ανάλογα με τον εάν είναι αναλογικοί ή ψηφιακοί και ο διαχωρισμός τους γίνεται με βάση το είδος του σήματος / τάσης (voltage) εξόδου.

Οι αναλογικοί αισθητήρες παράγουν σήματα εξόδου τα οποία είναι συνεχή σε σχέση με τον χρόνο και συνδέονται άμεσα με τα σήματα εισόδου. Χαρακτηριστικό παράδειγμα αναλογικού αισθητήρα είναι της θερμοκρασίας. Τα σήματα που παράγονται από τέτοιου είδους αισθητήρες αλλάζουν ομαλά και είναι συνεχή στο χρόνο, αυτά τα σήματα συνήθως είναι πολύ μικρά σε ισχύ έτσι χρειάζονται κάποιου είδους ενισχυτή. Ωστόσο αυτό το είδος αισθητήρων είναι συνήθως αργοί στις εναλλαγές και λιγότερο ακριβείς στις μετρήσεις.

Αντίθετα οι ψηφιακοί αισθητήρες παράγουν διακριτά σήματα εξόδου της λογικής 0 και 1 ή On και Off, είναι πιο άμεσοι στις μετρήσεις καθώς και πιο ακριβείς. Η ακρίβεια ωστόσο είναι ανάλογη με τον αριθμό των bits που χρησιμοποιούνται για να αντιπροσωπεύσουν μια έξοδο, όσο περισσότερα είναι τα bits τόσο μεγαλύτερη θα είναι η ακρίβεια. Τέλος να αναφέρουμε ότι τα αναλογικά σήματα μπορούν εύκολα να μετατραπούν σε ψηφιακά με τη χρήση ενός μετατροπέα αναλογικού σε ψηφιακού (analogue-to-digital converter ή ADC).

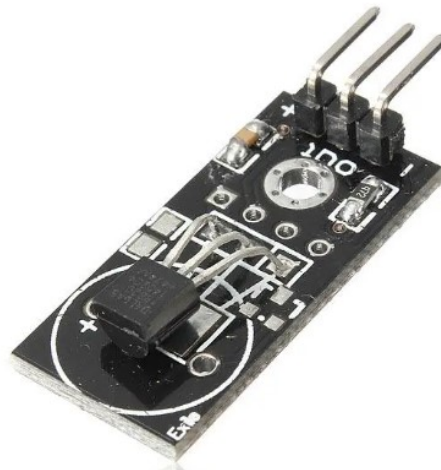


Σχήμα 3 : αναλογικό σήμα – ψηφιακό σήμα

Αισθητήρας Θερμοκρασίας

Η θερμοκρασία είναι ίσως μια από τις πιο καταμετρημένες τιμές γεγονός που πηγάζει από τη σημαντικότητα που έχει σε όλους τους τομείς των επιστημών και όχι μόνο. Αν αναλογιστούμε ότι διάφορες διαδικασίες είτε αυτές είναι βιολογικές, χημικές ή ακόμα και ηλεκτρονικές, επηρεάζονται σε μεγάλο βαθμό από τη θερμοκρασία, η καταμέτρηση της είναι καίριας σημασίας. Δεν είναι τυχαίο λοιπόν το γεγονός ότι υπάρχουν παραπάνω από ένας τύποι αισθητήρων θερμοκρασίας συμπεριλαμβανομένων των θερμοστοιχείων (thermocouple), RTD (Resistance temperature detector), thermistors (παρόμοιοι με τους RTD αλλά από διαφορετικά υλικά), ημιαγωγοί (semiconductor sensors).

Στην παρούσα εργασία εμείς θα χρησιμοποιήσουμε τον αισθητήρα DS18B20. Πρόκειται για έναν ψηφιακό αισθητήρα τεχνολογίας one-wire, που σημαίνει ότι χρειάζεται μόνο μία γραμμή δεδομένων και τη γείωση για να επικοινωνήσει με τη πλακέτα μας έχοντας έτσι τη δυνατότητα να αντλήσει ενέργεια από το τη γραμμή δεδομένων (parasite mode) χωρίς αυτό όμως να είναι απαραίτητο καθώς διαθέτει και γραμμή για ξεχωριστή πηγή ρεύματος.



- Επικοινωνία one-wire.
- Πηγή ρεύματος: 3.0V – 5.5V.
- Θερμοκρασίες λειτουργίας και καταγραφής: -55°C έως +125 °C.
- Ακρίβεια +/- 0.5 °C (από -10 °C έως 85 °C).

Αισθητήρας Επιτάχυνσης (Accelerometer)

Ο δεύτερος αισθητήρας τον οποίο προσθήσαμε στο σύστημά μας είναι ο MPU-6050. Πρόκειται για έναν αισθητήρα ο οποίος έχει ενσωματωμένα ένα επιταχυνσιόμετρο τριών διαστάσεων και ένα γυροσκόπιο επίσης τριών διαστάσεων.



Το γυροσκόπιο μετράει την ταχύτητα περιστροφής ή αλλιώς τον ρυθμό εναλλαγής της κλήσης στους άξονες X, Y, Z με τη χρήση της τεχνολογίας MEMS και οι τιμές που μας δίνει για έξοδο είναι σε μοίρες ανά δευτερόλεπτο για καθέναν από τους άξονες. Από την άλλη μεριά το επιταχυνσιόμετρο μετράει την επιτάχυνση της βαρύτητας (g) σε κάθε άξονα.

Η ευαισθησία και το εύρος τιμών στις μετρήσεις του αισθητήρα είναι πλήρως ρυθμιζόμενες και προγραμματιζόμενες σε κατηγορίες όπως φαίνονται στον παρακάτω πίνακα που μπορούμε να βρούμε στο φύλλο δεδομένων του αισθητήρα.

Gyro Full Scale Range	Gyro Sensitivity	Gyro Rate Noise	Accel Full Scale Range	Accel Sensitivity
(°/sec)	(LSB/°/sec)	dps/√Hz	(g)	LSB/g
±250	131	0.005	±2	16384
±500	65.5	0.005	±4	8192
±1000	32.8	0.005	±8	4096
±2000	16.4	0.005	±16	2048

Πίνακας 3 : Κατηγορίες εύρους και ευαισθησίας MPU6050

Τέλος να αναφέρουμε ότι έχει τη δυνατότητα να λαμβάνει μετρήσεις για κάθε άξονα και για κάθε αισθητήρα ταυτόχρονα, δίνοντας έτσι μεγάλες δυνατότητες στους προγραμματιστές, δεν είναι τυχαίο άλλωστε ότι τέτοιου είδους αισθητήρες χρησιμοποιούνται ευρέως σε έξυπνες συσκευές όπως κινητά τηλέφωνα, ταμπλέτες και ρολόγια.

3 - Περιγραφή Συστήματος

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τον τρόπο με τον οποίο συνδέσαμε όλα τα κομμάτια υλικού και λογισμικού ώστε να παραχθεί το τελικό αποτέλεσμα της διαδικτυακής μας εφαρμογής. Πιο αναλυτικά θα δούμε την αρχιτεκτονική του συστήματος μέσα από διαγράμματα, στη συνέχεια θα αναλύσουμε τη χρησιμότητα και τη λειτουργία κάθε αρχείου του κώδικα καθώς και μερικά κομμάτια κώδικα και των γλωσσών προγραμματισμού που χρησιμοποιήθηκαν καθώς και τα πρωτόκολλα επικοινωνίας.

3.1 Συνδεσμολογία Υλικού

Στην ενότητα αυτή θα δούμε τη συνδεσμολογία του υλικού δηλαδή τον τρόπο με τον οποίο οι επιμέρους αισθητήρες συνδέονται στην πλακέτα ανάπτυξης ώστε να λειτουργήσουν ομαλά χωρίς προβλήματα αποφεύγοντας βλάβη του υλικού ή ακόμα και καταστροφή του.

Στον παρακάτω πίνακα βλέπουμε την αντιστοίχιση των θυρών μιας πλακέτας Wemos D1 r2 με αυτές ενός Arduino Uno. Ο συγκεκριμένος πίνακας μας βοήθησε να καταλάβουμε πως η πλακέτα μας χρησιμοποιεί τις θύρες και με ποία ονομασία σε σύγκριση με το πιο γνωστό Arduino βοηθώντας μας στον προγραμματισμό.

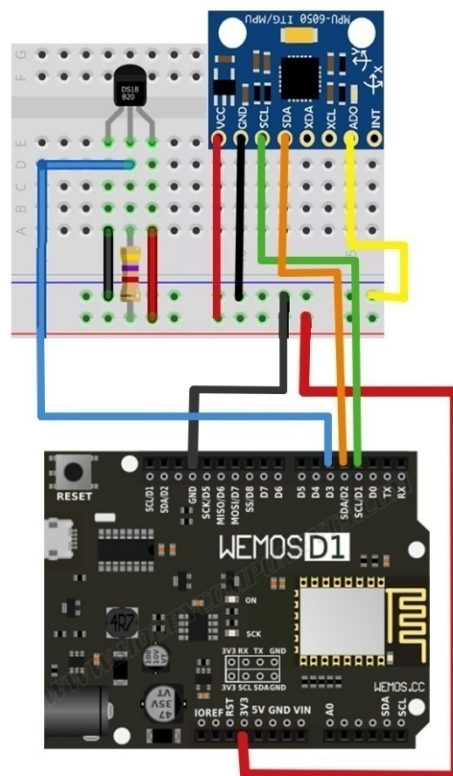
Arduino-UNO		→	WeMos-D1R2	
SCL	I2C: SCL		GPIO05	I2C: SCL
SDA	I2C: SDA	→	GPIO04	I2C: SDA
AREF		→		
GND		→	GND	
GPIO13	SPI: SCK	→	GPIO14	SCK
GPIO12	SPI: MISO	→	GPIO12	MISO
GPIO11	SPI: MOSI	→	GPIO13	MOSI
GPIO10	SPI: SS	→	GPIO15	SS
GPIO9		→	GPIO13	

Arduino-UNO		→	WeMos-D1R2	
GPIO8			GPIO12	
GPIO7		→	GPIO14	
GPIO6		→	GPIO2	
GPIO5		→	GPIO0	
GPIO4		→	GPIO04	
GPIO3		→	GPIO05	
GPIO2		→	GPIO16	
GPIO1	TX	→	GPIO01	TX0
GPIO0	RX	→	GPIO03	RX0

Πίνακας 4 : Διαφορές μεταξύ Arduino UNO και Wemos D1R2.

IOREF		WEMOS D1 R2 NodeMCU V0.9 Compatible	D1	SCL	GPIO5		
IOREF			D2	SDA	GPIO4		
RST			GND				
3V3			D5	SCK	GPIO14		
5V			D6	MISO	GPIO12		
GND			D7	MOSI	GPIO13	Rx0*	10K Pull down
GND			D8	SS	GPIO15	Tx0*	
VIN			D7	MOSI	GPIO13	Rx0*	
			D6	MISO	GPIO12		
A0		D5	SCK	GPIO14			
		D4		GPIO2	Tx1	10K Pull up	Built in
		D3		GPIO0		10K Pull up	
		D2	SDA	GPIO4			
		D1	SLC	GPIO5			
GPIO4	SDA	D0		GPIO16			
GPIO5	SCL	Tx		GPIO1			
		Rx		GPIO3			

Στην παρακάτω εικόνα βλέπουμε τις συνδέσεις που έγιναν ώστε να επικοινωνεί η πλακέτα μας με τους αισθητήρες.



3.2 Σχήμα Βάσης Δεδομένων

Παρακάτω θα δούμε και θα αναλύσουμε το σχήμα της βάσης δεδομένων που χρησιμοποιήθηκε καθώς και τους πίνακες τους οποίους περιέχει. Όπως αναφέραμε και σε προηγούμενο κεφάλαιο η βάση δεδομένων είναι μια MySQL βάση, με την οποία αλληλεπιδρούμε με εντολές SQL, και στη δικιά μας περίπτωση είναι εγκατεστημένη στον τοπικό μας υπολογιστή με τη βοήθεια του Wamp. Στη συνέχεια βλέπουμε κάποιες από τις εντολές:

```
CREATE DATABASE IF NOT EXISTS station DEFAULT CHARSET=utf8
```

- Αυτή η εντολή δημιουργεί τη βάση δεδομένων που θα χρησιμοποιηθεί αργότερα, με όνομα station εφόσον αυτή δεν υπάρχει ήδη.

```
CREATE TABLE IF NOT EXISTS temptest ( ID int NOT NULL AUTO_INCREMENT, Time  
TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP, Temperature int(10) NOT  
NULL, kitID int NOT NULL, PRIMARY KEY (ID), FOREIGN KEY (kitID) REFERENCES kits(ID))
```

- Αυτή η εντολή δημιουργεί τον πίνακα εάν δεν υπάρχει temptest ο οποίος θα περιέχει τα δεδομένα θερμοκρασιών, με τα γνωρίσματα ID το οποίο είναι ένας μοναδικός μη μηδενικός ακέραιος αριθμός που εκχωρείται σε κάθε στοιχείο του πίνακα, Time το οποίο είναι μία τιμή τύπου timestamp (π.χ. 2038-01-19 03:14:07.999999) η οποία δεν είναι μηδενική και παίρνει ως τιμή τη χρονική στιγμή που αποθηκεύεται το στοιχείο στον πίνακα, Temperature το οποίο είναι ένας μη μηδενικός ακέραιος αριθμός, KitID το οποίο είναι ένας ακέραιος μη μηδενικός αριθμός. Τέλος ορίζουμε το ID ως πρωτεύων κλειδί του πίνακα και το kitID ως ξένο κλειδί το οποίο αναφέρεται στο ID του πίνακα kits.

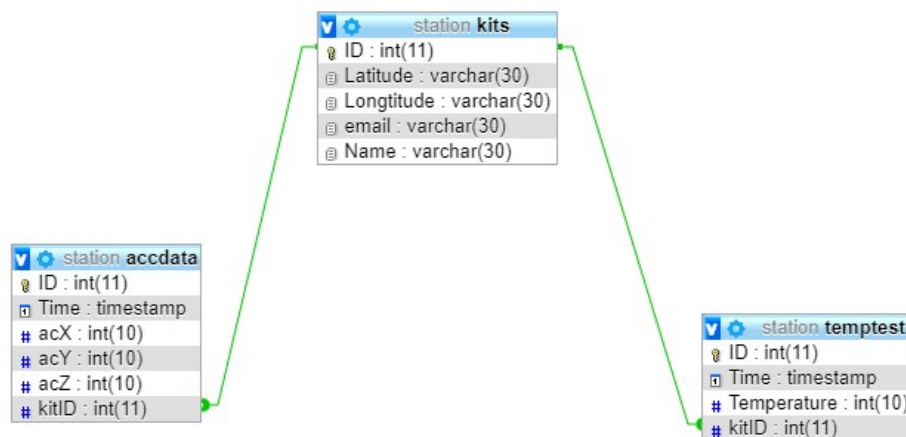
```
CREATE TABLE IF NOT EXISTS accdata ( ID int NOT NULL AUTO_INCREMENT, Time  
TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP, acX int(10) NOT NULL, acY  
int(10) NOT NULL, acZ int(10) NOT NULL, kitID int NOT NULL, PRIMARY KEY (ID),  
FOREIGN KEY (kitID) REFERENCES kits(ID) )
```

- Δημιουργεί τον πίνακα εάν δεν υπάρχει accdata ο οποίος θα περιέχει τα δεδομένα επιτάχυνσης, με τα γνωρίσματα ID το οποίο είναι ένας μοναδικός μη μηδενικός ακέραιος αριθμός που εκχωρείται σε κάθε στοιχείο του πίνακα και αυξάνεται αυτόματα ανάλογα με τις εισαγωγές, Time το οποίο είναι μία τιμή τύπου timestamp (π.χ. 2038-01-19 03:14:07.999999) η οποία δεν είναι μηδενική και παίρνει ως τιμή τη χρονική στιγμή που αποθηκεύεται το στοιχείο στον πίνακα, acX, acY, acZ τα οποία είναι ένας μη μηδενικός ακέραιος αριθμός, KitID το οποίο είναι ένας ακέραιος μη μηδενικός αριθμός. Τέλος ορίζουμε το ID ως πρωτεύων κλειδί του πίνακα και το kitID ως ξένο κλειδί το οποίο αναφέρεται στο ID του πίνακα kits.

```
SELECT      MAX(temptest.Temperature)      AS      maxTemperature,
MIN(temptest.Temperature) AS minTemperature, MAX(accddata.acX) AS maxX,
MIN(accddata.acX) AS minX, MAX(accddata.acY) AS maxY, MIN(accddata.acY) AS minY,
MAX(accddata.acZ) AS maxZ, MIN(accddata.acZ) AS minZ FROM temptest JOIN accdata
ON temptest.kitID=accdata.kitID=1 WHERE UNIX_TIMESTAMP(temptest.Time)*1000 >=
'$from_date' AND UNIX_TIMESTAMP(temptest.Time)*1000 <= '$to_date'
```

- Η συγκεκριμένη εντολή μας εμφανίζει τις μέγιστες και τις ελάχιστες τιμές των Temperature, acX, acY, acZ από τους πίνακες accdata και temptest αντίστοιχα, για το τελευταίο εικοσιτετράωρο.

Σε αυτό το σημείο να τονίσουμε ότι όλες οι εντολές SQL εκτελούνται στο εξυπηρετητή μέσω των κατάλληλων αρχείων PHP ώστε να επιτυγχάνεται ο επιθυμητός αυτοματισμός των διαδικασιών. Αυτό γίνεται εφικτό κατά κύριο λόγο μέσω των συναρτήσεων της PHP mysqli_connect() και mysqli_query() με τις οποίες συνδεόμαστε στον εξυπηρετητή και στη βάση και εκτελούμε το κατάλληλο SQL ερώτημα αντίστοιχα.



Σχήμα 4 : Σχήμα Βάσης δεδομένων MySQL

3.3 Περιγραφή κομματιών λογισμικού

Στην παρούσα ενότητα θα αναλύσουμε όλα τα αρχεία του κώδικα, δηλαδή ποια είναι λειτουργικότητά τους, ποια γλώσσα προγραμματισμού χρησιμοποιήθηκε για τη δημιουργία τους και πως συνδέονται με τα υπόλοιπα αρχεία. Παρακάτω βλέπουμε τα επιμέρους αρχεία:

index.html / indexgr.html : Πρόκειται για ένα αρχείο σε γλώσσα html το οποίο είναι υπεύθυνο για την απεικόνιση της αρχικής μας σελίδας σε ένα πρόγραμμα περιήγησης (web browser). Στην κεφαλίδα του αρχείου συνδέουμε τα αρχεία (βιβλιοθήκες) CSS **bootstrap.min.css** και **customcss.css**, τα οποία μας δίνουν τη δυνατότητα απεικόνισης του περιεχομένου σε μορφή bootstrap. Απαραίτητη προϋπόθεση για τα παραπάνω είναι να συμπεριλάβουμε και τα αρχεία (scripts) javascript **jquery.min.js**, **tether.min.js**, **bootstrap.min.js**. Τέλος για να μπορέσουμε να απεικονίσουμε δεδομένα από τη βάση δεδομένων συμπεριλαμβάνουμε και το αρχείο **homesetup.js**.

homesetup.js : Όπως είδαμε παραπάνω αυτό το αρχείο είναι γραμμένο σε javascript και μας βοηθά στο να φορτώσουμε δεδομένα από τη βάση μας, τα οποία είναι μέγιστες και ελάχιστες τιμές το τελευταίο εικοσιτετράωρο. Αυτό γίνεται ασύγχρονα με χρήση ενός Ajax request το οποίο στέλνει ένα HTTP POST αίτημα στο αρχείο **jsonindex.php** το οποίο μας επιστρέφει τα δεδομένα που χρειαζόμαστε σε μορφή JSON και με την κατάλληλη συνάρτηση τροποποιούμε τον html κώδικα του αρχείου **index.html** ώστε να εμφανίζονται τα δεδομένα στον περιηγητή.

jsonindex.php : Είναι ένα αρχείο γλώσσας PHP το οποίο όπως είδαμε παραπάνω συνδέεται στον εξυπηρετητή και κατ'επέκταση στη βάση δεδομένων και χτυπάει με ένα ερώτημα SQL το οποίο επιστρέφει τις μέγιστες και ελάχιστες τιμές το τελευταίο 24ωρο και τις μετατρέπει σε μορφή JSON.

find-stations.html : Είναι ένα αρχείο HTML το οποίο εμφανίζει στον browser τη σελίδα στην οποία μπορεί ο χρήστης να βρει τους σταθμούς στον χάρτη. Όπως και στο προηγούμενο HTML αρχείο έτσι και σε αυτό συμπεριλάβαμε όλα τα απαραίτητα CSS και Javascript αρχεία που είναι υπεύθυνα για το bootstrap. Ωστόσο προσθέσαμε και τα **mapSetting.js**, **canvasjs.min.js**, **linegraph.js**, **linegraphacc.js** καθώς και το API της Google που είναι απαραίτητο για την απεικόνιση του χάρτη (Google maps).

mapSetting.js : Ένα αρχείο javascript το οποίο είναι υπεύθυνο για την απεικόνιση του χάρτη και των σταθμών σε αυτόν στον περιηγητή τροποποιώντας κατάλληλα το αρχείο **find-stations.html**. Αυτό γίνεται ασύγχρονα με χρήση Ajax και στέλνοντας ένα GET αίτημα στον αρχείο **jsonmarkers.php** το οποίο επιστρέφει όλα τα «kit» σε μορφή JSON. Στη συνέχεια με την εκτέλεση της κατάλληλης συνάρτησης (**google.maps.event.addListener**) δίνουμε τη δυνατότητα στο χρήστη να πατήσει επάνω σε κάθε στίγμα (κίτ) και να δει παραπάνω λεπτομέρειες. Σε αυτό το σημείο να αναφέρουμε ότι οι συναρτήσεις **markerClick()**, **marker_change()**, **marker_load()** τροποποιούν κατάλληλα τον κώδικα του αρχείου **find-stations.html** ώστε να εμφανίζονται στοιχεία του σταθμού τον οποίο επέλεξε ο χρήστης, καθώς και το αρχείο **logo-load.html** το οποίο εμφανίζει τα λογότυπα της θερμοκρασίας και του

επιταχυνσιόμετρο τα οποία με τη σειρά τους όταν ο χρήστης πατήσει σε αυτά καλούν τις συναρτήσεις `temp_btn_form_load()` και `acc_btn_form_load()` αντίστοιχα. Με τις τελευταίες συναρτήσεις φορτώνεται στον κώδικα του `find-stations.html` τα αρχεία **temp-diagram-text.html** και **acc-diagram-text.html** αντίστοιχα.

dokimi.html / dokimiacc.html : Τα συγκεκριμένα αρχεία HTML φορτώνουν και απεικονίζουν τις κατάλληλες δομές (div) ώστε πάνω σε αυτές να εμφανιστούν τα διαγράμματα καθώς και αντίστοιχα συνοδευτικά κείμενα.

linegraph.js / linegraphacc.js : Είναι αρχεία Javascript τα οποία συμπεριλάβαμε στο αρχείο `find-stations.html` και είναι αυτά τα οποία δημιουργούν τα διαγράμματα των θερμοκρασιών και της επιτάχυνσης, καθώς και για την λειτουργία της φόρμας στην οποία ο χρήστης επιλέγει από τότε μέχρι τότε επιθυμεί να δει δεδομένα στα γραφήματα. Η δημιουργία των διαγραμμάτων να σημειώσουμε ότι γίνεται εφικτή με τη χρήση του αρχείου – βιβλιοθήκη **canvasjs.min.js**.

add-stations.html : Είναι ένα αρχείο HTML το οποίο είναι υπεύθυνο για την εμφάνιση τις σελίδας «εισαγωγή αισθητήρα» στον περιηγητή. Η οποία αποτελείται από μία φόρμα στην οποία ο χρήστης συμπληρώνει τα στοιχεία του. Επίσης συμπεριλαμβάνουμε ένα script το οποίο ενημερώνει τον χρήστη με κατάλληλα μηνύματα για τυχών λανθασμένες εισαγωγές.

add-form-val.php : Είναι ένα αρχείο σε γλώσσα PHP το οποίο κάνει τον έλεγχο/καθαρισμό των στοιχείων που εισήγαγε ο χρήστης στη παραπάνω φόρμα και στη συνέχεια εάν αυτά είναι σωστά και έχουν τη μορφή που πρέπει αποθηκεύονται στη βάση δεδομένων στον πίνακα «kits». Τέλος εμφανίζει στον χρήστη το αρχείο **successfully-added.html** το οποίο απλά εμφανίζει στον χρήστη το μήνυμα ότι ο σταθμός έχει προστεθεί με επιτυχία.

about.html : Ένα αρχείο HTML το οποίο εμφανίζει στον χρήστη τη σελίδα «σχετικά», η οποία περιέχει ένα σχετικό κείμενο και εικόνες.

contact.html : Πρόκειται για ένα αρχείο HTML το οποίο εμφανίζει στον browser την σελίδα «επικοινωνία» στην οποία βρίσκονται μία φόρμα και ένας χάρτης με ένα στίγμα να δείχνει το Ιόνιο Πανεπιστήμιο. Εδώ πάλι για να χρησιμοποιήσουμε το χάρτη, συμπεριλάβαμε το script για να κάνουμε χρήση του API της Google για το Google Maps, καθώς και ένα script για την αρχικοποίηση και τη δημιουργία του χάρτη στο κατάλληλο σημείο της σελίδας.

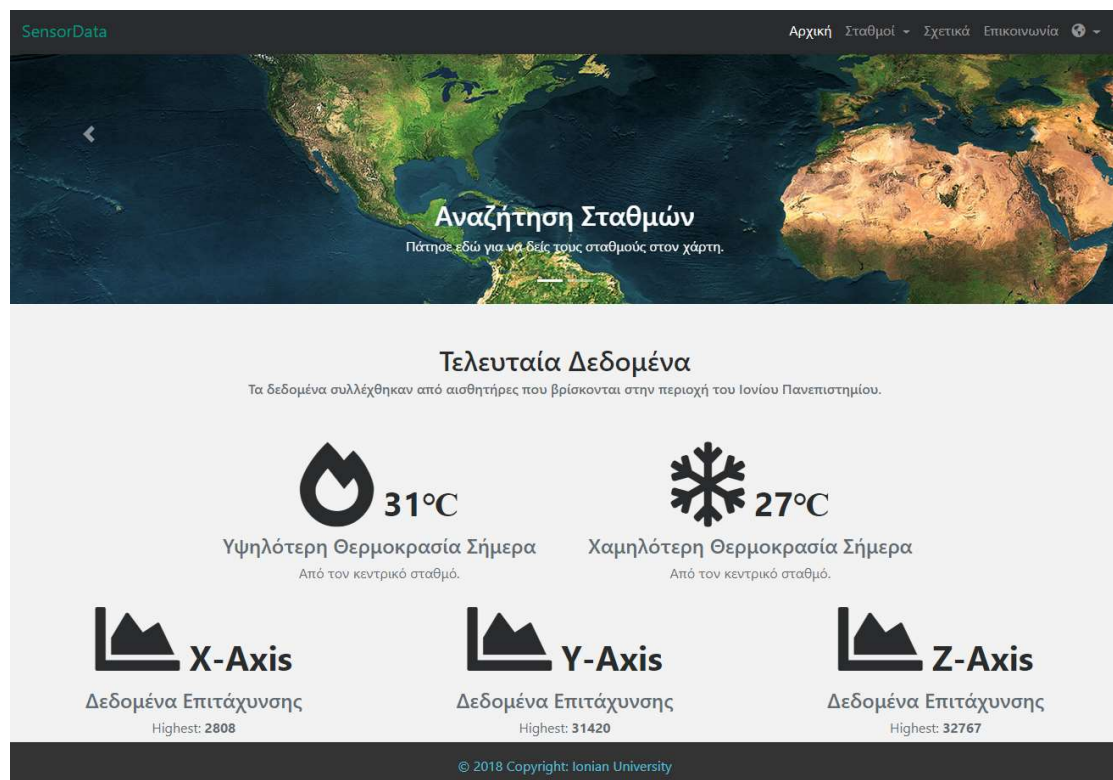
contact-form-val.php : Αρχείο γραμμένο σε PHP το οποίο ελέγχει και «καθαρίζει» τις εισόδους του χρήστη στη φόρμα του προηγούμενου αρχείου με την βοήθεια των συναρτήσεων `trim()`, `stripslashes()`, `htmlspecialchars()`, `mysqli_real_escape_string()` και στη συνέχεια στέλνει ενημερωτικό e-mail στο mail που εισήγαγε ο χρήστης. Τέλος φορτώνει το αρχείο **successfully-send.html** το οποίο εμφανίζει στο χρήστη ένα μήνυμα επιτυχίας αποστολής.

4 - Παρουσίαση Συστήματος

Σε αυτό το κεφάλαιο θα δούμε μέσα από εικόνες την τελική μορφή της διαδικτυακής μας εφαρμογής και τις δυνατότητες τις οποίες έχει ο χρήστης μέσα από αυτή.

Αρχική Σελίδα

Παρακάτω φαίνεται η αρχική σελίδα της εφαρμογής μας.



Ο χρήστης μέσα από αυτή την σελίδα μπορεί να κατευθυνθεί σε οποιαδήποτε άλλη σελίδα της εφαρμογής μας επιθυμεί μέσω του menu το οποίο περιλαμβάνει:

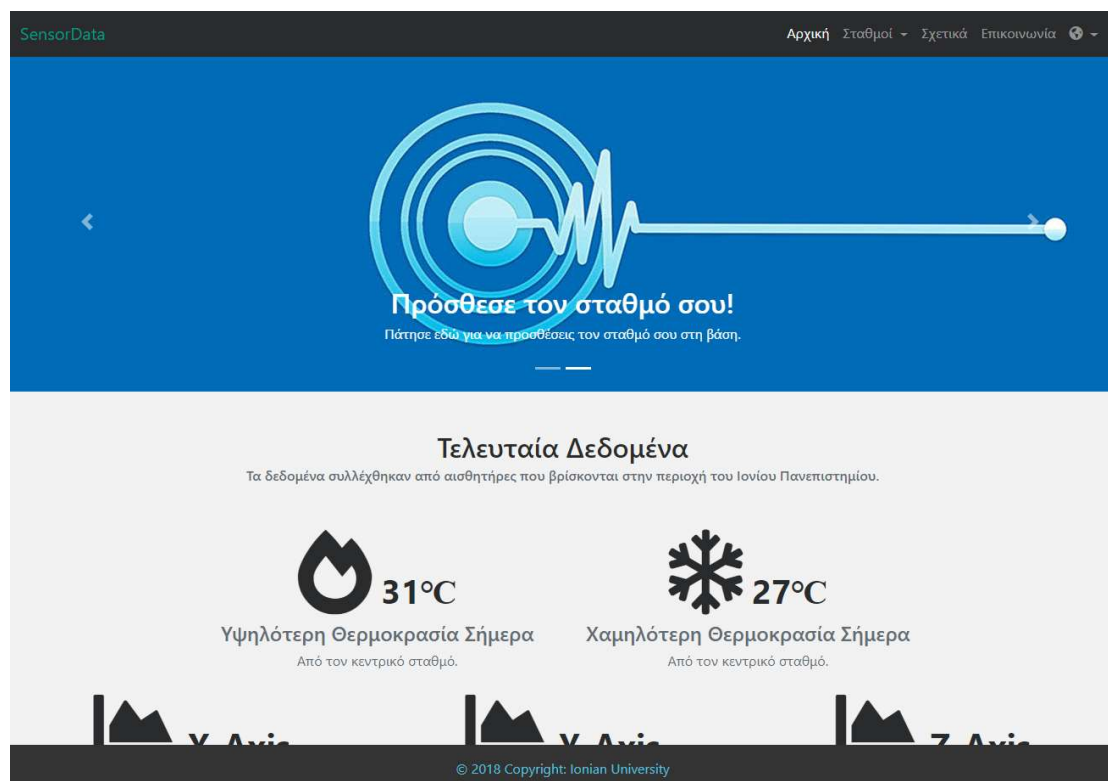
- Αρχική – η παρούσα σελίδα.
- Σταθμοί – με επιπλέον επιλογή.
 - Αναζήτηση Σταθμού

- Προσθήκη Σταθμού
- Σχετικά – Λίγα λόγια για την εφαρμογή.
- Επικοινωνία – Φόρμα επικοινωνίας.
- Εικονίδιο εναλλαγής γλώσσας (Αγγλικά και Ελληνικά).

Εδώ να σημειώσουμε ότι το μενού είναι διαθέσιμο στον χρήστη από οποιαδήποτε σελίδα της εφαρμογής μας.

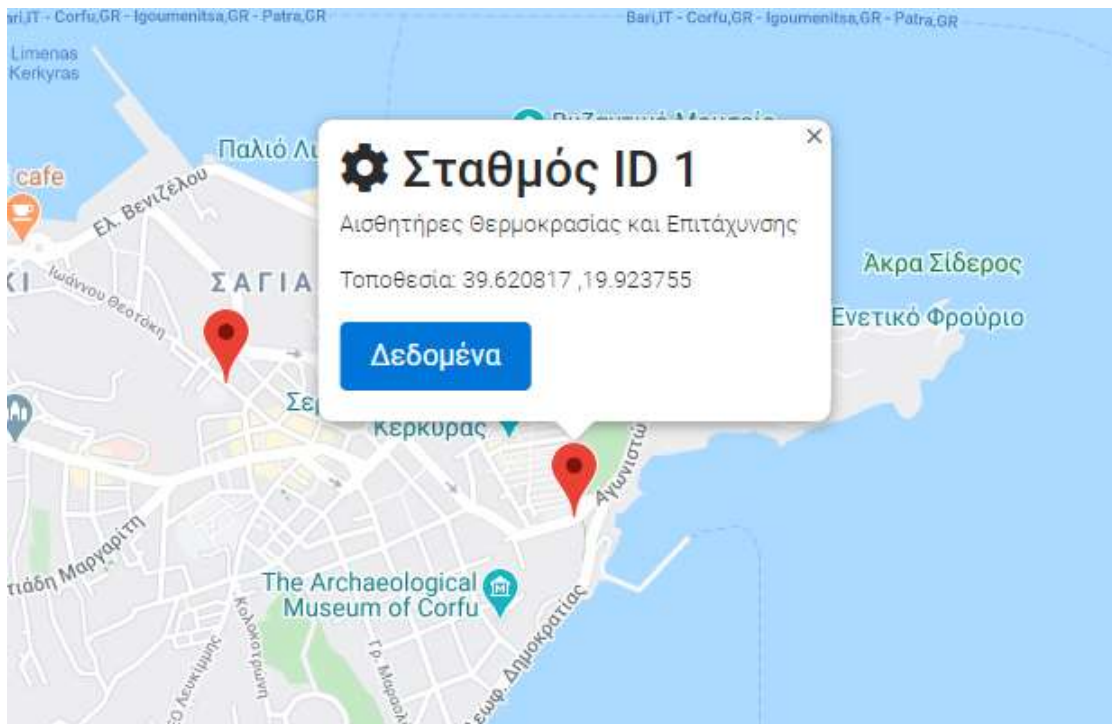
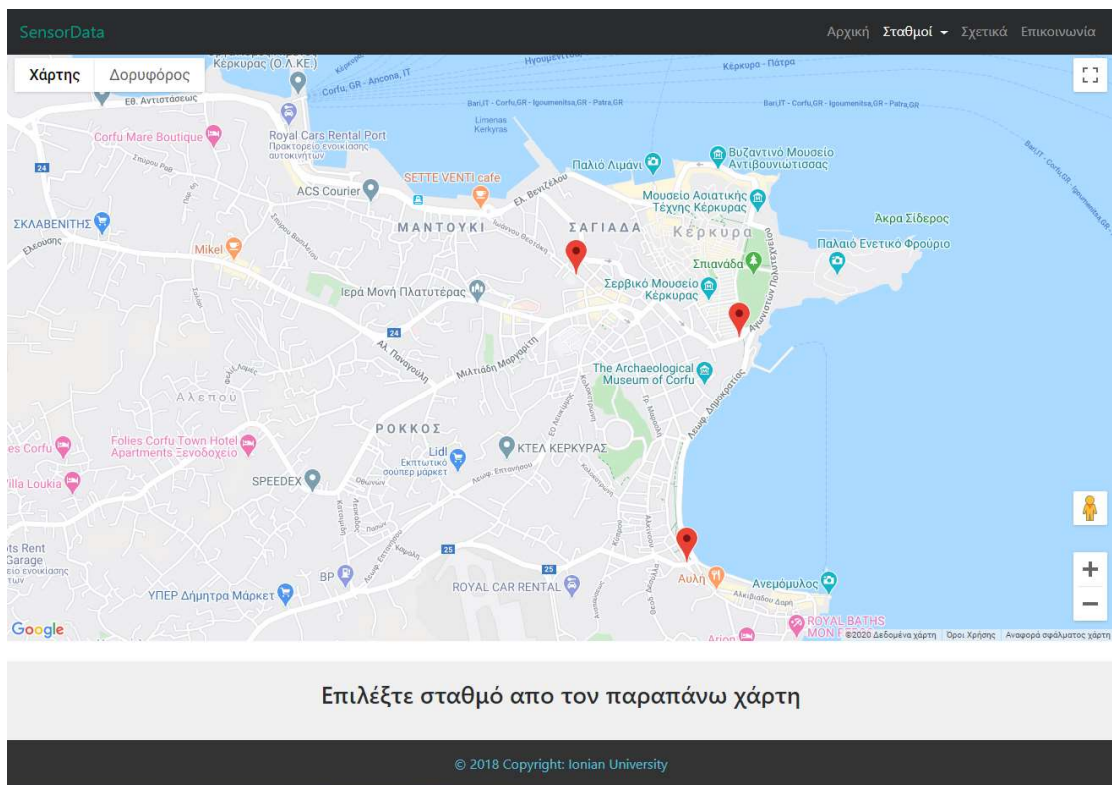
Κάτω από το μενού βλέπουμε δύο κυλιόμενες εικόνες οι οποίες αποτελούν και συνδέσμους για γρήγορη πρόσβαση στις ενότητες «Αναζήτηση Σταθμού» και «Προσθήκη Σταθμού», επίσης ο χρήστης μπορεί να επιλέξει ανάμεσα τους με βέλη τα οποία βρίσκονται αριστερά και δεξιά τους.

Κάτω από τις εικόνες-συνδέσμους βλέπουμε μερικά δεδομένα, που έχουν συλλεχτεί από τον κεντρικό σταθμό, με τα αντίστοιχα εικονίδια τους και την απαραίτητη επεξήγηση.



Αναζήτηση Σταθμού – Απεικόνιση Δεδομένων

Παρακάτω βλέπουμε την σελίδα «Αναζήτηση Σταθμού».



Εάν ο χρήστης επιλέξει και πατήσει πάνω σε έναν από τους σταθμούς, εμφανίζεται το παραπάνω παράθυρο στο οποίο μπορούμε να δούμε το αναγνωριστικό του σταθμού, τους αισθητήρες που είναι εγκατεστημένοι σε αυτόν, την τοποθεσία του καθώς και ένα κουμπί «Δεδομένα».

Εάν ο χρήστης πατήσει το παραπάνω κουμπί τότε στην σελίδα στην οποία βρίσκεται προστίθενται τα παρακάτω και ο χρήστης μετακινείται σε αυτά με αυτόματη ολίσθηση.

Σταθμός ID 1

Αισθητήρας
Θερμοκρασίας

Αισθητήρας
Επιτάχυνσης

39.620817,19.923755



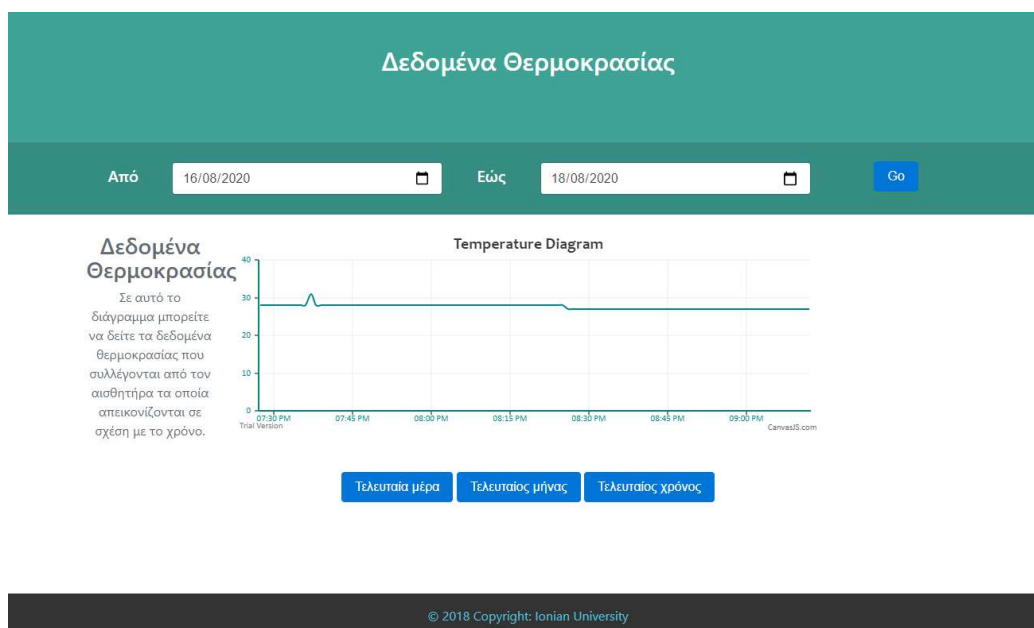
Δεδομένα Θερμοκρασίας
Δείτε τα δεδομένα θερμοκρασίας που απεικονίζονται σε διαγράμματα.
Απεικόνιση

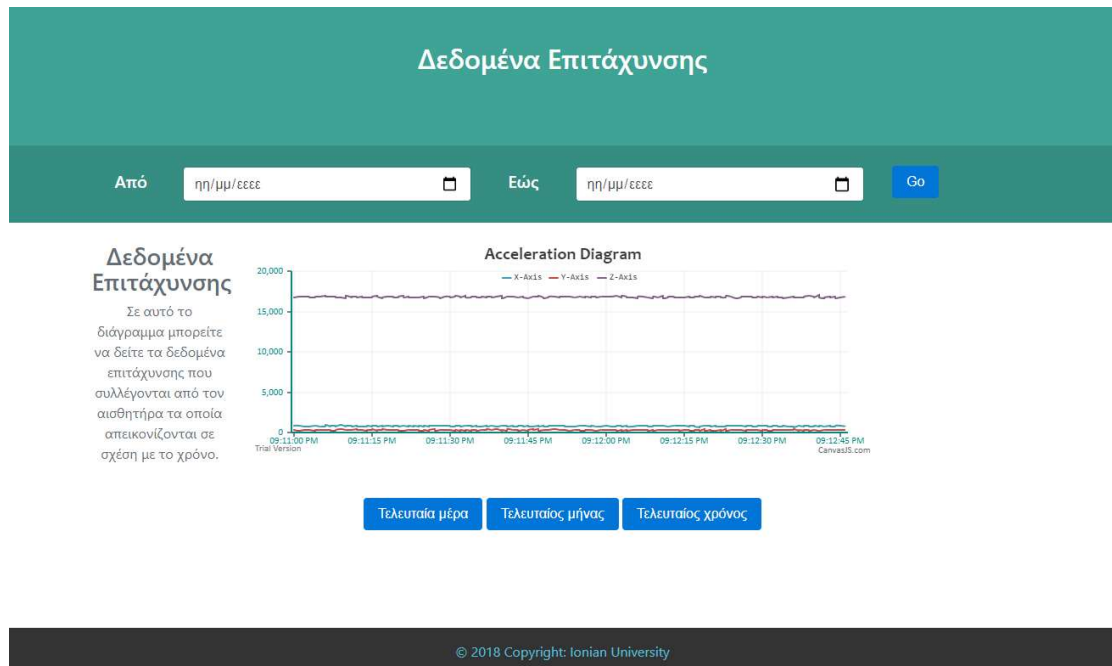


Δεδομένα Επιτάχυνσης
Δείτε τα δεδομένα επιτάχυνσης που απεικονίζονται σε διαγράμματα.
Απεικόνιση

© 2018 Copyright: Ionian University

Πατώντας ο χρήστης σε ένα από τα παραπάνω κουμπιά εμφανίζονται τα εκάστοτε δεδομένα σε διαγράμματα μαζί με κατάλληλη φόρμα επιλογής του εύρους, όπως φαίνεται παρακάτω.





Προσθήκη Σταθμού

Παρακάτω βλέπουμε την φόρμα την οποία ο χρήστης πρέπει να συμπληρώσει προκειμένου να προσθέσει το δικό του σταθμό στον χάρτη.

SensorData Αρχική Σταθμοί Σχετικά Επικοινωνία

Συμπληρώστε τη φόρμα με τα στοιχεία του σταθμού σας.

Προσθήκη Σταθμού

[Προσθήκη](#)

[Χρειάζεστε βοήθεια; Επικοινωνήστε μαζί μας εδώ.](#)

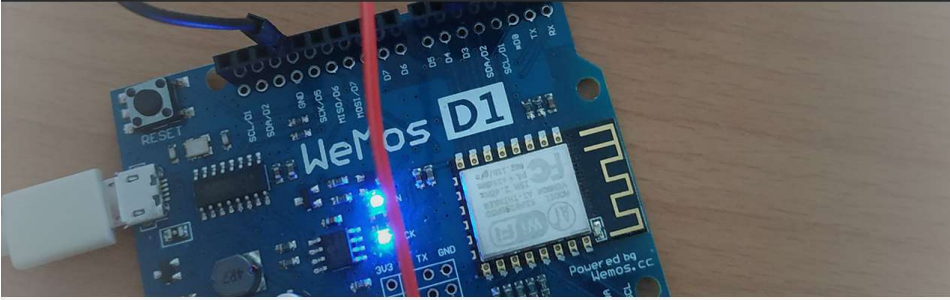
© 2018 Copyright: Ionian University

Σχετικά

Σε αυτή τη καρτέλα ο χρήστης μπορεί να δει μερικά λόγια για την εφαρμογή μας.

SensorData

Αρχική Σταθμοί Σχετικά Επικοινωνία



Σχετικά

Τι είναι το
SensorData.

Το SensorData είναι ένα έργο που δημιουργήθηκε από φοιτητές του Ιονίου Πανεπιστημίου και είναι μια διαδικτυακή εφαρμογή που συλλέγει δεδομένα από μικροεπεξεργαστές τύπου arduino εξοπλισμένους με διάφορους αισθητήρες. Στη συνέχεια, τα δεδομένα των αισθητήρων που συλλέγονται οπτικοποιούνται με τη μορφή διαγραμμάτων που εμφανίζουν τιμές με την πάροδο του χρόνου.

Κεντρικός
Σταθμός

Ο κύριος σταθμός μας βρίσκεται στο Ιόνιο Πανεπιστήμιο Κέρκυρας, τμήμα Πληροφορικής. Επί του παρόντος, ο σταθμός είναι κατασκευασμένος με χρήση μικροεπεξεργαστή τύπου arduino που ονομάζεται Wemos D1 που μπορεί να υποστηρίξει πολλούς τύπους αισθητήρων και έχει επίσης καλύτερες συνολικές προδιαγραφές και τιμή από ένα Arduino. Αυτή τη στιγμή ο σταθμός παρέχει δεδομένα θερμοκρασίας και επιταχυνσιόμετρο.

© 2018 Copyright: Ionian University

Επικοινωνία

Στην καρτέλα αυτή ο χρήστης βλέπει την φόρμα επικοινωνίας μέσα από την οποία μπορεί να επικοινωνήσει με τους διαχειριστές της εφαρμογής, καθώς και έναν χάρτη με το πού εδρεύει στο σύστημά μας.

SensorData

Αρχική Σταθμοί Σχετικά Επικοινωνία

Επικοινωνία

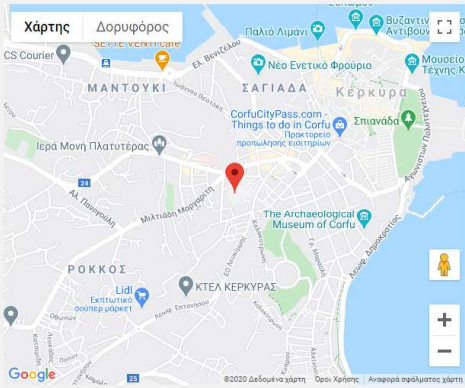
Όνομα

Διεύθυνση Email

Μήνυμα

Αποστολή

Χάρτης



© 2018 Copyright: Ionian University

5 – Συμπεράσματα – Μελλοντικές κατευθύνσεις

Μέσα από την υλοποίηση αυτής της πτυχιακής εργασίας και της διαδικτυακής εφαρμογής που αποτελεί το πρακτικό της τμήμα, μπορούμε να συμπεράνουμε ότι ένα σύστημα καταγραφής – αποθήκευσης – απεικόνισης δεδομένων αισθητήρων μπορεί να πραγματοποιηθεί χωρίς απαραίτητα τη χρήση ακριβών αισθητήρων και συστημάτων γενικότερα. Με τη χρήση χαμηλού κόστους υλικού μπορεί να δημιουργηθεί ένα αξιοπρεπές σύστημα το οποίο θα μπορεί να μας παρέχει χρήσιμα δεδομένα σε βάθος χρόνου για τη μελέτη του περιβάλλοντος.

Επιταχυνσιόμετρο

Η επιλογή του επιταχυνσιόμετρου στη παρούσα εργασία δεν ήταν τυχαία καθώς συνδέεται άμεσα με τη μελέτη της σεισμικής επικινδυνότητας μίας περιοχής. Αυτό συμβαίνει διότι όπως αναφέραμε και σε προηγούμενο κεφάλαιο, ένας τέτοιος τύπος αισθητήρα είναι σε θέση να καταγράψει την επιτάχυνση ενός σώματος στο οποίο είναι προσαρμοσμένος και στους τρεις άξονες. Επομένως γίνεται κατανοητό ότι με κατάλληλους μαθηματικούς τύπους και αλγορίθμους μπορούμε να απομονώσουμε τα δεδομένα που μας παρέχει και σχετίζονται με κάποιο σεισμό δίνοντας μας πληροφορίες σχετικά με το μέγεθος καθώς και τον τύπο της μετατόπισης (οριζόντια ή κάθετη) που μπορεί να προκαλέσει ο σεισμός. Αν λάβουμε λοιπόν υπόψη ότι ένα τέτοιο σύστημα μπορεί να τρέχει ακατάπαυστα για μεγάλο χρονικό διάστημα, σε βάθος χρόνου θα μπορούμε να έχουμε χρήσιμα δεδομένα που αφορούν την ποσότητα των σεισμών καθώς και το μέγεθος αυτών που έχουν γίνει στην εκάστοτε περιοχή, γεγονός που θα μπορεί να μας βοηθήσει να κρίνουμε το ποσοστό σεισμικής επικινδυνότητας της περιοχής.

Κόστος και Δεδομένα

Όπως αναφέραμε και σε προηγούμενα κεφάλαια, το κόστος των αισθητήρων είναι ανάλογο με τη ποιότητα ή ακρίβεια των δεδομένων, δηλαδή όσο μεγαλύτερο είναι το κόστος ενός αισθητήρα τόσο μεγαλύτερη είναι και η ακρίβεια των δεδομένων που προσφέρει. Στην περίπτωση όμως της μελέτης των σεισμών το κόστος είναι τεράστιο. Αν αναλογιστούμε ότι ένα συμβατικό σύστημα ανίχνευσης και καταγραφής σεισμών όπως ένα σεισμόμετρο μπορεί να ξεπερνά τις 10000 ευρώ και το γεγονός ότι ο

αριθμός των συστημάτων που απαιτούνται για να έχουμε μία ολοκληρωμένη εικόνα της περιοχής μπορεί να είναι αρκετά μεγάλος, τότε το εγχείρημα γρήγορα γίνεται ασύμφορο και κοστοβόρο, χωρίς να συνυπολογίσουμε τα κόστη συντήρησης και λειτουργίας.

Από την άλλη μεριά, όσο αναφορά το συγκεκριμένο τομέα, κάνοντας χρήση συστημάτων και αισθητήρων χαμηλού κόστους μπορούμε να πετύχουμε ακρίβεια πλήθους. Δηλαδή εφόσον τα συστήματα αυτά είναι πολύ πιο χαμηλά σε κόστος σε σχέση με τα συμβατικά, καθίσταται δυνατή η χρήση πολλών παραπάνω συστημάτων σε μία περιοχή δίνοντας μας πολύ περισσότερα δεδομένα ανά μονάδα χρόνου και κατ' επέκταση τη δυνατότητα να εντοπίσουμε και να αφαιρέσουμε δεδομένα που σχετίζονται με τυχόν αστοχίες του υλικού, πετυχαίνοντας έτσι μεγαλύτερη ακρίβεια. Ενώ παράλληλα πετυχαίνουμε και καλύτερη κάλυψη της περιοχής ή ακόμα και κάλυψη μεγαλύτερης περιοχής.

Μελλοντικές Επεκτάσεις

Στο σημείο αυτό να αναφέρουμε ότι η παρούσα εργασία υλοποιήθηκε με σκοπό το σύστημα μας να αποτελέσει τη βάση στην οποία θα βασιστούν μελλοντικές εφαρμογές αυτού του περιεχομένου. Μερικές από τις μελλοντικές αυτές επεκτάσεις θα μπορούσαν να περιλαμβάνουν σταθμούς με μεγαλύτερο αριθμό αισθητήρων διαφορετικών ιδιοτήτων, ανάλυση του μεγάλου όγκου αποθηκευμένων δεδομένων (Big data analytics) για την έκβαση συμπερασμάτων, χρήση αλγορίθμων για την εξοικονόμηση χώρου στη βάση δεδομένων και πολλά ακόμα.

Πιο συγκεκριμένα μερικές από τις εφικτές επεκτάσεις θα μπορούσαν να είναι οι παρακάτω:

- **Εφαρμογή καταγραφής σεισμικής επικινδυνότητας ανά περιοχή:** Όπως αναφέραμε και πιο πάνω με τη χρήση επιταχυνσιόμετρου μπορούμε να αντλήσουμε δεδομένα σχετικά με σεισμικές δονήσεις, το μέγεθος αυτών και τη διάρκεια. Λαμβάνοντας υπόψη τον μεγάλο όγκο δεδομένων που μπορούμε να έχουμε σε βάθος χρόνου θα είμαστε σε θέση να κατατάξουμε τις διάφορες περιοχές με βάση τη σεισμική επικινδυνότητα τους, γεγονός που θα μπορούσε να συμβάλει στη πρόληψη σημαντικών απωλειών που σχετίζονται με τους σεισμούς. Χαρακτηριστικό παράδειγμα του οποίου θα μπορούσε να είναι η κατασκευή κτηρίων με αντισεισμικές τεχνολογίες και υλικά σε περιοχές αυξημένης σεισμικής επικινδυνότητας.
- **Εφαρμογή καταγραφής υπερθέρμανσης του πλανήτη:** Η υπερθέρμανση του πλανήτη αποτελεί πλέον πραγματικότητα και γίνονται συνεχώς ενέργειες για να αντιστραφεί αυτό το φαινόμενο. Η χρήση δεδομένων θερμοκρασίας περιβάλλοντος ανά το χρόνο ήταν αυτή που βοήθησε τους επιστήμονες να προβούν σε αυτό το συμπέρασμα. Τα μεγάλα όγκοι δεδομένων λοιπόν που θα βρίσκονται στη διάθεσή μας σε βάθος χρόνου από αισθητήρες θερμοκρασίας όμοιους με αυτούς που χρησιμοποιήσαμε στη παρούσα εργασία, θα

μπορέσουν να μας δώσουν απαντήσεις στο ερώτημα αν κατευθυνόμαστε προς τη σωστή κατεύθυνση ή όχι.

- **Εφαρμογή ειδοποίησης σεισμού:** Όπως αναφέραμε και σε προηγούμενα κεφάλαια η συλλογή και η αποθήκευση των δεδομένων των αισθητήρων στη βάση δεδομένων του συστήματός μας γίνεται σε σχεδόν πραγματικό χρόνο (με μία μέγιστη απόκλιση 500ms). Αν αναλογιστούμε ακόμα ότι με την εξέλιξη των υπολογιστικών συστημάτων μπορούμε να εκτελούμε διάφορες πράξεις επί των δεδομένων σε κλάσματα του δευτερολέπτου, γίνεται κατανοητό ότι με βάση τα δεδομένα επιτάχυνσης, το πλήθος των συστημάτων και τους κατάλληλους αλγορίθμους, θα είμαστε σε θέση να ανιχνεύσουμε σε πολύ μικρό χρονικό διάστημα έναν σεισμό και στη συνέχεια να ενημερώσουμε για το συμβάν (π.χ. με μήνυμα στο κινητό τηλέφωνο), δίνοντας έτσι ένα χρονικό προβάδισμα που αν και μικρό μπορεί να προβεί σημαντικό για την έγκαιρη κινητοποίηση και επέμβαση τόσο των κοντινών πολιτών όσο και των ενδιαφερόμενων φορέων. Να σημειώσουμε εδώ ότι δεν πρόκειται για πρόβλεψη του σεισμού αλλά για έγκαιρη ειδοποίηση αυτών που βρίσκονται κοντά στο επίκεντρο (ή της πρώτης ανίχνευσης).
- **Εφαρμογή καταγραφής ατμοσφαιρικής ρύπανσης:** Στο πλαίσιο χρήσης παραπάνω και διαφορετικών αισθητήρων στο σύστημά μας, έγκειται και η χρήση αισθητήρων αερίων. Με τη χρήση τέτοιου είδους αισθητήρων θα είμαστε σε θέση να παράγουμε μετρίσιμο αποτέλεσμα σχετικά με την ύπαρξη ρυπογόνων αερίων, όπως είναι το μονοξείδιο του άνθρακα(CO), το διοξείδιο του θείου(SO₂), οι υδρογονάνθρακες (HC) και άλλα. Μέσα από την επεξεργασία αυτών των δεδομένων θα μπορούμε να μετρήσουμε το ποσοστό της ατμοσφαιρικής ρύπανσης σε μία περιοχή ενδιαφέροντος αλλά και αν η περιοχή αυτή δείχνει βελτίωση ή επιδείνωση του φαινομένου, βοηθώντας τους κατάλληλους φορείς να λάβουν μέτρα για την αντιμετώπιση του (π.χ. με την απαγόρευση κυκλοφορίας παλαιού τύπου πετρελαιοκίνητων οχημάτων).

6 - Παράρτημα

Αρχείο station-final.ino:

```
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <MPU6050.h>
#include <Wire.h>

const char* ssid    = "COSMOTE-B0BC5D";    // SSID of local network
const char* password = "xjFynpSY492hDVyK"; // Password on network

WiFiClient client;
char servername[]="192.168.1.4"; // remote server we will connect to
// εαν δεν συνδεεται IPAddress servername(192,168,1,4);

// Temperature Data wire is plugged into pin D1 on the ESP8266- GPIO 5
#define ONE_WIRE_BUS 0

// Setup a oneWire instance to communicate with any OneWire devices
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature
DallasTemperature DS18B20(&oneWire);
char temperatureString[7];

const int MPU_addr = 0x68;
MPU6050 mpu;
int16_t acX,acY,acZ,gX,gY,gZ,tmp;
int kitID=1;
String accdataPost;
char acXString[10];
char acYString[10];
char acZString[10];
float temperature;
unsigned long count = 0;
unsigned long prevcount = 0;
unsigned long hour = 0;

byte check_I2c(byte addr){
    byte error;
    Wire.beginTransmission(addr);
```

```

error = Wire.endTransmission();

if (error == 0){
  Serial.print("Device Found at 0x");
  Serial.println(addr,HEX);
}
else{
  Serial.print("No Device Found at 0x");
  Serial.println(addr,HEX);
  Serial.println(error);
}
return error;
}

void setup() {
  Serial.begin(115200);
  Serial.println("Connecting");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(300);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("Connected");

  DS18B20.begin();
  delay(1000);

  Wire.begin(D1,D2);
  while(!mpu.beginSoftwareI2C(D1,D2,MPU6050_SCALE_2000DPS,
MPU6050_RANGE_2G)){
    Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
    delay(500);
  }
  mpu.calibrateGyro();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B); // PWR_MGMT_1 register
  Wire.write(0); // set to zero (wakes up the MPU-6050)
  Wire.endTransmission(true);
  delay(1000);
  check_I2c(MPU_addr);
  delay(500);
}

void getTemperature(){

```

```

float temp;
do{
    DS18B20.requestTemperatures();
    temp = DS18B20.getTempCByIndex(0);
    dtostrf(temp, 2, 2, temperatureString);
    delay(100);
}while (temp == 85.0 || temp == (-127.0));
}

void loop() {
    //-----for accelerometer-----
    Wire.beginTransmission(MPU_addr);
    Wire.write(0x3B);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_addr,14,true);

    acX = Wire.read()<<8|Wire.read();
    acY = Wire.read()<<8|Wire.read();
    acZ = Wire.read()<<8|Wire.read();
    tmp = Wire.read()<<8|Wire.read();

    gX = Wire.read()<<8|Wire.read();
    gY = Wire.read()<<8|Wire.read();
    gZ = Wire.read()<<8|Wire.read();

    Serial.print("acX = "); Serial.println(acX);
    Serial.print(",");
    Serial.print("acY = "); Serial.println(acY);
    Serial.print(",");
    Serial.println("acZ = "); Serial.println(acZ);

    dtostrf(acX, 2, 2, acXString);
    dtostrf(acY, 2, 2, acYString);
    dtostrf(acZ, 2, 2, acZString);

    accdataPost = String("acX=") + acXString + "&acY=" + acYString + "&acZ=" + acZString
+ "&kitID=" + kitID;
    HTTPClient http;

    http.begin("http://" + String(servername) + ":80/thesisb4/accdatabcapture.php");
    //starts client connection
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    int httpCode = http.POST(accdatabPost); //Send the request

```

```

if (httpCode > 0) { //checks for connection
    if(httpCode >= 200 && httpCode <= 300) {
        Serial.println("connected to server");
        String payload = http.getString(); //Get the response payload
        Serial.println(httpCode);
        Serial.println(payload);
        Serial.println("ta esteile...");
    }
    else{
        Serial.println("httpCode not between 200 and 300..");
    }
}
else {
    Serial.println("connection failed"); //error message if no client connect
    Serial.println(httpCode);
    Serial.println();
}
http.end();
delay(500);
//-----for temperature-----
if( count >= hour){
    getTemperature();
    kitID = 1;
    String dataPost;
    dataPost = String("temperature=") + temperatureString + "&kitID=" + kitID;
    HTTPClient http;

    http.begin("http://" + String(servername) + ":80/thesisb4/datacapture.php"); //starts
client connection
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    int httpCode = http.POST(dataPost); //Send the request

    if (httpCode > 0) { //checks for connection
        if(httpCode >= 200 && httpCode <= 300) {
            Serial.println("connected to server");
            String payload = http.getString(); //Get the response payload
            Serial.println(httpCode);
            Serial.println(payload);
            Serial.println("ta esteile...");
        }
        else{
            Serial.println("httpCode not between 200 and 300..");
        }
    }
}

```

```

else {
    Serial.println("connection failed"); //error message if no client connect
    Serial.println(httpCode);
    Serial.println();
}
http.end();
hour = hour + 1;
}
count = millis()/60000;
if(count < prevcount){
    prevcount = 0;
    hour = 0;
}
prevcount = count;
}

```

Αρχείο dbcreate.php:

```

<?php
    header('Content-Type: application/x-www-form-urlencoded;
charset=utf-8');

    ini_set('default_charset', 'UTF-8');
    //echo "start";
    //echo "<br>";

    $con = mysqli_connect("127.0.0.1","root","");
    //echo "connected";
    //echo "<br>";
    if (!$con){
        die("Can not connect: " . mysqli_error());
    }

    mysqli_query($con,"SET NAMES utf8");

    if (mysqli_query($con,"CREATE DATABASE IF NOT EXISTS station
DEFAULT CHARSET=utf8")){
        //echo "Database created successfully";
        //echo "<br>";
    } else {
        //echo "Error: " . mysqli_error();
        //echo "<br>";
    }

    $db_selected = mysqli_select_db($con,"station");

```

```

        //echo "database selected";
        //echo "<br>";

        //echo "end";

        mysqli_close($con);
    ?>

```

Αρχείο datacapture.php:

```

<?php
include ('dbcreate.php');
$con = mysqli_connect("127.0.0.1","root","","station");
echo "<br>";
echo "connected";
echo "<br>";
if (!$con){
    die("Can not connect: " . mysqli_error());
}
//$db_selected = mysql_select_db("station",$con);
echo "database selected";
echo "<br>";
if(isset($_POST['temperature'])){
    $temperature = mysqli_real_escape_string($con,$_POST['temperature']);
}
else{
    $temperature = null;
    echo "no temperature supplied";
}
if(isset($_POST['kitID'])){
    $kitID = mysqli_real_escape_string($con,$_POST['kitID']);
}
else{
    $kitID = null;
    echo "no kitID supplied";
}

$sqltable = "CREATE TABLE IF NOT EXISTS temptest ( ID int NOT NULL
AUTO_INCREMENT, Time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
Temperature int(10) NOT NULL,

                                kitID int NOT NULL, PRIMARY KEY (ID), FOREIGN KEY
(kitID) REFERENCES kits(ID) )";

if (mysqli_query($con,$sqltable)) {

```

```

        echo "Table tempTEST created successfully";
        echo "<br>";
    } else {
        echo "Error creating table: " . mysqli_error($con);
        echo "<br>";
    }

    $sqlinsert = "INSERT INTO temptest( Temperature, kitID)
                VALUES( '$temperature', '$kitID' )";

    if (mysqli_query($con, $sqlinsert)) {
        echo "values inserted on table tempTEST";
        echo "<br>";
    } else {
        echo "Error filling table: " . mysqli_error($con);
        echo "<br>";
    }
    echo "end";

    mysqli_close($con);

?>

```

Αρχείο datacapture.php:

```

<?php
include ('dbcreate.php');
$con = mysqli_connect("127.0.0.1","root","","station");
echo "<br>";
echo "connected";
echo "<br>";
if (!$con){
    die("Can not connect: " . mysqli_error());
}
//$db_selected = mysql_select_db("station",$con);
//echo "database selected";
echo "<br>";

if(isset($_POST['acX'])){
    $acX = mysqli_real_escape_string($con,$_POST['acX']);
}
else{
    $acX = null;
    echo "no acX supplied";
}

```

```

}
if(isset($_POST['acY'])){
    $acY = mysqli_real_escape_string($con,$_POST['acY']);
}
else{
    $acY = null;
    echo "no acX supplied";
}
if(isset($_POST['acZ'])){
    $acZ = mysqli_real_escape_string($con,$_POST['acZ']);
}
else{
    $acZ = null;
    echo "no acX supplied";
}
if(isset($_POST['kitID'])){
    $kitID = mysqli_real_escape_string($con,$_POST['kitID']);
}
else{
    $kitID = null;
    echo "no kitID supplied";
}

```

```

$sqltable = "CREATE TABLE IF NOT EXISTS accdata ( ID int NOT NULL
AUTO_INCREMENT, Time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP, acX
int(10) NOT NULL, acY int(10) NOT NULL, acZ int(10) NOT NULL,

```

```

                                kitID int NOT NULL, PRIMARY KEY (ID), FOREIGN KEY
(kitID) REFERENCES kits(ID) )";

```

```

if (mysqli_query($con,$sqltable)) {
    echo "Table accdata created successfully";
    echo "<br>";
} else {
    echo "Error creating table: " . mysqli_error($con);
    echo "<br>";
}

```

```

$sqlinsert = "INSERT INTO accdata( acX, acY, acZ, kitID)
VALUES( '$acX', '$acY', '$acZ', '$kitID' )";

```

```

if (mysqli_query($con,$sqlinsert)) {
    echo "values inserted on table accdata";
    echo "<br>";
} else {

```

```

        echo "Error filling table: " . mysqli_error($con);
        echo "<br>";
    }
    echo "end";

    mysqli_close($con);

?>

```

Αρχείο accdatacapture.php:

```

<?php
include ('dbcreate.php');
$con = mysqli_connect("127.0.0.1","root","","station");
echo "<br>";
echo "connected";
echo "<br>";
if (!$con){
    die("Can not connect: " . mysqli_error());
}
//$db_selected = mysql_select_db("station",$con);
//echo "database selected";
echo "<br>";

if(isset($_POST['acX'])){
    $acX = mysqli_real_escape_string($con,$_POST['acX']);
}
else{
    $acX = null;
    echo "no acX supplied";
}
if(isset($_POST['acY'])){
    $acY = mysqli_real_escape_string($con,$_POST['acY']);
}
else{
    $acY = null;
    echo "no acX supplied";
}
if(isset($_POST['acZ'])){
    $acZ = mysqli_real_escape_string($con,$_POST['acZ']);
}
else{
    $acZ = null;
    echo "no acX supplied";
}

```

```

}
if(isset($_POST['kitID'])){
    $kitID = mysqli_real_escape_string($con,$_POST['kitID']);
}
else{
    $kitID = null;
    echo "no kitID supplied";
}

$sqltable = "CREATE TABLE IF NOT EXISTS accdata ( ID int NOT NULL
AUTO_INCREMENT, Time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP, acX
int(10) NOT NULL, acY int(10) NOT NULL, acZ int(10) NOT NULL,

                                kitID int NOT NULL, PRIMARY KEY (ID), FOREIGN KEY
(kitID) REFERENCES kits(ID) )";

if (mysqli_query($con,$sqltable)) {
    echo "Table accdata created successfully";
    echo "<br>";
} else {
    echo "Error creating table: " . mysqli_error($con);
    echo "<br>";
}

$sqlinsert = "INSERT INTO accdata( acX, acY, acZ, kitID)
                                VALUES( '$acX', '$acY', '$acZ', '$kitID' )";

if (mysqli_query($con,$sqlinsert)) {
    echo "values inserted on table accdata";
    echo "<br>";
} else {
    echo "Error filling table: " . mysqli_error($con);
    echo "<br>";
}
echo "end";

mysqli_close($con);

?>

```

Αρχείο homesetup.js:

```

$(document).ready(function() {

    var timestamp_tod = Math.round(new Date().getTime() / 1000);

```

```

var timestamp_yest = timestamp_tod - (24 * 3600);
var mili_to = timestamp_tod * 1000;
var mili_from = timestamp_yest * 1000;
var form_data = {from_date:mili_from, to_date:mili_to};
//αν θέλουμε JSON format:
var form_data_st = JSON.stringify(form_data);
//-----
//Στέλνουμε τα input του χρήστη με Ajax στο php file, και παίρνουμε πίσω τα
δεδομένα απο το Json.
$.ajax({
  type: "POST",
  //dataType: "json",
  url: "jsonindex.php",
  data: form_data,
  success: function(data){
    var element = document.getElementById("highest_temp");
    element.innerHTML = '<i class="fas fa-fire-alt" style="font-size:100px">'+</i>'+
<b>'+ data[0].maxTemperature + '&#8451;</b>';

    var element = document.getElementById("lowest_temp");
    element.innerHTML = '<i class="far fa-snowflake" style="font-
size:100px">'+</i>'+ <b>'+ data[0].minTemperature + '&#8451;</b>';

    var element = document.getElementById("maxX");
    element.innerHTML = ' Highest: <b>'+ data[0].maxX + '</b>';

    var element = document.getElementById("minX");
    element.innerHTML = ' Lowest: <b>'+ data[0].minX + '</b>';

    var element = document.getElementById("maxY");
    element.innerHTML = ' Highest: <b>'+ data[0].maxY + '</b>';

    var element = document.getElementById("minY");
    element.innerHTML = ' Lowest: <b>'+ data[0].minY + '</b>';

    var element = document.getElementById("maxZ");
    element.innerHTML = ' Highest: <b>'+ data[0].maxZ + '</b>';

    var element = document.getElementById("minZ");
    element.innerHTML = ' Lowest: <b>'+ data[0].minZ + '</b>';
  },
  error: function(xhr){
    alert('Error home.. status = ' + xhr.status + "message = " + xhr.statusText);
  }
});

```

```
});
```

Αρχείο jsonindex.php:

```
<?php

header( 'Content-Type: application/json');

$con = mysqli_connect("127.0.0.1","root","","station");
if (!$con){
    die("Can not connect: " . mysqli_error());
}

// $db_selected = mysqli_select_db($con,"station");
// ~~~~~

~~~~~

$from_date = input_val($con, $_POST['from_date']);
$to_date = input_val($con, $_POST['to_date']);

// ===== παινουμε την υψηλότερη, χαμηλότερη τιμή
τις τελευταίες 24 ώρες =====
$query = sprintf("SELECT MAX(temptest.Temperature) AS
maxTemperature, MIN(temptest.Temperature) AS minTemperature, MAX(accddata.acX)
AS maxX, MIN(accddata.acX) AS minX, MAX(accddata.acY) AS maxY, MIN(accddata.acY) AS
minY, MAX(accddata.acZ) AS maxZ, MIN(accddata.acZ) AS minZ FROM temptest JOIN
accddata ON temptest.kitID=accddata.kitID=1 WHERE
UNIX_TIMESTAMP(temptest.Time)*1000 >= '$from_date' AND
UNIX_TIMESTAMP(temptest.Time)*1000 <= '$to_date'");

// ~~~~~

~~~~~

$result = mysqli_query($con, $query);

$data = array();
while($row = mysqli_fetch_assoc($result)){
    $data[] = $row;
}

function input_val($con, $dates){
    $dates = trim($dates);
    $dates = stripslashes($dates);
    $dates = htmlspecialchars($dates);
    $dates = mysqli_real_escape_string($con,$dates);
```

```

        return $dates;
    }

    mysqli_close($con);

    echo json_encode($data);
    $data[] = [];

?>

```

Αρχείο index.html:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <!-- The above 3 meta tags *must* come first in the head; any other head content
must come *after* these tags -->
    <title>Sensor Data</title>

    <!-- Bootstrap -->
    <link rel="stylesheet" type="text/css" href="bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="customcss.css">
    <!-- Icons -->
    <link rel="stylesheet" href="all.min.css">

  </head>
  <body style="background: #f2f2f2; padding-bottom: 80px;">

    <!-- ===== Navigation bar
===== -->
    <nav class="navbar fixed-top navbar-toggleable-sm navbar-inverse bg-inverse">
      <button class="navbar-toggler navbar-toggler-right" type="button" data-
toggle="collapse" data-target="#navbarDropdown" aria-controls="navbarDropdown"
aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <a class="navbar-brand" href="index.html">SensorData</a>
      <div class="collapse navbar-collapse" id="navbarDropdown">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item active">

```

```

        <a class="nav-link" href="index.html">Home <span class="sr-
only">(current)</span></a>
    </li>
    <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
            Stations
        </a>
        <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
            <a class="dropdown-item" href="find-stations.html">Find Station</a>
            <a class="dropdown-item" href="add-stations.html">Add Station</a>
        </div>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="about.html">About</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="contact.html">Contact</a>
    </li>
    <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="language" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
            <i class="fas fa-globe-americas"></i>
        </a>
        <div class="dropdown-menu dropdown-menu-right" aria-
labelledby="language">
            <a class="dropdown-item" href="index.html">English</a>
            <a class="dropdown-item" href="indexgr.html">Ελληνικά</a>
        </div>
    </li>
</ul>
</div>
</nav>

<!-- ===== Bootstrap carousel
=====-->
<div id="mycarousel" class="carousel slide" data-ride="carousel">
    <ol class="carousel-indicators">
        <li data-target="#mycarousel" data-slide-to="0" class="active"></li>
        <li data-target="#mycarousel" data-slide-to="1"></li>
    </ol>
    <div class="carousel-inner" role="listbox">
        <div class="carousel-item active">
            <a href="find-stations.html">
                

```

```

        <div class="carousel-caption">
            <h2>Search Stations</h2>
            <p>Press here to see the stations in the map.</p>
        </div>
    </a>
</div>
<div class="carousel-item">
    <a href="add-stations.html">
        
        <div class="carousel-caption">
            <h2>Add Your Station</h2>
            <p>Press here to add your station to the database.</p>
        </div>
    </a>
</div>
</div>
<a class="carousel-control-prev" href="#mycarousel" role="button" data-
slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next" href="#mycarousel" role="button" data-
slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
</a>
</div>

<!--
=====
===== -->
<div class="container-fluid">
    <br>
    <br>
    <h2 align="center">Latest Sensor Data</h2>
    <h6 class="text-muted" align="center">Data collected from Arduino type sensor
which is located at Ionian University.</h6>
    <br>
    <br>
    <div class="row">
        <div class="col-md-2"></div>
        <div class="col-md-4" align="center">
            <h1 id="highest_temp"></h1>

```

```

        <h4 class="text-muted">Highest Temperature Today</h4>
        <p class="text-muted">From main station.</p>
    </div>
    <div class="col-md-4" align="center">
        <h1 id="lowest_temp"></h1>
        <h4 class="text-muted">Lowest Temperature Today</h4>
        <p class="text-muted">From main station.</p>
    </div>
    <div class="col-md-2"></div>
</div>
<div class="row">
    <div class="col-md-4" align="center">
        <h1 id="x_axis"><i class="fas fa-chart-area" style="font-size:100px"></i><b> X-
Axis</b></h1>
        <h4 class="text-muted">Acceleration Values Today</h4>
        <p class="text-muted" id="maxX"></p>
        <p class="text-muted" id="minX" style="margin-top:-18px;"></p>
        <p class="text-muted" style="margin-top:-18px;">From main station.</p>
    </div>
    <div class="col-md-4" align="center">
        <h1 id="y_axis"><i class="fas fa-chart-area" style="font-size:100px"></i><b> Y-
Axis</b></h1>
        <h4 class="text-muted">Acceleration Values Today</h4>
        <p class="text-muted" id="maxY"></p>
        <p class="text-muted" id="minY" style="margin-top:-18px;"></p>
        <p class="text-muted" style="margin-top:-18px;">From main station.</p>
    </div>
    <div class="col-md-4" align="center">
        <h1 id="z_axis"><i class="fas fa-chart-area" style="font-size:100px"></i><b> Z-
Axis</b></h1>
        <h4 class="text-muted">Acceleration Values Today</h4>
        <p class="text-muted" id="maxZ"></p>
        <p class="text-muted" id="minZ" style="margin-top:-18px;"></p>
        <p class="text-muted" style="margin-top:-18px;">From main station.</p>
    </div>
</div>
<br>
<br>
<br>
<br>
<!-- ===== footer
=====-->
<footer class="footer fixed-bottom mt-auto">
    <div class="container text-white">

```

```

    </div>
    <div class="footer-copyright text-center py-3 text-info">© 2018 Copyright: Ionian
University
    </div>
</footer>

<!--
=====
=====-->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
<script type="text/javascript" src="chartjs/js/tether.min.js"></script>
<script type="text/javascript" src="chartjs/js/bootstrap.min.js"></script>
<script type="text/javascript" src="chartjs/js/homesetup.js"></script>
</body>
</html>

```

Αρχείο find-stations.html:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <!-- The above 3 meta tags *must* come first in the head; any other head content
must come *after* these tags -->
    <title>Stations</title>

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
    <!-- Bootstrap -->
    <link rel="stylesheet" type="text/css" href="bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="customcss.css">
    <!-- Icons -->
    <link rel="stylesheet" href="all.min.css">

  </head>
  <body style="padding-bottom:70px;">

    <!-- ===== Navigation bar
===== -->

```

```

<nav class="navbar fixed-top navbar-toggleable-sm navbar-inverse bg-inverse">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-
toggle="collapse" data-target="#navbarDropdown" aria-controls="navbarDropdown"
aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <a class="navbar-brand" href="index.html">SensorData</a>
  <div class="collapse navbar-collapse" id="navbarDropdown">
    <ul class="navbar-nav ml-auto">
      <li class="nav-item">
        <a class="nav-link" href="index.html">Home <span class="sr-
only">(current)</span></a>
      </li>
      <li class="nav-item dropdown active">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
          Stations
        </a>
        <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
          <a class="dropdown-item" href="find-stations.html">Find Station</a>
          <a class="dropdown-item" href="add-stations.html">Add Station</a>
        </div>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="about.html">About</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="contact.html">Contact</a>
      </li>
    </ul>
  </div>
</nav>

```

```

<!-- ===== Google maps
=====-->

```

```

<div id="map-container" class="map-container" style="height: 700px;">
</div>

```

```

<!--
=====
===== -->
<div style="background: #fefefe;">
<div class="container-fluid">
  <br>
  <div class="col-md-12" align="center" id="" style="background: #f0f0f0">

```

```

<br>
<h3 id="h_before_logos">Select Station from the map above</h3>
<br>
<div class="row">
  <div class="col-md-3">
  </div>
  <div class="col-md-2">
    <p id="p_temp"></p>
  </div>
  <div class="col-md-2">
    <p id="p_accel"></p>
  </div>
  <div class="col-md-2">
    <p id="p_location"></p>
  </div>
  <div class="col-md-3">
  </div>
</div>
<div class="col-md-12" align="center" id="load_logos_div">
</div>
<div class="col-md-12" align="center" id="after_logos_div">
  <h3></h3>
</div>

</div>
</div>
<!--
=====
===== -->
<footer class="footer fixed-bottom mt-auto">
  <div class="container text-white">
  </div>
  <div class="footer-copyright text-center py-3 text-info">© 2018 Copyright: Ionian
University
  </div>

</footer>

<script src="https://maps.googleapis.com/maps/api/js?key=AlzaSyBw3eGKOGI_-
rMU5TG4ZyVljcLIRTJvmfw"></script>
<script type="text/javascript" src="chartjs/js/mapSetting.js"></script>
<script type="text/javascript" src="chartjs/js/tether.min.js"></script>
<script type="text/javascript" src="chartjs/js/bootstrap.min.js"></script>
<script type="text/javascript" src="chartjs/js/canvasjs.min.js"></script>

```

```

<script type="text/javascript" src="chartjs/js/linegraph.js"></script>
<script type="text/javascript" src="chartjs/js/linegraphacc.js"></script>

</body>
</html>

```

Αρχείο add-stations.html:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- The above 3 meta tags *must* come first in the head; any other head content
    must come *after* these tags -->
    <title>Stations</title>

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
    <!-- Bootstrap -->
    <link rel="stylesheet" type="text/css" href="bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="customcss.css">
    <!-- Icons -->
    <link rel="stylesheet" href="all.min.css">

  </head>
  <body style="background: #3a3a3a; padding-bottom: 80px;">

    <!-- ===== Navigation bar
    ===== -->
    <nav class="navbar fixed-top navbar-toggleable-sm navbar-inverse bg-inverse">
      <button class="navbar-toggler navbar-toggler-right" type="button" data-
toggle="collapse" data-target="#navbarDropdown" aria-controls="navbarDropdown"
aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <a class="navbar-brand" href="index.html">SensorData</a>
      <div class="collapse navbar-collapse" id="navbarDropdown">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item">
            <a class="nav-link" href="index.html">Home <span class="sr-
only">(current)</span></a>

```

```

    </li>
    <li class="nav-item dropdown active">
      <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        Stations
      </a>
      <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
        <a class="dropdown-item" href="find-stations.html">Find Station</a>
        <a class="dropdown-item" href="add-stations.html">Add station</a>
      </div>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="about.html">About</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="contact.html">Contact</a>
    </li>
  </ul>
</div>
</nav>

```

```

<!-- ===== Form ADD STATION
===== -->
<div class="container">
  <br>
  <h6 class="text-white" align="center">Please fill the form with your station
information.</h6>
  <br>
  <br>
  <div class="col-md-5 mx-auto rounded" style="background: #fefefe;">
    <br>
    <div class="logo mb-3">
      <div class="col-md-12 text-center">
        <h1>Add Station</h1>
      </div>
    </div>
    <form action="add-form-val.php" method="post" class="container"
name="add_form" onSubmit="return formval()">
      <div class="form-group">
        <label for="name"><i class="fas fa-user" style="font-size:25px"></i>
Name</label>
        <input type="text" class="form-control" id="name" placeholder="Enter Name"
required>
        <p type="hidden" class="text-danger" id="error0"></p>
      </div>

```

```

        <div class="form-group">
            <label for="email"><i class="fas fa-envelope" style="font-size:25px"></i>
Email address</label>
            <input type="email" class="form-control" name="email" placeholder="Enter
email" required>
        </div>
        <div class="form-group">
            <label for="latitude"><i class="fas fa-map-marker-alt" style="font-
size:25px"></i> Position Latitude</label>
            <input type="text" name="latitude" id="latitude" class="form-control"
placeholder="Enter Latitude (e.g. 19.231241)" required>
            <p type="hidden" class="text-danger" id="error1"></p>
        </div>
        <div class="form-group">
            <label for="longtitude"><i class="fas fa-map-marker-alt" style="font-
size:25px"></i> Position Longitude</label>
            <input type="text" name="longtitude" id="longtitude" class="form-control"
placeholder="Enter Longitude (e.g. 12.231241)" required >
            <p type="hidden" class="text-danger" id="error2"></p>
        </div>
        <div class="col-md-12 text-center mb-3">
            <button type="submit" class=" btn btn-block btn-primary tx-tfm"
name="add_btn">ADD</button>
        </div>
        <div class="col-md-12 ">
            <div class="form-group">
                <p class="text-center"><a href="contact.html" id="contact_link">Need
help? Contact us here.</a></p>
            </div>
        </div>
    </form>
</div>
<br>
<br>
</div>
<!--
=====
===== -->
    <footer class="footer fixed-bottom mt-auto">
        <div class="container text-white">
            </div>
            <div class="footer-copyright text-center py-3 text-info">© 2018 Copyright: Ionian
University
            </div>
        </footer>

```

```

<script type="text/javascript">
//===== validation before submit
=====
function formval(){
    var counter = 0;
    var regex = /^[ \p{L} ]+$/u;

    if(!(regex.test($('#name').val()))){
        $('#error0').html('Enter only letters and spaces.').show();
        counter+=1;
    }
    else{
        $('#error0').html('Enter only letters and spaces.').hide();
    }
    if (!($('#latitude').val() > -90 && $('#latitude').val() < 90 )) {
        $('#error1').html('Enter a number -90 to 90.').show();
        counter+=1;
    }
    else{
        $('#error1').html('Enter a number -90 to 90.').hide();
    }
    if (!($('#longitude').val() > -180 && $('#longitude').val() < 180) ) {
        $('#error2').html('Enter a number -180 to 180.').show();
        counter+=1;
    }
    else{
        $('#error2').html('Enter a number -180 to 180.').hide();
    }
    if(counter!=0){
        return false;
    }
}
</script>

<script type="text/javascript" src="chartjs/js/tether.min.js"></script>
<script type="text/javascript" src="chartjs/js/bootstrap.min.js"></script>

</body>
</html>

```

Αρχείο about.html:

```

<!DOCTYPE html>

```

```

<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <!-- The above 3 meta tags *must* come first in the head; any other head content
must come *after* these tags -->
    <title>Sensor Data</title>

    <!-- Bootstrap -->
    <link rel="stylesheet" type="text/css" href="bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="customcss.css">
    <!-- Icons -->
    <link rel="stylesheet" href="all.min.css">

  </head>
  <body style="background: #f2f2f2; padding-bottom: 80px;">

    <!-- ===== Navigation bar
===== -->
    <nav class="navbar fixed-top navbar-toggleable-sm navbar-inverse bg-inverse">
      <button class="navbar-toggler navbar-toggler-right" type="button" data-
toggle="collapse" data-target="#navbarDropdown" aria-controls="navbarDropdown"
aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <a class="navbar-brand" href="index.html">SensorData</a>
      <div class="collapse navbar-collapse" id="navbarDropdown">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item">
            <a class="nav-link" href="index.html">Home <span class="sr-
only">(current)</span></a>
          </li>
          <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
              Stations
            </a>
            <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
              <a class="dropdown-item" href="find-stations.html">Find Station</a>
              <a class="dropdown-item" href="add-stations.html">Add Station</a>
            </div>
          </li>
          <li class="nav-item active">

```

```

        <a class="nav-link" href="about.html">About</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="contact.html">Contact</a>
    </li>
</ul>
</div>
</nav>
<!--
=====
=====-->
    <div class="container-full-bg" style="height:400px; background-
image:url('about2000.jpg'); background-position:center;">
        <div class="container">
            </div>
        </div>
        <br>
        <h1 class="text-muted" align="center">About us</h1>
        <br>
        <!--
=====
===== -->
        <div>
            <div class="container-fluid">
                <div class="row">
                    <div class="col-md-6">
                        <h3 align="center">What is </h3><h3 class="text-info"
align="center">SensorData.</h3>
                        <p>SensorData is a project that created by students of Ionian University, and is a
web application that collects data from arduino type microprocessors equipped with
various sensors. The collected sensor data then visuallized in the form of charts
displaying values over time.</p>
                        <p></p>
                    </div>

                    <div class="col-md-6">
                        <h3 align="center">Main </h3><h3 class="text-info"
align="center">Station</h3>
                        <p>Our main station is located at Ionian University of Corfu, department of
Informatics. Currently the station is built with a use of arduino type microprocessor
called Wemos D1 that can support many types of sensors and it has also better overall
specs and price than a Arduino. At this time the station provides temperature and
accelerometer data.</p>
                    </div>
                </div>
            </div>

```

```

</div>
<!-- ===== footer
=====-->
<footer class="footer fixed-bottom mt-auto">
  <div class="container text-white">
    </div>
    <div class="footer-copyright text-center py-3 text-info">© 2018 Copyright: Ionian
University
    </div>
  </footer>

<!--
=====
=====-->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
  <script type="text/javascript" src="chartjs/js/tether.min.js"></script>
  <script type="text/javascript" src="chartjs/js/bootstrap.min.js"></script>
</body>
</html>

```

Αρχείο contact.html:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <!-- The above 3 meta tags *must* come first in the head; any other head content
must come *after* these tags -->
    <title>Sensor Data</title>

    <!-- Bootstrap -->
    <link rel="stylesheet" type="text/css" href="bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="customcss.css">
    <!-- Icons -->
    <link rel="stylesheet" href="all.min.css">

  </head>

  <body style="background: #f2f2f2; padding-bottom: 80px;">

```

```

<!-- ===== Navigation bar
===== -->
<nav class="navbar fixed-top navbar-toggleable-sm navbar-inverse bg-inverse">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-
toggle="collapse" data-target="#navbarDropdown" aria-controls="navbarDropdown"
aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <a class="navbar-brand" href="index.html">SensorData</a>
  <div class="collapse navbar-collapse" id="navbarDropdown">
    <ul class="navbar-nav ml-auto">
      <li class="nav-item">
        <a class="nav-link" href="index.html">Home <span class="sr-
only">(current)</span></a>
      </li>
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
          Stations
        </a>
        <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
          <a class="dropdown-item" href="find-stations.html">Find Station</a>
          <a class="dropdown-item" href="add-stations.html">Add Station</a>
        </div>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="about.html">About</a>
      </li>
      <li class="nav-item active">
        <a class="nav-link" href="contact.html">Contact</a>
      </li>
    </ul>
  </div>
</nav>
<!--
=====
=====-->
<br>
<br>
<h2 class="text-muted" align="center"><i class="far fa-paper-plane"></i>Contact
Us</h2>
<br>
<br>
<div class="container-fluid">
  <div class="row">

```

```

<div class="col-md-6">
  <form action="contact-form-val.php" method="post" class="container"
name="contact_form" onSubmit="contact_map()">
    <div class="form-group">
      <label for="name_c"><i class="fas fa-user" style="font-size:25px"></i>
Name</label>
      <input type="text" class="form-control" name="name_c" id="name_c"
placeholder="Enter Name" required>
    </div>
    <div class="form-group">
      <label for="email_c"><i class="fas fa-envelope" style="font-size:25px"></i>
Email address</label>
      <input type="email" class="form-control" name="email_c" id="email_c"
placeholder="Enter email" required>
    </div>
    <div class="form-group">
      <label for="message"><i class="far fa-edit" style="font-size:25px"></i>
Message</label>
      <textarea type="text" name="message" id="message" class="form-control"
placeholder="Enter your message here.." rows="5" required></textarea>
      <p type="hidden" class="text-danger" id="error_c"></p>
    </div>
    <div class="col-md-12 text-center mb-3">
      <button type="submit" class=" btn btn-block btn-primary tx-tfm"
id="send">Send <i class="far fa-paper-plane"></i></button>
    </div>
  </form>
</div>
<div class="col-md-6">
  <div class="map-container" id="map-container-c" style="height: 500px;">
  </div>
</div>
</div>

<!-- ===== footer
=====-->
<footer class="footer fixed-bottom mt-auto">
  <div class="container text-white">
  </div>
  <div class="footer-copyright text-center py-3 text-info">© 2018 Copyright: Ionian
University
  </div>
</footer>

```

```

<!-- ===== Map Load
=====-->
<script src="https://maps.googleapis.com/maps/api/js?key=AlzaSyBw3eGKOGI_-
rMU5TG4ZyVljcLIRTJvmfw"></script>
<script type="text/javascript">
    function contact_map(){

        var location = new google.maps.LatLng(39.620538, 19.914697);
        var mapoptions = {
            center: location,
            zoom: 15
        };
        var corfumap = new google.maps.Map(document.getElementById("map-
container-c"),mapoptions);

        var marker = new google.maps.Marker({
            position: location,
            map: corfumap,
            title: "Ionian University"
        });
    }
    google.maps.event.addDomListener(window, 'load', contact_map);
</script>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
<script type="text/javascript" src="chartjs/js/tether.min.js"></script>
<script type="text/javascript" src="chartjs/js/bootstrap.min.js"></script>
</body>
</html>

```

Αρχείο contact-form-val.php:

```

<?php
    use PHPMailer\PHPMailer\PHPMailer;
    use PHPMailer\PHPMailer\SMTP;
    use PHPMailer\PHPMailer\Exception;
    require 'PHPMailer\src\Exception.php';
    require 'PHPMailer\src\PHPMailer.php';
    require 'PHPMailer\src\SMTP.php';

    include ('dbcreate.php');

```

```

$con = mysqli_connect("127.0.0.1","root","","station");
echo "<br>";
echo "connected";
echo "<br>";
if (!$con){
    die("Can not connect: " . mysqli_error());
}
// $db_selected = mysqli_select_db($con,"station");
echo "database selected";
echo "<br>";

if(isset($_POST['name_c'])){
    $name = input_val($con,$_POST['name_c']);
}
else{
    $name = null;
    echo "no name supplied";
}
if(isset($_POST['email_c'])){
    $email = input_val($con,$_POST['email_c']);
}
else{
    $email = null;
    echo "no email supplied";
}
if(isset($_POST['message'])){
    $message = input_val($con,$_POST['message']);
}
else{
    $message = null;
    echo "Message is empty";
}

//=====*/
$mail = new PHPMailer(TRUE);

$mail->From = $email;
$mail->FromName = $name;

$mail->addAddress("gksproject01@gmail.com");

$mail->addReplyTo("gksproject01@gmail.com");

$mail->isHTML(true);

```

```

$mail->Subject = "Sensor data";
$mail->Body = "<i>". $message . "</i>";
$mail->AltBody = "This is text..";

try{
    $mail->send();
    echo "Message has been sent Successfully";
} catch (Exception $e){
    echo "Mailer Error: ". $mail->ErrorInfo;
}

//=====================================================
function input_val($con,$data){
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    $data = mysqli_real_escape_string($con,$data);
    return $data;
}

mysqli_close($con);

header("Location:successfully-send.html")
?>

```

Αρχείο successfully-send.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <!-- The above 3 meta tags *must* come first in the head; any other head content
must come *after* these tags -->
    <title>Stations</title>

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
    <!-- Bootstrap -->
    <link rel="stylesheet" type="text/css" href="bootstrap.min.css">

```

```

<link rel="stylesheet" type="text/css" href="customcss.css">
<!-- Icons -->
<link rel="stylesheet" href="all.min.css">

</head>
<body style="background: #3a3a3a;">

    <!-- ===== Navigation bar
===== -->
    <nav class="navbar fixed-top navbar-toggleable-sm navbar-inverse bg-inverse">
        <button class="navbar-toggler navbar-toggler-right" type="button" data-
toggle="collapse" data-target="#navbarDropdown" aria-controls="navbarDropdown"
aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <a class="navbar-brand" href="index.html">SensorData</a>
        <div class="collapse navbar-collapse" id="navbarDropdown">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item">
                    <a class="nav-link" href="index.html">Home <span class="sr-
only">(current)</span></a>
                </li>
                <li class="nav-item dropdown active">
                    <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                        Stations
                    </a>
                    <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                        <a class="dropdown-item" href="find-stations.html">Find Station</a>
                        <a class="dropdown-item" href="add-stations.html">Add station</a>
                    </div>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="about.html">About</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="contact.html">Contact</a>
                </li>
            </ul>
        </div>
    </nav>

```

```

<!--
=====
===== -->
<div class="container">
    <br>
    <br>
    <br>
    <br>
    <h1 class="text-white" align="center" type="text-white">Your message sended
Successfully!</h1>
    <h1 class="text-white" align="center" type="text-white">We will answer you
soon.</h1>
    <br>
    <h1 class="text-white" align="center" type="text-white"><i class="far fa-paper-
plane"></i></h1>
</div>

<script type="text/javascript" src="chartjs/js/tether.min.js"></script>
<script type="text/javascript" src="chartjs/js/bootstrap.min.js"></script>

</body>
</html>

```

Αρχείο jsondata.php:

```

<?php
header( 'Content-Type: application/json');

$con = mysqli_connect("127.0.0.1","root","","station");
    if (!$con){
        die("Can not connect: " . mysqli_error());
    }

//$db_selected = mysqli_select_db($con,"station");
//~~~~~
~~ eisodos formas
// Βλέπουμε αν έχει εισαχθεί τιμή
if(!isset($_POST['from_date']) && !isset($_POST['to_date']) || ($_POST['from_date']
== "" && $_POST['to_date'] == "")){
    $query = sprintf("SELECT kitID, UNIX_TIMESTAMP(Time)*1000 AS
Time, Temperature FROM temptest ORDER BY ID DESC LIMIT 10");
}

```

```

else if (isset($_POST['from_date']) && (!isset($_POST['to_date']) || $_POST['to_date']
== '')){
    $from_date = mysqli_real_escape_string($con,
$_POST['from_date']);
    $from_date = $from_date + 0; // μετατροπή του numeric string σε
int
    $query = sprintf("SELECT kitID, UNIX_TIMESTAMP(Time)*1000 AS
Time, Temperature FROM temptest WHERE UNIX_TIMESTAMP(Time)*1000 >=
'$from_date'");
}
else if (isset($_POST['to_date']) && (!isset($_POST['from_date']) ||
$_POST['from_date'] == '')){
    $to_date = mysqli_real_escape_string($con, $_POST['to_date']);
    $to_date = $to_date + 0; // μετατροπή του numeric string σε int
    $query = sprintf("SELECT kitID, UNIX_TIMESTAMP(Time)*1000 AS
Time, Temperature FROM temptest WHERE UNIX_TIMESTAMP(Time)*1000 <=
'$to_date'");
}
else if (isset($_POST['from_date']) && isset($_POST['to_date'])){
    $from_date = mysqli_real_escape_string($con,
$_POST['from_date']);
    $from_date = $from_date + 0;
    $to_date = mysqli_real_escape_string($con, $_POST['to_date']);
    $to_date = $to_date + 0;
    $query = sprintf("SELECT kitID, UNIX_TIMESTAMP(Time)*1000 AS
Time, Temperature FROM temptest WHERE UNIX_TIMESTAMP(Time)*1000 >=
'$from_date' AND UNIX_TIMESTAMP(Time)*1000 <= '$to_date'");
}
//~~~~~
~~

$result = mysqli_query($con, $query);

$data = array();
while($row = mysqli_fetch_assoc($result)){
    $data[] = $row;
}

//close connection
mysqli_close($con);

//now print the data
echo json_encode($data);
$data[] = [];
?>

```

Αρχείο jsonaccddata.php:

```
<?php
header( 'Content-Type: application/json');

$con = mysqli_connect("127.0.0.1","root","","station");
        if (!$con){
            die("Can not connect: " . mysqli_error());
        }

// $db_selected = mysqli_select_db($con, "station");
// ~~~~~ eisodos formas
~~~~~

// Βλέπουμε αν έχει εισαχθεί τιμή
if(!isset($_POST['from_date_acc']) && !isset($_POST['to_date_acc'])) ||
($_POST['from_date_acc'] == '' && $_POST['to_date_acc'] == ''){
    $query = sprintf("SELECT kitID, UNIX_TIMESTAMP(Time)*1000 AS
Time, acX, acY, acZ FROM accdata ORDER BY ID DESC LIMIT 200");
}
else if (isset($_POST['from_date_acc']) && (!isset($_POST['to_date_acc']) ||
$_POST['to_date_acc'] == '')){
    $from_date_acc = mysqli_real_escape_string($con,
$_POST['from_date_acc']);
    $from_date_acc = $from_date_acc + 0; // μετατροπή του numeric
string σε int
    $query = sprintf("SELECT kitID, UNIX_TIMESTAMP(Time)*1000 AS
Time, acX, acY, acZ FROM accdata WHERE UNIX_TIMESTAMP(Time)*1000 >=
'$from_date_acc'");
}
else if (isset($_POST['to_date_acc']) && (!isset($_POST['from_date_acc']) ||
$_POST['from_date_acc'] == '')){
    $to_date_acc = mysqli_real_escape_string($con,
$_POST['to_date_acc']);
    $to_date_acc = $to_date_acc + 0; // μετατροπή του numeric string
σε int
    $query = sprintf("SELECT kitID, UNIX_TIMESTAMP(Time)*1000 AS
Time, acX, acY, acZ FROM accdata WHERE UNIX_TIMESTAMP(Time)*1000 <=
'$to_date_acc'");
}
else if (isset($_POST['from_date_acc']) && isset($_POST['to_date_acc'])){
    $from_date_acc = mysqli_real_escape_string($con,
$_POST['from_date_acc']);
```

```

        $from_date_acc = $from_date_acc + 0;
        $to_date_acc = mysqli_real_escape_string($con,
$_POST['to_date_acc']);
        $to_date_acc = $to_date_acc + 0;
        $query = sprintf("SELECT kitID, UNIX_TIMESTAMP(Time)*1000 AS
Time, acX, acY, acZ FROM accdata WHERE UNIX_TIMESTAMP(Time)*1000 >=
'$from_date_acc' AND UNIX_TIMESTAMP(Time)*1000 <= '$to_date_acc'");
    }
    //~~~~~
    ~~~~~

$result = mysqli_query($con, $query);

$data = array();
while($row = mysqli_fetch_assoc($result)){
    $data[] = $row;
}

//close connection
mysqli_close($con);

//now print the data
echo json_encode($data);
$data[] = [];
?>

```

Αρχείο customcss.css:

```

body{
padding-top: 50px;
font-size: 16px;
}
.btn-primary {
margin-bottom: 5px;
}
/* navigation bar and dropdown custom color */
.navbar-nav > li > .dropdown-menu { background-color: #3a3a3a; }
.dropdown .dropdown-menu a{color: #c6c6c6; text-decoration: none;}

.navbar-inverse {
background-color: #333333;
color:#ffffff;
border-radius:0;

```

```

    min-height:auto;
}

.navbar-inverse .navbar-nav > .active > a, .navbar-nav > .active > a:hover, .navbar-nav
> .active > a:focus {
    color: #ffffff;
    background-color:transparent;
}

.navbar-inverse .navbar-nav > li > a:hover, .nav > li > a:focus {
    text-decoration: none;
    background-color: #3a3a3a;
}

.navbar-inverse .navbar-brand {
    color:#009b84;
    padding:none;
}

.navbar-inverse .navbar-toggle,
.navbar-inverse .nav .open>a, .navbar-inverse .nav .open>a:hover, .navbar-inverse .nav
.open>a:focus {
    background-color:transparent;
}

.navbar-inverse .icon-bar {
    border: 1px solid #fff;
}

/* carousel image responsive */
.carousel-item.active,
.carousel-item-next,
.carousel-item-prev {
    display: block;
}

/* responsive text queries */
@media screen and (max-width: 992px) {
    p,.btn,input,div,span,h4 {
        font-size: 95%;
    }
    h1 {
        font-size: 24px;
    }
    h2 {

```

```

        font-size: 22px;
    }
    h3 {
        font-size: 18px;
    }
}
.carousel-inner{
            height: auto;
        }
.carousel-inner > .item{
            height:400px;
        }
.carousel-inner > .item > img {
            margin: auto;
            width: auto;
        }

/*-----google maps-----*/
.map-container iframe{
            left:0;
            top:0;
            height:100%;
            width:100%;
            position:absolute;
        }
/*-----small screen space-----*/
@media (max-width: 767px) {
    .col-md-4 {
        margin-bottom:20px;
    }
}

/* footer */

.footer {
    background-color: #333333;
}

```

Αρχείο jsonmarkers.php:

```

<?php
header( 'Content-Type: application/json');

```

```

$con = mysqli_connect("127.0.0.1","root","","station");
    if (!$con){
        die("Can not connect: " . mysqli_error());
    }

// $db_selected = mysqli_select_db("station",$con);
// ~~~~~
~~

$query = sprintf("SELECT * FROM kits ORDER BY ID");

// ~~~~~
~~

$result = mysqli_query($con, $query) or die(mysqli_error());

$data = array();
while($row = mysqli_fetch_assoc($result)){
    $data[] = $row;
}

//close connection
mysqli_close($con);

//now print the data
echo json_encode($data);
$data[] = [];
?>

```

Αρχείο logo-load.html:

```

<!DOCTYPE HTML>

<div class="row" style="background: #777777;">
    <div class="col-md-2">
    </div>
    <div class="col-md-4">
        <br>
        <div class="card" style="width:300px">
            
            <div class="card-body">
                <h4 class="card-title">Temperature Data</h4>

```

```

        <p class="card-text">See the temperature data
visualized in charts.</p>
        <button type="button" id="temp_logo"
class="btn btn-primary" href="#after_logos_div" onclick=""><i class='fas fa-
temperature-low' style='font-size:30px'></i> <b>See Data</b></button>
    </div>
</div>
<br>
</div>
<div class="col-md-4">
    <br>
    <div class="card" style="width:300px">
        
        <div class="card-body">
            <h4 class="card-title">Accelerometer Data</h4>
            <p class="card-text">See the accelerometer data
visualized in charts.</p>
            <button type="button" id="acc_logo" class="btn
btn-primary" href="#after_logos_div" onclick=""><i class='fas fa-chart-bar' style='font-
size:30px'></i> <b>See Data</b></button>
        </div>
    </div>
    <br>
</div>
<div class="col-md-2">
</div>
</div>

```

Αρχείο ajaxformdates.js:

```

// onclick ok update chart..
function update_chart(){

    var from_date = new Date($("#from_date").val());
    var to_date = new Date($("#to_date").val());
    var mili_from = from_date.getTime();
    var mili_to = to_date.getTime();

    var form_data = {from_date:mili_from, to_date:mili_to};
    //αν θέλουμε JSON format:
    var form_data_st = JSON.stringify(form_data);

```

```
//-----

//Στέλνουμε τα input του χρήστη με Ajax στο php file.
$.ajax({
  type: "POST",
  //dataType: "json",
  //url: "formdate.php",
  url: "jsondata.php",
  data: form_data,
  //contentType: "application/json",
  success: function(){
    alert('sended!');
  },
  error: function(xhr){
    alert('Error! status = ' + xhr.status + "message = " + xhr.statusText);
  }
});
alert('JS IS UP AND RUNNING..')

$.getJSON('http://localhost/thesisb4/chartjs/jsondata.php',function(data){

  var dataPoints = [];

  for (var i = 0; i < data.length; i++) {

    dataPoints.push({ x: Number(data[i].Time), y: Number(data[i].Temperature) });
  }
  chart.options.data[0].dataPoints = dataPoints;
  chart.render();
});

}
```

Αρχείο linegraph.js:

```
function temp_btn_chart_load(id){
  //$(document).ready(function() {
    //Παίρνουμε τα δεδομένα απο το Json και δημιουργούμε το διάγραμμα με τη χρήση
    του canvasjs.
    $.getJSON('chartjs/jsondata.php',function(data){
      var dataPoints = [];
      var chart = new CanvasJS.Chart("chartContainer", {
```

```

theme: "theme3",
//backgroundColor: "#008780",
axisX:{
    //valueFormatString: "YYYY",
    labelFontColor: "#008780",
    lineColor: "#008780",
    gridDashType: "solid",
    gridThickness: 0.7,
    gridColor: "#e0e0e0"
},
axisY:{
    labelFontColor: "#008780",
    lineColor: "#008780",
    gridDashType: "solid",
    gridThickness: 0.7,
    gridColor: "#e0e0e0"
},
title:{
    text: "Temperature Diagram "
},
data: [
{
    lineColor: "#008780",
    type: "spline",
    xValueType: "dateTime",
    markerType: "none",
    connectNullData: "true",
    nullDataLineDashType: "solid",
    tooltipContent: "Time {x} : {y}",
    dataPoints: []
}
]
});

for (var i = 0; i < data.length; i++) {
    if(data[i].kitID == id){
        dataPoints.push({ x: Number(data[i].Time), y:
Number(data[i].Temperature) });
    }
}
chart.options.data[0].dataPoints = dataPoints;
chart.render();

$("#ok_button").click(function update_chart(){
    //Λαμβάνουμε τα δεδομένα του χρήστη απο τη φόρμα.

```

```

var from_date = $("#from_date").val();
var to_date = $("#to_date").val();
from_date = new Date($("#from_date").val());
to_date = new Date($("#to_date").val());

if (isNaN(from_date) && isNaN(to_date)){
    var mili_from = "";
    var mili_to = "";
    alert('Please select at least one date.');
```

}

```

else if (!isNaN(from_date) && isNaN(to_date)){
    var mili_from = from_date.getTime();
    var mili_to = "";
}

else if (isNaN(from_date) && !isNaN(to_date)){
    var mili_from = "";
    var mili_to = to_date.getTime();
}

else{
    var mili_from = from_date.getTime();
    var mili_to = to_date.getTime();
}

var form_data = {from_date:mili_from, to_date:mili_to};
//αν θέλουμε JSON format:
var form_data_st = JSON.stringify(form_data);
//-----
//Στέλνουμε τα input του χρήστη με Ajax στο php file, και παίρνουμε πίσω τα
δεδομένα απο το Json.
$.ajax({
    type: "POST",
    //dataType: "json",
    url: "chartjs/jsondata.php",
    data: form_data,
    success: function(data){
        var dataPoints = [];
        for (var i = 0; i < data.length; i++) {
            if(data[i].kitID == id){
                dataPoints.push({ x: Number(data[i].Time), y:
Number(data[i].Temperature) });
            }
        }
        chart.options.data[0].dataPoints = dataPoints;
        chart.render();
    },
    error: function(xhr){
```

```

        alert('Error 1! status = ' + xhr.status + "message = " + xhr.statusText);
    }
});
});

$("#last_day").click(function update_chart_day(){
    //Τελευταίες 24 ώρες.
    var timestamp_tod = Math.round(new Date().getTime() / 1000);
    var timestamp_yest = timestamp_tod - (24 * 3600);
    var mili_to = timestamp_tod * 1000;
    var mili_from = timestamp_yest * 1000;
    var form_data = {from_date:mili_from, to_date:mili_to};
    //αν θέλουμε JSON format:
    var form_data_st = JSON.stringify(form_data);
    //-----
    //Στέλνουμε τα input του χρήστη με Ajax στο php file, και παίρνουμε πίσω τα
    δεδομένα απο το Json.
    $.ajax({
        type: "POST",
        //dataType: "json",
        url: "chartjs/jsondata.php",
        data: form_data,
        success: function(data){
            dataPoints = [];
            for (var i = 0; i < data.length; i++) {
                if(data[i].kitID == id){
                    dataPoints.push({ x: Number(data[i].Time), y:
Number(data[i].Temperature) });
                }
            }
            chart.options.data[0].dataPoints = dataPoints;
            chart.render();
        },
        error: function(xhr){
            alert('Error 2! status = ' + xhr.status + "message = " + xhr.statusText);
        }
    });
});

$("#last_month").click(function update_chart_month(){
    //Τελευταίες 30 μέρες.
    var today = new Date();
    var last_month = new Date();
    last_month.setDate(last_month.getDate() - 30);
    var mili_to = today.getTime();

```

```

var mili_from = last_month.getTime();
var form_data = {from_date:mili_from, to_date:mili_to};
//αν θέλουμε JSON format:
var form_data_st = JSON.stringify(form_data);
//-----
//Στέλνουμε τα input του χρήστη με Ajax στο php file, και παίρνουμε πίσω τα
δεδομένα απο το Json.

```

```

$.ajax({
  type: "POST",
  //dataType: "json",
  url: "chartjs/jsondata.php",
  data: form_data,
  success: function(data){
    dataPoints = [];

    for (var i = 0; i < data.length; i++) {
      if(data[i].kitID == id){
        dataPoints.push({ x: Number(data[i].Time), y:
Number(data[i].Temperature) });
      }
    }
    chart.options.data[0].dataPoints = dataPoints;
    chart.render();
  },
  error: function(xhr){
    alert("Error 3! status = " + xhr.status + "message = " + xhr.statusText);
  }
});
});

```

```

$("#last_year").click(function update_chart_year(){
  //Τελευταίες 360 μέρες.
  var today = new Date();
  var last_year = new Date();
  last_year.setFullYear(last_year.getFullYear() - 1);
  var mili_to = today.getTime();
  var mili_from = last_year.getTime();
  var form_data = {from_date:mili_from, to_date:mili_to};
  //αν θέλουμε JSON format:
  var form_data_st = JSON.stringify(form_data);
  //-----
  //Στέλνουμε τα input του χρήστη με Ajax στο php file, και παίρνουμε πίσω τα
δεδομένα απο το Json.
  $.ajax({
    type: "POST",

```

```

//dataType: "json",
url: "chartjs/jsondata.php",
data: form_data,
success: function(data){
    dataPoints = [];

    for (var i = 0; i < data.length; i++) {
        if(data[i].kitID == id){
            dataPoints.push({ x: Number(data[i].Time), y:
Number(data[i].Temperature) });
        }
    }
    console.log(dataPoints);
    chart.options.data[0].dataPoints = dataPoints;
    chart.render();
},
error: function(xhr){
    alert('Error 4! status = ' + xhr.status + "message = " + xhr.statusText);
}
});
});
}

```

Αρχείο linegraphacc.js:

```

function acc_btn_chart_load(id){
//$(document).ready(function() {
    //Παίρνουμε τα δεδομένα απο το Json και δημιουργούμε το διάγραμμα με τη χρήση
του canvasjs.
    $.getJSON('chartjs/jsonaccdata.php',function(data){
        var dataPointsX = [];
        var dataPointsY = [];
        var dataPointsZ = [];
        var chart = new CanvasJS.Chart("accchartContainer", {

            theme: "theme3",
            zoomEnabled:true,
            animationEnabled: true,
            //backgroundColor: "#008780",
            toolTip: {
                shared: true
            },

```

```

legend: {
    cursor: "pointer",
    verticalAlign: "top",
    horizontalAlign: "center",
    dockInsidePlotArea: true,
    itemclick: toggleDataSeries
},
axisX:{
    //valueFormatString: "YYYY",
    labelFontColor: "#008780",
    lineColor: "#008780",
    gridDashType: "solid",
    gridThickness: 0.7,
    gridColor: "#e0e0e0"
},
axisY:{
    labelFontColor: "#008780",
    lineColor: "#008780",
    gridDashType: "solid",
    gridThickness: 0.7,
    gridColor: "#e0e0e0"
},
title:{
    text: "Acceleration Diagram "
},
data: [
{
    //lineColor: "#4286f4",
    type: "spline",
    xValueType: "dateTime",
    name: "X-Axis",
    showInLegend: true,
    markerType: "none",
    connectNullData: "true",
    nullDataLineDashType: "solid",
    toolTipContent: "{x} : {y}",
    dataPoints: dataPointsX
},
{
    //lineColor: "#ef6161",
    type: "spline",
    xValueType: "dateTime",
    name: "Y-Axis",
    showInLegend: true,
    markerType: "none",

```

```

        connectNullData: "true",
        nullDataLineDashType: "solid",
        toolTipContent: "{x} : {y}",
        dataPoints: dataPointsY
    },
    {
        //lineColor: "#ef9d62",
        type: "spline",
        xValueType: "dateTime",
        name: "Z-Axis",
        showInLegend: true,
        markerType: "none",
        connectNullData: "true",
        nullDataLineDashType: "solid",
        toolTipContent: "{x} : {y}",
        dataPoints: dataPointsZ
    }
]
});

for (var i = 0; i < data.length; i++) {
    if(data[i].kitID == id){
        dataPointsX.push({ x: Number(data[i].Time), y: Number(data[i].acX) });
    }
}
chart.options.data[0].dataPoints = dataPointsX;

for (var i = 0; i < data.length; i++) {
    if(data[i].kitID == id){
        dataPointsY.push({ x: Number(data[i].Time), y: Number(data[i].acY) });
    }
}
chart.options.data[1].dataPoints = dataPointsY;

for (var i = 0; i < data.length; i++) {
    if(data[i].kitID == id){
        dataPointsZ.push({ x: Number(data[i].Time), y: Number(data[i].acZ) });
    }
}
chart.options.data[2].dataPoints = dataPointsZ;

chart.render();
//===== εμφανίζει / κρυβει τις παραστασεις.
=====
function toggleDataSeries(e){

```

```

        if (typeof(e.dataSeries.visible) === "undefined" || e.dataSeries.visible) {
            e.dataSeries.visible = false;
        } else{
            e.dataSeries.visible = true;
        }
        chart.render();
    }

//=====

=====

$("#ok_button_acc").click(function update_chart_acc(){
    //Λαμβάνουμε τα δεδομένα του χρήστη απο τη φόρμα.
    var from_date_acc = $("#from_date_acc").val();
    var to_date_acc = $('#to_date_acc').val();
    from_date_acc = new Date($("#from_date_acc").val());
    to_date_acc = new Date($("#to_date_acc").val());

    if (isNaN(from_date_acc) && isNaN(to_date_acc)){
        var mili_from = "";
        var mili_to = "";
        alert('Please select at least one date.');
```

}

```

    else if (!isNaN(from_date_acc) && isNaN(to_date_acc)){
        var mili_from = from_date_acc.getTime();
        var mili_to = "";
    }
    else if(isNaN(from_date_acc) && !isNaN(to_date_acc)){
        var mili_from = "";
        var mili_to = to_date_acc.getTime();
    }
    else{
        var mili_from = from_date_acc.getTime();
        var mili_to = to_date_acc.getTime();
    }
    var form_data = {from_date_acc:mili_from, to_date_acc:mili_to};
    //αν θέλουμε JSON format:
    var form_data_st = JSON.stringify(form_data);
    //-----
    //Στέλνουμε τα input του χρήστη με Ajax στο php file, και παίρνουμε πίσω τα
    δεδομένα απο το Json.
    $.ajax({
        type: "POST",
        //dataType: "json",
        url: "chartjs/jsonaccddata.php",

```

```

data: form_data,
success: function(data){
    dataPointsX = [];
    dataPointsY = [];
    dataPointsZ = [];

    for (var i = 0; i < data.length; i++) {
        if(data[i].kitID == id){
            dataPointsX.push({ x: Number(data[i].Time), y: Number(data[i].acX)
});
        }
    }
    chart.options.data[0].dataPoints = dataPointsX;

    for (var i = 0; i < data.length; i++) {
        if(data[i].kitID == id){
            dataPointsY.push({ x: Number(data[i].Time), y: Number(data[i].acY)
});
        }
    }
    chart.options.data[1].dataPoints = dataPointsY;

    for (var i = 0; i < data.length; i++) {
        if(data[i].kitID == id){
            dataPointsZ.push({ x: Number(data[i].Time), y: Number(data[i].acZ)
});
        }
    }
    chart.options.data[2].dataPoints = dataPointsZ;

    chart.render();
},
error: function(xhr){
    alert('Error 4! status = ' + xhr.status + "message = " + xhr.statusText);
}
});
});

```

```

$("#last_day_acc").click(function update_chart_acc_day(){
    //Τελευταίες 24 ώρες.
    var timestamp_tod = Math.round(new Date().getTime() / 1000);
    var timestamp_yest = timestamp_tod - (24 * 3600);
    var mili_to = timestamp_tod * 1000;
    var mili_from = timestamp_yest * 1000;
    var form_data = {from_date_acc:mili_from, to_date_acc:mili_to};

```

```
//αν θέλουμε JSON format:
var form_data_st = JSON.stringify(form_data);
//-----
//Στέλνουμε τα input του χρήστη με Ajax στο php file, και παίρνουμε πίσω τα
δεδομένα απο το Json.
```

```
$.ajax({
    type: "POST",
    //dataType: "json",
    url: "chartjs/jsonaccddata.php",
    data: form_data,
    success: function(data){
        dataPointsX = [];
        dataPointsY = [];
        dataPointsZ = [];

        for (var i = 0; i < data.length; i++) {
            if(data[i].kitID == id){
                dataPointsX.push({ x: Number(data[i].Time), y: Number(data[i].acX)
});
            }
        }
        chart.options.data[0].dataPoints = dataPointsX;

        for (var i = 0; i < data.length; i++) {
            if(data[i].kitID == id){
                dataPointsY.push({ x: Number(data[i].Time), y: Number(data[i].acY)
});
            }
        }
        chart.options.data[1].dataPoints = dataPointsY;

        for (var i = 0; i < data.length; i++) {
            if(data[i].kitID == id){
                dataPointsZ.push({ x: Number(data[i].Time), y: Number(data[i].acZ)
});
            }
        }
        chart.options.data[2].dataPoints = dataPointsZ;

        chart.render();
    },
    error: function(xhr){
        alert('Error 5! status = ' + xhr.status + "message = " + xhr.statusText);
    }
});
```

```

});

$("#last_month_acc").click(function update_chart_acc_month(){
    //Τελευταίες 30 μέρες.
    var today = new Date();
    var last_month = new Date();
    last_month.setDate(last_month.getDate() - 30);
    var mili_to = today.getTime();
    var mili_from = last_month.getTime();
    var form_data = {from_date_acc:mili_from, to_date_acc:mili_to};
    //αν θέλουμε JSON format:
    var form_data_st = JSON.stringify(form_data);
    //-----
    //Στέλνουμε τα input του χρήστη με Ajax στο php file, και παίρνουμε πίσω τα
    δεδομένα απο το Json.
    $.ajax({
        type: "POST",
        //dataType: "json",
        url: "chartjs/jsonaccdata.php",
        data: form_data,
        success: function(data){
            dataPointsX = [];
            dataPointsY = [];
            dataPointsZ = [];

            for (var i = 0; i < data.length; i++) {
                if(data[i].kitID == id){
                    dataPointsX.push({ x: Number(data[i].Time), y: Number(data[i].acX)
});
                }
            }
            chart.options.data[0].dataPoints = dataPointsX;

            for (var i = 0; i < data.length; i++) {
                if(data[i].kitID == id){
                    dataPointsY.push({ x: Number(data[i].Time), y: Number(data[i].acY)
});
                }
            }
            chart.options.data[1].dataPoints = dataPointsY;

            for (var i = 0; i < data.length; i++) {
                if(data[i].kitID == id){
                    dataPointsZ.push({ x: Number(data[i].Time), y: Number(data[i].acZ)
});
                }
            }

```

```

    }
}
chart.options.data[2].dataPoints = dataPointsZ;

chart.render();
},
error: function(xhr){
    alert('Error 6! status = ' + xhr.status + "message = " + xhr.statusText);
}
});
});

$("#last_year_acc").click(function update_chart_acc_year(){
    //Τελευταίες 360 μέρες.
    var today = new Date();
    var last_year = new Date();
    last_year.setFullYear(last_year.getFullYear() - 1);
    var mili_to = today.getTime();
    var mili_from = last_year.getTime();
    var form_data = {from_date_acc:mili_from, to_date_acc:mili_to};
    //αν θέλουμε JSON format:
    var form_data_st = JSON.stringify(form_data);
    //-----
    //Στέλνουμε τα input του χρήστη με Ajax στο php file, και παίρνουμε πίσω τα
    δεδομένα απο το Json.
    $.ajax({
        type: "POST",
        //dataType: "json",
        url: "chartjs/jsonaccdata.php",
        data: form_data,
        success: function(data){
            dataPointsX = [];
            dataPointsY = [];
            dataPointsZ = [];

            for (var i = 0; i < data.length; i++) {
                if(data[i].kitID == id){
                    dataPointsX.push({ x: Number(data[i].Time), y: Number(data[i].acX)
                }
            }
            chart.options.data[0].dataPoints = dataPointsX;

            for (var i = 0; i < data.length; i++) {
                if(data[i].kitID == id){

```

```

        dataPointsY.push({ x: Number(data[i].Time), y: Number(data[i].acY)
    });
    }
}
chart.options.data[1].dataPoints = dataPointsY;

for (var i = 0; i < data.length; i++) {
    if(data[i].kitID == id){
        dataPointsZ.push({ x: Number(data[i].Time), y: Number(data[i].acZ)
    });
    }
}
chart.options.data[2].dataPoints = dataPointsZ;

chart.render();
},
error: function(xhr){
    alert('Error 7! status = ' + xhr.status + "message = " + xhr.statusText);
}
});
});
});
//});
}

```

7 - Βιβλιογραφία

Wang, D., Abdelzaher, T., Kaplan, L. (2015) 'Chapter 1 - A new information age', Dong Wang, Tarek Abdelzaher, Lance Kaplan (eds.) In: *Social Sensing*, Morgan Kaufmann, pp. 1-11.

Ευάγγελος Κιουντούζης, Μεθοδολογίες Ανάλυσης & Σχεδιασμού πληροφοριακών συστημάτων (3^η Έκδοση), Εκδόσεις Ε. Μπένου, Αθήνα 2009

Silberschatz, Korth, Sudarshan, Συστήματα Βάσεων Δεδομένων (4^η Έκδοση), Εκδόσεις Μ. Γκιούρδας

Luke Welling, Laura Thompson, Ανάπτυξη Web Εφαρμογών με PHP και MySQL (4^η Έκδοση), Εκδόσεις Μ. Γκιούρδας

Data Structures, Algorithms, and Applications in C++, Sartaj Sahni

P.V. Naveenkumar, S. Guruprasad, *A Study on Sensor Deployment strategies*, June 2014

H.M. Deitel, P.J. Deitel, C How to Program (3^η Έκδοση), Εκδόσεις Μ. Γκιούρδας

Stein S., Wysession M. E. An Introduction to Seismology, Earthquakes, and Earth Structure. London: Wiley-Blackwell, 2002. P. 400