ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Σχολή Τεχνολογίας

Τμήμα Ψηφιακών Συστημάτων

Σχεδίαση και Υλοποίηση Συστήματος Αυτοματισμών και IoT για εφαρμογές Πρωτογενούς Παραγωγής

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΝΤΑΤΗΣ ΚΩΝΣΤΑΝΤΊΝΟΣ (ΑΜ:3119120)

Επιβλέπων: ***Απόστολος Ξενάκης, Επίκουρος Καθηγητής***

ΛΑΡΙΣΑ, Νοέμβριος 2023

UNIVERSITY OF THESSALY

School of Technology

Digital Systems Department

Design and Implementation of Automation and IoT Systems for Primary Production Applications

BACHELOR THESIS

NTATIS KONSTANTINOS (RN: 3119120)

Supervisor: ***Apostolos Xenakis, Assistant Professor***

LARISSA, November 2023

Υπεύθυνη Δήλωση περί Ακαδημαϊκής Δεοντολογίας και Πνευματικών Δικαιωμάτων

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα πτυχιακή εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον, και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η Δηλών/ούσα

Ονοματεπώνυμο Φοιτητή

ΝΤΆΤΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Ημερομηνία

Νοέμβριος 2023

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων/πουσα	**Ονοματεπώνυμο Επιβλέποντα** Βαθμίδα/ιδιότητα επιβλέποντα, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος	**Ονοματεπώνυμο Μέλους 1** Βαθμίδα/ιδιότητα μέλους 1, Τμήμα/Ιδρυμα μέλους 1
Μέλος	**Ονοματεπώνυμο Μέλους 2** Βαθμίδα/ιδιότητα μέλους 2, Τμήμα/Ιδρυμα μέλους 2

Ημερομηνία έγκρισης: dd-mm-yyyy

Περίληψη

Η παρακάτω πτυχιακή εργασία δίνει κάποιες λύσεις στους ανθρώπους που ασχολούνται με την γεωργία . Φαίνεται η εξέλιξη που υπάρχει στον πρωτογενή τομέα σε βάθος χρόνου και αναλύεται μια τεχνολογία που δημιουργήθηκε με τις δυνατότητες του IoT για να διευκολύνει τους αγρότες , με βασικό περιεχόμενο ένα σύστημα για την συλλογή δεδομένων και λήψη αποφάσεων από ένα συγκεκριμένο περιβαλλοντικό χώρο και ενός λογισμικού που επιτρέπει στους χρήστες να έχουν τον πλήρη έλεγχο και γνώση της κατάστασης των φυτών . Αρχικά εμφανίζονται πληροφορίες σχετικά με τις ανάγκες του προβλήματος και έπειτα υπάρχει μια περιγραφή και ανάλυση του λογισμικού που έχει δημιουργηθεί μαζί με κάποια κομμάτια του κώδικα καθώς και όλες οι προκλήσεις που υπήρχαν μέχρι να επιτευχθεί το επιθυμητό αποτέλεσμα . Στο επόμενο κεφάλαιο παρουσιάζεται το σύστημα καταγραφής δεδομένων και αυτόματων αποφάσεων (Arduino) για το περιβάλλον που έχει τοποθετηθεί αυτός ο μηχανισμός καθώς και τα πραγματικά υλικά που χρειάστηκαν (αισθητήρες) . Επίσης γίνεται και μία μικρή αναφορά στο προγραμματισμό που υπήρξε για αυτό το σύστημα (Firmware) έτσι ώστε να λειτουργεί σύμφωνα με τις υπόλοιπες απαιτήσεις του λογισμικού . Λίγο πριν το τέλος βρίσκουμε τον τρόπο επίτευξης της διασύνδεσης του Hardware με το Software καθώς και το πώς επιτυγχάνεται η ενεργειακή επάρκειά του συστήματος , και με την σειρά τους φαίνονται τα συμπεράσματα αυτής της εργασίας και η βιβλιογραφία που χρειάστηκε .

Abstract

The following thesis appears to offer some solutions to those involved in agriculture. It showcases the evolution in the primary sector over time and analyzes a technology created to facilitate farmers, focusing on a system for data collection and decision-making with IoT from a specific environmental setting, along with software that allows users to have full control and knowledge of the plant conditions. Initially, it presents information regarding the needs of the problem, followed by a description and analysis of the software that has been developed, including some pieces of the code, as well as all the challenges encountered until the desired result was achieved. The next chapter introduces the data recording and automatic decision-making system (Arduino) for the environment where this mechanism has been placed, as well as the actual materials required (sensors). There is also a brief mention of the programming (Firmware) that existed for this system so that it operates according to the other software requirements. Shortly before the end, we find how the integration of Hardware with Software is achieved, as well as how the system's energy efficiency is accomplished, followed by the conclusions of this work and the bibliography . .

Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου προς τον κ. Απόστολο Ξενάκη , Επίκουρου Καθηγητή στα Ασύρματα Δίκτυα Αισθητήρων με Εφαρμογές στην Πρωτογενή Παραγωγή, για το ενδιαφέρον και τη βοήθειά του στην ολοκλήρωση της διπλωματικής εργασίας μου.

Επίσης θα ήθελα να εκφράσω τις ευχαριστίες μου προς τον κ. Αθανάσιο Ντάτη για την στήριξη και την πολύτιμη βοήθεια που μου προσέφερε για την ολοκλήρωση της διπλωματικής εργασίας .

Τέλος , θέλω από τα βάθη της καρδιάς μου να ευχαριστήσω την οικογένειά μου, για την ανεκτίμητη στήριξή τους, τόσο στην παρούσα εργασία, όσο και στην ολοκλήρωση των σπουδών μου.

Ντάτης Κωνσταντίνος

17-11-2023

Περιεχόμενα

1 Εισαγωγή

Η Σχεδίαση και Υλοποίηση Συστήματος Αυτοματισμών και IoT για εφαρμογές Πρωτογενούς Παραγωγής αναφέρεται σε ένα δίκτυο στο οποίο φυσικά στοιχεία, όπως φυτά, περιβαλλοντικά στοιχεία, εργαλεία παραγωγής και διάφορα εικονικά «αντικείμενα» στο γεωργικό σύστημα, συνδέονται με το διαδίκτυο μέσω εξοπλισμού αντίληψης γεωργικών πληροφοριών ορισμένα πρωτόκολλα για την ανταλλαγή πληροφοριών και την επικοινωνία. Σκοπεύει να βοηθήσει στην παρακολούθηση και διαχείριση γεωργικών αντικειμένων και διαδικασιών , την λήψη αποφάσεων όταν υφίσταται ανάγκη , αλλά και μελλοντικά να γίνει εισαγωγή την μηχανικής μάθησης για να μπορεί να υπάρχει πλήρης έλεγχος και πρόβλεψης της απόδοσης των κτημάτων. Η διασύνδεση «άνθρωπος-μηχανή-πράγματα» του γεωργικού IoT μπορεί να βοηθήσει τους ανθρώπους να αναγνωρίσουν, να διαχειριστούν και να ελέγξουν διάφορα γεωργικά στοιχεία, διαδικασίες και συστήματα με πιο εκλεπτυσμένο και δυναμικό τρόπο. Μπορεί επίσης να ενισχύσει σημαντικά την κατανόηση του ανθρώπου για τα βασικά μέρη της ζωής των φυτών, να βοηθήσει στην ικανότητα ελέγχου σύνθετων γεωργικών συστημάτων και να βοηθήσει στον χειρισμό αγροτικών εκτάκτων αναγκών. Επί του παρόντος, η παγκόσμια έρευνα για την αγροτική τεχνολογία IoT είναι εκτεταμένη και εντατική, αλλά οι εφαρμογές βρίσκονται γενικά στο στάδιο της πειραματικής επίδειξης. Αυτό το έγγραφο συνοψίζει συστηματικά την ερευνητική κατάσταση ενός συστήματος IoT. για γεωργία ακριβείας και την σχεδίαση και υλοποίηση ενός συστήματος συλλογής – καταγραφής και αποθήκευσης δεδομένων που μπορούν να επεξεργασθούν από ένα λογισμικό για να δώσουν την ευκαιρία στον χρήστη για πλήρη έλεγχο των χωραφιών του μέσω μίας (κυρίως) κινητής (mobile) εφαρμογής .

2 Ιστορική Αναδρομή , Στόχοι και Σκοπός

Έξυπνη Γεωργία :

Ο αγρότης δεν είχε πρόσβαση στην επιστημονική γνώση και στις σύγχρονες τεχνολογίες. Αυτό είχε ως αποτέλεσμα να χορηγεί στο χωράφι του εισροές (φάρμακα, λιπάσματα, νερό) εμπειρικά και πολλές φορές μιμούμενος τον γείτονα του συνάδελφο του παραγωγό. Συνέπεια των παραπάνω ήταν η αλόγιστη χρήση εισροών το οποίο πολλές φορές είχε αρνητικό αντίκτυπο στην τσέπη του, στα ποιοτικά χαρακτηριστικά του παραγόμενου προϊόντος και φυσικά με την περιβαλλοντική επιβάρυνση.

Η Γεωργία Ακριβείας είναι ένα σύστημα διαχείρισης αγροκτημάτων το οποίο χρησιμοποιώντας την πληροφορική και τα ηλεκτρονικά εφαρμοσμένα στη γεωργία, βοηθά τον γεωργό στη λήψη αποφάσεων για την καλύτερη διαχείριση του αγροκτήματος (Gemtos et al., 2002).Οι κύριοι στόχοι της γεωργίας ακριβείας είναι: η αύξηση της απόδοσης των καλλιεργειών ,η βελτίωση της ποιότητας των παραγόμενων προϊόντων , η πιο αποδοτική χρήση των αγρό χημικών, η εξοικονόμηση της ενέργειας και τέλος η προστασία του εδάφους και των νερών από την ρύπανση.

Στην Ελλάδα έχουν γίνει εφαρμογές σε βαμβάκι, χειμερινά σιτηρά και καλαμπόκι, αλλά και σε οπωρώνες μηλιάς, αχλαδιάς και αμπελιού. Στη χώρα μας και γενικότερα στον Ευρωπαϊκό Νότο υπάρχει μια καθυστέρηση στην εφαρμογή των συστημάτων αυτών. Αυτό αποδίδεται στις επικρατούσες συνθήκες που χαρακτηρίζονται: α) Από μικρές γεωργικές εκμεταλλεύσεις. β) Από γεωργούς με χαμηλό μορφωτικό επίπεδο. γ) Από γεωργούς προσκολλημένους στις παραδοσιακές μεθόδους παραγωγής και στις επιδοτήσεις των προϊόντων. δ) Από έλλειψη αναπτυγμένης τεχνολογίας εφαρμογής των μεθόδων Γεωργίας Ακριβείας για τις καλλιέργειες του Ευρωπαϊκού Νότου, κυρίως για τα φρούτα και λαχανικά.

Η ραγδαία κλιματική αλλαγή και ο αυξανόμενος παγκόσμιος πληθυσμός έχουν εντείνει το πρόβλημα της έλλειψης τροφίμων και ώθησαν τους ανθρώπους να κάνουν εκτενή έρευνα για την επισιτιστική ασφάλεια χρησιμοποιώντας τεχνολογία όπως τα θερμοκήπια. Η θερμοκηπιακή καλλιέργεια είναι μια από τις κύριες πηγές λαχανικών και φρούτων στον τομέα της εσωτερικής γεωργίας .Ένα θερμοκήπιο είναι μια κλειστή δομή που παρέχει ένα εφικτό περιβάλλον για την καλλιέργεια των καλλιεργειών και έναν

αποτελεσματικό τρόπο διαχείρισης των πόρων, συμπεριλαμβανομένης της ενέργειας .Ένα σύγχρονο και τεχνολογικά εξελιγμένο θερμοκήπιο μπορεί να αυξήσει τις αποδόσεις έως και 10 φορές αυτές της γεωργίας ανοιχτού αγρού . Επιπλέον, τα θερμοκήπια προσφέρουν σημαντικά πλεονεκτήματα, όπως παραγωγή καλλιεργειών όλο το χρόνο ανεξάρτητα από τις εξωτερικές κλιματικές συνθήκες, φυσική απόφραξη από επιθέσεις εντόμων και ελάχιστη κατανάλωση νερού και λιπασμάτων .

Internet of Things :

Το Διαδίκτυο των πραγμάτων (Internet of Things) αποτελεί το δίκτυο επικοινωνίας πληθώρας συσκευών, οικιακών συσκευών, αυτοκινήτων , γεωργικών καθώς και κάθε αντικειμένου που ενσωματώνει ηλεκτρονικά μέσα, λογισμικό, αισθητήρες και συνδεσιμότητα σε δίκτυο ώστε να επιτρέπεται η σύνδεση και η ανταλλαγή δεδομένων. Απλούστερα, η φιλοσοφία του IoT είναι η σύνδεση όλων των ηλεκτρονικών συσκευών μεταξύ τους (τοπικό δίκτυο) ή με δυνατότητα σύνδεσης στο διαδίκτυο (παγκόσμιο ιστό).

Βασικό χαρακτηριστικό είναι η σύνδεση όλων των αντικειμένων μεταξύ τους με απώτερο σκοπό τη δυνατότητα του χρήστη να τα ελέγχει από έναν υπολογιστή ή κινητό. Ο όρος Internet of Things αποδόθηκε τη δεκαετία του 1990 από τον Kevin Ashton.

Το Ίντερνετ των πραγμάτων (Internet of Things) είναι μία από τις τρεις κορυφαίες τεχνολογικές εξελίξεις της επόμενης δεκαετίας (μαζί με το mobile Internet και την αυτοματοποίηση του knowledge work) και αποτελεί το επόμενο μεγάλο βήμα στον χώρο της τεχνολογίας .Ο όρος Internet of Things επινοήθηκε στα τέλη της δεκαετίας του 1990 από τον επιχειρηματία Kevin Ashton.

3 ΠΕΡΙΓΡΑΦΗ ΛΟΓΙΣΜΙΚΟΥ

Όπως αναφέρθηκε και παραπάνω , μελετήθηκε πως κάθε άνθρωπος είναι πιο πρόθυμος να χρησιμοποιεί καθημερινά μια εφαρμογή στο κινητό του , από το να είναι αναγκαίο να εισέρχεται σε κάποια ιστοσελίδα για να βρει την πληροφορία που αναζητεί. Στην περίπτωση αυτή χρειάζεται να αναλογιστούμε το πόσες πλατφόρμες/συσκευές υπάρχουν στον κόσμο . Οι ευρέως γνωστές συσκευές δουλεύουν σε : Android , IOS , Web (Windows / Linux). Οπότε όταν ένας προγραμματιστής θέλει να φτιάξει μια εφαρμογή θα πρέπει να αναλογιστεί το σε τι κοινό απευθύνεται .Αν το κοινό που απευθύνεται περιέχει και τις 3 παραπάνω πλατφόρμες είναι αναγκαίο για κάθε μια από αυτές να δημιουργήσει ξεχωριστή εφαρμογή στην ανάλογη γλώσσα προγραμματισμού (συνήθως : Java & XML για Android, , Objective-C για IOS , JavaScript για Web)

Έτσι η React Native αποτελεί μια πρωτοπόρο τεχνολογία για τη δημιουργία εφαρμογών για κινητά, καθώς προσφέρει τη δυνατότητα στους προγραμματιστές να χρησιμοποιούν τη γνώση τους στη JavaScript χωρίς την ανάγκη να μάθουν τις εξειδικευμένες γλώσσες προγραμματισμού όπως η Objective-C για iOS ή Java για Android. Αυτό σημαίνει ότι η ίδια βάση κώδικα μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε διάφορες πλατφόρμες, εξασφαλίζοντας συνέπεια και μειώνοντας τον χρόνο ανάπτυξης.

Πέρα από τη φορητότητα του κώδικα, η React Native διαθέτει απόδοση σχεδόν ισοδύναμη με τις παραδοσιακές εφαρμογές που κατασκευάζονται αποκλειστικά για κάθε πλατφόρμα. Αυτό εξασφαλίζει ότι οι χρήστες απολαμβάνουν μια ομαλή και αποτελεσματική εμπειρία, ανεξαρτήτως της συσκευής που χρησιμοποιούν. Η μεγάλη και ενεργή κοινότητα του React Native είναι επίσης ένας σημαντικός παράγοντας, καθώς προσφέρει συνεχή υποστήριξη, πολλές βιβλιοθήκες και εργαλεία που βοηθούν τους προγραμματιστές να παράγουν κορυφαίες εφαρμογές.

Ωστόσο, παρ' όλες τις πλεονεκτήματα του, το React Native φέρει και ορισμένες προκλήσεις. Η συμβατότητα με τις διαφορετικές πλατφόρμες μπορεί να απαιτήσει επιπλέον δουλειά, και οι προγραμματιστές θα πρέπει να δώσουν προσοχή στις

κατευθυντήριες γραμμές και πρότυπα σχεδίασης UI της κάθε πλατφόρμας για να διασφαλίσουν την ομοιογένεια της εμπειρίας του χρήστη.

Έτσι η εφαρμογή που θα παρουσιαστεί παρακάτω είναι γραμμένη σε React Native και θα δούμε κάποια από τα δομικά της στοιχεία, τις λειτουργείες της αλλά και το τί αποτέλεσμα μας δίνει στο τέλος

C++ , Για προγραμματισμό σε Arduino:

Το Arduino χρησιμοποιεί μια παραλλαγή της C++ για την ανάπτυξη των προγραμμάτων του. Μέσω αυτής της γλώσσας, ο χρήστης μπορεί να ελέγχει τη συμπεριφορά των ηλεκτρονικών εξαρτημάτων που συνδέονται με την πλατφόρμα. Η γλώσσα αυτή προσφέρει τη δυνατότητα για σύνθετες λειτουργίες και αντικειμενοστραφή προγραμματισμό. Με την κατάλληλη εκμάθηση και πρακτική, ο προγραμματισμός σε Arduino με C++ μπορεί να ανοίξει έναν κόσμο δυνατοτήτων, από απλές εφαρμογές όπως η αυτοματοποίηση του κήπου μέχρι πιο περίπλοκα προγράμματα όπως οι ρομποτικές εφαρμογές.

SQL (Structured Query Language)

Η SQL (Structured Query Language) είναι μία γλώσσα προγραμματισμού που χρησιμοποιείται για τη διαχείριση και ανάκτηση δεδομένων από σχεσιακές βάσεις δεδομένων. Από τη δημιουργία πινάκων και σχεσιακών δομών, μέχρι την εκτέλεση ερωτημάτων και την ανάλυση δεδομένων, η SQL παρέχει τα εργαλεία για να διαχειριζόμαστε με αποτελεσματικότητα τις πληροφορίες. Είναι η βασική γλώσσα πίσω από πολλές μεγάλες εφαρμογές βάσεων δεδομένων και είναι απαραίτητη γνώση για όσους επιδιώκουν μια καριέρα στη βιομηχανία της πληροφορικής.

Και τέλος υπάρχει και η Python …:

Μία από τις πλέον δημοφιλείς γλώσσες προγραμματισμού και αυτό δεν είναι τυχαίο. Ένας από τους κύριους λόγους είναι η ικανότητά της να δημιουργεί αποτελεσματικά scripts. Είτε πρόκειται για αυτοματισμούς, είτε για διαδικαστικές διεργασίες, η Python παρέχει ένα απλό και καθαρό συντακτικό, καθιστώντας τον κώδικα ευανάγνωστο και εύκολο στη συντήρηση. Οι βιβλιοθήκες και τα frameworks που είναι διαθέσιμα για Python καλύπτουν μια ευρεία γκάμα λειτουργιών, από τον διαχειριστικό αυτοματισμό,

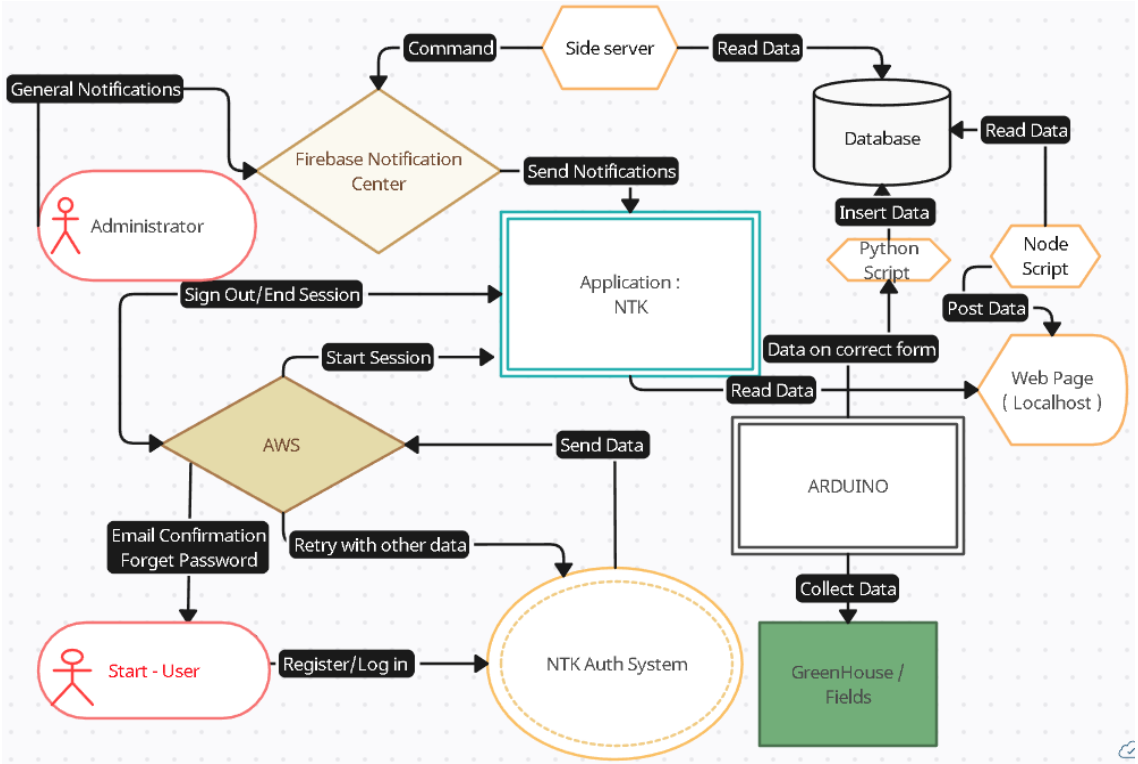
μέχρι την ανάλυση δεδομένων και το web scraping. Ως εκ τούτου, η Python είναι εξαιρετική επιλογή για όσους επιθυμούν να δημιουργήσουν scripts που ενισχύουν την παραγωγικότητα, εξοικονομώντας χρόνο και πόρους

.

3.1 System Modeling Λογισμικού

Η ενσωμάτωση του IoT στην πρακτική της γεωργίας ακριβείας αναδεικνύει την αξία της τεχνολογίας στην παραγωγική διαδικασία . Η παρούσα ανάλυση παρέχει μια σαφή εικόνα της ροής των δεδομένων και της αλληλεπίδρασης μεταξύ των διάφορων πλατφορμών, αποκαλύπτοντας τις δυνατότητες αυτοματοποίησης και βελτιστοποίησης των γεωργικών διαδικασιών. Καθώς η τεχνολογία προχωρά, αναμένεται η περαιτέρω ενσωμάτωση προηγμένων αναλυτικών εργαλείων και αλγορίθμων μηχανικής μάθησης, ανοίγοντας νέους ορίζοντές για την ακρίβεια και την προσαρμοστικότητα των γεωργικών εφαρμογών. Η εξέλιξη αυτή σηματοδοτεί μια νέα εποχή στην γεωργία όπου η πρόσβαση σε ακριβή, εγκαίρως ενημερωμένα δεδομένα και η δυνατότητα γρήγορης ανάλυσης και απόκρισης θα είναι η βάση για όλες τις γεωργικές δραστηριότητες.

Το τρέχων κεφάλαιο παρέχει μια περιεκτική επισκόπηση του τρόπου με τον οποίο η εφαρμογή NTK επιτυγχάνει αυτή τη ροή δεδομένων και αναδεικνύει την κρίσιμη σημασία της ακριβούς και αποδοτικής διαχείρισης πληροφοριών για την επίτευξη της βέλτιστης αγροτικής παραγωγής.



Εικόνα 3.1 Διάγραμμα ροής δεδομένων & διαδικασιών

Ανάλυση Συστήματος :

Η κεντρική εφαρμογή, που στην περίπτωσή μας ονομάζεται NTK, λειτουργεί ως ο εγκέφαλος του ολοκληρωμένου γεωργικού συστήματος, ενσωματώνοντας δεδομένα από πολλαπλές πηγές και επιτρέποντας την αυτοματοποιημένη λήψη αποφάσεων. Το σύστημα χωρίζεται σε τρεις βασικούς άξονες: την πλατφόρμα διαχείρισης, την πλατφόρμα συλλογής δεδομένων και την πλατφόρμα ελέγχου και ανάλυσης.

Πλατφόρμα Διαχείρισης:

Στην κορυφή της αρχιτεκτονικής βρίσκεται ο Διαχειριστής, ο οποίος είναι υπεύθυνος για την επίβλεψη και τον έλεγχο του συστήματος. Μέσω της πλατφόρμας AWS και Firbase, ο Διαχειριστής έχει την δυνατότητα να διαχειριστεί τα στοιχεία ταυτότητας των χρηστών, να εκδίδει εντολές και να στέλνει στους χρήστες γενικές ειδοποιήσεις.. Η επικοινωνία με το κέντρο ειδοποιήσεων Firebase επιτρέπει την αποστολή ειδοποιήσεων προς τους χρήστες, διασφαλίζοντας την έγκαιρη ενημέρωση για τυχόν σημαντικές αλλαγές ή ανάγκες του συστήματος..

Πλατφόρμα Συλλογής Δεδομένων:

Η συλλογή δεδομένων από το περιβάλλον του θερμοκηπίου ή των καλλιεργειών γίνεται μέσω της πλατφόρμας Arduino, η οποία είναι εξοπλισμένη με αισθητήρες για την καταγραφή κρίσιμων παραμέτρων όπως η υγρασία, η θερμοκρασία και η σύνθεση του εδάφους. .Τα δεδομένα από τους αισθητήρες αναρτώνται σε έναν τοπικό διακομιστή (localhost) μέσω ενός Node Script και εισάγονται στη βάση δεδομένων μέσω ενός Python Script, εξασφαλίζοντας την ακρίβεια και τη συνέπεια των συλλεγόμενων πληροφοριών.

Πλατφόρμα Ελέγχου και Ανάλυσης:

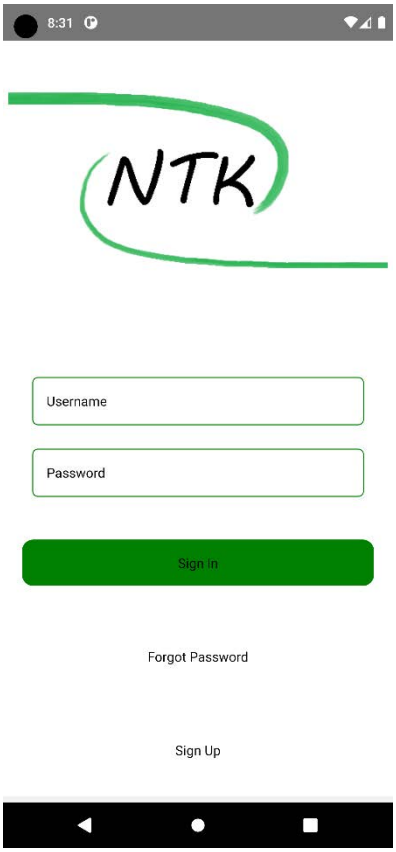
Η πλατφόρμα NTK αναλαμβάνει τον έλεγχο και την ανάλυση των δεδομένων. Μέσω του συστήματος εξουσιοδότησης NTK, οι χρήστες μπορούν να κάνουν εγγραφή και να συνδέονται στην εφαρμογή, ενώ ταυτόχρονα, το σύστημα επιτρέπει την ανάκτηση δεδομένων από τη βάση δεδομένων και την ανάλυση τους για τη λήψη αποφάσεων. Επιπρόσθετα, η διασύνδεση του Hardware με το Software διασφαλίζει την ομαλή λειτουργία του συστήματος και την εκτέλεση των εντολών που προέρχονται από τον Διαχειριστή.

Προοπτικές:

Στο μέλλον, αναμένεται ότι τα συστήματα IoT θα συνεχίσουν να εξελίσσονται, προσφέροντας ακόμη πιο ολοκληρωμένες και εξατομικευμένες λύσεις. Η ένταξη τεχνολογιών όπως η τεχνητή νοημοσύνη θα επιτρέψει την ανάπτυξη προγνωστικών μοντέλων που θα βελτιώνουν τη διαχείριση των καλλιεργειών και την πρόληψη πιθανών προβλημάτων. Η προοδευτική αυτή προσέγγιση θα ενισχύσει τη δυνατότητα των γεωργών να ανταποκρίνονται σε περιβαλλοντικές αλλαγές και να μεγιστοποιούν την αποδοτικότητα των πόρων τους. . Η σύνδεση και η συνεργασία μεταξύ των διαφόρων πλατφορμών και συστημάτων είναι κρίσιμη για την επιτυχία της γεωργίας ακριβείας. Η διασφάλιση της ομαλής ροής δεδομένων από τον τομέα της συλλογής μέχρι την ανάλυση και την εφαρμογή είναι θεμελιώδης. .

3.2 Παρουσίαση εφαρμογής και κώδικα λογισμικού

Στον κόσμο της σύγχρονης γεωργίας, η ανάγκη για προηγμένες τεχνολογίες και ακριβείς μεθόδους επεξεργασίας δεδομένων είναι εντονότερη από ποτέ. Η ενσωμάτωση του Διαδικτύου των Πραγμάτων (Internet of Things - IoT) στη γεωργική πρακτική δεν αποτελεί πλέον μια προαιρετική προσθήκη αλλά μια απαραίτητη πτυχή που εξασφαλίζει την αποδοτικότητα, τη βιωσιμότητα και την κερδοφορία των γεωργικών επιχειρήσεων. Στο πλαίσιο αυτό, η παρούσα ανάλυση επιχειρεί να αποτυπώσει τη ροή των δεδομένων μέσα από την αρχιτεκτονική ενός ολοκληρωμένου γεωργικού συστήματος βασισμένου στο IoT, αποκαλύπτοντας τον τρόπο αλληλεπίδρασης των διαφόρων πλατφορμών και την είσοδο των κρίσιμων στοιχείων.

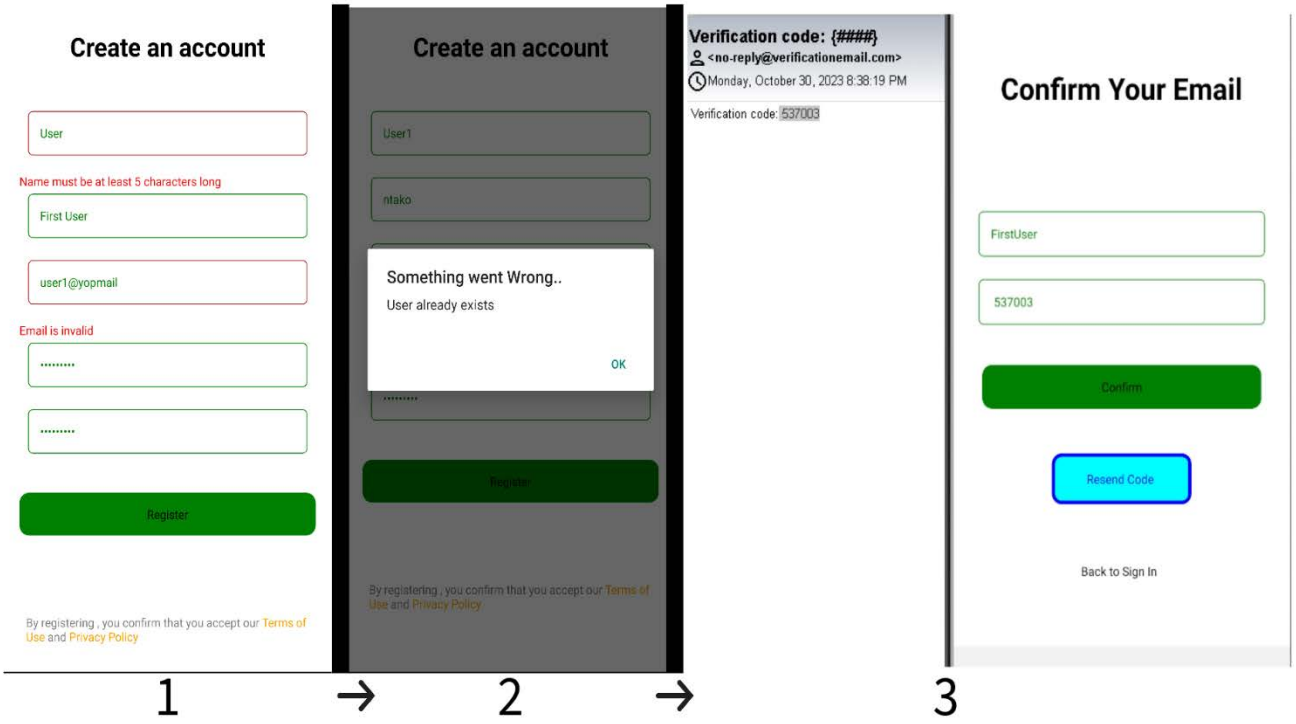


Εικόνα 3.2: Οθόνη σύνδεσης στην εφαρμογή.

Η εφαρμογή που δημιουργήθηκε έχει σκοπό να δώσει στους χρήστες που ασχολούνται με τον πρωτογενή τομέα την ευκαιρία να έχουν την απόλυτη ενημέρωση σχετικά με το κάθε φυτό που ενδιαφέρει τον κάθε χρήστη .

Αρχικά έχει ένα ολοκληρωμένο σύστημα εγγραφής και σύνδεσης και ταυτοποίησης του χρήστη στην εφαρμογή έτσι ώστε στον κάθε ένα να προβάλλονται με ασφάλεια οι πληροφορίες που τον ενδιαφέρουν (υπάρχει ένα σύστημα που παρέχει τις πληροφορίες που αναφέρονται όμως αυτό θα το μελετήσουμε πιο διεξοδικά σε

επόμενο κεφάλαιο). Αυτό το κομμάτι της εγγραφής και της σύνδεσης έχει το design (frontend) που έχω δημιουργήσει μόνος μου όμως το σύστημα αυθεντικοποίησης (το backend δηλαδή) είναι από τις εγκεκριμένες υπηρεσίες cloud της Amazon ,AWS(Amazon Web Services) , και πιο συγκεκριμένα από το AWS Amplify, Cognito & IAM. Μέσα από αυτό μπορούμε να διασφαλίσουμε στο μέγιστο την ασφάλεια για τα στοιχεία του χρήστη αλλά και για την αξιοπιστία σχετικά με την λειτουργικότητα των υπηρεσιών .



Εικόνα 3.3: Οθόνη εγγραφής & σφαλμάτων [1,2] και επιβεβαίωση e-mail [3]

Παραπάνω στην φωτογραφία [3.3.1] φαίνεται η διαδικασία εγγραφής , όπου ταυτόχρονα με την συμπλήρωση των στοιχείων είναι δυνατή η εμφάνιση των λαθών έτσι ώστε ο χρήστης να συμπληρώσει εύκολα χωρίς κόπο τα στοιχεία που απαιτούνται για την εγγραφή . Επιπλέον στην επόμενη φωτογραφία [3.3.2] εμφανίζεται η σύνδεση με την βάση δεδομένων που διαχειρίζεται τα στοιχεία των χρηστών και εμφανίζει το σφάλμα κατά την υποβολή , καθώς το όνομα υπάρχει ήδη από άλλον χρήστη .

Τέλος στην τελευταία φωτογραφία φαίνεται το email επιβεβαίωσης που στέλνεται αυτόματα κατά την υποβολή της αίτησης εγγραφής για να επιβεβαιώσει ο χρήστης το ηλεκτρονικό ταχυδρομείο του .

Κώδικας :

Εκτός από την εμφάνιση είναι αναγκαίο να δείξουμε και τον τρόπο που δημιουργήθηκε όλο αυτό . Φυσικά όπως αναφέραμε και παραπάνω ο κώδικας είναι σε React Native , και θα δείξουμε και αναλύσουμε αρκετά κομμάτια αυτού .

Αρχικά να κάνουμε ξεκάθαρο πως για να λειτουργήσουν όλα αυτά σε κάθε screen που εμφανίζουμε είναι αναγκαίο να εισάγουμε αρκετές βιβλιοθήκες κάθε φορά . Κάποιες από τις βασικές βιβλιοθήκες που σχεδόν κάθε σελίδα χρειάζεται είναι οι :

Import {View, Image, StyleSheet, ScrollView} from “react-native” ;

Αυτές είναι οι βασικές λειτουργείες που μας επιτρέπουν να εμφανίζουμε στοιχεία στην οθόνη , και ας τις αναλύσουμε λίγο παραπάνω έτσι ώστε να μπορούμε να κατανοήσουμε κάποιες εικόνες κώδικα που θα δοθούν και μελλοντικά

View : μας επιτρέπει να εμφανίζουμε στοιχεία στην οθόνη , είναι κάτι αντίστοιχο με το div που συναντάμε σε άλλες γλώσσες π.χ html.

Image : μας δίνει την δυνατότητα να εισάγουμε εικόνες στην οθόνη μας

StyleSheet : Αυτό ευθύνεται για το Styling και την μορφοποίηση της του κάθε View η ότι αλλού αντικειμένου θέλουμε .

ScrollView :Αν δεν υπήρχε αυτή η εντολή δεν θα μπορούσαμε να είχαμε την scrollable (κυλιόμενη) οθόνη και να μετακινούμαστε προς τα πάνω και κάτω .

Τώρα λίγο πιο συγκεκριμένα …

```
const onSignInPressed = async data => {
    if (loading) {
      return;
    }
    setLoading(true); //to give the "permision to press and call corectly tghe but-
ton"
    try {
      const response = await Auth.signIn(data.username, data.password);
      //console.log(response); //Shows all data on console
    } catch (e) {
      Alert.alert('Something went wrong', e.message);
    }
    setLoading(false); //So we can press again the button if we want
    // console.log(data);
  };
  const onForgotPasswordPressed = () => {
    navigation.navigate('ForgotPasswordScreen');
  };
```

```
const onSignUpPressed = () => {
  navigation.navigate('SignUpScreen');
```

Κώδικας 1.1:BackEnd για screen switching

Αυτός ο κώδικάς (1.1) είναι από το backend που γράφτηκε , για το πώς αλλάζουν οι οθόνες του κινητού ανάλογα με τις κινήσεις το χρήστη .

Αρχικά αν ο χρήστης βάλει τα στοιχεία του και πατήσει sign in τότε απευθείας μέσω της συνάρτησης που εισάγουμε από το AWS Amplify

(response = await Auth.signIn(data.username , data.password)) , στέλνονται τα στοιχεία στην βάση δεδομένων και εκεί γίνεται έλεγχος αν ταυτίζονται τα στοιχεία με κάποιον χρήστη.

Αν κάτι πάει λάθος (λάθος στοιχεία , να μην υπάρχει σύνδεση στο internet κλπ) εμφανίζεται το κατάλληλο μήνυμα όπως φαίνεται από την εντολή

Alert.alert("Something went wrong" , e.message);

Επιπλέον φαίνεται πώς γίνεται και η πλοήγηση από τις παρακάτω εντολές . Παραδείγματος χάρη αν ο χρήστης πατήσει την επιλογή Forgot Password τότε θα πλοηγήθει (navigation.navigate("ForgotPasswordScreen")) στην κατάλληλη σελίδα για να γίνει επαναφορά του κωδικού του .

Παρακάτω στον Κώδικα 1.2 ,φαίνεται το backend στο πως έχουν δημιουργηθεί όλα τα πλαίσια έτσι ώστε ο κάθε χρήστης να μπορεί να συμπληρώσει τα στοιχεία του. Έχει δημιουργηθεί ένα σύστημα με κάποιους κανόνες (rules) όπου όταν ο χρήστης αρχίζει και βάζει τα στοιχεία του αυτόματα πραγματοποιείται έλεγχος για την ορθότητα και εγκυρότητα των στοιχείων. Προσοχή καθώς εδώ δεν γίνεται σύγκριση των στοιχείων με την βάση δεδομένων για ταυτοποίηση του χρήστη . Η εφαρμογή έχει κάποιες απαιτήσεις για την μεγαλύτερη ασφάλεια ως προς την παραβίαση των προσωπικών στοιχείων του κάθε χρήστη και για αυτό παράλληλα με την συμπλήρωση των στοιχείων αν κάτι δεν καλύπτει τις απαιτήσεις της εφαρμογής εμφανίζει σε πραγματικό χρόνο ποιο είναι το σφάλμα έτσι ώστε να το διορθώσει ο χρήστης (Εικόνα 3.3[1]).. Σε περίπτωση που ο χρήστης δεν διορθώσει αυτά τα σφάλματα δεν γίνεται να πατηθεί το κουμπί για να μεταβεί στις επόμενες σελίδες .

```
<ScrollView>
      <View style={styles.root}>
        <Image source={Logo} style={styles.Logo} resizeMode="contain" />
        <CustomInput
          name="username"
          placeholder="Username"
          placeholderTextColor="black"
          control={control}
          rules={{required: 'Username is required'}}
        />
        <CustomInput
          name="password"
          placeholder="Password"
          placeholderTextColor="black"
          control={control}
          secureTextEntry={true}
          rules={{
            required: 'Password is required ',
            minLength: {
              value: 4,
              message: 'Password needs more than 4 characters',
            },
          }}
        />
        <CustomButton
          text={loading ? 'Loading...' : 'Sign In'}
          onPress={handleSubmit(onSignInPressed)}
        />
```

Κώδικας1.2 : Πλαίσια συμπλήρωσης στοιχείων

Μετά την επιτυχημένη σύνδεση ο χρήστης έρχεται αντιμέτωπος με την αρχική σελίδα (οθόνη) .

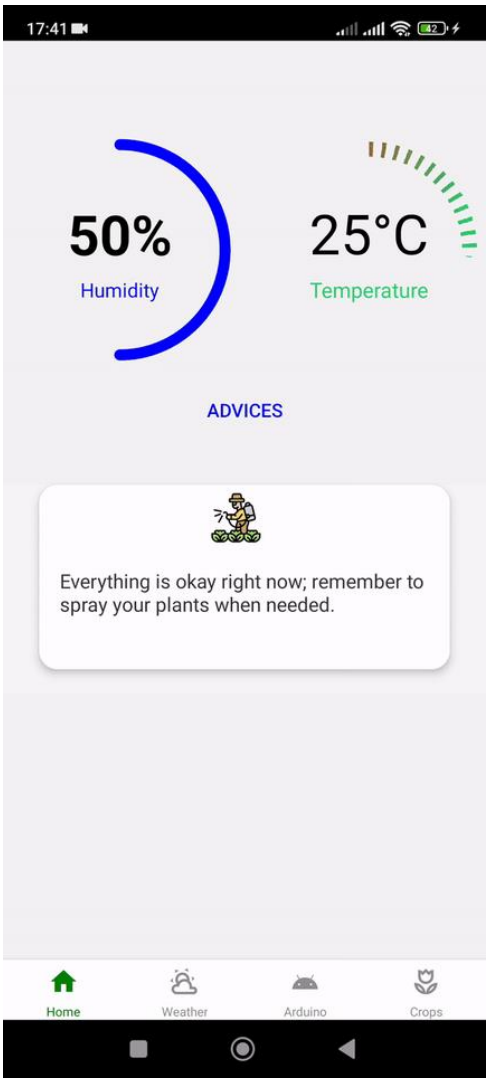
Όπως φαίνεται και στην Εικόνα:3.4 στο κάτω μέρος της οθόνης υπάρχει ένα πολύ απλό menu .Στην πραγματικότητα εδώ εμφανίζεται μόνο οι μισές επιλογές που υπάρχουν στην εφαρμογή . Αρχικά βλέπουμε τις επιλογές :

Home - Weather – Arduino – Crops .

Τις υπόλοιπες 4 επιλογές για να τις εμφανίσουμε είναι αναγκαίο να κάνουμε μια χειρονομία (gesture) , κύλιση από το αριστερό μέρος της οθόνης προς τα δεξιά

Έτσι βλέπουμε ακόμα 4 επιλογές οι οποίες είναι :

1. **To Do List**
2. **Contact Us**
3. **Settings**
4. **Sign Out**



Εικόνα 3.4:GIF από το κυλιόμενο MENU

Για να συμβεί αυτό χρειάστηκε να δημιουργηθούν 2 Μενού (BottomNavigation & DrawerNavigator).

Ας αναλύσουμε λίγο το πρώτο ..

Στο Bottom Navigaton (Κώδικας 1.2.1) έχουμε : έναν Tab.Navigator , το οποίο δημιουργεί το πλαίσιο και έχει δικές του επιλογές για styling .Μας δίνει την ευκαιρία να βάλουμε μέσω του Tab.Screen κάθε μία επιλογή του μενού όπως φαίνεται σύμφωνα με τα ονόματα που υπάρχουν. Επίσης υπάρχει και το MaterialCommunityIcons όπου είναι μια βιβλιοθήκη που περιέχει διάφορα icons (έχει γίνει λήψη από το πακέτο react-native-vector-icons/MaterialCommunityIcons)

```
BottomTabNavigator = () => {
  return (
    <Tab.Navigator
      initialRouteName="Feed"
      screenOptions={{
        screenOptions: {headerShown: false},
        tabBarActiveTintColor: 'green',
      }}>
      <Tab.Screen
        name="Home!"
        component={HomeScreen}
        options={{
          headerShown: false, //show or hide header
          tabBarLabel: 'Home',
          tabBarIcon: ({color, size}) => (
            <MaterialCommunityIcons name="home" color={color} size={size}/>
          ),
        }}
      />
```

Κώδικας 1.2.1: Βασικό Menu εφαρμογής

Έπειτα έχουμε το DrawerNavigator :

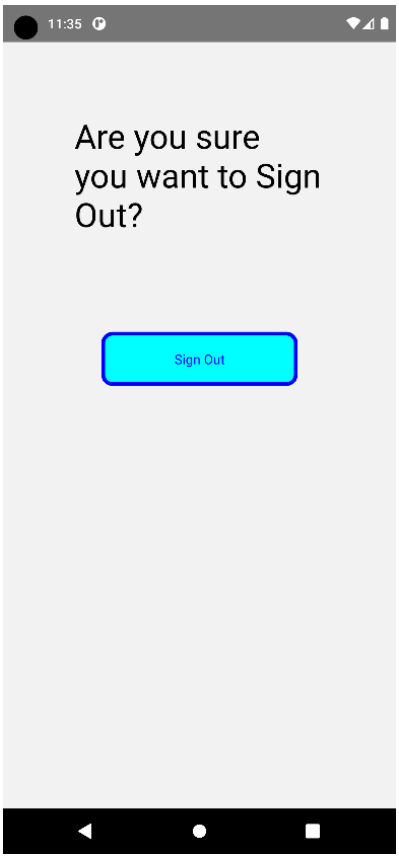
```
DrawerTabNavigator = () => {
  return (
    <Drawer.Navigator screenOptions={{headerShown: false}}>
      <Drawer.Screen name="Home" component={BottomTabNavigator} />
      <Drawer.Screen name="To Do List" component={ToDoList} />
      <Drawer.Screen name="Contact Us" component={ContactUs} />
      <Drawer.Screen name="Settings" component={ProfileScreen} />
      <Drawer.Screen name="Sign Out" component={SignOut} />
    </Drawer.Navigator>

 );

};
```

Κώδικας 1.3 :Δευτερεύων Μενού εφαρμογής (πλαινό)

Δεν χρειάζεται κάποια ανάλυση καθώς απλώς φαίνονται οι επιλογές που μας δίνονται . Ωστόσο , αξίζει να αναφερθεί πως για να γίνει πραγματικότητα η απόκρυψη – εμφάνιση του πλαϊνού μενού είναι απαραίτητη η προσθήκη της βιβλιοθήκης

'react-native-gesture-handler' που δίνει την ευκαιρία στον προγραμματιστή να βάλει και άλλες έξυπνες κινήσεις αν ο ίδιος το επιθυμεί .

Τέλος στην ίδια σελίδα κώδικα υπάρχει και η επιλογή αποσύνδεσης του χρήστη μέσω της επιλογής Sign Out. Αυτό μπορεί να πραγματοποιηθεί με την βοήθεια των επιλογών του backend της AWS , κάνοντας εισαγωγή την βιβλιοθήκη { Auth } από το AWS Amplify .



Εικόνα 3.5:Οθόνη Εξόδου

```
const SignOut = () => {
  const signOut = () => {
    Auth.signOut();
  };
  return (
    <View style={{alignItems: 'center'}}>
      <Text style={{color: 'black', fontSize: 35, padding: 75}}>
        Are you sure you want to Sign Out?</Text>
      <CustomButton text={'Sign Out'} type="SECONDARY" onPress={signOut} />
    </View>
  );
```

Κώδικας 1.4 : Έξοδος χρήστη – Τερματισμός συνεδρίας

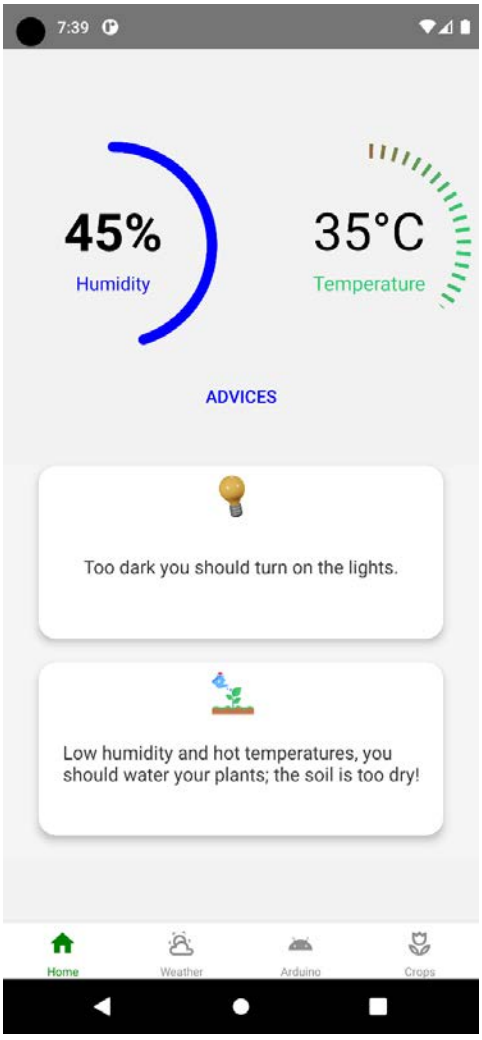
Απλώς γράφοντας την εντολή : Auth.signOut() τερματίζεται η συνεδρία που είχε ξεκινήσει κατά την είσοδο του χρήστη.

Σειρά έχει η ανάλυση της πρώτης και βασικής οθόνης που εμφανίζεται κατά την είσοδο & σύνδεση του χρήστη στην εφαρμογή.

Παρατηρούμε (Εικόνα:3.6) τους 2 νοητούς κύκλους με διαφορετικό Styling όπου ανάλογα με τις ενδείξεις που λαμβάνουμε από το σύστημα καταγραφής πληροφοριών

μας ενημερώνουν σχετικά.Το σχετικά «ιδιαίτερο » που έχει δημιουργηθεί είναι πως το χρώμα των κύκλων αλλάζει δυναμικά σύμφωνα με τις τιμές που λαμβάνονται , πχ εάν η θερμοκρασία (Temperature) πέσει κάτω από -4 °C το χρώμα γίνεται σκούρο μπλε ενώ αντιθέτως αν φτάσει σε υψηλή ακραία τιμή (πάνω από 45°C) γίνεται έντονο κόκκινο .

Από κάτω έχουμε το πεδίο των συμβουλών (Advices). Σε αυτό το σημείο γίνονται οι αναγκαίες συγκρίσεις των λαμβανόμενων δεδομένων και βγαίνουν τα κατάλληλα μηνύματα κάθε φορά έτσι ώστε ο χρήστης να κάνει τις απαραίτητες ενέργειες έτσι ώστε τα φυτά του να απολαμβάνουν τις καλύτερες δυνατές συνθήκες διαβίωσης και να μεγιστοποιήσουν την απόδοση τους κατά την παραγωγή .



Εικόνα 3.6:Βασική οθόνη προβολής δεδομένων & συμβουλών σε συνδεδεμένο χρήστη

Σε περίπτωση που κάτι πηγαίνει εντελώς λάθος (περίπτωση φωτιάς) τότε όλα τα μηνύματα εξαφανίζονται και εμφανίζεται μόνο αυτό που τον ενημερώνει άμεσα για την

πιθανότητα φωτιάς αλλά στέλνεται και το κατάλληλο notification (σε περίπτωση που υπάρχει σύνδεση στο διαδίκτυο) έτσι ώστε να υπάρχει η γρηγορότερη δυνατή ενημέρωση.

Αυτή η αρχική σελίδα έχει ρυθμιστεί να κάνει ανανέωση (refresh) κάθε 20 δευτερόλεπτα έτσι ώστε να γίνεται έγκυρη ενημέρωση των στοιχείων .

Εκτός από το UI της εφαρμογής , ας δούμε και το κομμάτι του κώδικα που χρειάστηκε :

```
const getColorForValue = value => {
  if (value >= -100 && value <= -15) {
    return '#00008B'; }
  if (value >= -14 && value <= 0) {
    return '#0000FF'; }
  if (value >= 1 && value <= 15) {
    return '#00FFFF';}
  if (value >= 16 && value <= 100) {
    return '#FF0000';}
  return 'gray'; // default color if none of the conditions are met (for safety)
};

const CustomCircularProgressTemp = ({value, title}) => (
  <CircularProgress
    value={value}
    valueSuffix={'°C'}
    radius={90}
    inActiveStrokeOpacity={0.0}
    activeStrokeWidth={15}
    inActiveStrokeWidth={20}
    progressValueStyle={{fontWeight: '400', color: 'black'}}
    inActiveStrokeColor="gray"
    duration={3000}
    dashedStrokeConfig={{
      count: 50,
      width: 4, }}
    title={title}
    titleFontSize={17}
    activeStrokeSecondaryColor={getColorForValue(value)}
  />);
```

Κώδικας 1.5 : Εμφάνιση δυναμικής αλλαγής χρωμάτων κύκλου θερμοκρασίας

Έχουμε το κομμάτι της δυναμικής αλλαγής χρώματος του κύκλου της θερμοκρασίας (Κώδικας 1.5) . Βλέπουμε πως στην αρχή υπάρχουν τα κατάλληλα if όπου κάνει τις

συγκρίσεις για το κάθε πότε χρειάζεται να γίνεται αλλαγή στο χρώμα (ξεκινάμε από την ακραία τιμή -100 και φτάνουμε ως την τιμή 100).

Έπειτα υπάρχουν οι επιλογές διαμόρφωσης του κύκλου .

Όσον αφορά το κομμάτι των Advices (Κώδικας 1.6) η διαδικασία είναι αρκετά απλή καθώς υπάρχει ένα content (ένα `κουτί`) στο οποίο υπάρχουν μέσα διάφορα if statements και κάθε φορά που κάποιο από τα statement είναι true τότε γίνεται ένα push (προώθηση) στην αρχική σελίδα

```
if (wateLevelValue < 1) {
    content.push(
      <View key="water-condition" style={styles.box}>
        <Image source={img7} style={styles.icon} />
        <Text style={styles.textInsideBox}>
          Water level is low, you should refill the water tank!.
        </Text>
      </View>,
    );
  }

  if (averageHumidityHourlyValue < 45 && averageTemperatureHourlyValue > 30) {
    content.push(
      <View key="hourly-condition" style={styles.box}>
        <Image source={img1} style={styles.icon} />
        <Text style={styles.textInsideBox}>
          You should do something , the last hour the plants are not okay...
        </Text>
      </View>,
    );
  }
```

Κώδικας 1.6 :Προώθηση συμβουλών στον χρήστη

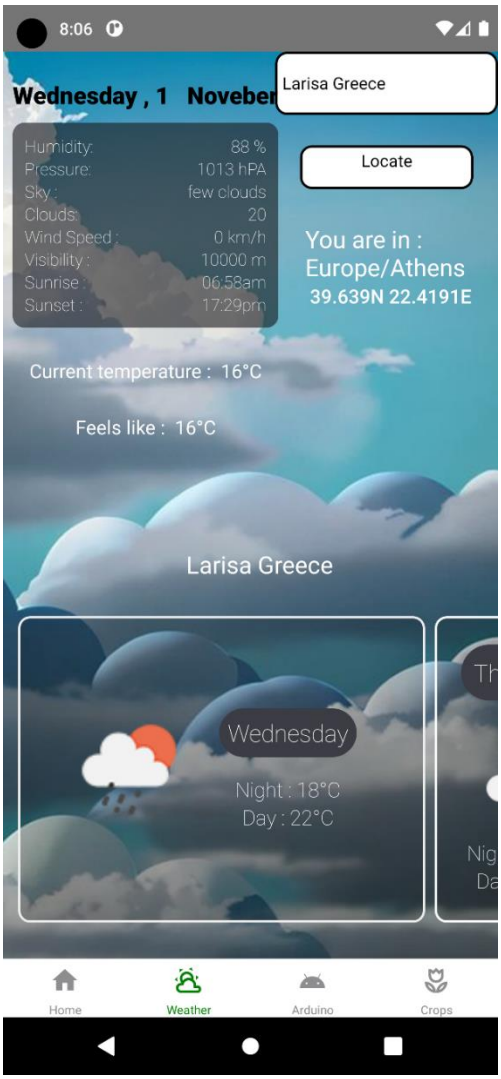
Η επόμενη επιλογή του μενού (Bottom Navigation) είναι η επιλογή Weather(Εικόνα 3.7).

Αυτή η σελίδα θα μπορούσε να είναι μια ολόκληρη εφαρμογή από μόνη της καθώς έχει ένα ολοκληρωμένο σύστημα πρόβλεψης καιρού (όπως πχ το meteo.gr) αφού μπορεί να προσφέρει πληροφορίες καιρού για όποιο σημείο του πλανήτη θέλει ο χρήστης .

Πάνω δεξιά υπάρχει το πλαίσιο που ο χρήστης δίνει το όνομα της τοποθεσίας που θέλει και πατώντας το κουμπί Locate γίνεται έλεγχος για αν η τοποθεσία είναι υπαρκτή και αν ισχύει αυτό φθάνουν αμέσως οι πληροφορίες του καιρού (Οι πληροφορίες λαμβάνονται μέσω της σελίδας OpenWeather , μέσω σχετικής άδειας που παραχωρήθηκε) .

Στο αριστερό κομμάτι διακρίνουμε τα πιο επίκαιρα στοιχεία που υπάρχουν και περιέχουν:

- Υγρασία
- Ατμοσφαιρική Πίεση
- Σύννεφα στον ουρανό /Πιθανότητα βροχής
- Ταχύτητα Αέρα
- Ορατότητα
- Τοπική ώρα ανατολής ηλίου
- Τοπική ώρα δύσης ηλίου



Εικόνα 3.7: Οθόνη πρόβλεψης καιρού

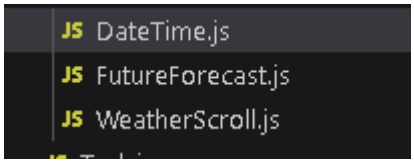
Στο κάτω μέρος της οθόνης βλέπουμε τα κουτιά που αναφέρουν τις μελλοντικές προβλέψεις του καιρού , που είναι για τις επόμενες 7 ημέρες ,όπου φαίνεται μέσω του εικονιδίου αν θα υπάρχει συνάφειά ,βροχή , ηλιοφάνεια .Προφανώς γίνεται κύλιση προς τα δεξιά για να εμφανιστούν τα στοιχεία των μεθεπόμενων ημερών.

Τι ιδιαίτερο έχει όμως αυτή η σελίδα και ήταν αναγκαίο να δημιουργηθεί και να συμπεριληφθεί εντός της εφαρμογής ;

Αυτό που αξίζει να αναφερθεί είναι στον τρόπο επιλογής τοποθεσίας .Όπως φαίνεται όταν δίνεται το όνομα της περιοχής (Στην φωτογραφία αναγράφεται η: Larisa Greece)

βλέπουμε πως εμφανίζεται η γενική τοποθεσία της εκάστοτε περιοχής αλλά και η ζώνη ώρας. .Το σημαντικό κομμάτι είναι ακριβώς από κάτω , όπου αναγράφεται το γεωγραφικό μήκος και πλάτος.. Έκτος από το όνομα μιας περιοχής δίνεται η δυνατότητα στον χρήστη να δώσει το το γεωγραφικό μήκος και πλάτος ενός σημείου της γης και να λάβει εξειδικευμένες πληροφορίες για το συγκεκριμένο σημείο . Αν αναλογιστούμε πως αυτή η εφαρμογή απευθύνεται στον πρωτογενή τομέα , είναι εύκολα αντιληπτό πως η πρόβλεψη του καιρού για ένα συγκεκριμένο σημείο είναι ιδιαίτερα σημαντική αφού τα χωράφια/αγροκτήματα τις περισσότερες φορές είναι απομακρυσμένα από πόλεις & χωριά.

Ας δούμε και την προγραμματιστική πλευρά …



Χρειάστηκε να δημιουργηθούν 3 αρχεία για να μπορούν να καλυφθούν οι ανάγκες της παραπάνω σελίδας .Αναφορικά έχουμε το αρχείο DateTime το FutureForecast και τέλος το WeatherScroll . Αυτά τα τρία είναι υπο-αρχεία του βασικού μέρους που ονομάζεται WeatherScreen.

Ας αρχίσουμε την επεξήγηση από το τελευταίο , στο αρχείο WeatherScreen γίνεται εισαγωγή της βιβλιοθήκης “react-geocode”. Αυτή έχει την δυνατότητα να παίρνει την ονομασία μιας τοποθεσίας και να την μετατρέπει σε γεωγραφικό μήκος και πλάτος (Κώδικας 1.7)

```
const [address, setAddress] = useState('');
const locateWeather = () => {
  fromAddress(address)
    .then(({results}) => {
      const {lat, lng} = results[0].geometry.location;
      fetchDataFromApi(lat, lng);
      console.log(lat, lng);
    })
    .catch(console.error);
};
```

Κώδικας 1.7 : Μετατροπή ονομασίας τοποθεσίας σε συντεταγμένες

Η μεταβλητή address είναι αυτή που δίνει ο χρήστης (όνομα τοποθεσίας). Μέσω της βιβλιοθήκης geocode , χρησιμοποιούμε την geometry.location όπου μας αποθηκεύει το latitude και το longitude (lat , lng σε μεταβλητές του κώδικα)

Από κάτω έχουμε την κλήση που γίνεται στην OpenWeather (Κώδικας 1.8)για να μας αποστείλει τα στοιχεία που θέλουμε . Έχουμε περάσει ως ορίσματα το latitude & longitude που έχουμε πάρει από παραπάνω και όπως φαίνεται μας επιστρέφεται ένα "αρχείο" σε μορφή json όπου έπειτα μπορούμε να πάρουμε από εκεί τα δεδομένα και να τα εμφανίσουμε στην σελίδα μας

```
const fetchDataFromApi = (latitude, longitude) => {
    fetch(
      `https://api.openweathermap.org/data/3.0/onecall?lat=${latitude}&lon=${longi-
tude}&exclude=hourly,minutely&units=metric&appid=${API_KEY}`,
    )
      .then(res => res.json())
      .then(data => {
        setData(data);
      });
  };
```

Κώδικας 1.8 : Κλήση API σε OpenWeather με σύστημα συντεταγμένων

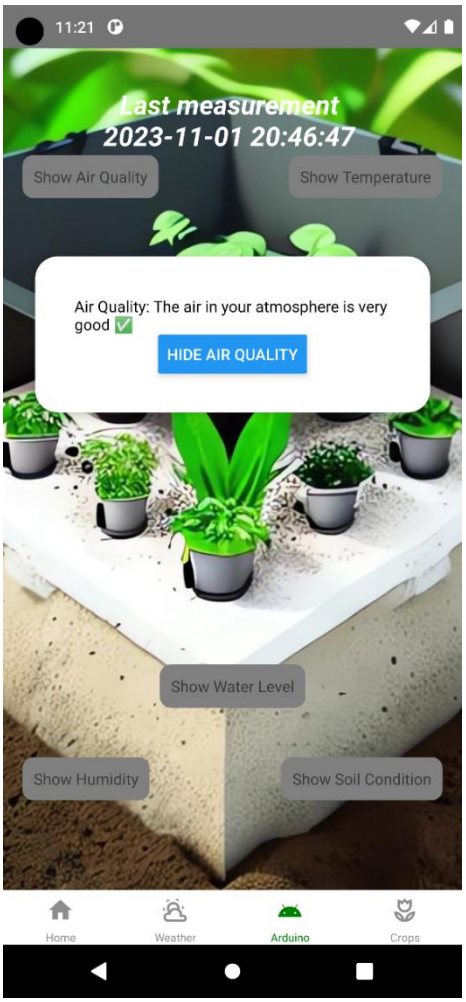
Το WeatherScreen έχει εμπεριέχει 3 αρχεία μέσα του όπου ευθύνονται για την σωστή προβολή και το styling της σελίδας και δεν έχουν ιδιαίτερο νόημα να προβληθούν καθώς το προγραμματιστικό ενδιαφέρον στερεύει .

Αμέσως μετά έχουμε την οθόνη που προβάλλονται τα στοιχεία από το σύστημα Arduino(Εικόνα 3.8) που έχουμε στήσει αλλά θα αναλύσουμε σε μελλοντικό κεφάλαιο(κεφάλαιο 4).

Προγραμματίστηκα δεν έχει κάτι ιδιαίτερο εκτός από το design και το styling της σελίδας..

Αυτό αποτελεί μελλοντικό κομμάτι για το πώς θα μπορούσε μέσω μηχανικής μάθησης και A.I(Artificial Intelligence) να παράγει από μόνο του πληροφορίες (όπως πρόβλεψη τιμών παραγωγής) και να παίρνει αποφάσεις για να επιτευχθούν οι βέλτιστες συνθήκες που χρειάζονται.

Όσο για το περιεχόμενο διακρίνουμε κάποια κουμπιά στο background.Όταν ο χρήστης πατάει σε κάποιο από αυτά εμφανίζεται ένα νέο κουτί (box) όπου είτε αναγράφει την τιμή κάποιας μέτρησης είτε κάποιο μήνυμα σχετικά με τις συνθήκες του περιβάλλοντος η του χώματος

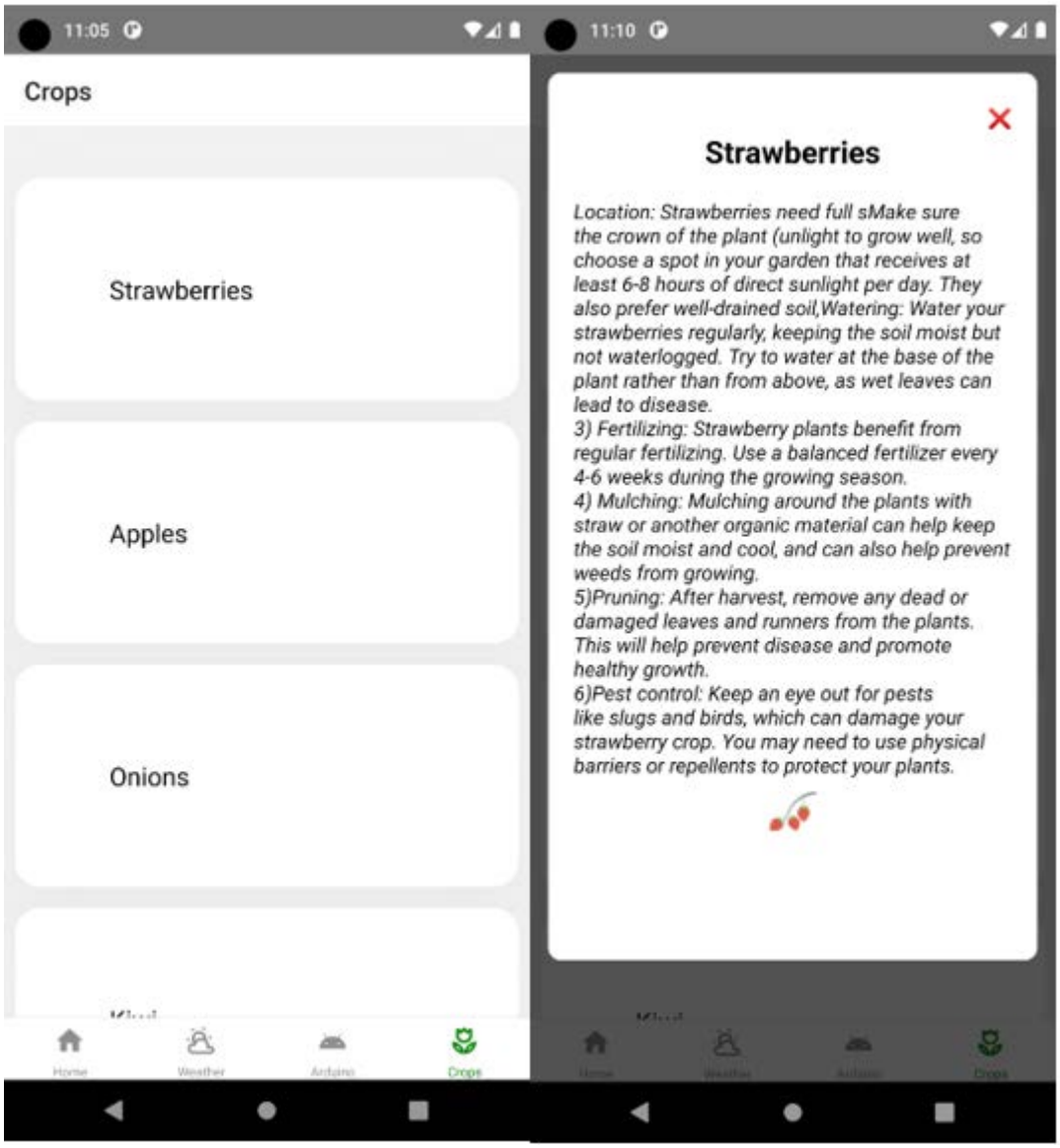


Εικόνα 3.8:Προβολή οθόνης όλων των λαμβανόμενων δεδομένων

Συνεχίζοντας , η επόμενη επιλογή στο μενού είναι η Crops (καλλιέργειες)

(Εικόνα 3.9).Εκεί ο χρήστης μπορεί να διαβάζει οδηγίες , νέα αλλά και νέες έρευνες που μπορεί να υπάρχουν έτσι ώστε να υπάρχει εύκολη και γρήγορη πρόσβαση .

Κάνοντας κλικ σε οποιαδήποτε επιλογή που ήδη υπάρχει εμφανίζεται ένα ολοκληρωμένο κείμενο με τις σχετικές πληροφορίες όπως φαίνεται και στην επόμενη φωτογραφία .

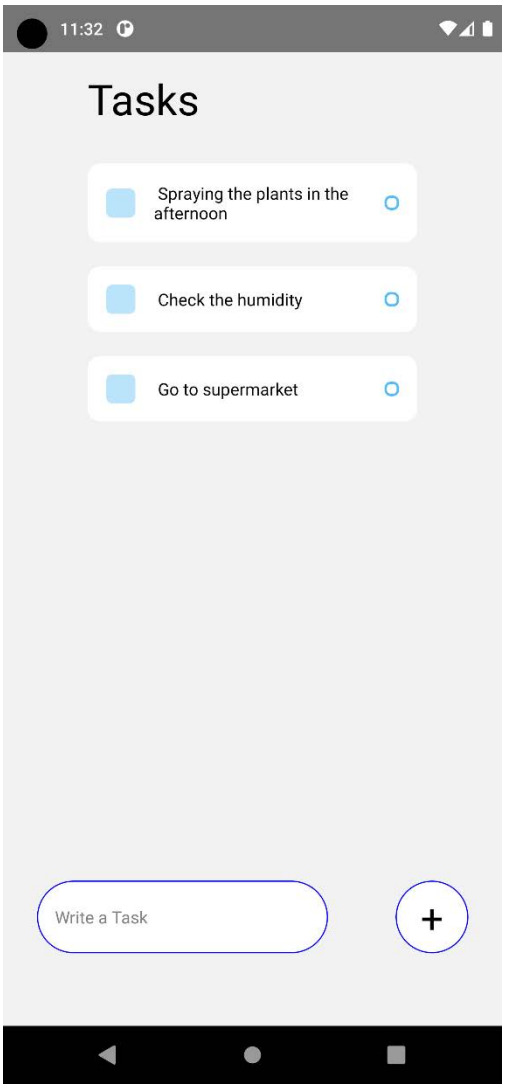


Εικόνα 3.9: Οθόνες από τις οδηγίες για κάποιες καλλιέργειές

Οι επιλογές από το Bottom Navigation τελείωσαν , οπότε σειρά έχει το Drawer Navigator

Αρχικά υπάρχει η επιλογή το To Do List (Εικόνα 3.10), που είναι ένα σημειωματάριο .Εκεί ο χρήστης μπορεί εύκολα να σημειώσει το τι εργασίες έχει να κάνει (χωρίς αυτό να είναι απαραίτητα σχετικά με την γεωργία , μπορεί να χρησιμοποιηθεί και για την καθημερινότητα του).

Τέλος εάν θέλει να αφαιρέσει αυτά που έχει "καρφιτσώσει " μπορεί ο χρήστης να πατήσει στο μπλε κύκλο που βρίσκεται στα δεξιά κάθε σημείωσης .



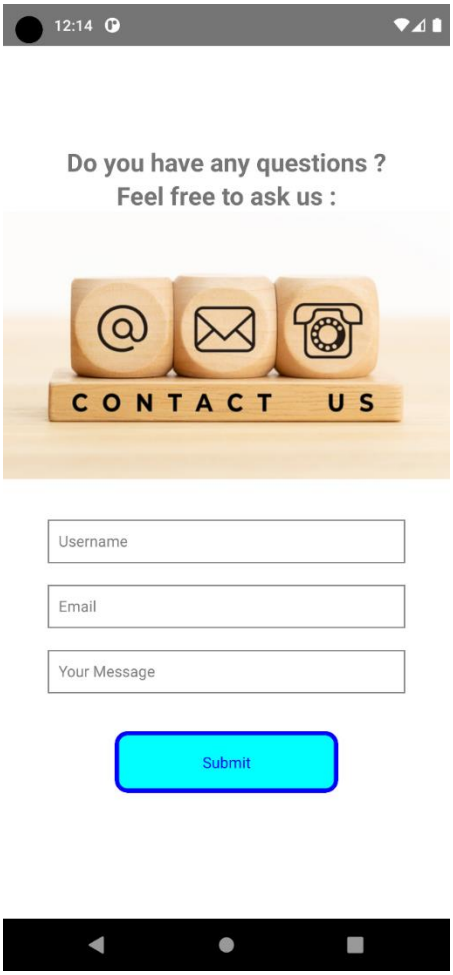
Εικόνα 3.10:Οθόνη με πεδίο σημειώσεων χρήστη

```
const ToDoList = () => {
  const [task, setTask] = useState();
  const [taskItems, setTaskItems] = useState([]);

  const handleAddTask = () => {
    Keyboard.dismiss(); // This will dismiss the keyboard after adding a task
    setTaskItems([...taskItems, task]);// This will add the new task to the array
    setTask(null);// This will clear the input after adding a task
  };

  const completeTask = index => {
    let itemsCopy = [...taskItems];// This will copy the array
    itemsCopy.splice(index, 1);// This will remove the item from the array
    setTaskItems(itemsCopy);// This will set the new array
  };
<View style={styles.items}>
        {/* Here is where the tasks will go ! */}
```

```
{taskItems.map((item, index) => {
  return (
    <TouchableOpacity key={index} onPress={() => completeTask(index)}>
```

Κώδικας 1.9: Κώδικας από οθόνη σημειωματάριου

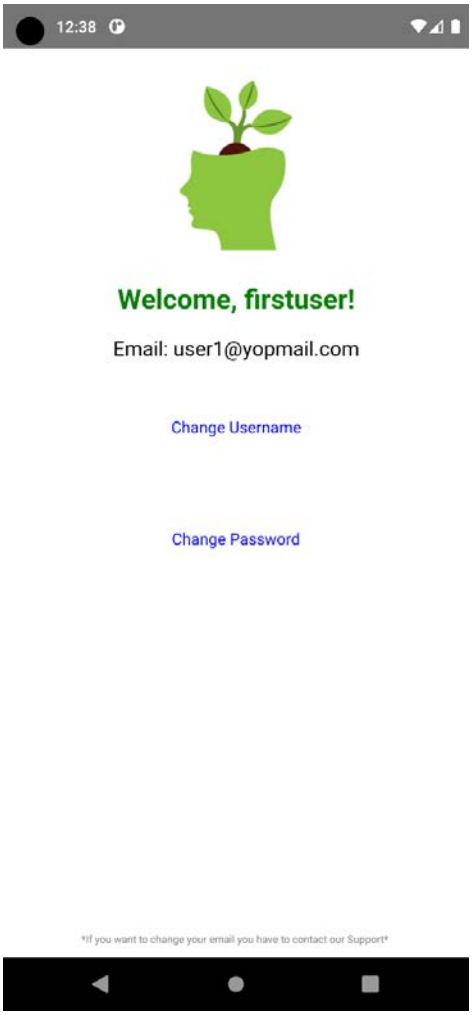
Στη συνέχεια στο μενού βρίσκουμε την επιλογή Contact Us(Εικόνα 3.11) , χωρίς να χρειάζεται να πούμε πολλά για αυτό ο κάθε χρήστης μπορεί να στείλει κάποια ερώτηση που έχει στον προγραμματιστή της εφαρμογής μέσω αυτής της σελίδας.. Ιδιαίτερα απλά τα πράγματα εδώ , 3 πλαίσια που είναι αναγκαία να συμπληρωθούν σωστά για να γίνει η αποστολή του ερωτήματος .



Εικόνα 3.11: Οθόνη επικοινωνίας χρηστών με Admin

Αξίζει να σημειώσουμε ότι σε αυτό υλοποιήθηκε κυρίως το frontend (styling) της σελίδας καθώς το μεγαλύτερο κομμάτι του backend διατίθεται από τις υπηρεσίες Cloud της AWS (Amazon Simple Email Service (SES)) , οπότε αυτό πάλι προσδίδει τεράστια επίπεδα ασφάλειας και αξιοπιστίας ως προς τον χρήστη και τον προγραμματιστή .

Λίγο πριν το τέλος βρίσκουμε τα Settings (Ρυθμίσεις)(Εικόνα 3.12) .Όπως είναι ευρέως γνωστό εκεί ο χρήστης μπορεί να διαχειριστεί τα στοιχεία του λογαριασμού του. Επειδή το project είναι σε αρχικό στάδιο , δεν ήταν απαραίτητη η προσθήκη εξιδεικευμένων αλλαγών οπότε έχουμε αρκεστεί σε μοναδικές επιλογές όπως αλλαγή Ονόματος Χρήστη & Αλλαγή Κωδικού Χρήστη .



Εικόνα 3.12:Ρυθμίσεις Profile χρήστη

Φυσικά δεν είναι μία απλή διαδικασία αφού ο χρήστης καλείται να κάνει επιβεβαίωση μέσω email των στοιχείων του πριν κάνει κάποια αλλαγή έτσι ώστε να μην τίθεται θέμα υποκλοπής στοιχείων .

*Χαμηλά βλέπουμε και ένα μήνυμα για αλλαγή του email , όμως αυτό μπορεί να γίνει μόνο μέσω ενέργειας του προγραμματιστή , συνεπώς ο χρήστης καθοδηγείται στην οθόνη του Contact Us *

Και κάπως έτσι παρουσιάστηκε η περιήγηση στην εφαρμογή από τον χρήστη αλλά και μερικά από τα έξυπνα σημεία του κώδικα για το πως λειτουργούν αυτές οι σελίδες .

Φυσικά ο κώδικας που παρουσιάστηκε δεν είναι ούτε μια σταγόνα από τον ωκεανό που χρειάστηκε για να δημιουργηθούν αυτά και είναι βέβαιο πως κάποιος που δεν έχει γνώσεις προγραμματισμού σε React Native δεν μπορεί να αντιληφθεί πλήρως τον τρόπο λειτουργείας του κώδικα .Σίγουρα όμως ο οποιασδήποτε με βασικές γνώσεις προγραμματισμού μπορεί να μπει στο κλίμα λειτουργείας και να αντιληφθεί την ιδέα και τον τρόπο σκέψης πίσω από όλα αυτά .

3.3 Προβλήματα που αντιμετωπίστηκαν

Η δημιουργία μιας εφαρμογής κινητού με το React Native είναι μια ελκυστική διαδικασία, ωστόσο, μπορεί να συνοδεύεται από ποικίλα τεχνικά εμπόδια. Αρχάριοι προγραμματιστές συχνά αντιμετωπίζουν δυσκολίες που κυμαίνονται από θέματα στο build process έως προβλήματα συμβατότητας και απαιτήσεις συστήματος.

Προβλήματα Κατά τη Διάρκεια του Build:

Οι αρχάριοι προγραμματιστές συχνά αντιμετωπίζουν προβλήματα κατά τη διάρκεια του build process. Τα build errors μπορεί να προκύπτουν λόγω λανθασμένων εξαρτήσεων, ασυμβατότητας εκδόσεων και λανθασμένης διαμόρφωσης. Επιπρόσθετα, οι αλλαγές στις εκδόσεις των APIs μπορούν να προκαλέσουν σφάλματα που απαιτούν χρονοβόρες ενημερώσεις κώδικα.

Προβλήματα με Node Modules και Cache:

Προβλήματα με τα node modules και τα cache errors είναι επίσης συχνά. Είναι κοινό ένας αρχάριος προγραμματιστής να αντιμετωπίζει περιπτώσεις όπου τα modules δεν εγκαθίστανται σωστά ή το cache του συστήματος παρουσιάζει προβλήματα, απαιτώντας επανεκκίνηση της διαδικασίας ανάπτυξης.

Αλλαγή από Expo:

Η πλατφόρμα Expo αποτελεί μια χρήσιμη εργαλειοθήκη για την ανάπτυξη εφαρμογών React Native. Ωστόσο, μερικές φορές, λόγω περιορισμών και προβλημάτων στην

επικαιροποίηση, οι προγραμματιστές αναγκάζονται να απομακρύνονται από το Expo και να προχωρούν σε native λύσεις.

Απαιτήσεις Συστήματος:

Η ανάπτυξη εφαρμογών σε React Native απαιτεί την εγκατάσταση και τη σωστή ρύθμιση εργαλείων όπως το Gradle και την Java. Αυτό μπορεί να αποτελέσει μια πρόκληση για αρχάριους προγραμματιστές, καθώς η διαδικασία εγκατάστασης και ρύθμισης μπορεί να είναι περίπλοκη και χρονοβόρα.

Συμπέρασμα:

Παρά τις προκλήσεις, η ανάπτυξη εφαρμογών με το React Native παραμένει μια δυναμική και ευέλικτη επιλογή. Οι αρχάριοι προγραμματιστές μπορούν να ωφεληθούν από την εκμάθηση και την κατανόηση των πιθανών προβλημάτων και των λύσεων τους, γεγονός που θα τους βοηθήσει να αναπτύξουν πιο αποτελεσματικά και αξιόπιστα προϊόντα.

4 Σύστημα συλλογής δεδομένων – Hardware σε Arduino

4.1 Γενική επεξήγηση

Κάθε καινοτομία, για να λειτουργήσει ομαλά, χρειάζεται πολλούς παράγοντες να εξελιχθούν. Στο συγκεκριμένο project είναι αναγκαία η τυπική αναπαράσταση ενός συστήματος καταγραφής και αποθήκευσης δεδομένων σε πραγματικό χρόνο εντός ενός μικρού χώρου (μοκέτα). Εάν επιχειρούσαμε το ίδιο σε μεγαλύτερες χωροταξικές εγκαταστάσεις, θα απαιτούνταν διαφορετικός εξοπλισμός προκειμένου να μεγιστοποιηθεί η αξιοπιστία και η απόδοση του συστήματος.

Αρχικά, θα παρουσιάσουμε τη λειτουργία και τον σκοπό του συστήματος και στη συνέχεια θα αναλύσουμε τον απαραίτητο εξοπλισμό και τον τρόπο προγραμματισμού του, ώστε να ικανοποιεί τις δικές μας ανάγκες.

Το σύστημα κατάφερε να συλλέγει και να αποθηκεύει με ακρίβεια πληροφορίες σχετικά με το περιβάλλον στο οποίο τοποθετήθηκε και, τελικά, να πραγματοποιεί αυτόματες αποφάσεις όταν οι συνθήκες απαιτούν παρέμβαση.

Το σύστημα μπορεί να καταγράφει με μεγάλη ακρίβεια δεδομένα όπως:

- Υγρασία της ατμόσφαιρας ,
- Θερμοκρασία ,
- Ποιότητα του αέρα ,
- Ειδοποίηση για κίνδυνο φωτιάς
- Υγρασία του εδάφους ,
- Ποσοστά υδατικού περιεχομένου (%),
- Ποσότητα φωτός που λαμβάνει το σύστημα

Και τελικά να μπορεί να πάρει κάποιες αποφάσεις (μέχρι στιγμής η μόνη δυνατή είναι):

- Αυτόματο πότισμα
- Αυτόματο σύστημα κατάσβεσης φωτιάς

Σε μελλοντικό κεφάλαιο θα εξετάσουμε επιπλέον χαρακτηριστικά, αλλά σε αυτό το σημείο περιγράφουμε τι έχει υλοποιηθεί σε επίπεδο μοκέτας..

4.2 Επεξήγηση και ανάλυση υλικών

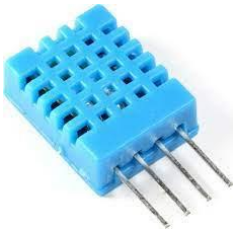
Η επιλογή του Arduino Uno R3 ως επεξεργαστική μονάδα για το πρότζεκτ προσφέρει μια πολύ καλή βάση για την πρωτογενή ανάπτυξη και τον έλεγχο των αισθητήρων. Η συγκεκριμένη πλακέτα είναι ιδανική για πρωτότυπα και μικρό-κλίμακα έργα, παρέχοντας ευκολία στην προγραμματιστική διαδικασία και στη φυσική διασύνδεση των εξαρτημάτων.

Το αρνητικό αυτής της πλακέτας είναι το κόστος (σε περίπτωση αγοράς της αυθεντικής) που το έτος 2023 ξεπερνάει τα 30 $ καθώς και ότι απουσιάζει η δυνατότητα απομακρυσμένης σύνδεσης με κάποιο δίκτυο (Wi-Fi ή Bluetooth) .

Εικόνα 4.1:Arduino Uno R3

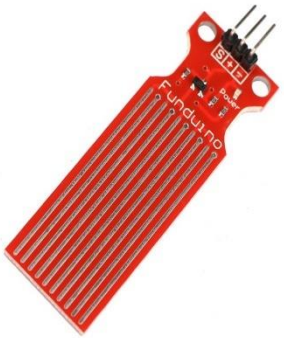
Έπειτα σειρά έχουν οι αισθητήρες .

- DHT11: Παρέχει αξιόπιστες μετρήσεις για υγρασία και θερμοκρασία, στοιχεία βασικά για τη διαχείριση περιβαλλοντικών συνθηκών. Το κόστος του αισθητήρα είναι περίπου στο 1.5 $



- ZP07-4 , Εντοπίζει και μετράει αέρια που περιέχουν CO σε διάφορα περιβάλλοντα, είναι σημαντικός για την προστασία από δηλητηριώδη αέρια αλλά και καπνό (φωτιά). Ο σχεδιασμός του επιτρέπει την ανίχνευση ακόμα και μικρών συγκεντρώσεων CO, προσφέροντας υψηλή ακρίβεια . Το κόστος του είναι περίπου στα 3.5 $

- Water Level Sensor , Ανιχνεύει το επίπεδο του νερού σε δεξαμενές, βοηθώντας στην πρόληψη υπερχειλίσεων ή στη διαχείριση πόρων ύδατος. Κατάλληλος για εφαρμογές αυτοματισμού, όπως συστήματα ποτίσματος και έλεγχος υδροδότησης.. Το κόστος είναι ιδιαίτερα χαμηλό αφού κάποιος μπορεί να τον βρει στην τιμή γύρω στο 1 $.



- Soil Moisture Sensor , Ελέγχει την υγρασία του εδάφους, ζωτικής σημασίας για την καλλιέργεια και διαχείριση φυτών.Παρέχει ακριβείς μετρήσεις, καθιστώντας το εργαλείο πολύτιμο για την αγροτική τεχνολογία και την έρευνα εδαφώνΕνσωματώνεται σε συστήματα αυτοματοποιημένου ποτίσματος, διασφαλίζοντας την ιδανική υγρασία για κάθε φυτό.Ακριβώς όπως και παραπάνω το κόστος είναι περίπου στο 1 $.

- Light Resistor , Μετρά την ένταση του φωτός στο περιβάλλον, ένας κρίσιμος παράγοντας για συστήματα ελέγχου φωτισμού. Ιδανικός για προσαρμογές σε συνθήκες φωτός, προς5φέροντας δυνατότητες για εφαρμογές σε αγροτική τεχνολογία και έξυπνα σπίτια. .Το κόστος για αυτό είναι το μικρότερο από όλα τα παραπάνω και δεν ξεπερνάει τα 0.05 $ ανά κομμάτι .

- Τέλος έχουμε ένα Water Pulmb , αυτό είναι μια αντλία νερού με δυνατότητα εξολοκλήρου βύθισης στο νερό .Αυτό είναι το μόνο μηχανικό μέρος που υπάρχει στην μοκέτα που λειτουργεί αυτόνομα . Το κόστος είναι γύρω στα 3 $.

Η προοπτική χρήσης της πλακέτας ESP32 στο μέλλον εξετάζεται λόγω των σημαντικά ανωτέρων χαρακτηριστικών της σε σχέση με το Arduino Uno R3. Η ESP32 προσφέρει ενσωματωμένη υποστήριξη για Wi-Fi και Bluetooth, καθώς και μεγαλύτερη επεξεργαστική ισχύ, περισσότερες ψηφιακές και αναλογικές είσοδοι/έξοδοι και αυξημένες δυνατότητες ασφάλειας με ενσωματωμένες κρυπτογραφικές λειτουργίες. Αν και δεν έχει γίνει χρήση της στην τρέχουσα φάση, η ESP32 αποτελεί μια υποσχόμενη επιλογή για μελλοντικές επεκτάσεις, δίνοντας τη δυνατότητα για πιο σύνθετες και αξιόπιστες εφαρμογές.

Εικόνα 4.2:ESP32

Σίγουρα υπάρχουν πολλές επιλογές αισθητήρων που ο κάθε ένας θα μπορούσε να επιλέξει κάποιον άλλον όμως οι συγκεκριμένοι επιλέχθηκαν σύμφωνα με την διαθεσιμότητα εντός Ελλάδας αλλά και κόστους . Αν υπολογίσουμε το κόστος αυτών μαζί με τα περιφερειακά που χρειάστηκαν (breadboard , καλώδια , Relay 5V, resistors ,led) για να υλοποιηθεί το σύστημα έχουμε ένα κόστος γύρω στα 50 $.

4.3 Προγραμματισμός Συστήματος

Η εν λόγω διαδικασία υλοποιήθηκε με τη χρήση της γλώσσας προγραμματισμού C++. Η επιλογή αυτής της γλώσσας δικαιολογείται από την ευρύτητα και τη διαθεσιμότητα υποστηρικτικών πόρων, καθώς και την εκτεταμένη διαδικτυακή κοινότητα που προσφέρει πληθώρα οδηγών και βοηθημάτων. Η προσέγγιση του προγραμματισμού κάθε αισθητήρα ήταν στοχευμένη , με συγκεκριμένες εντολές απαιτούμενες για την αρχική λειτουργία τους, ενώ η επιπλέον παραμετροποίηση προσαρμόζεται στις εκάστοτε απαιτήσεις της εκάστοτε εφαρμογής. Έτσι, η ανάπτυξη του προγραμματισμού αποδείχθηκε μια ευθύγραμμη διαδικασία, που διευκολύνεται αισθητά από την υποστηρικτική δομή της γλώσσας και τη διαθέσιμη κοινότητα.

Παρακάτω παρουσιάζεται η κύρια λειτουργικότητα του κώδικα που αναπτύξαμε.

Αρχικά, ορίζουμε τους ακροδέκτες που θα χρησιμοποιήσουμε για τη σύνδεση των αισθητήρων και των εκτελεστικών μηχανισμών με το Arduino(Κώδικας 2.1):

```
const uint8_t DHT_PIN = 13;                 // Πινακίδα για τον αισθητήρα
DHT
const uint8_t SOIL_MOISTURE_PIN = 6;        // Πινακίδα για τον αισθητήρα
υγρασίας του εδάφους
const uint8_t RELAY_PIN = 7;                // Πινακίδα για τον έλεγχο του
ρελέ
```

Κώδικας 2.1 :Δήλωση θέσεων αισθητήρων σε Arduino

Στην καρδιά του συστήματος, η loop() διαχειρίζεται τη λήψη μετρήσεων και τον έλεγχο των συσκευών βάσει αυτών των μετρήσεων(Κώδικας 2.2):

```
void loop() {
  // Διαβάζουμε τις τιμές από τους αισθητήρες...
  int soil_condition = digitalRead(SOIL_MOISTURE_PIN);  // Υγρασία εδάφους

  // Αποφασίζουμε αν χρειάζεται να ενεργοποιήσουμε την άντληση νερού
  if (soil_condition == HIGH && scaledLevel > 2) {
    digitalWrite(RELAY_PIN, LOW);           // Ενεργοποίηση της αντλίας
  } else {
    digitalWrite(RELAY_PIN, HIGH);          // Απενεργοποίηση της αντλίας
  }
  // ...
}
```

Κώδικας 2.2: Λήψη και διαχείριση τιμών από μετρήσεις αισθητήρων

Η κατάσταση κάθε παραμέτρου (π.χ., υγρασία, θερμοκρασία, ποιότητα αέρα) εμφανίζεται στην οθόνη LCD και μέσω της σειριακής θύρας, επιτρέποντας την εύκολη παρακολούθηση και διαχείριση του θερμοκηπίου:

```
lcd.print("Temp: ");  // Εμφανίζουμε τη θερμοκρασία στο LCD
lcd.print(tempC);
```

Συμπέρασμα Η ανάπτυξη αυτού του κώδικα επιτρέπει στον χρήστη να επιτηρεί και να διαχειρίζεται τις συνθήκες ενός θερμοκηπίου με ακρίβεια και αυτοματισμό, βελτιστοποιώντας το περιβάλλον για την ανάπτυξη των φυτών. Ο αυτοματισμός αυτός μειώνει την ανάγκη για συνεχή φυσική επιθεώρηση και επέμβαση, καθώς οι σημαντικές λειτουργίες ελέγχονται ηλεκτρονικά

4.3.1 Δυσκολίες στην Εγκατάσταση του Συστήματος

Η σωστή εγκατάσταση ενός αισθητήρα είναι μια λεπτομερής διαδικασία που ξεκινά με την προσεκτική σύνδεση των καλωδίων. Κάθε αισθητήρας απαιτεί τη συνδεσμολογία τουλάχιστον τριών καλωδίων, η οποία πρέπει να γίνει με ακρίβεια για να εξασφαλιστεί η αξιοπιστία και η σταθερή λειτουργία του. Ενώ η διαδικασία αυτή δεν ενέχει κίνδυνο για ατυχήματα, ένα λάθος στην καλωδίωση μπορεί να προκαλέσει ζημιές στον αισθητήρα ή να εμποδίσει την εύρυθμη λειτουργία του. Επομένως, η ενέργεια της εγκατάστασης απαιτεί προσοχή και μεθοδικότητα για να διασφαλίσει ότι η ηλεκτρική τροφοδοσία και οι συνδέσεις είναι σύμφωνες με τις τεχνικές προδιαγραφές του εκάστοτε αισθητήρα.

Εάν δεν προηγηθεί λεπτομερής έρευνα πριν από την εγκατάσταση των αισθητήρων, αυξάνεται ο κίνδυνος υλικών βλαβών. Η επιλογή της σωστής τροφοδοσίας είναι επίσης ένα κρίσιμο στάδιο, καθώς κάθε αισθητήρας απαιτεί συγκεκριμένη ισχύ και προδιαγραφές. Μια ενδεικτική περίπτωση είναι ο αισθητήρας DHT11, ο οποίος μπορεί να υποστεί βλάβη ή ακόμα να προκαλέσει υπερθέρμανση και φωτιά εάν τον τροφοδοτήσουμε με τάση υψηλότερη των 3.3V. Η πρόληψη τέτοιων περιστατικών είναι δυνατή μόνο μέσω της μεθοδικής και αναλυτικής μελέτης των τεχνικών χαρακτηριστικών.

Πρόσθετα, η σωστή διαχείριση της καλωδίωσης αντιπροσωπεύει μια σημαντική πρόκληση. Η ακριβής συνδεσμολογία των καλωδίων απαιτεί προσοχή, προκειμένου να αποφευχθούν πιθανά λάθη στις συνδέσεις που θα μπορούσαν να επηρεάσουν την ακρίβεια των μετρήσεων. Αν και δεν υφίσταται άμεσος κίνδυνος ατυχημάτων, λανθασμένες συνδέσεις μπορούν να υπονομεύσουν την αξιοπιστία του συστήματος.

.

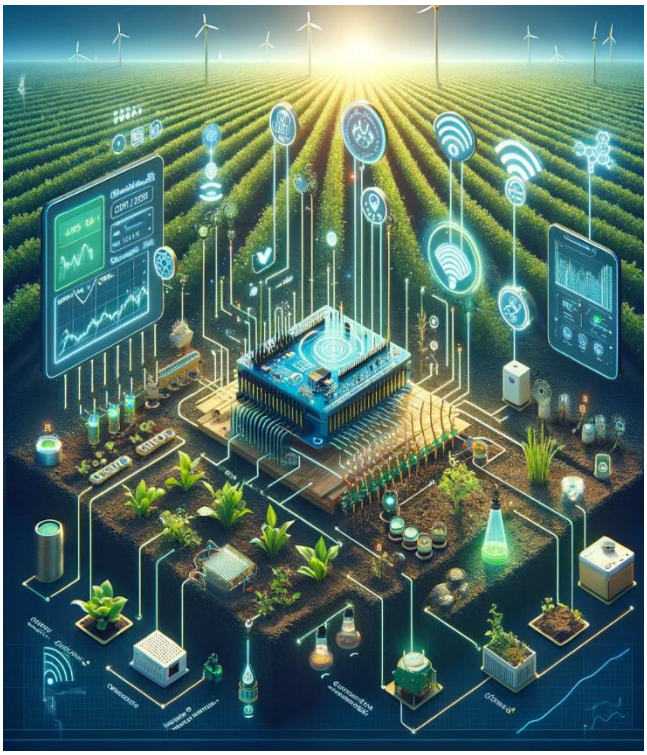
5 Διασύνδεση Εφαρμογής -Υλικών

5.1 Θεωρητικοί Τρόποι διασύνδεσης

Τα πλακίδια Arduino από μόνα τους δεν διαθέτουν ασύρματη δυνατότητα, η ενσωμάτωση με ασύρματες μονάδες όπως τα ESP32 και ESP8266 μπορεί να δώσει τη δυνατότητα τηλεπικοινωνίας και απομακρυσμένου ελέγχου. Το κύριο επίτευγμα τέτοιων συστημάτων είναι η συλλογή, αποθήκευση, και παρουσίαση δεδομένων στον χρήστη μέσω μιας διαισθητικής εφαρμογής, επιτρέποντάς του να λαμβάνει ενημερωμένες αποφάσεις σχετικά με τη διαχείριση των φυτών του.

Η εφαρμογή που έχει αναπτυχθεί λειτουργεί σαν ένας γέφυρα μεταξύ των τεχνολογικά προηγμένων αισθητήρων που είναι τοποθετημένοι στο περιβάλλον των φυτών και του τελικού χρήστη, που ενδέχεται να βρίσκεται χιλιόμετρα μακριά. Οι προκλήσεις που πρέπει να αντιμετωπιστούν σε ένα τέτοιο σύστημα είναι πολλαπλές: από την ασφαλή και αξιόπιστη μετάδοση δεδομένων μέχρι την εγγυημένη διατήρηση των δεδομένων σε περίπτωση διακοπής σύνδεσης. Αυτό το κομμάτι στοχεύει να παρουσιάσει την θεωρητική υλοποίηση αλλά και την υπάρχουσα αρχιτεκτονική που θα δούμε στο επόμενο κεφάλαιο , να αναλύσει τις τεχνολογικές επιλογές και να προτείνει βελτιώσεις για την απρόσκοπτη λειτουργία του συνόλου.

Εικόνα 5.1Εικονική εικόνα για την διασύνδεση Software & Hardware



Εικόνα 5.2.1Εικονική εικόνα για την διασύνδεση Software & Hardware

Η αναλυτική διερεύνηση θα ξεκινήσει με μια περιγραφή του συστήματος και της εφαρμογής που έχει δημιουργηθεί, προχωρώντας στην ανάλυση της επικοινωνίας μεταξύ των συστατικών μερών και καταλήγοντας στις προτεινόμενες λύσεις για τη βέλτιστη λειτουργία του συνόλου. Το τελικό μέρος θα αναφέρεται στις προκλήσεις που συναντάμε και τις στρατηγικές αντιμετώπισης τους, προσφέροντας ένα ολοκληρωμένο πλάνο για την αξιοποίηση και τη βελτίωση της παρακολούθησης φυτών μέσω IoT. .

Η εφαρμογή έχει σχεδιαστεί με γνώμονα την ευκολία χρήσης και την ευχρηστία. Διαθέτει έναν καθαρό και οργανωμένο User Interface (UI), που επιτρέπει στους χρήστες να βλέπουν γρήγορα τις πιο σημαντικές πληροφορίες και να διενεργούν αναλύσεις με βάση τα συλλεγόμενα δεδομένα. Επιπλέον, η εφαρμογή παρέχει ειδοποιήσεις σε περίπτωση που κάποιες μετρήσεις βγουν εκτός των προκαθορισμένων ορίων, επιτρέποντας έτσι την άμεση αντίδραση και προσαρμογή των συνθηκών περίθαλψης των φυτών.

Το σύστημα αυτό, ενώ είναι εξαιρετικά λειτουργικό και αποτελεσματικό, αντιμετωπίζει σημαντικές προκλήσεις, όπως η ανάγκη για σταθερή και αξιόπιστη σύνδεση στο Internet, η διαχείριση της ενέργειας για τους αισθητήρες και την ασύρματη μονάδα, και η ασφάλεια των μεταδιδόμενων δεδομένων. Οι προτεινόμενες λύσεις θα εστιάσουν στην επίλυση αυτών των ζητημάτων, ενώ θα προτείνουν τρόπους για τη βελτίωση της συνολικής απόδοσης και της εμπειρίας του χρήστη.

5.2 Τεχνικές Προκλήσεις και Διαδικασίες Επικοινωνίας

Η σχεδίαση και η υλοποίηση ενός ολοκληρωμένου συστήματος επικοινωνίας που περιλαμβάνει αισθητήρες, μικρό ελεγκτές και ασύρματη μετάδοση δεδομένων είναι μια περίπλοκη διαδικασία με πολλαπλές τεχνικές προκλήσεις. Αυτές περιλαμβάνουν την αξιοπιστία της επικοινωνίας, τη διαχείριση της ενέργειας, την ακεραιότητα και την ασφάλεια των δεδομένων, και την ομαλή ένταξη των διάφορων υποσυστημάτων.

- **Αξιοπιστία της Επικοινωνίας**: Η διακοπή της επικοινωνίας μπορεί να οδηγήσει σε απώλεια σημαντικών δεδομένων και να παραβιάσει την έγκαιρη αντίδραση στις ανάγκες του θερμοκηπίου .
- **Διαχείριση της Ενέργειας**: Η ανεξαρτησία των αισθητήρων από συνεχή πηγές τροφοδοσίας επιβάλλει τη βελτιστοποίηση της κατανάλωσης ενέργειας. Το σύστημα χρησιμοποιεί μηχανισμούς ύπνου (sleep modes) για τους αισθητήρες και τις ασύρματες μονάδες, ξυπνώντας μόνο για την αποστολή μετρήσεων ανά δύο λεπτά , μειώνοντας έτσι την ενέργεια που απαιτείται. Επίσης η χρήση ενναλακτικών πηγών ενέργειας βοηθά στο να παραμένει το σύστημα 100 % αυτόνομο και να λειτουργεί ανεξαρτήτως συνεχούς παροχής ρεύματος ή όχι
- **Ασφάλεια Δεδομένων:** Τα δεδομένα που μεταδίδονται μπορεί να είναι ευαίσθητα ή να αποτελούν στόχο για επιθέσεις. Για την προστασία τους, εφαρμόζονται πρωτόκολλα κρυπτογράφησης και ασφαλείς διαδικασίες ελέγχου ταυτότητας και αυθεντικότητας.
- **Ομαλή Ένταξη των Υποσυστημάτων:** Η συνέργεια μεταξύ των αισθητήρων, των μικρό ελεγκτών και του λογισμικού που συλλέγει και αναλύει τα δεδομένα είναι κρίσιμη. Η αρχιτεκτονική του συστήματος επιτρέπει την εύκολη ενσωμάτωση και αναβάθμιση των επιμέρους στοιχείων.

6 Σύνδεση Arduino με Υπολογιστή Μέσω Ενσύρματης Επικοινωνίας

Η ενσύρματη επικοινωνία μεταξύ του Arduino και ενός υπολογιστή προσφέρει έναν αξιόπιστο και ασφαλή τρόπο για τη μεταφορά δεδομένων. Χρησιμοποιώντας τη σειριακή θύρα USB του Arduino, μπορούμε να κατασκευάσουμε ένα σταθερό κανάλι επικοινωνίας για την άμεση μετάδοση δεδομένων από τους αισθητήρες στην κεντρική επεξεργασία δεδομένων.

Αρχικά, η συνδεσιμότητα καθιερώνεται μέσω της ρύθμισης του Arduino IDE και της δημιουργίας σειριακής πύλης που θα επιτρέπει στον υπολογιστή να αναγνωρίζει το Arduino ως μια σειριακή συσκευή. Αυτή η μέθοδος είναι ιδιαίτερα ωφέλιμη σε περιβάλλοντα όπου η ασύρματη επικοινωνία μπορεί να διαταραχθεί ή να μην είναι δυνατή.

6.1 Χρήση Script για Αυτοματοποιημένη Μεταφόρτωση Δεδομένων

Εικόνα 6.1Python σε Arduino

Μετά τη σύσταση της φυσικής σύνδεσης, το επόμενο βήμα είναι η ανάπτυξη και χρήση scripts που θα διαχειρίζονται την παραλαβή των δεδομένων από το Arduino και την αναβάθμισή τους σε μια βάση δεδομένων. Αυτά τα scripts μπορούν να γραφτούν σε γλώσσες όπως Python, JavaScript (μέσω Node.js) ή ακόμη και σε bash shell, ανάλογα με το λειτουργικό σύστημα του υπολογιστή και τις προτιμήσεις του χρήστη. Στην δική μας περίπτωση χρησιμοποιούμε την γλώσσα Python και JavaScript(Node.js) που θα δούμε παρακάτω .Τα scripts αυτά θα αναλάβουν να ανοίξουν τη σειριακή θύρα, να διαβάσουν τα εισερχόμενα δεδομένα, να τα επεξεργαστούν αν είναι απαραίτητο (π.χ., μετατροπή από σειριακή μορφή σε

αναγνωρίσιμες μονάδες μέτρησης), και τέλος, να τα μεταφέρουν στη βάση δεδομένων μέσω της υλοποίησης κατάλληλων δικτύων (API ή απευθείας SQL κλήσεων.)

Για την βέλτιστη επίδοση, τα scripts μπορούν επίσης να περιλαμβάνουν λογική για την ανίχνευση και τη διόρθωση σφαλμάτων, να διασφαλίζουν την ακεραιότητα των δεδομένων, και να παρέχουν έναν μηχανισμό για την αυτοματοποιημένη ανάκτηση σε περίπτωση που η σύνδεση διακοπεί ή τα δεδομένα δεν μεταφορτωθούν επιτυχώς.

6.2 Υλοποιημένα Scripts σε Python

```python
def read_from_arduino():
    # Δημιουργία σύνδεσης με το Arduino μέσω σειριακής θύρας
    ser = serial.Serial('COM3', 9600, timeout=0)
    try:
        # Αναμονή για σταθεροποίηση δεδομένων
        time.sleep(120)
        # Ανάγνωση και επεξεργασία των δεδομένων
        data = ser.readline().decode('utf-8').strip()
        # Επιστροφή των δεδομένων ως λίστα
        return data.split(',')
    except Exception as e:
        # Εμφάνιση μηνύματος σφάλματος
        print(f"Error: {str(e)}")
        return None
```

Κώδικας 3.1 :Python για αναγνώση και μεταφορά δεδομένων από Arduino

Αυτός ο Κώδικας 3.1 Python εκτελεί την ενέργεια ανάγνωσης δεδομένων από ένα Arduino μέσω σειριακής επικοινωνίας. Ξεκινά με την αρχικοποίηση μιας σειριακής θύρας με τον αναγνωριστικό αριθμό 'COM3' και ρυθμίζει την ταχύτητα στα 9600 baud. Πριν προβεί στην ανάγνωση των δεδομένων, υπάρχει μια παύση 120 δευτερολέπτων, για να εξασφαλιστεί ότι το Arduino έχει σταθεροποιήσει τα εξερχόμενα δεδομένα. Εν συνεχεία, πραγματοποιείται η ανάγνωση της γραμμής δεδομένων από το Arduino, αποκωδικοποιείται από το δυαδικό σε κείμενο UTF-8 και αφαιρούνται πιθανά περιττά κενά ή άλλοι χαρακτήρες. Τα δεδομένα στη συνέχεια χωρίζονται με βάση το κόμμα και επιστρέφονται σε μορφή λίστας. Σε περίπτωση που συμβεί κάποιο λάθος κατά τη διαδικασία, εμφανίζεται ένα μήνυμα σφάλματος και η συνάρτηση επιστρέφει None, δηλαδή κανένα αποτέλεσμα.

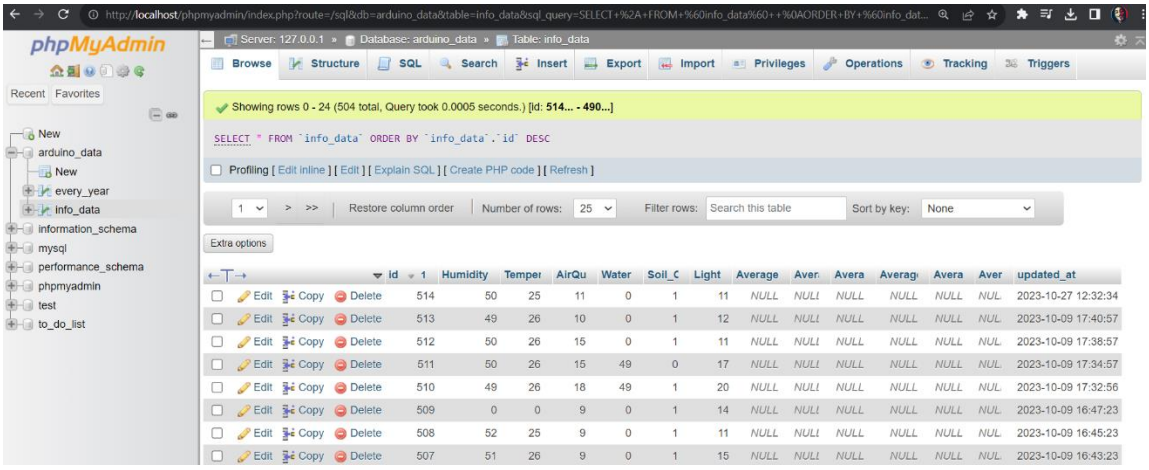
Εισαγωγή Δεδομένων στη Βάση Δεδομένων

```python
def insert_into_db(data):
    # Σύνδεση με τη βάση δεδομένων MySQL
    cnx = mysql.connector.connect(user='root', password='', database='arduino_data')
    cursor = cnx.cursor()
    try:
        # Δημιουργία και εκτέλεση της εντολής INSERT
        query = "INSERT INTO info_data (Humidity, Temperature, ...) VALUES (%s, %s, ...)"
        cursor.execute(query, data)
        # Δέσμευση των αλλαγών στη βάση
        cnx.commit()
    except Exception as err:
        # Εμφάνιση μηνύματος σφάλματος
        print(f"Database error: {str(err)}")
    finally:
        # Κλείσιμο σύνδεσης με τη βάση
        cursor.close()
        cnx.close()
```

Κώδικας 3.2 : Κώδικας σε Node.js για εισαγωγή δεδομένων στην βάση δεδομένων

Ο παραπάνω Κώδικας 3.2 αντιστοιχεί σε μια συνάρτηση Python που χρησιμοποιείται για την εισαγωγή δεδομένων σε μια βάση δεδομένων MySQL(Εικόνα 6.2). Η συνάρτηση ονομάζεται insert_into_db και δέχεται ως παράμετρο ένα πίνακα data, το οποίο περιέχει τα δεδομένα προς εισαγωγή.

Αρχικά, γίνεται σύνδεση με τη βάση δεδομένων μέσω της συνάρτησης connect του module mysql.connector, χρησιμοποιώντας τις πιστοποιήσεις του χρήστη root και χωρίς κωδικό πρόσβασης . Αυτό συμβαίνει διότι γίνεται σε localhost . Στη συνέχεια, δημιουργείται ένα αντικείμενο cursor, το οποίο επιτρέπει την εκτέλεση SQL εντολών.



Εικόνα 6.2 Localhost Database (phpMyAdmin)

Η κύρια λειτουργία της συνάρτησης είναι η δημιουργία μιας SQL εντολής INSERT, η οποία εισάγει τα δεδομένα που περιέχονται στον πίνακα (table) info_data της βάσης δεδομένων. Σημειώνεται ότι η λίστα των πεδίων (Humidity, Temperature, ...) και οι τιμές πρέπει να συμπληρωθούν ανάλογα με τη δομή του πίνακα και τα δεδομένα που θα εισαχθούν.Εφόσον εκτελεστεί η εντολή INSERT χωρίς σφάλματα, οι αλλαγές δεσμεύονται στη βάση με τη μέθοδο commit. Σε περίπτωση σφάλματος, εκτυπώνεται ένα μήνυμα με τις λεπτομέρειες του σφάλματος. Τέλος, η συνάρτηση εξασφαλίζει ότι η σύνδεση με τη βάση και το αντικείμενο cursor θα κλείσουν σωστά, ανεξάρτητα από το αν η διαδικασία εισαγωγής ολοκληρώθηκε με επιτυχία ή όχι.

Τα δύο παραπάνω αρχεία είναι συναρτήσεις που ενσωματώνονται σε ένα νέο αρχείο όπου όταν το πρώτο αρχείο έχει τραβήξει τα αναγκαία δεδομένα τότε γίνεται η εισαγωγή στην βάση δεδομένων .

```
from read_from_arduino import read_from_arduino
from insert_into_db import insert_into_db

while True:
    data = read_from_arduino()
    if data is not None:
        insert_into_db(data)
```

Κώδικας 3.3: Συγχώνευση αρχείων

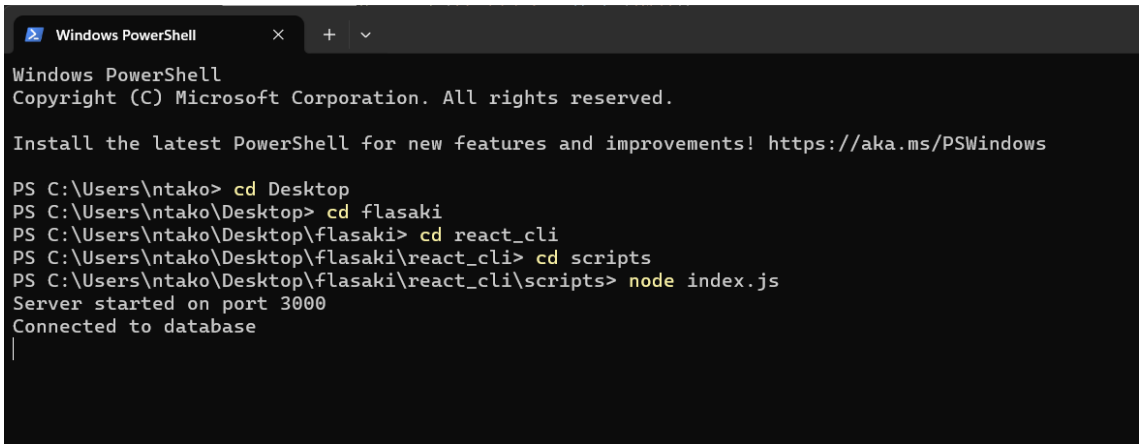
6.3 Υλοποιημένα Scripts σε Node.js

Ανάκτηση Δεδομένων από τη Βάση

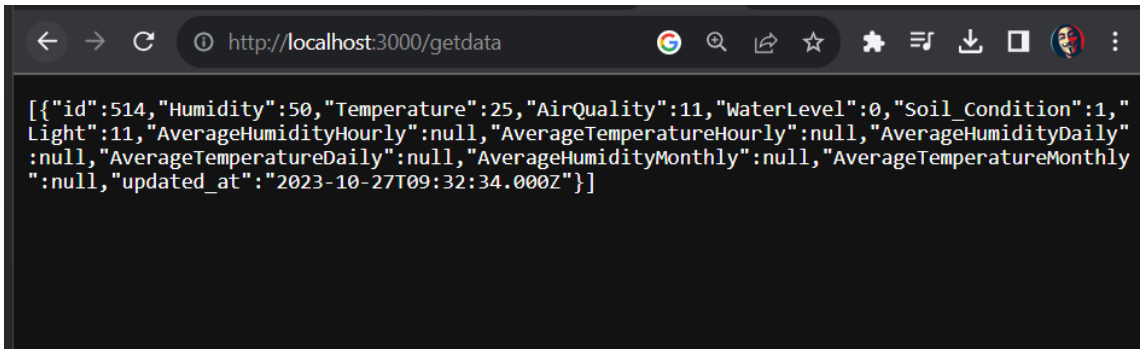
```python
@app.route('/getdata', methods=['GET'])
def get_data():
    cnx = mysql.connector.connect(user='root', password='', database='arduino_data')
    cursor = cnx.cursor()
    query = "SELECT * FROM info_data ORDER BY updated_at DESC LIMIT 1"
    cursor.execute(query)
    # Ανάκτηση της τελευταίας εγγραφής
    result = cursor.fetchone()
    cursor.close()
    cnx.close()
    # Επιστροφή των δεδομένων σε JSON μορφή
    return jsonify(result)
```

Κώδικας 3.3 :Ανάκτηση δεδομένων από βάση και μεταφόρτωση σε σελίδα

Ο συγκεκριμένος κώδικας υλοποιεί μία διαδικασία ανάκτησης δεδομένων από μία βάση δεδομένων σε εναν web server. Μέσα σε μία λειτουργία που ανταποκρίνεται σε αιτήματα GET στη διεύθυνση '/getdata', ο κώδικας συνδέεται με τη βάση δεδομένων MySQL και εκτελεί ένα ερώτημα που επιλέγει την πιο πρόσφατη εγγραφή από τον πίνακα info_data. Αφού εκτελέσει το ερώτημα, λαμβάνει την εγγραφή, την κλείνει μαζί με τη σύνδεση στη βάση δεδομένων για να αποφύγει την άσκοπη κατανάλωση πόρων, και τέλος επιστρέφει την εγγραφή σε μορφή JSON .



Εικόνα 6.3 Εκκίνηση Node Script σε Terminal



Εικόνα 6.4:Τοπική Web hosted ιστοσελίδα που μεταφορτώνονται τα δεδομένα

6.4 Προτεινόμενες αναβαθμίσεις και Στρατηγικές Βελτίωσης

Η βελτιστοποίηση ενός έξυπνου συστήματος επιτήρησης φυτών εξαρτάται από την αποδοτική λειτουργία και την εύκολη διαχείριση από τον τελικό χρήστη. Ακολουθούν κάποιες προτεινόμενες λύσεις:

- **Εξέλιξη των Αλγορίθμων Επεξεργασίας Δεδομένων:** Η ανάπτυξη πιο προηγμένων αλγορίθμων για την ανάλυση δεδομένων θα επιτρέψει την πιο αποτελεσματική προβλεπτική συντήρηση και την πρόληψη πιθανών προβλημάτων.
- **Εκπαίδευση Μέσω Μηχανικής Μάθησης:** Η χρήση τεχνικών μηχανικής μάθησης μπορεί να βοηθήσει το σύστημα να μαθαίνει από τις συνθήκες του περιβάλλοντος και να προσαρμόζει ανάλογα τις συστάσεις του.
- **Ενσωμάτωση Κοινωνικών Λειτουργιών:** Προσθέτοντας κοινωνικές λειτουργίες, όπως η δυνατότητα κοινής χρήσης της προόδου των φυτών στα κοινωνικά δίκτυα, θα αυξηθεί η εμπλοκή και το ενδιαφέρον των χρηστών.
- **Προσαρμοστικότητα και Προσωποποίηση:** Το σύστημα θα πρέπει να μπορεί να προσαρμόζεται στις ανάγκες και τις προτιμήσεις του κάθε χρήστη, προσφέροντας προσωποποιημένες συμβουλές και ρυθμίσεις.
- **Νέοι αισθητήρες** : Καθημερινά κυκλοφορούν στην αγορά νέοι αισθητήρες τεχνολογικά πιο εξελιγμένοι με μικρά κόστη που μπορούν να αποδώσουν ακόμα καλύτερα από τους ήδη υπάρχοντες αισθητήρες

- **Τακτικός έλεγχος ενημερώσεων**: Σύμφωνα με έμπειρους προγραμματιστές , οι κινητές συσκευές λαμβάνουν πολύ συχνά διάφορες ενημερώσεις που μπορούν να κάνουν το σύστημα ακόμα ταχύτερο .Όμως αυτό μπορεί να κρύβει και παγίδες αφού είναι εύκολο τα λογισμικά να έχουν προβλήματα διασύνδεσης μεταξύ τους λόγω των εκδόσεων τους.
- **Καλύτερος Σχεδιασμός χώρου** : Μπορεί να γίνει καλύτερη μελέτη για το πώς στήνονται τα μηχανήματα και οι αισθητήρες στον χώρο για μεγαλύτερη απόδοση

Μέσω της εφαρμογής αυτών των λύσεων, το έξυπνο σύστημα επιτήρησης φυτών μπορεί να επιτύχει μια υψηλότερη βαθμίδα λειτουργικότητας και να προσφέρει μια ικανοποιητική και ευχάριστη εμπειρία στον τελικό χρήστη.

7 Συνθήκες και ανάγκες Θερμοκηπίων & Ενεργειακή αυτονομία

7.1 Εισαγωγή στην ιδέα των θερμοκηπίων

Η ανάπτυξη των φυτών εξαρτάται σε μεγάλο βαθμό από τη θερμοκρασία περιβάλλοντος και κάθε φυτό έχει τη βέλτιστη θερμοκρασία ανάπτυξης. Σύμφωνα με πρόσφατες μελέτες, η πιθανή βέλτιστη θερμοκρασία ανάπτυξης του μαρουλιού, της ντομάτας και του αγγουριού είναι 14 °C, 20 °C και 26 °C, αντίστοιχα (Yildirim and Bilir, 2017). Όπως και για τη θερμοκρασία, τα φυτά απαιτούν διάφορες εντάσεις φωτός και εκθέσεις για βέλτιστη ανάπτυξη. Η υγρασία επηρεάζει επίσης τα φυτά. Ένα φυτό χάνει νερό μέσω της διαπνοής σε ξηρό περιβάλλον, ενώ μπορεί να αναπτύξει μυκητιάσεις σε υψηλή υγρασία. Η αναπνοή και η φωτοσύνθεση των καλλιεργειών εξαρτώνται επίσης από τις συγκεντρώσεις CO 2 μέσα στο θερμοκήπιο . Επειδή η ανάπτυξη και η ποιότητα μιας καλλιέργειας μπορεί να παρεμποδιστεί από χαμηλές συγκεντρώσεις CO 2 , πρέπει να διατηρηθούν επαρκή επίπεδα CO 2 και υγρασίας.

Έτσι, οι βέλτιστες συνθήκες διαδραματίζουν σημαντικό ρόλο στη διασφάλιση της άριστης απόδοσης ενός θερμοκηπίου. Διάφορες μορφές ελεγχόμενου κλίματος παρέχουν βέλτιστες συνθήκες για θέρμανση, ψύξη, φωτισμό, πότισμα, αφύγρανση και συγκέντρωση CO 2 ανάλογα με το κλίμα μιας συγκεκριμένης τοποθεσίας, τον τύπο της καλλιέργειας και τη δομή του θερμοκηπίου .Για παράδειγμα, οι τροπικές και άνυδρες περιοχές έχουν άφθονους ηλιακούς πόρους για την καλλιέργεια των καλλιεργειών. Ωστόσο, λόγω των ακραίων θερμοκρασιών και της σχετικής υγρασίας, απαιτούνται μέθοδοι ψύξης για τη διασφάλιση των κατάλληλων συνθηκών ανάπτυξης, όπως εξαναγκασμένος αερισμός, ψύξη με εξάτμιση, θόλωση, θωράκιση, σκίαση και αντλίες θερμότητας . Για παράδειγμα, σε μια μελέτη περίπτωσης στη Σαουδική Αραβία, ένα

ετήσιο ρεκόρ κατανάλωσης για έναν ανεμιστήρα και το μαξιλάρι ψύξης επιτεύχθηκε στις 56 kWh/m 2 - έτος (Al-Ibrahim et al., 2006). Ομοίως, το σύστημα εξαναγκασμένου αερισμού, που χρησιμοποιήθηκε σε άλλη μελέτη περίπτωσης στην Ελλάδα για την ψύξη του θερμοκηπίου, κατανάλωνε 20 kWh/m 2 -έτος (Trypanagnostopoulos et al., 2017).

Στην ελεγχόμενη γεωργία θερμοκηπίου, οι εφαρμογές θέρμανσης και ψύξης κατέγραψαν το υψηλότερο ποσοστό κατανάλωσης ενέργειας. Πρέπει να σημειωθεί ότι οι ενεργειακές απαιτήσεις θερμοκηπίου εξαρτώνται και από τις καλλιέργειες που επιλέγονται. Για παράδειγμα, στη Σμύρνη της Τουρκίας, η ντομάτα, το αγγούρι και το μαρούλι απαιτούσαν 38, 79 και 10 kWh/m 2 -έτος, αντίστοιχα (Yildirim and Bilir, 2017).

Εικόνα 7.1 ‘ Έξυπνο ’ θερμοκήπιο με φωτοβολταϊκά και ηλεκτρονικό έλεγχο

Επιπλέον, ο συμπληρωματικός φωτισμός είναι μερικές φορές απαραίτητος για καλλιέργειες που καλλιεργούνται στα βόρεια θερμοκήπια, όπου η ηλιακή ακτινοβολία είναι ανεπαρκής για τη φωτοσύνθεση των καλλιεργειών. Οι Bambara και Athienitis (2019) υπολόγισαν το ετήσιο φορτίο συμπληρωματικών φώτων σε 123 kWh/m 2 - έτος για ένα θερμοκήπιο στην Οττάβα του Οντάριο, Καναδάς και οι Vadiee και Martin (2013) υπολόγισαν τα φορτία ηλεκτρικής κατανάλωσης των συμπληρωματικών φώτων για ένα θερμοκήπιο στο Στοκχόλμη, Σουηδία να είναι 115 kWh/m 2 -έτος.

Για την αντιμετώπιση της υπερθέρμανσης του πλανήτη και την παροχή βιώσιμης πηγής ενέργειας για τις ενεργειακές απαιτήσεις του θερμοκηπίου, θα μπορούσαν να

χρησιμοποιηθούν εναλλακτικοί φιλικοί προς το περιβάλλον πράσινες πηγές ενέργειας για την παραγωγή ενέργειας, όπως αιολική, ηλιακή & υδροηλεκτρική. Μία από τις πιο ανανεώσιμες πηγές ενέργειας για εφαρμογές θερμοκηπίου είναι η ηλιακή ενέργεια. Ένα θερμοκήπιο χτίζεται συνήθως σε ανοιχτό πεδίο, επομένως έχει άφθονη ηλιακή ακτινοβολία για να καλύψει τη θεμελιώδη ανάγκη της καλλιέργειας για φωτοσύνθεση. Επομένως, τέτοιες τοποθεσίες είναι κατάλληλες για ηλιακή τεχνολογία και χρήσιμες για παραγωγή ενέργειας. Επιπλέον, οι ηλιακές τεχνολογίες είναι ένα πολύ ενεργό και τεράστιο πεδίο στις εφαρμογές της ενέργειας του θερμοκηπίου. Πολλοί ερευνητές προσπάθησαν να ενσωματώσουν διαφορετικούς τύπους ηλιακών τεχνολογιών στο θερμοκήπιο. Για παράδειγμα, ο Gorjian et al. (2020) παρουσίασε μια ανασκόπηση για την ενσωμάτωση εφαρμογών ηλιακής ενέργειας σε γεωργικά θερμοκήπια λαμβάνοντας υπόψη τις μελέτες των τελευταίων δεκαετιών. Σε αυτήν την ανασκόπηση, οι συγγραφείς χώρισαν την εφαρμογή ηλιακής ενέργειας σε ενεργητική και παθητική διαμόρφωση για το θερμοκήπιο και επικεντρώθηκαν κυρίως στην άποψη του ενεργειακού ενημερωτικού δελτίου με λιγότερο λεπτομερή τον αντίκτυπο του ηλιακού συλλέκτη στις καλλιέργειες θερμοκηπίου.

Πραγματοποιήθηκε μια ολοκληρωμένη ανασκόπηση για να ερευνηθούν οι δυνατότητες της τεχνολογίας ηλιακής ενέργειας για τη γεωργία θερμοκηπίου και να συζητηθούν οι νέες και εφικτές ηλιακές τεχνολογίες που θα μπορούσαν να εφαρμοστούν. Σε αυτήν την έρευνα, το 70% των αντιπροσωπευτικών εφαρμογών ηλιακής τεχνολογίας σχεδιάστηκαν ειδικά για θερμοκήπια τα τελευταία 5 χρόνια (2016–2021). Αυτές οι ηλιακές εφαρμογές κατηγοριοποιήθηκαν σύμφωνα με τα χαρακτηριστικά των ηλιακών τεχνολογιών (PV, STC και PV/T). Επίσης, αξιολογήθηκε η σχέση μεταξύ των στατικών ηλιακών συλλεκτών και της απόδοσης των καλλιεργειών για να προσδιοριστεί η κατάλληλη ηλιακή κάλυψη για τη στέγη του θερμοκηπίου. Εξετάστηκαν δυναμικοί, συγκεντρωτικοί και ημιδιαφανείς τύποι ηλιακών τεχνολογιών. Τέλος, συζητήθηκαν τα οφέλη, οι περιορισμοί και η καταλληλόλητα κάθε ηλιακής τεχνολογίας.

7.2 Φωτοβολταικη ενέργεια για ηλεκτρική αυτονομία

Φωτοβολταϊκές τεχνολογίες:

Οι τεχνολογίες φωτοβολταϊκών (PV) μετατρέπουν το ηλιακό φως απευθείας σε ηλεκτρική ενέργεια μέσω του φωτοβολταϊκού φαινομένου. Οι τεχνολογίες PV ταξινομούνται σε συμβατικές φωτοβολταϊκές μονάδες και ημιδιαφανείς μονάδες, οι οποίες χρησιμοποιούνται για εφαρμογές θερμοκηπίου. Οι περισσότερες συμβατικές φωτοβολταϊκές μονάδες είναι κατασκευασμένες από πυρίτιο, μονοκρυσταλλικό και πολυκρυσταλλικό πυρίτιο, ενώ οι ημιδιαφανείς φωτοβολταϊκές μονάδες είναι κατασκευασμένες είτε από οργανικά ηλιακά κύτταρα, είτε από ευαισθητοποιημένα με χρωστικές ηλιακές κυψέλες είτε από άλλες αναδυόμενες τεχνολογίες (Jean et al., 2015, Obeidat ,

Λίγο πιο συγκεκριμένα …

Στο συγκεκριμένο σύστημα που δημιουργήθηκε για την συλλογή δεδομένων και αυτόματων αποφάσεων & ενεργειών, μελετήθηκε και η ενεργειακή αυτονομία που θα χρειάζονταν έτσι ώστε να είναι δυνατή η πλήρως αυτόνομη λειτουργεία του συστήματος και σε συνθήκες που δεν είναι εφικτή η σύνδεση στο διαδίκτυο και στον υπολογιστή και γενικώς το ρεύμα.

Αρχικά το σύστημα είναι αναγκαίο να τροφοδοτηθεί με ηλεκτρισμό , συνεπώς για αυτόν τον λόγο θα χρειαστούμε ένα Powerbank και ένα ηλιακό Πάνελ (Solar Panel).Το κόστος του ηλιακού πάνελ πλησιάζει τα 8$ για απόδοση 10Watt-h.

Εάν προχωρήσουμε σε κάποιους υπολογισμούς θα δούμε πώς το σύστημα χρειάζεται

Arduino Uno R3: **50 mA** ,DHT11: **2.5 mA**, Soil Moisture Sensor: **35 mA**, Water Level, Sensor : **20 mA,** Air Quality Sensor ZP07-04: **150 mA** ,Light Resistor: **1 mA** ,5V Relay: **70 mA**

Συνολική κατανάλωση: 50 + 2.5 + 35 + 20 + 150 + 1 + 70 = **328.5 mA**

Αυτό είναι ανάγκη να το μετατρέψουμε σε Watt , για να μπορέσουμε να δούμε αν το πάνελ μπορεί να καλύψει τις ανάγκες μας . Υπολογίζεται με τον τύπο $P=V \times I$, όπου P είναι η ισχύς σε Watt, V είναι η τάση σε Volt, και I είναι το ρεύμα σε Amperes (A).

Κάνοντάς τις πράξεις καταλήγουμε σε

Συνολική Ισχύς σε Watt: 0.25 + 0.0125 + 0.175 + 0.1 + 0.75 + 0.005 + 0.35 = **1.6425Watt**

Αυτή η τιμή αφορά την ωριαία μέγιστη δυνατή κατανάλωση .Κάνουμε στρογγυλοποίηση σε αυτή την τιμή στα **1.7Watt-h** έτσι ώστε να βρούμε την ετήσια κατανάλωση του συστήματος . Αν σκεφτούμε πως η μέρα έχει 24 ώρες , 30 ημέρες ο μήνας , 12 μήνες τον χρόνο έχουμε:

24*30*12 *1,7Watt-h = **14688 Watt-y**

Δηλαδή κάθε έτος η μέγιστη δυνατή κατανάλωση μπορεί να φτάσει τις **14.68kW** (Και για να το μετατρέψουμε και σε ευρώ , αν πάρουμε την τωρινή μέση τιμή ρεύματος (0.1$ *kW) ,έχουμε ετήσιο κόστος 1.46 $).

Για να δούμε τώρα την απόδοση του ηλιακού φωτοβολταϊκού πάνελ . Η μέγιστη δυνατή φόρτιση σε πλήρη έκθεση στον ήλιο φτάνει τα 10Watt . Θεωρούμε πως στην Ελλάδα (για λόγους ευκολίας) έχουμε 6 μήνες καλοκαίρι και 6 μήνες χειμώνα , με την μέση τιμή καθαρής ηλιοφάνειας το καλοκαίρι να είναι 11 ώρες ενώ τον χειμώνα 3:30 ώρες τον χειμώνα .Αν κάνουμε τον θεωρητικό υπολογισμό έχουμε :

Καλοκαίρι : 11*30*6*10Watt-h= 39600Watt-y
Χειμώνας : 3.30*30*6*10=5940Watt-y
Σύνολο : 39600+5940=**45540Watt-y**

Άρα με μία παραγωγή **45.54kW** σε ετήσια βάση βλέπουμε πως οι ανάγκες του συστήματος (<14.68kW) υπερκαλύπτονται .Όμως πρέπει να λάβουμε υπόψη πως οι ώρες ηλιοφάνειας (κυρίως τον χειμώνα) είναι μεταβαλλόμενες μέρα με την μέρα , οπότε δεν μπορούμε να πούμε με σιγουριά πως το σύστημα θα λειτουργεί αυτόνομα χωρίς να μείνει από ρεύμα για πάντα .Για αυτό ένας τακτικός έλεγχος του υπολοίπου της μπαταρίας και η τυχών φόρτιση της θα ήταν η καλύτερη λύση !

8 Συμπεράσματα

Η Σχεδίαση και Υλοποίηση Συστήματος Αυτοματισμών και IoT για εφαρμογές Πρωτογενούς Παραγωγής ήτανε μια εργασία με τεράστιο φόρτο εργασίας καθώς έπρεπε να καλύψει τις ανάγκες από ένα application που θα μπορεί ο κάθε χρήστης να έχει όλες τις απαραίτητες πληροφορίες και δεδομένων για τα φυτά του μέσω από το διαδίκτυο των πραγμάτων(IoT) . Η συλλογή – αποθήκευση και αποστολή των δεδομένων & οι αυτοματισμοί για την λήψη αποφάσεων από το σύστημα Arduino ήταν κύριας σημασίας αφού εκεί βρίσκεται όλη η ‘καρδία ’ της τεχνολογίας και του λογισμικού που

αναπτύχθηκε .Οι δυσκολίες από διάφορα τεχνικά και προγραμματίστηκα προβλήματα ήταν μια πρόκληση για μένα , όμως με την επίλυση αυτών μου δόθηκε η ευκαιρία να αποκτήσω γνώση και εμπειρία που μελλοντικά θα μου είναι ιδιαιτέρως χρήσιμη .Σίγουρα αν ξανά ξεκινούσα την ίδια πτυχιακή εργασία θα έκανα πολλά πράγματα διαφορετικά όπως : να επιλέξω μια διαφορετική πλακέτα , διαφορετικούς αισθητήρες , άλλους τρόπους υλοποίησης του κώδικα για το λογισμικό που δημιουργήθηκε όμως ο σκοπός και ο στόχος του αποτελέσματος θα ήταν ακριβώς ο ίδιος , να υπάρξει ένα αρχικό λογισμικό με μελλοντικές προοπτικές εξέλιξης που θα βοηθά ανθρώπους που ασχολούνται με τον πρωτογενή τομέα να έχουν όλες τις αναγκαίες πληροφορίες και τον έλεγχο των εγκαταστάσεων τους στην τσέπη τους .Σίγουρα είναι ένας τομέας που μελλοντικά θα με ενδιέφερε να ασχοληθώ και να εξελίξω ακόμα περισσότερο αυτή την ιδέα.

Εάν κάποιος άλλος θα ήθελε να ασχοληθεί με την συγκεκριμένη ιδέα και υλοποίηση θα μπορούσε να προσθέσει ακόμα περισσότερους αυτοματισμούς , μεγαλύτερη ασφάλεια και αποδοτικότητα σε επίπεδο τροφοδοσίας και κώδικα . Επίσης η ένταξη της μηχανικής μάθησης και της τεχνίτης νοημοσύνης σίγουρα θα « απογείωνε » το συγκεκριμένο project με τεράστιες προοπτικές τεχνολογικής εξέλιξης και κερδοφορίας .

Βιβλιογραφία

- Γεωργία Ακριβείας -- Σπύρος Φουντάς & Θεοφάνης Γέμτος [2015]
- Η σημασία των τεχνολογιών IoT στην έξυπνη γεωργία –Tractor GPS[2021]
- Εφαρμογές του Δδιαδικτύου των Πραγμάτων στην έξυπνη γεωργία με τεχνολογίες ασύρματης δικτύωσης -- Καλομοίρη Νικόλαου , Πανεπιστήμιο Πατρών [2021]
- https://reactnative.dev/docs/getting-started -- React Native Introduction + [09/2023]
- https://docs.amplify.aws/react-native/build-a-backend/auth/set-up-auth/ AWS Authentication Documentation [2023]
- https://openweathermap.org/api OpenWeatherOfficial Dashboard documentation [2022-2023]

- https://developers.google.com/maps/documentation/geocoding/overview Google For Developers [Geocoding 2023]
- Online course Developer – React Native-- notJust Development [2022]
- https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf -- DHT11 Humidity & Temperature Sensor
- https://docs.arduino.cc/hardware/uno-rev3 UNO R3 – Arduino.CC (Documentation + Manual)
- https://lastminuteengineers.com/soil-moisture-sensor-arduino-tutorial/ -- How Soil Moisture Sensor Works and Interface it with Arduino
- https://circuitdigest.com/microcontroller-projects/interfacing-water-level-sensor-with-arduino -- How does a Water Level Sensor Work and How to Interface it with Arduino?
- https://www.winsen-sensor.com/d/files/zp07-mp503-10-grade-manual-air-quality-detection-module-1_3-terminal-forward.pdf ZP07 Air-Quality Detection Module – Winsen (Manual) [2019]
- https://docs.expo.dev/ --Expo Go Documentation
- https://docs.joomla.org/XAMPP -- XAMPP From Joomla! Documentation [09/2022]
- https://www.phpmyadmin.net/docs/ -- Bringing MySQL to the web [2003-2023]
- Mukesh Kumar a b, Didier Haillot a, Stéphane Gibout b--Survey and evaluation of solar technologies for agricultural greenhouse application -- [01/2022]

Παράρτημα Α

Φωτογραφία μοκέτας αισθητήρων