



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Ανάπτυξη διαδικτυακής εφαρμογής ΙΟΤ
πραγματικού χρόνου για την εποπτεία και
διαχείριση της άρδευσης οικιακών φυτών

Πέτρος Ζέρβας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΤΖΙΑΛΛΑΣ ΓΡΗΓΟΡΙΟΣ
Καθηγητής

Λαμία έτος 2025



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Ανάπτυξη διαδικτυακής εφαρμογής ΙΟΤ
πραγματικού χρόνου για την εποπτεία και
διαχείριση της άρδευσης οικιακών φυτών

Πέτρος Ζέρβας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΤΖΙΑΛΛΑΣ ΓΡΗΓΟΡΙΟΣ
Καθηγητής

Λαμία έτος



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

Development of a Real-Time IoT Web Application for Monitoring and Managing the Irrigation of Indoor Plants

Petros Zervas

FINAL THESIS

ADVISOR

TZIALLAS GRIGORIOS
Professor

Lamia year

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

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία:/...../20.....

Ο – Η Δηλ.

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

ΠΕΡΙΛΗΨΗ

Η συγκεκριμένη πτυχιακή, έχει ως θέμα την ανάπτυξη μιας διαδικτυακής εφαρμογής ΙΟΤ σε πραγματικό χρόνο, για την εποπτεία και διαχείριση της άρδευσης οικιακών φυτών. Για το στόχο αυτό, υλοποιήθηκε μια συσκευή, η οποία αποτελείται από μια πλακέτα NodeMCU esp8266, μαζί με έναν αισθητήρα και ένα breadboard. Τα δεδομένα αυτά καταγράφονται σε πραγματικό χρόνο και μεταφέρονται, μέσω του ΙΟΤ πρωτοκόλλου MQTT, σε έναν server NodeJS όπου, ανάλογα με την ένδειξη, εμφανίζονται μηνύματα για την υγρασία φυτών. Έπειτα τα δεδομένα αυτά αποθηκεύονται σε μια βάση δεδομένων MongoDB με χρήση της υπηρεσίας MongoDB Atlas. Τέλος έχει πραγματοποιηθεί η υλοποίηση μίας ιστοσελίδας, όπου καταγράφεται η αυξομείωση των τιμών μέσω ενός γραφήματος, καθώς και μια ενδεικτική λίστα με οικιακά φυτά.

ABSTRACT

This thesis project focuses on developing a real-time IoT web application for monitoring and managing the irrigation of houseplants. To achieve this goal, a device was implemented, consisting of a NodeMCU esp8266 board, along with a sensor and a breadboard. This data is recorded in real time and transmitted, via the MQTT IoT protocol, to a NodeJS server where, depending on the reading, messages about the humidity of plants are displayed. This data is then stored in a MongoDB database using the MongoDB Atlas service. Finally, a web page has been implemented, where the fluctuation of values is recorded through a graph, as well as an indicative list of houseplants.

Table of Contents

ΠΕΡΙΛΗΨΗ	I
ABSTRACT	III
ΚΕΦΑΛΑΙΟ 1 INTERNET OF THINGS.....	2
1.1 ΤΙ ΕΙΝΑΙ ΤΟ INTERNET OF THINGS.....	2
1.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ IoT.....	4
1.3 IoT HARDWARE	
1.3.A ΤΙ ΕΙΝΑΙ ΤΟ NODE MCU ESP 8266	
1.4 IoT ΠΡΩΤΟΚΟΛΛΑ	
1.4.A Πρωτόκολλο HTTP (HyperText Transfer Protocol)	
1.4.B Πρωτόκολλο AMQP (Advanced Message Queuing Protocol)	
1.4.C Πρωτόκολλο ZigBee	
1.4.D Πρωτόκολλο Matter	
1.4.E Πρωτόκολλο NB-IoT (NarrowBand Internet of Things)	
1.4.F Πρωτόκολλο LoRaWAN (Long Range WAN)	
1.4.G Πρωτόκολλο CoAP (Constrained Application Protocol)	
1.4.H Πρωτόκολλο MQTT (Message Queuing Telemetry Transport)	
1.5 IoT BROKER ΕΜQX	
ΚΕΦΑΛΑΙΟ 2 ΔΙΑΔΙΚΑΣΙΑ ΕΦΑΡΜΟΓΗΣ.....	112
2.1 Back-End	
2.1.A Arduino IDE	
2.1.B NodeJS	
2.1.C MongoDB & MongoDB Atlas	
2.2 Front-End	
2.2.A Πρωτόκολλο WebSocket	
ΚΕΦΑΛΑΙΟ 3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	15
3.1 Υλοποίηση	
3.2 Περιγραφή	

ΚΕΦΑΛΑΙΟ 4 ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....16

- 4.1 Κατασκευή του μετρητή**
- 4.2 Εκτέλεση και προβολή αποτελεσμάτων στο Arduino IDE**
 - 4.2.A Βιβλιοθήκη ESP8266WiFi**
 - 4.2.B Βιβλιοθήκη PubSubClient**
 - 4.2.C Βιβλιοθήκη ArduinoJSON**
 - 4.2.D Εκτέλεση στο Arduino IDE**
- 4.3 Σύνδεση με τον NodeJS server**
 - 4.3.A Βιβλιοθήκες**
 - 4.3.B Εκτέλεση του server**
- 4.4 Υλοποίηση της ιστοσελίδας**

ΚΕΦΑΛΑΙΟ 5 ΕΠΙΛΟΓΟΣ..... 58

- 5.1 Συμπεράσματα**
- 5.2 Βελτιστοποιήσεις**

ΒΙΒΛΙΟΓΡΑΦΙΑ..... 59

ΚΕΦΑΛΑΙΟ 1: Internet Of Things

1.1 Τι είναι το Internet Of Things

Το Internet Of Things (Διαδίκτυο των Πραγμάτων), ως έννοια αναφέρεται σε δύο πράγματα. Αρχικά, στις συσκευές οι οποίες που είναι συνδεδεμένες με αισθητήρες, με την ικανότητα επεξεργασίας και ανταλλαγής δεδομένων, αλλά και στις τεχνολογίες που επιτρέπουν την σύνδεση αυτών των συσκευών είτε με το διαδίκτυο, είτε μεταξύ τους. Αποτελεί ένα από τα σημαντικότερα κομμάτια της νέας βιομηχανικής επανάστασης και πλέον υπάρχουν εκατομμύριες συνδεδεμένες συσκευές [1]. Μερικές από τις εφαρμογές του IoT βρίσκονται στα εξής πεδία:

- 1) Στην ιατρική, με τον όρο **Διαδίκτυο των Ιατρικών Πραγμάτων** (IoMT) ή “Εξυπνη υγειονομική περίθαλψη”, αναφερόμενο στο δίκτυο των συνδεδεμένων IoT συσκευών και λογισμικών, για τη συλλογή και ανάλυση δεδομένων με σκοπό την παροχή πιο αποτελεσματικών διαγνώσεων και θεραπείας. [2]
- 2) Στο επικοινωνιακό σύστημα οχημάτων, περιγράφει την ασύρματη επικοινωνία, ανάμεσα σε ένα όχημα και το περιβάλλον του. Ο επιστημονικός του είναι επικοινωνία **Vehicle-To-Everything** (V2X) και αποτελείται από τις εξής υποκατηγορίες. [3]
 - i) **Επικοινωνία όχημα με όχημα** (V2V), δηλαδή την ανταλλαγή δεδομένων σε πραγματικών χρόνων ανάμεσα στα οχήματα. [4]
 - ii) **Επικοινωνία όχημα με υποδομή** (V2I), την σύνδεση και επικοινωνία των οχημάτων με την υποδομή ενός δρόμου, δηλαδή τις πινακίδες, τα φανάρια και την σηματοδότηση. [5]
 - iii) **Επικοινωνία όχημα με πεζό** (V2P), την επικοινωνία ανάμεσα σε ένα όχημα και τον πεζό, για την αποφυγή ατυχημάτων, ιδιαίτερα σε περιπτώσεις όπου ο πεζός είναι σε αναπηρικό αμαξίδιο ή με ποδήλατο. [6]
 - iv) **Επικοινωνία όχημα με δίκτυο** (V2N), το όχημα συνδέεται με υπηρεσίες νέφους, για την λήψη δεδομένων σε περίπτωση μποτιλιαρίσματος, πλοήγησης και αλλαγές στον καιρό.
 - v) **Επικοινωνία όχημα με πλέγμα** (V2G), ένα ηλεκτρικό όχημα επικοινωνεί με το ηλεκτρικό δίκτυο για την βελτίωση της φόρτισης και διανομής ενέργειας. [7]

3) Στη βιομηχανία, με τον όρο **Διαδίκτυο των βιομηχανικών πραγμάτων** (IIoT) [8], που απαρτίζεται από όλες τις συνδεδεμένες βιομηχανικές συσκευές, για τη συλλογή και ανάλυση δεδομένων με σκοπό την λήψη αποφάσεων για την βελτίωση της βιομηχανικής διαδικασίας σε τομείς όπως η παραγωγικότητα, η αποδοτικότητα και η ασφάλεια. Αξιοποιεί ένα ευρύ τεχνολογικό φάσμα, το οποίο συλλεγεί και αναλύει δεδομένα με σκοπό την μετατροπή τους σε χρήσιμες πληροφορίες. Μερικές από αυτές τις τεχνολογίες είναι:

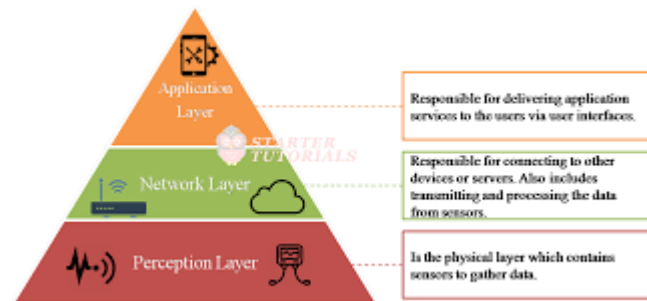
- i) **Νεφουπολογιστική** (cloud computing), όπου υπηρεσίες και πόροι να ληφθούν μέσω του διαδικτύου χωρίς την ανάγκη σύνδεσης με κάποιον server.
- ii) **Κυβερνοασφάλεια** (Cybersecurity), για την προστασία των δεδομένων και των συστημάτων από μη εξουσιοδοτημένη πρόσβαση, καθώς τα βιομηχανικά συστήματα είναι συχνός στόχος κυβερνοεπιθέσεων.
- iii) **Τεχνητή νοημοσύνη** (AI) και **μηχανική μάθηση**, κατάλληλο για την αποτελεσματική πρόβλεψη βλαβών και την βελτιστοποίηση της παραγωγής και αυτοματοποίησης των διαδικασιών.
- iv) **Δίκτυα**, είναι ο πυλώνας της επικοινωνίας μεταξύ των συστημάτων του IIoT. Η επικοινωνία τους μπορεί να γίνει είτε μέσω ενσύρματων δικτύων, είτε μέσω ασύρματων δικτύων, όπως το Wi-Fi και το Bluetooth.

4) Στον αγροτικό τομέα. Ως μέρος της νέας Πράσινης Επανάστασης, μετά από αυτές της αναπαραγωγής των φυτών και της γενετικής, συσκευές όπως οι αισθητήρες, ενεργοποιητές, drones, ρομπότ και κάμερες αποσκοπούν στο να βοηθήσουν τους αγρότες σε όλο τον κόσμο να παρακολουθούν τα χωράφια τους. [9]

1.2 Αρχιτεκτονική του ΙΟΤ

Η πρώτη και πιο απλή δομή αρχιτεκτονικής ΙοΤ που δημιουργήθηκε είναι η αρχιτεκτονική των τριών επιπέδων.(3-layer architecture) [10]. Η δομή της περιλαμβάνει:

- 1) Επίπεδο Αντίληψης (Perception Layer):** Το πρώτο επίπεδο στην αρχιτεκτονική ΙοΤ. Η κύρια λειτουργία είναι η αλληλεπίδραση με το περιβάλλον για τη συλλογή δεδομένων, όπως θερμοκρασία, υγρασία και ήχο. Έπειτα τα δεδομένα αυτά μεταφέρονται σε επόμενα επίπεδα για περαιτέρω λειτουργίες.
- 2) Επίπεδο Δικτύου (Network Layer):** Υπεύθυνο για τη σύνδεση τερματικών μεταξύ τους καθώς και σύνδεσής τους με το διαδίκτυο. Ως ο συνδετικός κρίκος αυτής της δομής, αποθηκεύει και μεταφέρει τα δεδομένα από το επίπεδο αντίληψης στο επίπεδο εφαρμογής.
- 3) Επίπεδο Εφαρμογής (Application Layer):** Είναι το επίπεδο στο οποίο συνδιαλέγεται ο χρήστης. Παρέχει τα κατάλληλα μηνύματα και υπηρεσίες ανάλογα με τα δεδομένα που έχουν συλλεχθεί στα δύο προηγούμενα επίπεδα.



(Εικόνα 1.1): Αρχιτεκτονική 3 επιπέδων

Λόγω της ανάπτυξης και δημοτικότητας του ΙοΤ, της αυξημένης ανάγκης συσκευών που αλληλοεπιδρούν μεταξύ τους αλλά και της ανάγκης επίλυσης των θεμάτων ασφάλειας που προκύπτουν από την υπάρχουσα δομή, προστέθηκαν άλλα δύο επίπεδα, δημιουργώντας την δομή των 5 επιπέδων. [10]

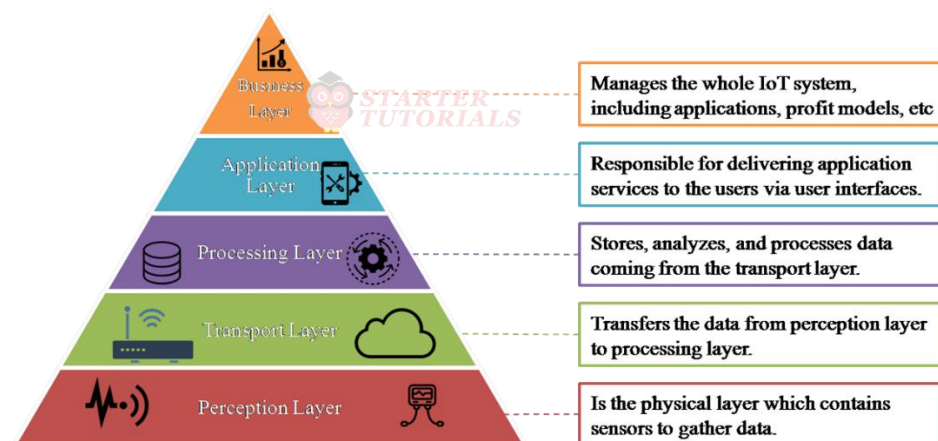
Τα επιπλέον αυτά επίπεδα είναι:

- 1) Επίπεδο Επεξεργασίας (Processing Layer):** Αποθηκεύει και αναλύει τα δεδομένα που λαμβάνει από το επίπεδο δικτύου. Στο συγκεκριμένο επίπεδο χρησιμοποιούνται τεχνολογίες νεφουπολογιστικής για ευελιξία και Οι ενέργειες που λαμβάνονται είναι μεταξύ άλλων η συνάθροιση

δεδομένων (data aggregation), η μετάφραση πρωτοκόλλου και η ενίσχυση της ασφάλειας. Όλα αυτά γίνονται για τη προετοιμασία των δεδομένων για το επόμενο επίπεδο

2) Επίπεδο Επιχείρησης (Business Layer): Το πιο γνωστό και εμπορικό από όλα τα επίπεδα, αφορά τη σωστή χρήση των δεδομένων, μέσω διεπαφών και διαγραμμάτων ροής, τα δεδομένα αυτά βοηθούν στις λείψεις αποφάσεων.

Τέλος να σημειωθεί, στο κομμάτι της ασφάλειας, πως το επίπεδο του δικτύου είναι και το πιο ευαίσθητο σε επιθέσεις. Για το λόγο αυτό, στα σύγχρονα μοντέλα, έχουν προστεθεί πρωτόκολλα όπως το TCP για την αξιόλογη μεταφορά δεδομένων, χωρίς απώλειες. Η νέα του ονομασία είναι **επίπεδο μεταφοράς (Transport Layer)** αν και η κλασική του ονομασία χρησιμοποιείται κανονικά.



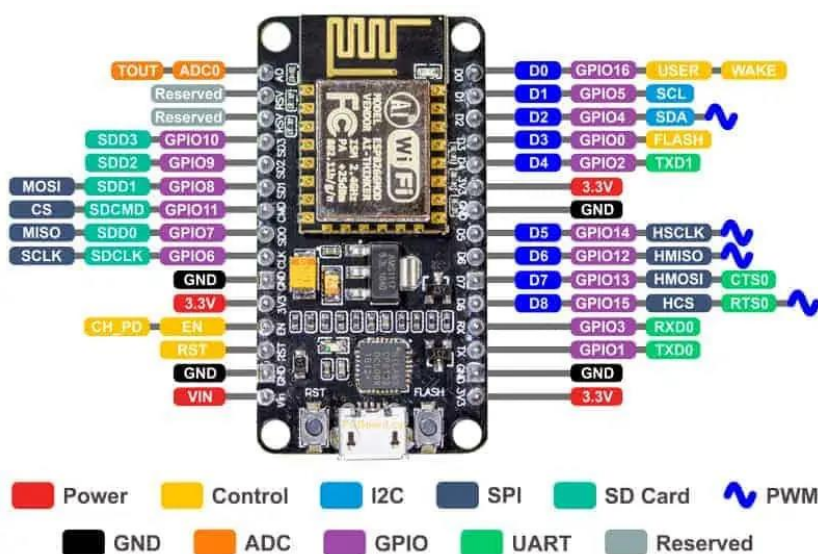
(Εικόνα 1.2): Αρχιτεκτονική 5 επιπέδων

1.3 IoT Hardware

Για την δημιουργία IoT οικοσυστημάτων, απαιτείται η χρήση του κατάλληλου hardware. Συσκευών δηλαδή που συλλέγουν, επεξεργάζονται και μεταδίδουν δεδομένα, ενώ παράλληλα αλληλοεπιδρούν με το περιβάλλον τους. Μερικά από τα πιο χαρακτηριστικά hardwares είναι:

- 1) **Αισθητήρες:** Λαμβάνουν δεδομένα από τις αλλαγές στο περιβάλλον τους και δημιουργούν ηλεκτρικούς παλμούς για να επικοινωνήσουν.
- 2) **Ενεργοποιητές:** Πραγματοποιούν ενέργειες μετά από εντολές από κάποιο κεντρικό έλεγχο, συνήθως μετά από αλλαγές που παρατηρούνται στους αισθητήρες.
- 3) **Μονάδα μικροελεκτή (MCU):** Μικροί υπολογιστές όπου έχουν ενσωματωμένα microchips, κεντρικές μονάδες επεξεργασίας καθώς και μνήμες RAM και ROM.
- 4) **Μονάδα μικροεπεξεργαστή (MPU):** Εκτελούν τις λειτουργίες των κεντρικών μονάδων επεξεργασίας σε ένα ή πολλαπλά κυκλώματα.
- 5) **Έξυπνες συσκευές:** Συσκευές με την ικανότητα να επικοινωνούν μεταξύ τους.

1.3.A Τι είναι το Node MCU ESP8266



Είναι μια πολύ δημοφιλής, οικονομική και open-source πλακέτα κατάλληλη για τον σχεδιασμό και την ανάπτυξη εφαρμογών IoT, αυτοματισμούς σε σπίτια καθώς και να συνδέει συσκευές. Από τα βασικότερα χαρακτηριστικά του είναι [17]:

- 1) Ο μικρό-ελεγκτής ESP8266. Η συνδεσιμότητα στο Wi-Fi έχει ενσωματωμένο το standard 802.11 b/g/n, στη ζώνη συχνοτήτων 2.4 GHz. Η ταχύτητά του είναι από 150 μέχρι 300 Mbps, σε ιδανικές συνθήκες, χρησιμοποιώντας τεχνικές όπως MIMO (Multiple-Input and Multiple-Output). Επίσης έχει επεξεργαστή Tensilica L106 RISC (Reduced Instruction Set Computing), κάτι το οποίο το κάνει ιδανικό για εκτέλεση μικρού set εντολών, για γρήγορη εκτέλεση και χαμηλή κατανάλωση, με την κεντρική μονάδα επεξεργασίας του να είναι στα 32 bits και η ταχύτητα ρολογιού του στα 80-160 MHz. Η μνήμη του μικρό-ελεγκτή είναι στα 32 με 128 KB RAM, με αποθηκευτικό χώρο flash 1 έως 4 MB. Τέλος, η τροφοδοσία του γίνεται μέσω USB και λειτουργεί στα 3.3V. [18]
- 2) Αποκλειστικά στοιχεία ως NodeMCU πλακέτα. Το πρώτο από αυτά είναι οι είσοδοι GPIO (General Purpose Input/Output). Χρησιμοποιούνται ως είσοδοι και έξοδοι και ελέγχονται από λογισμικό. Για την πλακέτα ESP8266 οι αριθμοί και η λειτουργία της κάθε εισόδου είναι η εξής:

NodeMCU Label	ESP8266 GPIO	Λειτουργία
D0	GPIO16	Συνδεδεμένο σε ενσωματωμένο LED χαμηλής κατανάλωσης
D1	GPIO5	Κατάλληλη για γενική είσοδο / έξοδο
D2	GPIO4	Κατάλληλη για γενική είσοδο / έξοδο
D3	GPIO0	Επανεκκίνηση Λειτουργίας
D4	GPIO2	Επανεκκίνηση Λειτουργίας
D5	GPIO14	Κατάλληλη για γενική είσοδο / έξοδο

D6	GPIO12	Κατάλληλη για γενική είσοδο / έξοδο
D7	GPIO13	Κατάλληλη για γενική είσοδο / έξοδο
D8	GPIO15	Επανεκκίνηση Λειτουργίας
RX	GPIO3	Είσοδος για λήψη δεδομένων τύπου UART
TX	GPIO1	Είσοδος για μετάδοση δεδομένων τύπου UART
A0	ADC0	Αναλογική Είσοδος, εύρους 0 με 3.3V

[17]

Το δεύτερο είναι η διεπαφή USB-to-serial chip CH340, καθώς γεφυρώνει την επικοινωνία μεταξύ της θύρας USB του υπολογιστή και διεπαφής UART (Universal Asynchronous Receiver Transmitter) του μικρό-ελεγκτή.

- 3) Ευκολία στο λογισμικό. Μπορεί να προγραμματιστεί εύκολα σε περιβάλλοντα όπως Arduino IDE, ESP-IDF (Espressif IoT Development Framework) για προγραμματισμό σε C και C++, Espruino για προγραμματισμό σε JavaScript, NodeMCU Editor για προγραμματισμό σε Lua, καθώς και MicroPython για προγραμματισμό σε Python. Παράλληλα, με την υποστήριξη Wi-Fi, μπορεί να χειριστεί τα πιο δημοφιλή πρωτόκολλα, όπως HTTP, MQTT, WebSocket και CoAP, για την αλληλεπίδραση με IoT πλατφόρμες και συσκευές. [18]

1.4 Πρωτόκολλα IoT

1.4.A Πρωτόκολλο HTTP (HyperText Transfer Protocol)

Δημιουργήθηκε στο CERN το 1989 και είναι από τα σημαντικότερα πρωτόκολλα στο επίπεδο εφαρμογής. Αποτελεί ένα σημαντικό επικοινωνιακό θεμέλιο για τον παγκόσμιο ιστό, καθώς διαμορφώνει και μεταδίδει τα μηνύματα μεταξύ πελάτη και server. Συγκεκριμένα ο πελάτης εκδίδει ένα έτοιμα, το οποίο λαμβάνεται και επεξεργάζεται από τον server και μετά στέλνει μια απάντηση. [11]

1.4.B Πρωτόκολλο AMQP (Advanced Message Queuing Protocol)

Ένα δυικό πρωτόκολλο στο επίπεδο εφαρμογής το οποίο δημιουργήθηκε για την υποστήριξη μεγάλου εύρους αποτελεσματικών και ασφαλών μηνυμάτων ανάμεσα σε συστήματα. Είναι κατάλληλο για την ασύγχρονη επικοινωνία μεταξύ αποστολέα και παραλήπτη ακόμη και όταν λειτουργούν ανεξάρτητα. [12]

1.4.C Πρωτόκολλο ZigBee

Το ZigBee είναι ένα πρωτόκολλο επικοινωνίας υψηλού επιπέδου και ένα από τα πιο δημοφιλή πρωτόκολλα MeSH (Medical Subject Headings), ειδικό για την δημιουργία και εκτέλεση εφαρμογών αυτοματοποίησης χαμηλής ισχύος και ροής δεδομένων, καθώς και μικρής εμβέλειας σε σπίτια. [12]

1.4.D Πρωτόκολλο Matter

Ένα δωρεάν και open-source πρωτόκολλο, το οποίο δημιουργήθηκε από την Amazon, την Google, την Apple και την CSA (Connectivity Standards Alliance), με σκοπό την ένωση οικοσυστημάτων έξυπνων σπιτιών. [13]

1.4.E Πρωτόκολλο NB-IoT (Narrowband Internet of Things)

Αναπτύχθηκε από την 3GPP (3rd Generation Partnership Project) και είναι τύπος IoT επικοινωνίας χαμηλής ισχύς σε ευρεία περιοχή (LPWA). Είναι κατάλληλο για συλλογή και ανταλλαγή δεδομένων μεταξύ τερματικών με λιγότερους πόρους από άλλες τεχνολογίες όπως GPRS, 3G, LTE. [14]

1.4.F Πρωτόκολλο LoRaWAN (Long Range WAN)

Είναι ένα στάνταρ πρωτόκολλο επικοινωνίας για ασύρματα δίκτυα ευρείας περιοχής και χαμηλής ισχύς. Σε IoT εφαρμογές παρατηρείται μεγάλη ποικιλία σε ασύρματες τεχνολογίες και σε τοπικό αλλά και σε ευρύ δίκτυο. [12]

1.4.G Πρωτόκολλο CoAP (Constrained Application Protocol)

Το CoAP είναι ένα εξιδεικευμένο πρωτόκολλο μεταφοράς, για περιορισμένους κόμβους και δίκτυα στο IoT. Είναι σχεδιασμένο για τη λειτουργία και σύνδεση συσκευών με περιορισμένο επεξεργαστή, μνήμη και πόρους, κάτι το οποίο το κάνει κατάλληλο για επικοινωνία M2M (Machine to Machine). [12]

1.4.H Πρωτόκολλο MQTT (Message Queuing Telemetry Transport)

Το πιο σύνηθες πρωτόκολλο σε χρήση για επικοινωνία M2M, καθώς είναι σχεδιασμένο ακριβώς για συσκευές με χαμηλό εύρος ζώνης και υψηλής καθυστέρησης. Το μοντέλο του, Publish / Subscribe, επιτρέπει σε πολλούς clients να έρχονται σε επαφή μεταξύ τους, με την επικοινωνία τους να γίνεται μέσω ενός διαμεσολαβητή (Broker). Οι clients αρχικά εκδίδουν μηνύματα σε έναν broker. Ο broker έπειτα λαμβάνει και φιλτράρει τα μηνύματα, με βάση το topic στα οποία είναι εγγεγραμμένα, και τα στέλνει στους πελάτες που είναι και αυτοί εγγεγραμμένοι στο ίδιο topic. Από τα σημαντικότερα χαρακτηριστικά του πρωτοκόλλου MQTT είναι η εγγύηση στην παροχή των

μηνυμάτων που λέγεται QoS (Quality of Service). Η συγκεκριμένη λειτουργία έχει 3 επίπεδα:

- 1) **QoS (0)**: Το χαμηλότερο επίπεδο υπηρεσίας όπου τα μηνύματα παραδίδονται στον subscriber μέχρι μία φορά και δεν χρειάζεται αναγνώριση. Ο publisher στέλνει τα μηνύματα και δεν τα αποθηκεύει.
- 2) **QoS (1)**: Εδώ υπάρχει βεβαίωση πως τα μηνύματα παραδίδονται στον subscriber τουλάχιστον μία φορά. Ο publisher αποθηκεύει τα μηνύματα μέχρι να λάβει αναγνώριση από τον broker. Σε διαφορετική περίπτωση ο publisher ξαναστέλνει τα μηνύματα.
- 3) **QoS(2)**: Σε αυτό το επίπεδο τα μηνύματα παραδίδονται στον πελάτη ακριβώς μια φορά. Με αυτό το τρόπο δεν υπάρχει περιθώριο για να σταλθεί το ίδιο μήνυμα 2 φορές. [15]

1.5 IoT Broker EMQX



EMQX

Ο EMQX είναι ένας διαμεσολαβητής (broker), για το πρωτόκολλο MQTT, και χρησιμοποιείται ευρέως σε εφαρμογές IoT. Χαρακτηρίζεται από την υψηλή του απόδοση, μιας και έχει την ικανότητα να χειρίζεται μεγάλους όγκους δεδομένων, διατηρώντας παράλληλα καθυστέρηση σε χαμηλά επίπεδα, το οποίο είναι ζωτικής σημασίας για της IoT εφαρμογές. Ως ένας διαμεσολαβητής ανοιχτού κώδικα, μπορεί να προσαρμοσθεί και να επεκταθεί σύμφωνα με τις ανάγκες της εφαρμογής. Μια απλή σύνδεση στο συγκεκριμένο διαμεσολαβητή πραγματοποιείται στην πύλη επικοινωνίας TCP: 1883. Ανάλογα το πρωτόκολλο επικοινωνίας και τον δείκτη ασφάλειας που αναζητά ο χρήστης, υπάρχουν οι κατάλληλες πύλες [19]. Συγκεκριμένα:

- i) Πύλη WebSocket: 8083
 - ii) Ασφαλή πύλη WebSocket: 8084
 - iii) Πύλη SSL/TLS: 8883
 - iv) Πύλη QUIC: 14567
- [20]

ΚΕΦΑΛΑΙΟ 2: Διαδικασία Εφαρμογής

2.1 Back-End

Παρακάτω, θα αναλυθούν τα εργαλεία και τα περιβάλλοντα που χρησιμοποιήθηκαν για την υλοποίηση του Back-End μέρος της πτυχιακής

2.1.1 Arduino IDE



Arduino IDE

Το Arduino IDE (Integrated Development Environment) είναι το πιο διαδεδομένο περιβάλλον για τη συγγραφή κώδικα για πλακέτες Arduino, καθώς και άλλες πλακέτες συμβατές με αυτό. Αυτό το απλό και φιλικό προς τον χρήστη περιβάλλον, είναι ένας από τους κύριους λόγους που η δημιουργία projects σε πλακέτες Arduino έχει γίνει πολύ δημοφιλής. Τα βασικά χαρακτηριστικά που έχει το Arduino IDE, τα οποία είναι κοινά και με οποιοδήποτε άλλο προγραμματιστικό περιβάλλον, περιλαμβάνουν την ανίχνευση λαθών (debugging) του κώδικα καθώς και την αυτόματη ολοκλήρωση (autocompletion) Τα ιδιαίτερα του χαρακτηριστικά που κάνουν το Arduino IDE ξεχωριστό, περιλαμβάνουν [21]:

- 1) Χειρισμός Βιβλιοθηκών (Library Manager):** Υποστηρίζει τη χρήση βιβλιοθηκών κώδικα επεκτείνοντας της λειτουργίες του Arduino και διευκολύνουν την ανάπτυξη εφαρμογών. [21]
- 2) Χειρισμός πολλαπλών πλακετών (Boards Manager):** Παρέχεται η δυνατότητα πλοήγησης και εγκατάστασης πακέτων για οποιαδήποτε πλακέτα συμβατή με το περιβάλλον. [21]

- 3) **Σειριακή Οθόνη (Serial Monitor):** Ένα «παράθυρο», όπου επιτρέπει στον χρήστη, κατά την εκτέλεση του προγράμματος, την επικοινωνία με την πλακέτα και την εμφάνιση μηνυμάτων ή δεδομένων [21]
- 4) **Σειριακός Σχεδιογράφος (Serial Plotter):** Ένα χρήσιμο εργαλείο που μετατρέπει τα δεδομένα, όπως φαίνονται στη σειριακή οθόνη, σε οπτική μορφή μέσω γραφήματος. Κατάλληλο για παρατήρηση και αξιολόγηση των αυξομειώσεων των τιμών. [21]

2.1.2 NodeJS



Το NodeJS είναι ένα ασύγχρονο περιβάλλον εκτέλεσης κώδικα JavaScript, από τη πλευρά του server. Δημιουργήθηκε το 2009 και βασίζεται στη μηχανή JavaScript V8 του Google Chrome, η οποία εκτελεί τον κώδικα JavaScript. Με το NodeJS ο κώδικας τρέχει απευθείας στο λειτουργικό σύστημα και όχι μέσα σε έναν browser. Έτσι παρέχεται η δυνατότητα πρόσβασης σε διάφορες λειτουργίες του συστήματος όπως το σύστημα αρχείων και το δίκτυο. [22]

Συγγράφοντας κώδικα JavaScript με NodeJS, είναι ο σύνηθες τρόπος για τη δημιουργία δυναμικών εφαρμογών πραγματικού χρόνου, συνδυάζοντάς το με frameworks όπως το Express.js, οι οποίες απαιτούν συνεχή επικοινωνία με τον server.



Η MongoDB είναι μια πολύ διαδεδομένη βάση δεδομένων τύπου NoSQL που χρησιμοποιείται από πολλές σύγχρονες εφαρμογές. Αποθηκεύει τα δεδομένα σε ευέλικτα έγγραφα σε μορφή JSON και χρησιμοποιεί διάφορα μοντέλα για την αποθήκευση δεδομένων, όπως είναι το μοντέλο εγγράφων, το μοντέλο κλειδιού-τιμής και το μοντέλο γραφημάτων. Χαρακτηρίζεται από την ευελιξία της καθώς η μορφή JSON επιτρέπει την αποθήκευση πολύπλοκων και ετερογενών δεδομένων, χωρίς την αυστηρή δομή των παραδοσιακών βάσεων δεδομένων. Λόγω της κλιμακωσιμότητάς της, μπορεί να χειριστεί μεγάλο όγκο δεδομένων και με την οριζόντια κλιμάκωση ανταποκρίνεται σε αυξανόμενες απαιτήσεις. Τέλος η υψηλή της απόδοση την κάνει κατάλληλη για εφαρμογές πραγματικού χρόνου. [23]



Το MongoDB Atlas είναι μια υπηρεσία βάσης δεδομένων, νέφους και βασίζεται στο δημοφιλές σύστημα βάσεων δεδομένων MongoDB. Σχεδιάστηκε για να απλοποιήσει τη διαχείριση και την κλιμάκωση βάσεων δεδομένων, προσφέροντας μια ευέλικτη και κλιμακωτή λύση για αποθήκευση δεδομένων. Καλύπτοντας ένα μεγάλο εύρος αναγκών, από μικρές ατομικές εφαρμογές, μέχρι μεγάλες εφαρμογές για εταιρίες. [24]

2.2 Front-End

Στο δεύτερο μέλος της υλοποίησης, το κομμάτι front-end, θα αναλυθεί το πρωτόκολλο που χρησιμοποιήθηκε για την επικοινωνία μεταξύ του server και της ιστοσελίδας.

2.2.1 Πρωτόκολλο WebSocket

Το πρωτόκολλο WebSocket αποτελεί μια σημαντική εξέλιξη στον τρόπο επικοινωνίας, καθώς παρέχει την συνεχή και αδιάκοπη επικοινωνία μεταξύ πελάτη και διακομιστή, μέσω μιας και μόνο σύνδεσης με το πρωτόκολλο TCP. Σχεδιασμένο ώστε οι επικοινωνίες τους να γίνεται σε πραγματικό χρόνο, αποτελεί την ιδανική λύση για την αποτελεσματική και ανταποκρίσιμη μεταφορά δεδομένων σε web εφαρμογές. [16]

Κεφάλαιο 3: Αρχιτεκτονική της Εφαρμογής

3.1 Υλοποίηση

Στα πλαίσια αυτής της πτυχιακής, αναπτύχθηκε μια IoT εφαρμογή, η οποία μεταφέρει τις ενδείξεις από έναν αισθητήρα σε έναν server σε πραγματικό χρόνο, αποθηκεύοντας τα σε μια βάση δεδομένων. Έπειτα η αυξομείωση των δεδομένων φαίνονται στο γράφημα μιας ιστοσελίδας.

3.2 Περιγραφή

Ο σκοπός της πτυχιακής είναι η κατανόηση των τεχνολογιών IoT καθώς και τις εφαρμογές τους. Απαραίτητο στη κατανόηση ήταν η ανάπτυξη εφαρμογής IoT πραγματικού χρόνου. Στο πρώτο μέρος, το back-end της εφαρμογής, κατασκευάστηκε μια συσκευή για την καταμέτρηση των επιπέδων υγρασίας στο χώμα οικόσιτων φυτών, με την πλακέτα NodeMCU ESP8266. Έπειτα στο περιβάλλον Arduino IDE η πλακέτα προγραμματίστηκε να εκτυπώνει το ποσοστό της υγρασίας κάθε 5". Στη συνέχεια δημιουργήθηκε ένα αρχείο σε μορφή JSON το οποίο έχει μία ενδεικτική λίστα με οικόσιτα φυτά. Τα ποσοστά υγρασίας μεταφέρονται από το Arduino IDE, σε έναν server NodeJS γραμμένο στη γλώσσα JavaScript, σε πραγματικό χρόνο, με την επικοινωνία τους να γίνεται μέσω του διαμεσολαβητή MQTT. Το μέρος αυτό της εφαρμογής,

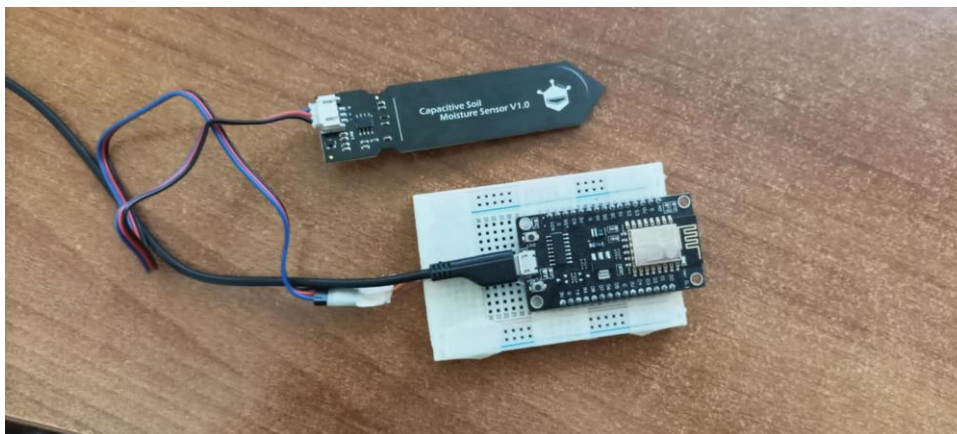
ολοκληρώνεται με την αποθήκευση της υγρασίας, ανάλογα με το φυτό, σε μια βάση δεδομένων MongoDB και πιο συγκεκριμένα με την υπηρεσία νέφους MongoDB Atlas. Στο δεύτερο μέρος, το front-end της εφαρμογής, έχει υλοποιηθεί μία ιστοσελίδα, η οποία απαρτίζεται από δύο μέρη. Το πρώτο μέρος είναι το ταμπλό (dashboard) της ιστοσελίδας και περιλαμβάνει ένα τερματικό, που εκτυπώνει τα ποσοστά της υγρασίας κατά την εκτέλεση της εφαρμογής, με τις αυξομειώσεις να αποτυπώνονται σε μορφή γραφήματος. Το δεύτερο μέρος είναι ένας κατάλογος με τα φυτά της ενδεικτικής λίστας, και αναγράφονται γενικές πληροφορίες καθώς και ειδικές οδηγίες φροντίδας και περιποίησης για κάθε φυτό ξεχωριστά.

Κεφάλαιο 4: Εκτέλεση της Εφαρμογής

Παρακάτω θα γίνει η επίδειξη της εφαρμογής. Από την δημιουργία του μετρητή μέχρι την προβολή των αποτελεσμάτων στο γράφημα της ιστοσελίδας.

4.1 Κατασκευή του μετρητή

Για τις ανάγκες της εφαρμογής, κατασκευάστηκε ένας μετρητής, για την καταγραφή της υγρασίας για οικόσιτα φυτά. Εκτός από την πλακέτα NodeMCU esp8266, χρησιμοποιήθηκε ένας αναλογικός αισθητήρας που λειτουργεί με διεπαφή τριών ακίδων. Οι τρεις ακίδες περιλαμβάνουν έναν κοινό συλλέκτη τάσης (VCC) στα 3.3V με 5.5V, την γείωση (GND) και το σήμα, αναλογικό σε αυτή τη περίπτωση (SIG). Τέλος τα εξαρτήματα τοποθετήθηκαν σε ένα breadboard 400 εισόδων.

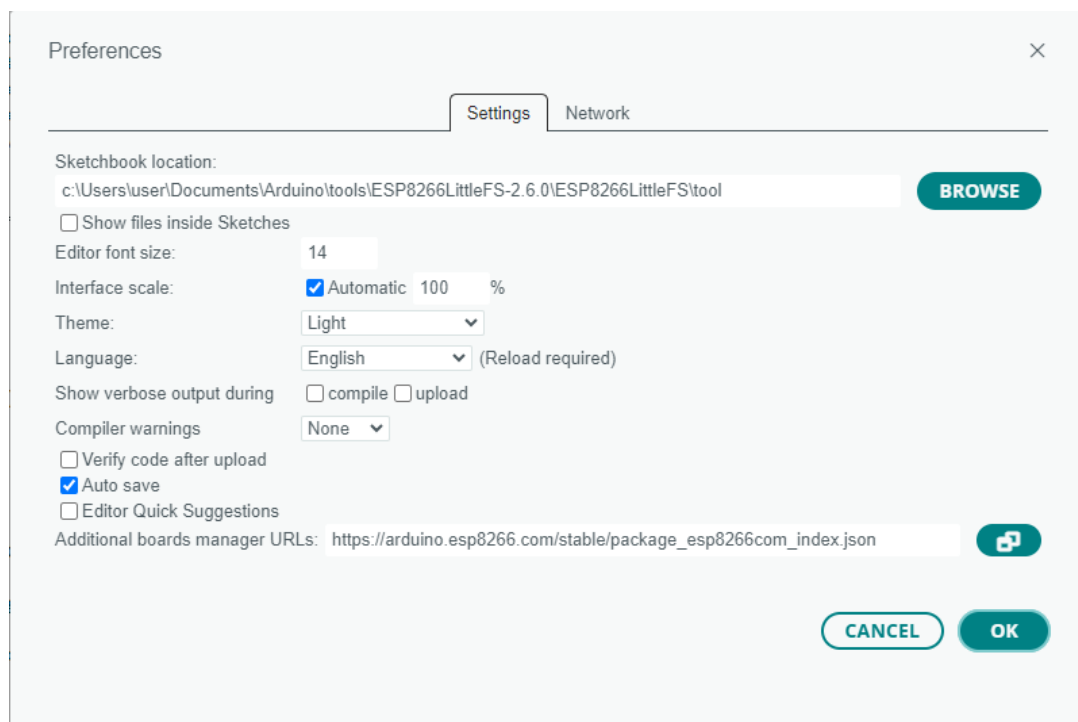


(Εικόνα 4.1): Τα εξαρτήματα τοποθετημένα στο breadboard

Τα χρώματα στο καλώδιο το αισθητήρα προσδιορίζουν μια ξεχωριστή ακίδα στη διεπαφή τριών ακίδων. Το μπλε καλώδιο είναι για την σύνδεση με την ακίδα αναλογικού σήματος A0 (SIG), το κόκκινο καλώδιο συνδέεται στην τάση της πλακέτας (VCC) και το μαύρο καλώδιο συνδέεται στη γείωση (GND). Για την ανάγνωση και ηλεκτροδότηση, η πλακέτα συνδέεται σε θύρα USB.

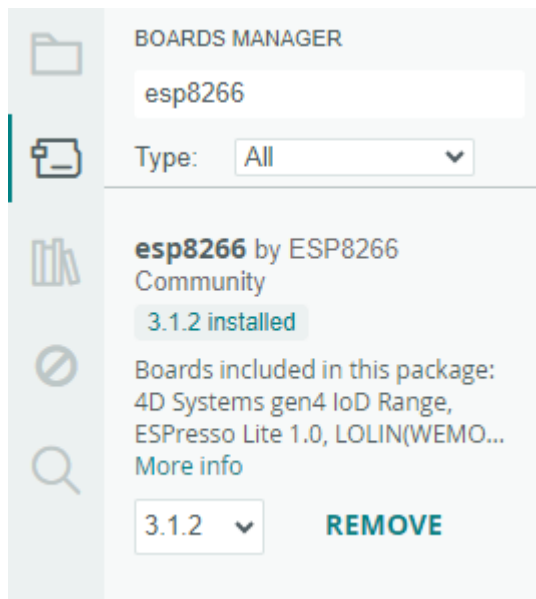
4.2 Εκτέλεση και προβολή αποτελεσμάτων στο Arduino IDE

Για να επικοινωνήσει το περιβάλλον Arduino IDE με την πλακέτα NodeMCU esp8266, θα πρέπει να ακολουθηθούν κάποιες ενέργειες πρώτα. Αρχικά γίνεται η προσθήκη του συνδέσμου http://arduino.esp8266.com/stable/package_esp8266com_index.json που αναγνωρίζει την πλακέτα. Ανοίγοντας το Arduino IDE, μετά **Files** και τέλος **Preferences**, γίνεται η αντιγραφή του συνδέσμου στο πεδίο **Additional boards manager URL's**, όπως φαίνεται στη παρακάτω εικόνα



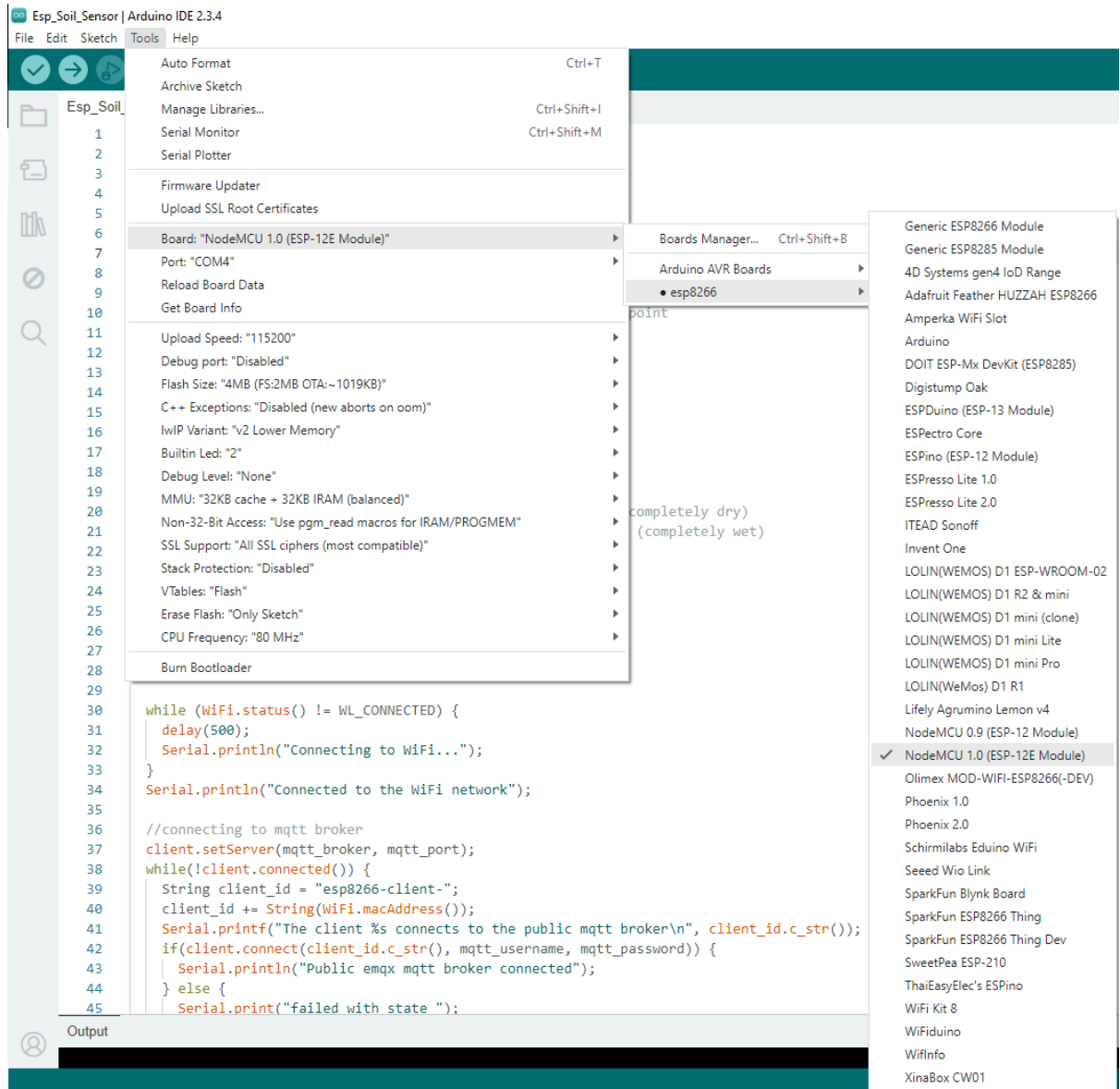
(Εικόνα 4.2)

Το επόμενο βήμα είναι η εγκατάσταση της πλακέτας **esp8266** από την κοινότητα ESP8266, στο **BOARDS MANAGER**.

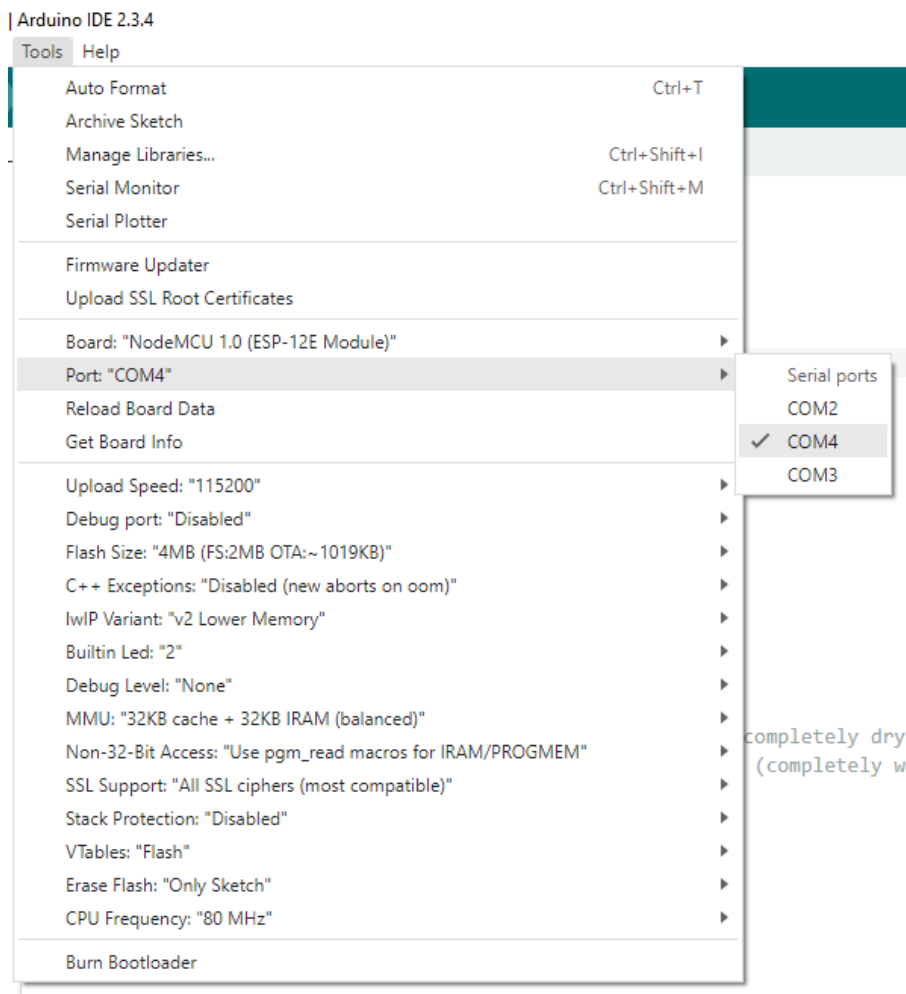


(Εικόνα 4.3): Εγκατάσταση τη πλακέτας

Στη συνέχεια επιλέγεται η NodeMCU πλακέτα **NodeMCU 1.0 (ESP-12E Module)** (Εικόνα 4.4) και η πύλη (τύπου COM για τα Windows) **COM4** (Εικόνα 4.5)



(Εικόνα 4.4): Επιλογή της πλακέτας



(Εικόνα 4.5): Επιλογή της πύλης

Το πρώτο βήμα για την συγγραφή του προγράμματος είναι η εγκατάσταση των βιβλιοθηκών: **ESP8266WiFi, PubSubClient, ArduinoJson**.

4.2.1 Βιβλιοθήκη ESP8266WiFi

Η βιβλιοθήκη **ESP8266WiFi**, επιτρέπει στη πλακέτα esp8266 να στέλνει και να λαμβάνει δεδομένα με τη σύνδεση στο τοπικό WiFi. Η δήλωση της βιβλιοθήκης στον κώδικα γίνεται με την εντολή:

#include <ESP8266WiFi.h> [25]. Μετά από τη δήλωση στον κώδικα, αναγράφονται τα κομμάτια τα οποία δηλώνουν το όνομα και τον κωδικό του WiFi (**Εικόνα 4.6**), καθώς και την διαδικασία και τον έλεγχο της σύνδεσης. (**Εικόνα 4.7**).

```

5 //wifi settings
6 const char *ssid = "your ssid";
7 const char *password = "your password";
8

```

Εικόνα(4.6)

```

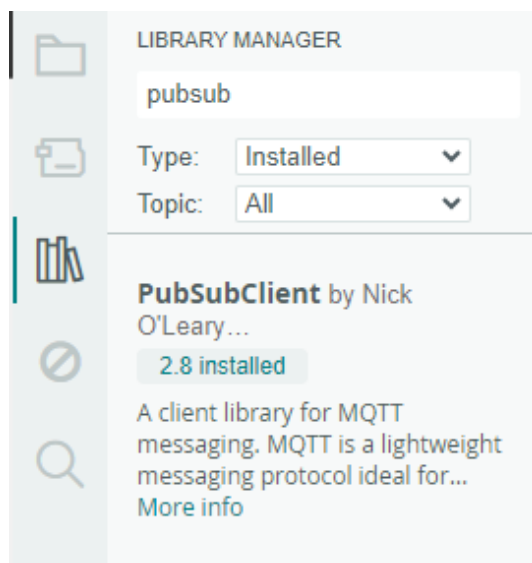
26 void setup() {
27     Serial.begin(9600);
28     WiFi.begin(ssid, password);
29
30     while (WiFi.status() != WL_CONNECTED) {
31         delay(500);
32         Serial.println("Connecting to WiFi...");
33     }
34     Serial.println("Connected to the WiFi network");
35

```

Εικόνα(4.7)

4.2.2 Βιβλιοθήκη PubSubClient

Απαραίτητη βιβλιοθήκη, για επικοινωνία μέσω του πρωτοκόλλου MQTT όπου οι συσκευή χρειάζεται να δημοσιεύει και να εγγράφεται σε μηνύματα μέσω διαμεσολαβητών [26]. Στο περιβάλλον, η εγκατάσταση γίνεται μέσω του **Library Manager**. (Εικόνα 4.8)



(Εικόνα 4.8)

Παρομοίως με την προηγούμενη βιβλιοθήκη, η δήλωσή της γίνεται με την εντολή: **#include <PubSubClient.h>**

Για τις ανάγκες της εφαρμογής, χρησιμοποιείται ο δημόσιος διαμεσολαβητής **emqx**

```
9 //mqtt broker settings
10 const char *mqtt_broker = "broker.emqx.io"; // emqx broker endpoint
11 const char *mqtt_topic = "esp8266/test"; // mqtt topic
12 const char *mqtt_username = "emqx_test";
13 const char *mqtt_password = "emqx_test";
14 const int mqtt_port = 1883; // mqtt tcp port
15
```

(Εικόνα 4.9)

- i) **broker.emqx.io**, είναι το τερματικό σημείο του δημοσίου διαμεσολαβητή
- ii) **esp8266/test**, είναι το **topic** στο οποίο θα γίνεται η δημοσίευση και η εγγραφή των μηνυμάτων.
- iii) **Πύλη 1883**, είναι η πύλη επικοινωνίας μεταξύ πελάτη και διακομιστή
- iv) **emqx_test**, είναι το όνομα χρήστη και ο κωδικός για την αυθεντικοποίηση του πελάτη

Τέλος είναι απαραίτητο το κομμάτι κώδικα για τον έλεγχο της σύνδεσης με τον διαμεσολαβητή και κατάλληλα μηνύματα σε περίπτωση λάθους. **(Εικόνα 4.10)**

```
36 //connecting to mqtt broker
37 client.setServer(mqtt_broker, mqtt_port);
38 while(!client.connected()) {
39     String client_id = "esp8266-client-";
40     client_id += String(WiFi.macAddress());
41     Serial.printf("The client %s connects to the public mqtt broker\n", client_id.c_str());
42     if(client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
43         Serial.println("Public emqx mqtt broker connected");
44     } else {
45         Serial.print("failed with state ");
46         Serial.print(client.state());
47         delay(2000);
48     }
49 }
50
```

(Εικόνα 4.10)

4.2.3 Βιβλιοθήκη ArduinoJSON

Εξαιρετικά δημοφιλής βιβλιοθήκη η οποία επιτρέπει στα προγράμματα Arduino να χειρίζονται αρχεία σε μορφή JSON [27]. Η εγκατάσταση της στο περιβάλλον γίνεται με τον ίδιο τρόπο στην **(Εικόνα 4.8)**, με την δήλωση της στο πρόγραμμα να γίνεται μέσω της εντολής: **#include <ArduinoJson.h>**.

```
73 //json serialize
74 DynamicJsonDocument data(256);
75 data["moisture_percentage"] = moisturePercentage;
76
77 //publish moisture
78 char json_string[256];
79 serializeJson(data, json_string);
80
81 Serial.println(json_string);
82 client.publish(mqtt_topic, json_string, false);
83 }
84 }
```

(Εικόνα 4.11)

4.2.4 Εκτέλεση στο Arduino IDE

Για να αναγνωριστεί ο αισθητήρας από την πλακέτα, ορίζεται στο πρόγραμμα με την μεταβλητή **A0**. Η εφαρμογή είναι προγραμματισμένη ώστε να βγάζει νέα τιμή χωρίς όμως να διαγράφει την προηγούμενη. Η πρακτική προβολή των αποτελεσμάτων είναι σε ποσοστό, οπότε για την μετατροπή τους ορίστηκε ως το 0, η τιμή όπου το χώμα είναι πολύ υγρό και 1023, η τιμή όπου το χώμα είναι πολύ ξηρό. **(Εικόνα 4.12) & (Εικόνα 4.13)**. Νέες ενδείξεις βγαίνουν κάθε 5". **(Εικόνα 4.14)**

```
16 #define sensorPIN A0
17 unsigned long previousMillis = 0;
18
19 // Define moisture thresholds for mapping
20 const int dryValue = 1023; // Value when sensor is in air (completely dry)
21 const int wetValue = 0; // Value when sensor is in water (completely wet)
--
```

(Εικόνα 4.12)

```

51
52 float getMoisturePercentage(int rawValue) {
53     // Convert raw value to percentage
54     float percentage = map(rawValue, dryValue, wetValue, 0, 100);
55     // Constrain the value between 0 and 100
56     percentage = constrain(percentge, 0, 100);
57     return percentage;
58 }

```

(Εικόνα 4.13)

```

60 void loop() {
61     client.loop();
62     unsigned long currentMillis = millis();
63
64     // data are published every 5 seconds
65     if (currentMillis - previousMillis >= 5000) {
66         previousMillis = currentMillis;
67
68         // Read raw moisture value
69         int rawMoistureValue = analogRead(sensorPIN);
70         // Convert to percentage
71         float moisturePercentage = getMoisturePercentage(rawMoistureValue);
72

```

(Εικόνα 4.14)

Το πρόγραμμα **Esp_Soil_Sensor.ino** σε όλη του τη μορφή:
(Εικόνα 4.15) & (Εικόνα 4.16)



```
1 #include <ESP8266WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4
5 //wifi settings
6 const char *ssid = "your ssid";
7 const char *password = "your password";
8
9 //mqtt broker settings
10 const char *mqtt_broker = "broker.emqx.io"; // emqx broker endpoint
11 const char *mqtt_topic = "esp8266/test"; // mqtt topic
12 const char *mqtt_username = "emqx_test";
13 const char *mqtt_password = "emqx_test";
14 const int mqtt_port = 1883; // mqtt tcp port
15
16 #define sensorPIN A0
17 unsigned long previousMillis = 0;
18
19 // Define moisture thresholds for mapping
20 const int dryValue = 1023; // Value when sensor is in air (completely dry)
21 const int wetValue = 0; // Value when sensor is in water (completely wet)
22
23 WiFiClient espClient;
24 PubSubClient client(espClient);
25
26 void setup() {
27   Serial.begin(9600);
28   WiFi.begin(ssid, password);
29
30   while (WiFi.status() != WL_CONNECTED) {
31     delay(500);
32     Serial.println("Connecting to WiFi...");
33   }
34   Serial.println("Connected to the WiFi network");
35
36   //connecting to mqtt broker
37   client.setServer(mqtt_broker, mqtt_port);
38   while(!client.connected()) {
39     String client_id = "esp8266-client-";
40     client_id += String(WiFi.macAddress());
41     Serial.printf("The client %s connects to the public mqtt broker\n", client_id.c_str());
42     if(client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
43       Serial.println("Public emqx mqtt broker connected");
44     } else {
45       Serial.print("failed with state ");
```

(Εικόνα 4.15)



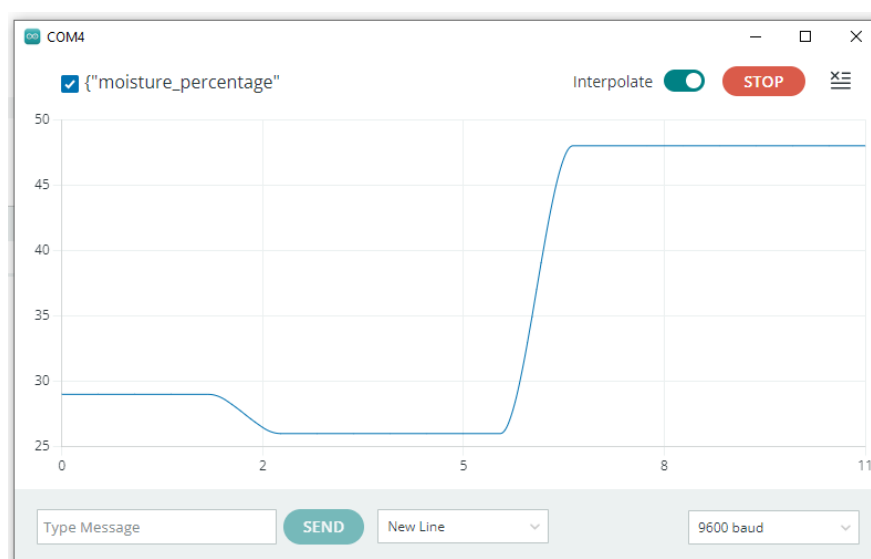
```
40 client_id += String(WiFi.macAddress());
41 Serial.printf("The client %s connects to the public mqtt broker\n", client_id.c_str());
42 if(client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
43     Serial.println("Public emqx mqtt broker connected");
44 } else {
45     Serial.print("failed with state ");
46     Serial.print(client.state());
47     delay(2000);
48 }
49 }
50 }
51
52 float getMoisturePercentage(int rawValue) {
53     // Convert raw value to percentage
54     float percentage = map(rawValue, dryValue, wetValue, 0, 100);
55     // Constrain the value between 0 and 100
56     percentage = constrain(percentge, 0, 100);
57     return percentage;
58 }
59
60 void loop() {
61     client.loop();
62     unsigned long currentMillis = millis();
63
64     // data are published every 5 seconds
65     if (currentMillis - previousMillis >= 5000) {
66         previousMillis = currentMillis;
67
68         // Read raw moisture value
69         int rawMoistureValue = analogRead(sensorPIN);
70         // Convert to percentage
71         float moisturePercentage = getMoisturePercentage(rawValue);
72
73         //json serialize
74         DynamicJsonDocument data(256);
75         data["moisture_percentage"] = moisturePercentage;
76
77         //publish moisture
78         char json_string[256];
79         serializeJson(data, json_string);
80
81         Serial.println(json_string);
82         client.publish(mqtt_topic, json_string, false);
83     }
84 }
```

(Εικόνα 4.16)

Στις παρακάτω εικόνες θα παρατεθούν τα αποτελέσματα της εκτέλεσης. Μέσω του συριακού ελεγκτή εκπομπών **(Εικόνα 4.17)** και του συριακού σχεδιογράφου **(Εικόνα 4.18)**.

```
Connected to the WiFi network
The client esp8266-client-D8:F1:5B:00:42:CE connects to the public mqtt broker
Public emqx mqtt broker connected
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":29}
{"moisture_percentage":26}
{"moisture_percentage":26}
{"moisture_percentage":26}
{"moisture_percentage":26}
{"moisture_percentage":48}
{"moisture_percentage":48}
{"moisture_percentage":48}
```

(Εικόνα 4.17)



(Εικόνα 4.18)

4.3 Σύνδεση με τον NodeJS Server

Το επόμενο στάδιο της εφαρμογής είναι η κατασκευή του NodeJS server. Για την δημιουργία του, γίνεται η εκτέλεση της εντολής **npm init -y** στη γραμμή εντολών του NodeJS. Σκοπός του server είναι να παίζει τον ρόλο του συνδετικού κρίκου ανάμεσα στην IoT συσκευή, την ενδεικτική λίστα των οικόσιτων φυτών σε μορφή JSON, την βάση δεδομένων σε MongoDB και το γράφημα της ιστοσελίδας. Οι βιβλιοθήκες που χρειάστηκαν είναι οι εξής.

4.3.1 Βιβλιοθήκες

- 1) **Βιβλιοθήκη fs (Fyle System):** Μια από τις ενσωματωμένες βιβλιοθήκες κατά τη δημιουργία του server, η βιβλιοθήκη fs διαβάσει και γράφει αρχεία σε συγχρονισμένη ή ασύγχρονη μορφή. Στο πρόγραμμα διαβάζει το αρχείο **plants.json**, την ενδεικτική λίστα με τα οικόσιτα φυτά.

```
29 // Load plant data
30 const plantData = JSON.parse(fs.readFileSync('./plants.json', 'utf8'));
31
```

(Εικόνα 4.19)

- 2) **Βιβλιοθήκη readline:** Παρομοίως με την fs, η βιβλιοθήκη readline είναι και αυτή μια από τις ενσωματωμένες βιβλιοθήκες κατά τη δημιουργία του server. Με τη βοήθεια αυτής της βιβλιοθήκης δημιουργείται μια διεπαφή, όπου ο χρήστης κατά την εκτέλεση του προγράμματος στο τερματικό, ερωτάται ποιο φυτό θέλει να επιλέξει για τον έλεγχο της υγρασίας του.

```

46   async selectPlant() {
47       const rl = readline.createInterface({
48           input: process.stdin,
49           output: process.stdout
50       });
51
52       this.log('Available plants:');
53       plantData.plants.forEach((plant, index) => {
54           this.log(`${index + 1}. ${plant.name}`);
55       });
56
57       return new Promise((resolve) => {
58           rl.question('Select a plant to monitor (enter number): ', (answer) => {
59               const plantIndex = parseInt(answer, 10) - 1;
60
61               if (plantIndex >= 0 && plantIndex < plantData.plants.length) {
62                   this.selectedPlant = plantData.plants[plantIndex];
63                   this.log(`Selected plant: ${this.selectedPlant.name}`);
64                   resolve();
65               } else {
66                   this.log('Invalid selection. Using default plant.');

```

(Εικόνα 4.20)

- 3) **Βιβλιοθήκη mqtt**: Καθιστά την επικοινωνία μεταξύ του αισθητήρα, και των ενδείξεών του στο περιβάλλον Arduino IDE, με τον server εφικτή μέσω του πρωτοκόλλου MQTT και του διαμεσολαβητή EMQX. Στη γραμμή εντολών η εγκατάσταση γίνεται με την εντολή **npm install mqtt**. Για την επικοινωνία τους, ο server πρέπει να υπάγεται στις ίδιες ρυθμίσεις που αναγράφονται στην (Εικόνα 4.20) του Arduino IDE. Τα αρχεία **connectionCred.js** (Εικόνα 4.21) και **server.js** (Εικόνα 4.22) δείχνουν τη σύνδεση.

```

Back-End > JS connectionCred.js > [🔗] connectionOptions > 🔑 host
1   const connectionOptions = {
2       protocol: 'mqtt',
3       port: 1883,
4       host: 'broker.emqx.io'
5   }
6
7   module.exports = {
8       connectionOptions
9   };

```

(Εικόνα 4.21)

```

16   mqtt: {
17     options: {
18       clientId: `mqtt_client_${Math.random().toString(16).slice(2, 8)}`,
19       username: 'emqx_test',
20       password: 'emqx_test'
21     }
22   },

```

(Εικόνα 4.22)

Έπειτα γίνεται η σύνδεση με τον broker για το topic **esp8266/test** (Εικόνα 4.23) και ο χειρισμός των μηνυμάτων του πρωτοκόλλου (Εικόνα 4.24).

```

connectMQTT() {
  const { protocol, host, port } = connectionOptions;
  const mqttUrl = `${protocol}://${host}:${port}`;

  this.mqttClient = mqtt.connect(mqttUrl, {
    ...config.mqtt.options,
    connectTimeout: 4000,
    reconnectPeriod: 1000
  });

  this.mqttClient.on('connect', () => {
    this.log('Connected to MQTT broker');
    this.mqttClient.subscribe('esp8266/test');
  });

  this.mqttClient.on('reconnect', (error) => {
    this.log(`Reconnecting to MQTT broker:`, error);
  });

  this.mqttClient.on('error', (error) => {
    this.log(`MQTT connection error:`, error);
  });

  this.mqttClient.on('message', (topic, payload) => this.handleMQTTMessage(payload));
}

```

(Εικόνα 4.23)

```

async handleMQTTMessage(payload) {
  try {
    const data = JSON.parse(payload);
    // Updated to use moisture_percentage from the ESP8266
    const moisturePercentage = data.moisture_percentage;

    if (!this.selectedPlant) {
      this.log('No plant selected. Please select a plant first.');
```

(Εικόνα 4.24)

- 4) **Βιβλιοθήκη mongodb:** Βιβλιοθήκη για την δημιουργία βάσης δεδομένων MongoDB και πιο συγκεκριμένα για τη αποθήκευση των ενδείξεων στην υπηρεσία MongoDB Atlas. Η εγκατάσταση της στον server γίνεται μέσω της εντολής **npm install mongodb**. Η δημιουργία της βάσης δεδομένων μέσω του MongoDB Atlas, γίνεται με τη βοήθεια μιας σειράς σύνδεσης η οποία προστίθεται στον κώδικα. Μετά από αυτό το βήμα δημιουργείται μία βάση δεδομένων και μία συλλογή, όπου εκεί θα αποθηκεύονται οι μετρήσεις. Για τις ανάγκες της εφαρμογής, η βάση δεδομένων ονομάζεται **plantMonitor** και συλλογή λέγεται **soilMoistureReadings**.

```

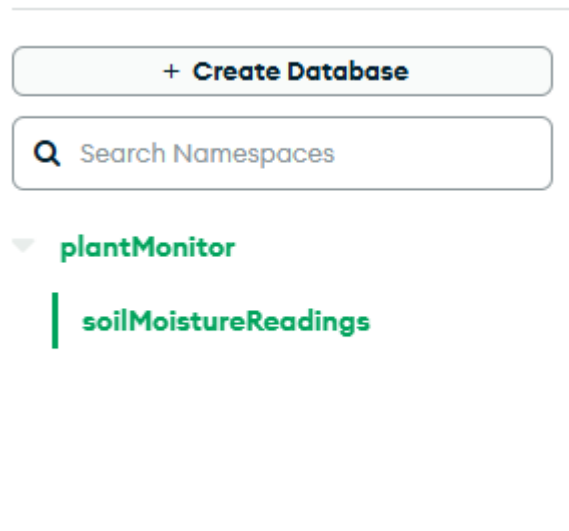
10 const config = {
11   mongodb: {
12     uri: "mongodb+srv://<db_user>:<db_password>@monitoringcluster.ff92g.mongodb.net/?retryWrites=true&w=majority",
13     dbName: 'plantMonitor',
14     collection: 'soilMoistureReadings'
15   },

```

(Εικόνα 4.25)

Στην (Εικόνα 4.25) η μεταβλητή **uri** αποθηκεύει τη σειρά σύνδεσης, και οι μεταβλητές **dbName** και **collection** αποθηκεύουν τις τιμές **plantMonitor** και

soilMoistureReadings αντίστοιχα. Ακριβώς δηλαδή τα στοιχεία για την βάση δεδομένων και την συλλογή στο MongoDB Atlas. **(Εικόνα 4.26)**



(Εικόνα 4.26)

Κατά την εκτέλεση γίνεται ο έλεγχος για την σύνδεση του server με την βάση δεδομένων **(Εικόνα 4.27)** και τα δεδομένα αποθηκεύονται σύμφωνα με το όνομα τους, την επιστημονική τους ονομασία, την ένδειξη τους και τη χρονική στιγμή που πραγματοποιήθηκε η ένδειξη. **(Εικόνα 4.28)**

```
async connectMongoDB() {  
  try {  
    this.mongoClient = new MongoClient(config.mongodb.uri);  
    await this.mongoClient.connect();  
    this.db = this.mongoClient.db(config.mongodb.dbName);  
    this.collection = this.db.collection(config.mongodb.collection);  
    this.log('Connected to MongoDB');  
  } catch (error) {  
    this.log('MongoDB connection error:', error.message);  
  }  
}
```

(Εικόνα 4.27)

```

    async saveMoistureReading(moisturePercentage, plant) {
      const reading = {
        plantName: plant.name,
        scientificName: plant.scientificName,
        moisturePercentage,
        timestamp: new Date()
      };
      await this.collection.insertOne(reading);
      this.log('Reading saved to MongoDB');
    }
  }

```

(Εικόνα 4.28)

- 5) **Βιβλιοθήκη express:** Ιδανική βιβλιοθήκη για τη δημιουργία ηλεκτρονικής διεπαφής για το NodeJS, με την ιδιότητα του να χειρίζεται HTTP αιτήσεις. Παράλληλα, βοηθάει στην εξυπηρέτηση στατικών αρχείων, όπως αυτά της ιστοσελίδας. Γίνεται εγκατάσταση μέσω της εντολής **npm install express**.
- 6) **Βιβλιοθήκη WebSocket:** Μπορεί η βιβλιοθήκη Express να χειρίζεται τον στατικά αρχεία, η βιβλιοθήκη WebSocket είναι υπεύθυνη για την επικοινωνία διπλής κατεύθυνσης σε πραγματικό χρόνο, μεταξύ του back-end και του front-end.

```

setupExpress() {
  this.app = express();
  this.app.use(express.static('public'));
  this.app.listen(config.server.port, () => {
    this.log(`HTTP server running at http://localhost:${config.server.port}`);
  });
}

setupWebSocket() {
  const wss = new WebSocket.Server({ port: config.server.wsPort });
  wss.on('connection', (ws) => {
    this.clients.add(ws);
    ws.on('close', () => this.clients.delete(ws));
  });
}

```

(Εικόνα 4.29)

```

23     server: {
24         port: 3000,
25         wsPort: 8080
26     }
27 };

```

(Εικόνα 4.30)

Στις εικόνες (4.29) & (4.30) γίνεται η εγκατάσταση του **Express** και του **WebSocket** στις πύλες **3000** και **8080** αντίστοιχα.

4.3.2 Εκτέλεση του server

Στην εκτέλεση του προγράμματος, αφού γίνει η σύνδεση με τον διαμεσολαβητή mqtt και με τη βάση δεδομένων και αφού γίνει η αρχικοποίηση του **Express** και του **WebSocket**, γίνεται ο έλεγχος της ένδειξης υγρασίας, ανάλογα με το φυτό που επιλέχθηκε εκτυπώνοντας το κατάλληλο μήνυμα.

```

158     checkMoistureLevel(moisturePercentage, plant) {
159         // Convert threshold values to percentages if they aren't already
160         const { minimum, optimal, maximum } = plant.moistureThresholds;
161         const alerts = plantData.generalGuidelines.alertThreshold;
162
163         if (moisturePercentage < minimum) {
164             this.log(`Alert: Moisture ${moisturePercentage}% below minimum ${minimum}%. ${alerts.belowMinimum}`);
165         } else if (moisturePercentage > maximum) {
166             this.log(`Alert: Moisture ${moisturePercentage}% above maximum ${maximum}%. ${alerts.aboveMaximum}`);
167         } else {
168             this.log(`Moisture ${moisturePercentage}% is within acceptable range (optimal: ${optimal}%)`);
169         }
170     }

```

(Εικόνα 4.31)

Τέλος τα δεδομένα με τη μέθοδο **broadcast** μεταδίδονται στην ιστοσελίδα. Ένα αποτέλεσμα που θα εξηγηθεί παρακάτω.

```

    log(...args) {
      const message = args.join(' ');
      console.log(message);
      this.broadcast({ type: 'log', message });
    }

    broadcast(message) {
      this.clients.forEach(client => {
        if (client.readyState === WebSocket.OPEN) {
          client.send(JSON.stringify(message));
        }
      });
    }
  }
}

```

(Εικόνα 4.32)

Ο κώδικας **server.js**:

```

1  const mqtt = require('mqtt'); 305.1k (gzipped: 84.6k)
2  const fs = require('fs'); 211 (gzipped: 166)
3  const readline = require('readline'); 217 (gzipped: 170)
4  const { MongoClient, Timestamp } = require('mongodb'); 773.2k (gzipped: 203.7k)
5  const express = require('express');
6  const WebSocket = require('ws'); 46.6k (gzipped: 13.3k)
7  const { connectionOptions } = require('./connectionCred');
8
9  // Configuration object
10 const config = {
11   mongodb: {
12     uri: "mongodb+srv://<db_user>:<db_password>@monitoringcluster.ff92g.mongodb.net/?retryWrites=true&w=majority",
13     dbName: 'plantMonitor',
14     collection: 'soilMoistureReadings'
15   },
16   mqtt: {
17     options: {
18       clientId: `mqtt_client_${Math.random().toString(16).slice(2, 8)}`,
19       username: 'emqx_test',
20       password: 'emqx_test'
21     }
22   },
23   server: {
24     port: 3000,
25     wsPort: 8080
26   }
27 };

```

(Εικόνα 4.33)

```

29 // Load plant data
30 const plantData = JSON.parse(fs.readFileSync('./plants.json', 'utf8'));
31
32 class PlantMonitor {
33   constructor() {
34     this.clients = new Set();
35     this.selectedPlant = null;
36     this.setupExpress();
37     this.setupWebSocket();
38   }
39
40   async start() {
41     await this.selectPlant();
42     await this.connectMongoDB();
43     this.connectMQTT();
44   }
45
46   async selectPlant() {
47     const rl = readline.createInterface({
48       input: process.stdin,
49       output: process.stdout
50     });
51
52     this.log('Available plants:');
53     plantData.plants.forEach((plant, index) => {
54       this.log(`${index + 1}. ${plant.name}`);
55     });
56
57     return new Promise((resolve) => {
58       rl.question('Select a plant to monitor (enter number): ', (answer) => {
59         const plantIndex = parseInt(answer, 10) - 1;
60
61         if (plantIndex >= 0 && plantIndex < plantData.plants.length) {
62           this.selectedPlant = plantData.plants[plantIndex];
63           this.log(`Selected plant: ${this.selectedPlant.name}`);
64           resolve();
65         } else {
66           this.log('Invalid selection. Using default plant.');
```

(Εικόνα 4.34)

```

75     setupExpress() {
76         this.app = express();
77         this.app.use(express.static('public'));
78         this.app.listen(config.server.port, () => {
79             this.log(`HTTP server running at http://localhost:\${config.server.port}`);
80         });
81     }
82
83     setupWebSocket() {
84         const wss = new WebSocket.Server({ port: config.server.wsPort });
85         wss.on('connection', (ws) => {
86             this.clients.add(ws);
87             ws.on('close', () => this.clients.delete(ws));
88         });
89     }
90
91     async connectMongoDB() {
92         try {
93             this.mongoClient = new MongoClient(config.mongoddb.uri);
94             await this.mongoClient.connect();
95             this.db = this.mongoClient.db(config.mongoddb.dbName);
96             this.collection = this.db.collection(config.mongoddb.collection);
97             this.log('Connected to MongoDB');
98         } catch (error) {
99             this.log('MongoDB connection error:', error.message);
100         }
101     }

```

(Εικόνα 4.35)

```

103     connectMQTT() {
104         const { protocol, host, port } = connectionOptions;
105         const mqttUrl = `${protocol}://${host}:${port}`;
106
107         this.mqttClient = mqtt.connect(mqttUrl, {
108             ...config.mqtt.options,
109             connectTimeout: 4000,
110             reconnectPeriod: 1000
111         });
112
113         this.mqttClient.on('connect', () => {
114             this.log('Connected to MQTT broker');
115             this.mqttClient.subscribe('esp8266/test');
116         });
117
118         this.mqttClient.on('reconnect', (error) => {
119             this.log(`Reconnecting to MQTT broker:`, error);
120         });
121
122         this.mqttClient.on('error', (error) => {
123             this.log(`MQTT connection error:`, error);
124         });
125
126         this.mqttClient.on('message', (topic, payload) => this.handleMQTTMessage(payload));
127     }

```

(Εικόνα 4.36)

```

129   async handleMQTTMessage(payload) {
130       try {
131           const data = JSON.parse(payload);
132           // Updated to use moisture_percentage from the ESP8266
133           const moisturePercentage = data.moisture_percentage;
134
135           if (!this.selectedPlant) {
136               this.log('No plant selected. Please select a plant first.');
```

```

137               return;
138           }
139
140           // Send percentage data to graph
141           this.broadcast({
142               type: 'moisture',
143               value: moisturePercentage,
144               timestamp: new Date().toISOString()
145           });
146
147           // Check moisture levels using percentage values
148           this.checkMoistureLevel(moisturePercentage, this.selectedPlant);
149           await this.saveMoistureReading(moisturePercentage, this.selectedPlant);
150
151           // Log the percentage value
152           this.log(`Soil Moisture: ${moisturePercentage}%`);
153       } catch (error) {
154           this.log('Error processing message:', error.message);
155       }
156   }
157
158   checkMoistureLevel(moisturePercentage, plant) {
159       // Convert threshold values to percentages if they aren't already
160       const { minimum, optimal, maximum } = plant.moistureThresholds;
161       const alerts = plantData.generalGuidelines.alertThreshold;
```

```

162
163       if (moisturePercentage < minimum) {
164           this.log(`Alert: Moisture ${moisturePercentage}% below minimum ${minimum}%. ${alerts.belowMinimum}`);
165       } else if (moisturePercentage > maximum) {
166           this.log(`Alert: Moisture ${moisturePercentage}% above maximum ${maximum}%. ${alerts.aboveMaximum}`);
167       } else {
168           this.log(`Moisture ${moisturePercentage}% is within acceptable range (optimal: ${optimal}%)`);
169       }
170   }

```

(Εικόνα 4.37)

```

172     async saveMoistureReading(moisturePercentage, plant) {
173         const reading = {
174             plantName: plant.name,
175             scientificName: plant.scientificName,
176             moisturePercentage,
177             timestamp: new Date()
178         };
179         await this.collection.insertOne(reading);
180         this.log('Reading saved to MongoDB');
181     }
182
183     log(...args) {
184         const message = args.join(' ');
185         console.log(message);
186         this.broadcast({ type: 'log', message });
187     }
188
189     broadcast(message) {
190         this.clients.forEach(client => {
191             if (client.readyState === WebSocket.OPEN) {
192                 client.send(JSON.stringify(message));
193             }
194         });
195     }
196 }
197
198 // Start monitoring
199 const monitor = new PlantMonitor();
200 monitor.start();

```

(Εικόνα 4.38)

Το τελικό αρχείο **package.json**:

```

1  {
2      "dependencies": {
3          "body-parser": "^1.20.3",
4          "cors": "^2.8.5",
5          "espress": "^0.0.0",
6          "express": "^4.21.2",
7          "fs": "^0.0.1-security",
8          "mongodb": "^6.12.0",
9          "mongoose": "^8.9.2",
10         "mqtt": "^5.10.3",
11         "readline": "^1.3.0",
12         "ws": "^8.18.0"
13     },
14     "name": "thesisproject",
15     "version": "1.0.0",
16     "main": "server.js",
17     "scripts": {
18         "test": "echo \"Error: no test specified\" && exit 1",
19         "start": "node server.js"
20     },
21     "keywords": [],
22     "author": "",
23     "license": "ISC",
24     "description": ""
25 }
26

```

(Εικόνα 4.39)

Η λίστα plants.json:

```
1 {
2   "plants": [
3     {
4       "name": "Snake Plant",
5       "scientificName": "Sansevieria trifasciata",
6       "moistureThresholds": {
7         "minimum": 20,
8         "optimal": 30,
9         "maximum": 50
10      },
11      "wateringInstructions": {
12        "frequency": "Every 2-3 weeks",
13        "notes": "Allow soil to dry between waterings"
14      }
15    },
16    {
17      "name": "Peace Lily",
18      "scientificName": "Spathiphyllum wallisii",
19      "moistureThresholds": {
20        "minimum": 40,
21        "optimal": 60,
22        "maximum": 80
23      },
24      "wateringInstructions": {
25        "frequency": "Weekly",
26        "notes": "Keep soil consistently moist"
27      }
28    },
29    {
30      "name": "Cactus",
31      "scientificName": "Cactaceae",
32      "moistureThresholds": {
33        "minimum": 10,
34        "optimal": 20,
35        "maximum": 30
36      },
37      "wateringInstructions": {
38        "frequency": "Every 3-4 weeks",
39        "notes": "Prefers dry conditions"
40      }
41    }
42  ]
43 }
```

(Εικόνα 4.40)

```

42   {
43     "name": "Monstera",
44     "scientificName": "Monstera deliciosa",
45     "moistureThresholds": {
46       "minimum": 30,
47       "optimal": 50,
48       "maximum": 70
49     },
50     "wateringInstructions": {
51       "frequency": "Every 1-2 weeks",
52       "notes": "Allow top inch of soil to dry between waterings"
53     }
54   },
55   {
56     "name": "Aloe Vera",
57     "scientificName": "Aloe barbadensis miller",
58     "moistureThresholds": {
59       "minimum": 10,
60       "optimal": 20,
61       "maximum": 30
62     },
63     "wateringInstructions": {
64       "frequency": "Every 3 weeks",
65       "notes": "Allow soil to dry out completely between waterings"
66     }
67   },
68   {
69     "name": "Fiddle Leaf Fig",
70     "scientificName": "Ficus lyrata",
71     "moistureThresholds": {
72       "minimum": 30,
73       "optimal": 40,
74       "maximum": 60
75     },
76     "wateringInstructions": {
77       "frequency": "Weekly",
78       "notes": "Keep soil evenly moist but not soggy"
79     }
80   },

```

(Εικόνα 4.41)

```

81  {
82      "name": "Spider Plant",
83      "scientificName": "Chlorophytum comosum",
84      "moistureThresholds": {
85          "minimum": 30,
86          "optimal": 40,
87          "maximum": 60
88      },
89      "wateringInstructions": {
90          "frequency": "Every 1-2 weeks",
91          "notes": "Allow soil to dry out slightly between waterings"
92      }
93  },
94  {
95      "name": "Boston Fern",
96      "scientificName": "Nephrolepis exaltata",
97      "moistureThresholds": {
98          "minimum": 50,
99          "optimal": 60,
100         "maximum": 80
101     },
102     "wateringInstructions": {
103         "frequency": "Weekly",
104         "notes": "Keep soil consistently moist, do not let dry out"
105     }
106 },
107 {
108     "name": "Jade Plant",
109     "scientificName": "Crassula ovata",
110     "moistureThresholds": {
111         "minimum": 10,
112         "optimal": 20,
113         "maximum": 30
114     },
115     "wateringInstructions": {
116         "frequency": "Every 2-3 weeks",
117         "notes": "Allow soil to dry between waterings"
118     }
119 },

```

(Εικόνα 4.42)

```

120  {
121      "name": "Rubber Plant",
122      "scientificName": "Ficus elastica",
123      "moistureThresholds": {
124          "minimum": 20,
125          "optimal": 40,
126          "maximum": 60
127      },
128      "wateringInstructions": {
129          "frequency": "Every 1-2 weeks",
130          "notes": "Allow top inch of soil to dry between waterings"
131      }
132  }
133 },
134 "generalGuidelines": {
135     "readingFrequency": "Every 6 hours",
136     "calibrationNotes": "Moisture values are in percentage (0-100%)",
137     "alertThreshold": {
138         "belowMinimum": "Immediate watering required",
139         "aboveMaximum": "Potential overwatering, check drainage"
140     }
141 }
142 }

```

(Εικόνα 4.43)

Τα αποτελέσματα του τερματικού. Το φυτό που χρησιμοποιείται ως παράδειγμα είναι ο κάκτος:

i) Επιλογή του κάκτου:

```
Select a plant to monitor (enter number):
PS D:\thesisproject\back-end> node server.js
Available plants:
1. Snake Plant
2. Peace Lily
3. Cactus
4. Monstera
5. Aloe Vera
6. Fiddle Leaf Fig
7. Spider Plant
8. Boston Fern
9. Jade Plant
10. Rubber Plant
Select a plant to monitor (enter number): HTTP server running at http://localhost:3000
3
Selected plant: Cactus
Connected to MongoDB
Connected to MQTT broker
```

(Εικόνα 4.44)

ii) Μηνύματα ανάλογα με τα επίπεδα υγρασίας.:

```
Soil Moisture: 51%
Moisture 28% is within acceptable range (optimal: 20%)
Reading saved to MongoDB
Soil Moisture: 28%
Moisture 28% is within acceptable range (optimal: 20%)
Reading saved to MongoDB
Soil Moisture: 28%
Moisture 29% is within acceptable range (optimal: 20%)
Reading saved to MongoDB
Soil Moisture: 29%
Alert: Moisture 34% above maximum 30%. Potential overwatering, check drainage
Reading saved to MongoDB
Soil Moisture: 34%
Alert: Moisture 40% above maximum 30%. Potential overwatering, check drainage
Reading saved to MongoDB
Soil Moisture: 40%
```

(Εικόνα 4.45)

iii) Προβολή των αποτελεσμάτων στη βάση δεδομένων

plantMonitor.soilMoistureReadings

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 1.94KB TOTAL DOCUMENTS: 3

Find Indexes Schema Anti-Patterns ⓘ

Generate queries from natural language in Compass

Filter Type a query: { field: 'value' }

```
{
  "_id": "ObjectId('67a10f122b435132e35a9c0e')",
  "plantName": "Cactus",
  "scientificName": "Cactaceae",
  "moisturePercentage": 51,
  "timestamp": "2025-02-03T18:46:42.332+00:00"
}
```

```
{
  "_id": "ObjectId('67a10f172b435132e35a9c0f')",
  "plantName": "Cactus",
  "scientificName": "Cactaceae",
  "moisturePercentage": 28,
  "timestamp": "2025-02-03T18:46:47.329+00:00"
}
```

```
{
  "_id": "ObjectId('67a10f1c2b435132e35a9c10')",
  "plantName": "Cactus",
  "scientificName": "Cactaceae",
  "moisturePercentage": 28,
  "timestamp": "2025-02-03T18:46:52.331+00:00"
}
```

(Εικόνα 4.46)

plantMonitor.soilMoistureReadings

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 1.94KB TOTAL DOCUMENTS: 3

Find Indexes Schema Anti-Patterns ⓘ

Generate queries from natural language in Compass

Filter Type a query: { field: 'value' }

```
{
  "_id": "ObjectId('67a10f212b435132e35a9c11')",
  "plantName": "Cactus",
  "scientificName": "Cactaceae",
  "moisturePercentage": 29,
  "timestamp": "2025-02-03T18:46:57.329+00:00"
}
```

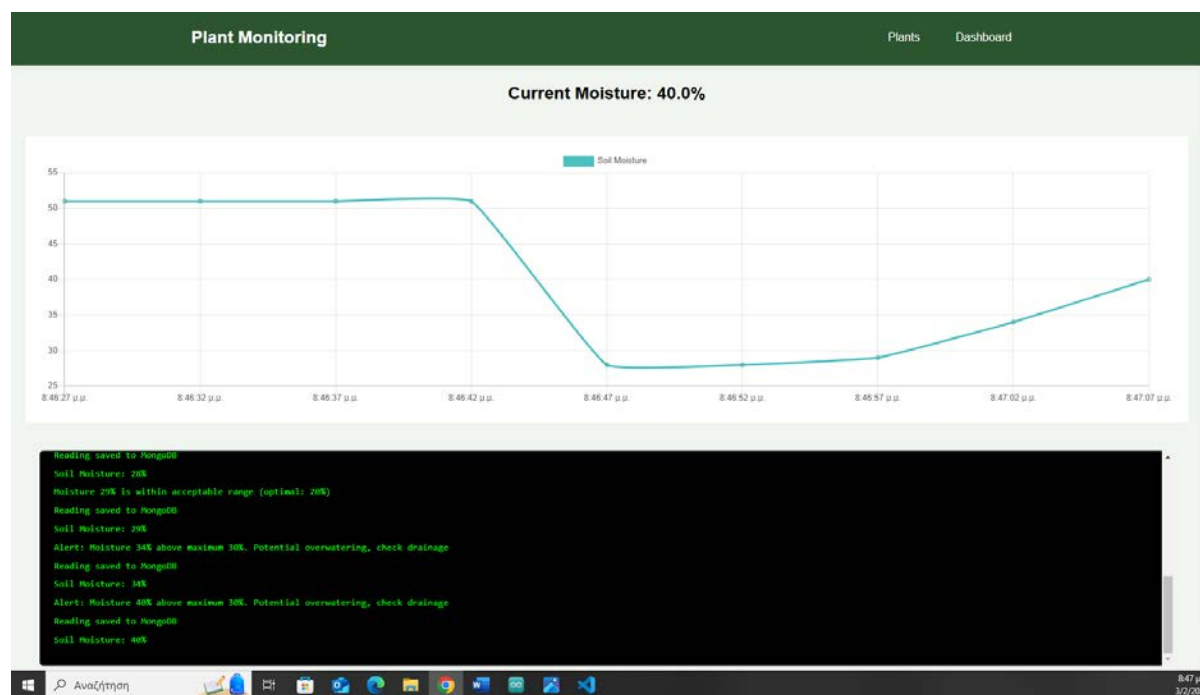
```
{
  "_id": "ObjectId('67a10f262b435132e35a9c12')",
  "plantName": "Cactus",
  "scientificName": "Cactaceae",
  "moisturePercentage": 34,
  "timestamp": "2025-02-03T18:47:02.333+00:00"
}
```

```
{
  "_id": "ObjectId('67a10f2b2b435132e35a9c13')",
  "plantName": "Cactus",
  "scientificName": "Cactaceae",
  "moisturePercentage": 40,
  "timestamp": "2025-02-03T18:47:07.341+00:00"
}
```

(Εικόνα 4.47)

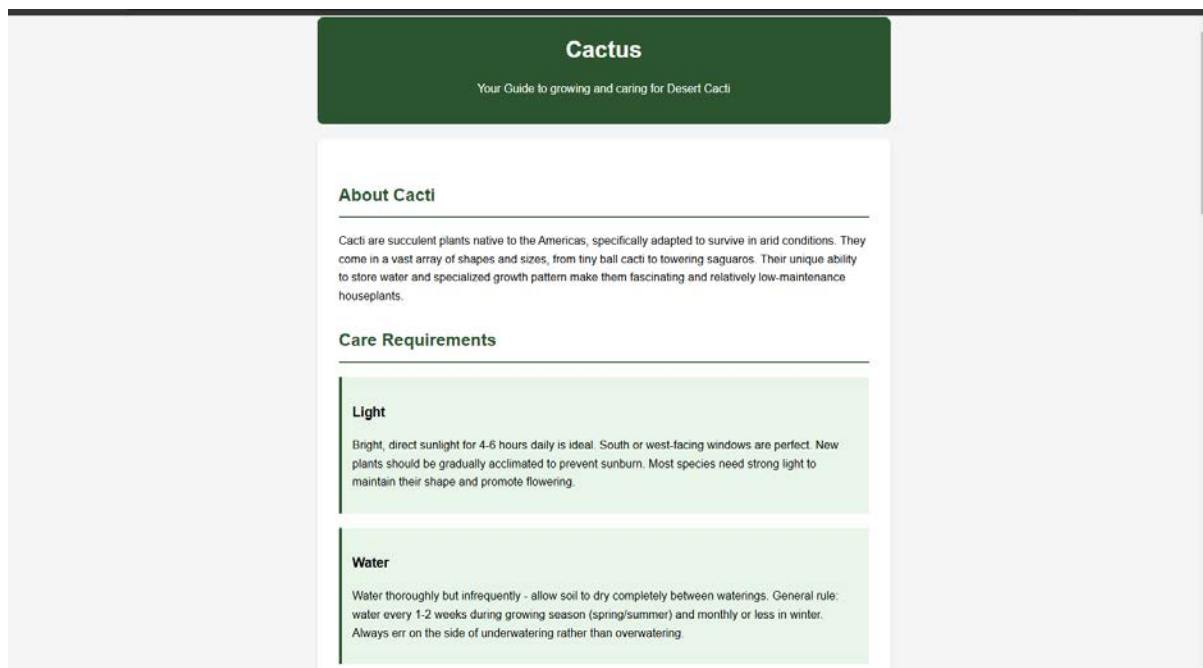
4.4 Υλοποίηση της ιστοσελίδας

Στο ταμπλό της ιστοσελίδας, το τερματικό αποτυπώνει τις ενδείξεις για τα ποσοστά υγρασίας του κάκτου και οι αυξομειώσεις στις τιμές αποτυπώνονται στο γράφημα. **(Εικόνα 4.48)**

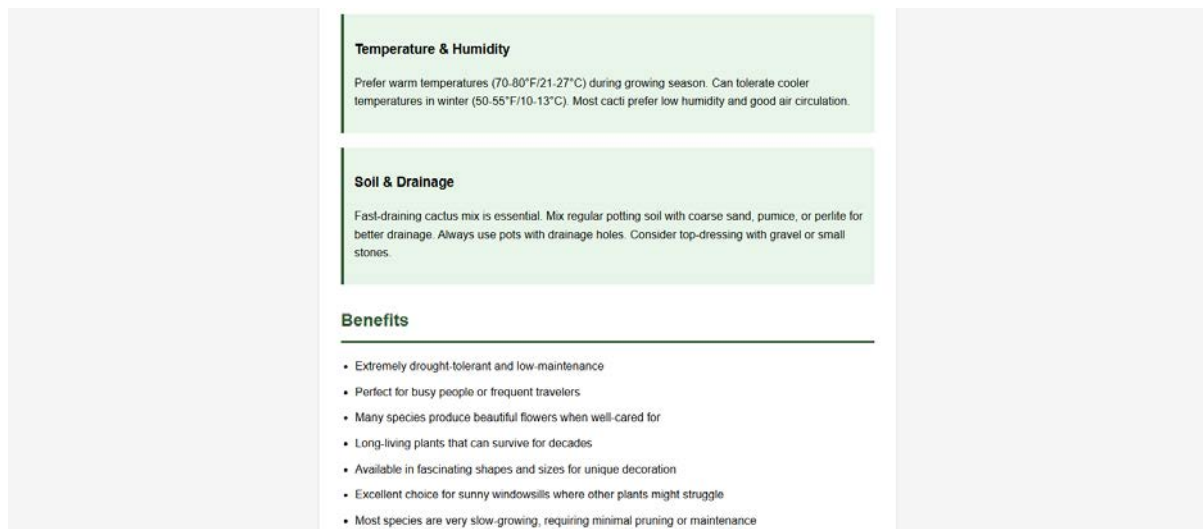


(Εικόνα 4.48)

Το δεύτερο κομμάτι της ιστοσελίδας είναι το μενού που περιλαμβάνει τις πληροφορίες για τα φυτά. Οι εικόνες **(4.49)** έως **(4.51)** περιλαμβάνουν την σελίδα για τις πληροφορίες και ανάγκες ενός κάκτου.



(Εικόνα 4.49)



(Εικόνα 4.50)

Common Problems

- Rotting base or roots - Almost always indicates overwatering
- Winkled or shriveled appearance - Signs of underwatering
- Pale, yellow, or bleached appearance - Too much direct sun without acclimation
- Elongated, stretched growth - Insufficient light (etiolation)
- White, cottony patches - Mealybugs infestation
- Corking (brown, bark-like tissue at base) - Natural aging process, not harmful

Seasonal Care

Growing Season (Spring/Summer)

- Rotate the plant regularly to ensure even growth
- Remove any dead or yellowing fronds at the base to maintain appearance
- Clean fronds occasionally with a damp cloth to remove dust
- Repot every 2-3 years or when roots begin growing through drainage holes
- Consider moving outdoors to a shaded location during warm months
- Divide large plants in spring to create new specimens

Dormant Season (Fall/Winter)

- Reduce watering significantly
- Stop fertilizing
- Maintain cooler temperatures
- Continue providing bright light

(Εικόνα 4.51)

Ο κώδικας **plantMonitor.html** που δείχνει τη κεντρική σελίδα:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Plant Monitoring</title>
7   <link rel="stylesheet" href="./plantMonitoring.css">
8   <script src="https://cdn.jsdelivr.net/npm/chart.js@2.9.4/dist/chart.js"></script>
9 </head>
10 <body>
11   <header>
12     <div class="container">
13       <h1>Plant Monitoring</h1>
14       <nav>
15         <ul>
16           <li class="dropdown">
17             <a href="#">Plants</a>
18             <ul class="dropdown-menu">
19               <li><a href="/plants/aloevera.html">Aloe Vera</a></li>
20               <li><a href="/plants/bostonfern.html">Boston Fern</a></li>
21               <li><a href="/plants/cactus.html">Cactus</a></li>
22               <li><a href="/plants/fiddleleaffig.html">Fiddle Leaf Fig</a></li>
23               <li><a href="/plants/jadeplant.html">Jade Plant</a></li>
24               <li><a href="/plants/monstera.html">Monstera</a></li>
25               <li><a href="/plants/peacylily.html">Peace Lily</a></li>
26               <li><a href="/plants/rubberplant.html">Rubber Plant</a></li>
27               <li><a href="/plants/snakeplant.html">Snake Plant</a></li>
28               <li><a href="/plants/spiderplant.html">Spider Plant</a></li>
29             </ul>
30           </li>
31           <li><a href="#">Dashboard</a></li>
32         </ul>
33       </nav>
34     </div>
35   </header>
36
37   <div class="dashboard">
38     <div class="current-moisture">
39       Current Moisture: <span id="moistureValue">--</span>%
40     </div>
41     <div class="chart-container">
42       <canvas id="moistureChart"></canvas>
43     </div>
44     <div id="terminal">
45
46     </div>
47   </div>
48   <script src="/plantMonitoring.js"></script>
49 </body>
```

(Εικόνα 4.52)

Κώδικας plantMonitor.css:

```
ront-End > # plantMonitoring.css > .dashboard
1  /* Basic Reset */
2  * {
3      margin: 0;
4      padding: 0;
5      box-sizing: border-box;
6  }
7
8  /* Body Styles */
9  body {
10     font-family: Arial, sans-serif;
11     line-height: 1.6;
12     background-color: #f0f5f0;
13 }
14
15 /* Header Styles */
16 header {
17     background-color: #2c5530;
18     padding: 1rem 0;
19     box-shadow: 0 2px 5px rgba(0,0,0,0.1);
20 }
21
22 .container {
23     max-width: 1200px;
24     margin: 0 auto;
25     padding: 0 20px;
26     display: flex;
27     justify-content: space-between;
28     align-items: center;
29 }
30
31 h1 {
32     color: white;
33     font-size: 24px;
34 }
35
36 /* Navigation Styles */
37 nav ul {
38     list-style: none;
39     display: flex;
40 }
41
42 nav ul li {
43     position: relative;
44     margin-left: 20px;
45 }
```

(Εικόνα 4.53)

```

47 nav ul li a {
48     color: white;
49     text-decoration: none;
50     padding: 10px 15px;
51     display: block;
52     transition: all 0.3s ease;
53 }
54
55 nav ul li a:hover {
56     background-color: #3a6b3f;
57     border-radius: 4px;
58 }
59
60 /* Dropdown Styles */
61 .dropdown-menu {
62     display: none;
63     position: absolute;
64     top: 100%;
65     left: 0;
66     background-color: white;
67     border-radius: 4px;
68     box-shadow: 0 2px 5px rgba(0,0,0,0.2);
69     width: 200px;
70     z-index: 1000;
71 }
72
73 .dropdown:hover .dropdown-menu {
74     display: block;
75 }
76
77 .dropdown-menu li {
78     margin: 0;
79 }
80
81 .dropdown-menu li a {
82     color: #2c5530;
83     padding: 12px 20px;
84     border-bottom: 1px solid #eee;
85 }
86
87 .dropdown-menu li:last-child a {
88     border-bottom: none;
89 }
90
91 .dropdown-menu li a:hover {
92     background-color: #e8f5e9;
93     color: #2c5530;
94 }

```

(Εικόνα 4.54)

```

97  @media (max-width: 768px) {
98      .container {
99          flex-direction: column;
100         text-align: center;
101     }
102
103     nav {
104         margin-top: 1rem;
105     }
106
107     nav ul {
108         flex-direction: column;
109     }
110
111     nav ul li {
112         margin: 10px 0;
113     }
114
115     .dropdown-menu {
116         position: static;
117         width: 100%;
118         box-shadow: none;
119         margin-top: 10px;
120     }
121
122     .dropdown:hover .dropdown-menu {
123         display: none;
124     }
125
126     .dropdown.active .dropdown-menu {
127         display: block;
128     }
129 }
130
131 #terminal {
132     background-color: #000;
133     color: #00ff00;
134     font-family: monospace;
135     padding: 20px;
136     border-radius: 5px;
137     margin: 20px;
138     height: 300px;
139     overflow-y: auto;
140 }
141 .log-entry {
142     margin: 5px 0;
143     white-space: pre-wrap;
144 }

```

(Εικόνα 4.55)

```

145 .dashboard {
146     display: flex;
147     flex-direction: column;
148     gap: 20px;
149     padding: 20px;
150 }
151 .chart-container {
152     background-color: white;
153     padding: 20px;
154     border-radius: 5px;
155     height: 400px;
156 }
157 .current-moisture {
158     font-size: 24px;
159     font-weight: bold;
160     text-align: center;
161     margin-bottom: 20px;
162 }

```

(Εικόνα 4.56)

Ο κώδικας **plantMonitor.js** όπου με τη βοήθεια του πρωτοκόλλου WebSocket γίνεται η επικοινωνία δυο κατευθύνσεων μεταξύ back-end και front-end.

```
1 document.querySelectorAll('.dropdown').forEach(dropdown => {
2     dropdown.addEventListener('click', (e) => {
3         if (window.innerWidth <= 768) {
4             e.preventDefault();
5             dropdown.classList.toggle('active')
6         }
7     });
8 });
```

(Εικόνα 4.57)

```
10 const terminal = document.getElementById('terminal');
11 const moistureValue = document.getElementById('moistureValue');
12 const ctx = document.getElementById('moistureChart').getContext('2d');
13
14 // Initialize the chart
15 const chart = new Chart(ctx, {
16     type: 'line',
17     data: {
18         labels: [],
19         datasets: [{
20             label: 'Soil Moisture ',
21             data: [],
22             borderColor: 'rgb(75, 192, 192)',
23             tension: 0.1,
24             fill: false
25         }]
26     },
27     // Chart Configuration
28     options: {
29         responsive: true,
30         maintainAspectRatio: false,
31         scales: {
32             y: {
33                 beginAtZero: true,
34                 max: 100,
35                 title: {
36                     display: true,
37                     text: 'Moisture Level (%)'
38                 }
39             },
40             x: {
41                 title: {
42                     display: true,
43                     text: 'Time'
44                 }
45             }
46         },
47         animation: {
48             duration: 0
49         }
50     }
51 });
52
53 // Maximum number of points to show on the graph
54 const MAX_POINTS = 20;
55
56 // Connect to WebSocket
57 const ws = new WebSocket('ws://localhost:8080');
```

(Εικόνα 4.58)

```

59     ws.onmessage = (event) => {
60         const data = JSON.parse(event.data);
61         if (data.type === 'moisture') {
62             // Direct access to the moisture value
63             updateChart(data.value);
64         } else if (data.type === 'log') {
65             // Add log entry to terminal
66             const logEntry = document.createElement('div');
67             logEntry.className = 'log-entry';
68             logEntry.textContent = data.message;
69             terminal.appendChild(logEntry);
70             terminal.scrollTop = terminal.scrollHeight;
71         }
72     };
73
74     function updateChart(moisture) {
75         // Update current moisture value
76         moistureValue.textContent = moisture.toFixed(1);
77
78         // Add new data point
79         const now = new Date();
80         const timeString = now.toLocaleTimeString();
81
82         chart.data.labels.push(timeString);
83         chart.data.datasets[0].data.push(moisture);
84
85         // Remove old data points if we exceed MAX_POINTS
86         if (chart.data.labels.length > MAX_POINTS) {
87             chart.data.labels.shift();
88             chart.data.datasets[0].data.shift();
89         }
90
91         // Update the chart
92         chart.update();
93     }
94
95     ws.onclose = () => {
96         const logEntry = document.createElement('div');
97         logEntry.className = 'log-entry';
98         logEntry.style.color = '#ff0000';
99         logEntry.textContent = 'Connection to server lost...';
100         terminal.appendChild(logEntry);
101     };

```

(Εικόνα 4.59)

Τέλος ο κώδικας **cactus.html** για τη σελίδα του κάκτου:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Cactus</title>
7   <link rel="stylesheet" href="../plants.css">
8 </head>
9 <body>
10  <header>
11    <h1>Cactus</h1>
12    <p>Your Guide to growing and caring for Desert Cacti</p>
13  </header>
14
15  <div class="container">
16    <section>
17      <h2>About Cacti</h2>
18      <p>Cacti are succulent plants native to the Americas,
19        specifically adapted to survive in arid conditions.
20        They come in a vast array of shapes and sizes, from tiny ball cacti to towering saguaros.
21        Their unique ability to store water and specialized growth pattern make them fascinating and relatively low-maintenance houseplants.</p>
22    </section>
23
24    <section>
25      <h2>Care Requirements</h2>
26      <div class="care-tip">
27        <h3>Light</h3>
28        <p>Bright, direct sunlight for 4-6 hours daily is ideal.
29          South or west-facing windows are perfect.
30          New plants should be gradually acclimated to prevent sunburn. Most species need strong light to maintain their shape and promote flowering.</p>
31      </div>
32
33      <div class="care-tip">
34        <h3>Water</h3>
35        <p>Water thoroughly but infrequently - allow soil to dry completely between waterings.
36          General rule: water every 1-2 weeks during growing season (spring/summer) and monthly or less in winter.
37          Better be on the side of underwatering rather than overwatering.</p>
38      </div>
39
40      <div class="care-tip">
41        <h3>Temperature & Humidity</h3>
42        <p>Prefer warm temperatures (70-80°F/21-27°C) during growing season.
43          Can tolerate cooler temperatures in winter (50-55°F/10-13°C).
44          Most cacti prefer low humidity and good air circulation.</p>
45      </div>
```

(Εικόνα 4.60)

```
47      <div class="care-tip">
48        <h3>Soil & Drainage</h3>
49        <p>Fast-draining cactus mix is essential.
50          Mix regular potting soil with coarse sand, pumice, or perlite for better drainage.
51          Always use pots with drainage holes.
52          Consider top-dressing with gravel or small stones.</p>
53      </div>
54    </section>
55
56    <section>
57      <h2>Benefits</h2>
58      <ul>
59        <li>Extremely drought-tolerant and low-maintenance</li>
60        <li>Perfect for busy people or frequent travelers</li>
61        <li>Many species produce beautiful flowers when well-cared for</li>
62        <li>Long-living plants that can survive for decades</li>
63        <li>Available in fascinating shapes and sizes for unique decoration</li>
64        <li>Excellent choice for sunny windowsills where other plants might struggle</li>
65        <li>Most species are very slow-growing, requiring minimal pruning or maintenance</li>
66      </ul>
67    </section>
68
69    <section>
70      <h2>Common Problems</h2>
71      <ul>
72        <li>Rotting base or roots - Almost always indicates overwatering</li>
73        <li>Wrinkled or shriveled appearance - Signs of underwatering</li>
74        <li>Pale, yellow, or bleached appearance - Too much direct sun without acclimation</li>
75        <li>Elongated, stretched growth - Insufficient light (etiolation)</li>
76        <li>White, cottony patches - Mealybugs infestation</li>
77        <li>Corking (brown, bark-like tissue at base) - Natural aging process, not harmful</li>
78      </ul>
79    </section>
```

(Εικόνα 4.61)

```

81     <section>
82         <h2>Seasonal Care</h2>
83         <div class="care-tip">
84             <p><b>Growing Season (Spring/Summer)</b></p>
85             <ul>
86                 <li>Rotate the plant regularly to ensure even growth</li>
87                 <li>Remove any dead or yellowing fronds at the base to maintain appearance</li>
88                 <li>Clean fronds occasionally with a damp cloth to remove dust</li>
89                 <li>Repot every 2-3 years or when roots begin growing through drainage holes</li>
90                 <li>Consider moving outdoors to a shaded location during warm months</li>
91                 <li>Divide large plants in spring to create new specimens</li>
92             </ul>
93         </div>
94         <p><b>Dormant Season (Fall/Winter)</b></p>
95         <ul>
96             <li>Reduce watering significantly</li>
97             <li>Stop fertilizing</li>
98             <li>Maintain cooler temperatures</li>
99             <li>Continue providing bright light</li>
100         </ul>
101     </div>
102 </div>
103 </section>
104
105 <section>
106     <h2>Safety Note</h2>
107     <div class="care-tip">
108         <p>Always handle cacti with care - use thick gloves or folded newspaper when repotting or handling.
109             Some species have very fine glochids (tiny barbed spines) that can be difficult to remove from skin.
110             Keep out of reach of children and pets.</p>
111     </div>
112 </section>
113
114 <section>
115     <h2>Propagation Tips</h2>
116     <div class="care-tip">
117         <p>Cacti can be propagated though:</p>
118         <li>Offsets or "pups" - carefully remove and allow cut to callus before planting</li>
119         <li>Stem cuttings - cut a segment, let it callus, then plant in dry soil</li>
120         <li>Seeds - requires patience and specific conditions for germination</li>
121     </div>
122 </section>
123 </body>
124 </html>

```

(Εικόνα 4.62)

Ο κώδικας **plants.css**:

```
1  body {
2      font-family: Arial, sans-serif;
3      line-height: 1.6;
4      max-width: 800px;
5      margin: 0 auto;
6      padding: 20px;
7      background-color: #f5f5f5;
8  }
9
10 header {
11     text-align: center;
12     padding: 20px;
13     background-color: #2c5530;
14     color: white;
15     border-radius: 8px;
16     margin-bottom: 20px;
17 }
18
19 .container {
20     background-color: white;
21     padding: 30px;
22     border-radius: 8px;
23     box-shadow: 0 2px 4px rgba(0,0,0,0.1);
24 }
25
26 h1 {
27     margin: 0;
28 }
```

(Εικόνα 4.63)

```

30  h2 {
31      color: #2c5530;
32      border-bottom: 2px solid #2c5530;
33      padding-bottom: 10px;
34      margin-top: 30px;
35  }
36
37  .care-tip {
38      background-color: #e8f5e9;
39      padding: 15px;
40      border-left: 4px solid #2c5530;
41      margin: 20px 0;
42  }
43
44  ul {
45      padding-left: 20px;
46  }
47
48  li {
49      margin-bottom: 10px;
50  }

```

(Εικόνα 4.64)

ΚΕΦΑΛΑΙΟ 5 Επίλογος

5.1 Συμπεράσματα

Κλείνοντας, γίνεται αντιληπτό πως η τεχνολογία δεν θα μπορούσε να κάνει το επόμενο βήμα μπροστά, χωρίς την χρήση των τεχνολογιών IoT. Η εφαρμογή αυτή δεν είναι παρά ένα μικρό κομμάτι των απεριόριστων δυνατοτήτων αυτών των τεχνολογιών, οι οποίες μπορούν να κάνουν τη καθημερινότητά μας πιο απλή και εύκολη.

5.2 Βελτιστοποιήσεις

Η εφαρμογή αυτή, όπως έχει υποθεί, έχει ως στόχο την επίβλεψη οικιακών φυτών. Σε μεγαλύτερη κλίμακα όμως μπορεί να χρησιμοποιηθεί για τη επίβλεψη εκτάσεων γης για καλλιεργήσιμα φυτά. Έχοντας αυτό το σκοπό μερικά πράγματα που θα μπορούσαν να βελτιωθούν, είναι τα εξής.

- 1) Δημιουργία αλγορίθμου πρόβλεψης μελλοντικής υγρασίας, ειδικά με τις μεταβολές του καιρού.
- 2) Επικοινωνία με άλλες συσκευές, όπως «έξυπνα» ποτιστήρια για αυτόματο πότισμα
- 3) Από όλα αυτά δεν θα μπορούσε να λείπει, η μετατροπή της εφαρμογής από τη στατική της μορφή, σε κινητή για την πρόσβαση σε πληροφορίες και λήψη αποφάσεων οπουδήποτε.

BIBΛΙΟΓΡΑΦΙΑ

- [1] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A literature review," *Journal of Computer and Communications*, vol. 3, no. 5, pp. 164–173, 2015.
- [2] S. Razdan and S. Sharma, "Internet of Medical Things (IoMT): Overview, Emerging Technologies, and Case Studies," 2022. doi: 10.1080/02564602.2021.1927863.
- [3] Z. Špitálová, "VEHICLE-TO-EVERYTHING COMMUNICATION," 2023. doi: 10.26552/com.C.2023.017.
- [4] S. Zeadally, J. Guerrero, and J. Contreras, "A tutorial survey on vehicle-to-vehicle communications," *Telecommun Syst*, vol. 73, no. 3, 2020, doi: 10.1007/s11235-019-00639-8.
- [5] D. Kanthavel, S. K. B. Sangeetha, and K. P. Keerthana, "An empirical study of vehicle to infrastructure communications - An intense learning of smart infrastructure for safety and mobility," *International Journal of Intelligent Networks*, vol. 2, 2021, doi: 10.1016/j.ijin.2021.06.003.
- [6] A. Rouchitsas and H. Alm, "External Human–Machine Interfaces for Autonomous Vehicle-to-Pedestrian Communication: A Review of Empirical Work," 2019. doi: 10.3389/fpsyg.2019.02757.
- [7] I. A. Umoren, M. Z. Shakir, and H. Tabassum, "Resource Efficient Vehicle-to-Grid (V2G) Communication Systems for Electric Vehicle Enabled Microgrids," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, 2021, doi: 10.1109/TITS.2020.3023899.
- [8] S. F. Ahmed *et al.*, "Industrial Internet of Things enabled technologies, challenges, and future directions," *Computers and Electrical Engineering*, vol. 110, 2023, doi: 10.1016/j.compeleceng.2023.108847.
- [9] J. Fan, Y. Zhang, W. Wen, S. Gu, X. Lu, and X. Guo, "The future of Internet of Things in agriculture: Plant high-throughput phenotypic platform," *J Clean Prod*, vol. 280, 2021, doi: 10.1016/j.jclepro.2020.123651.
- [10] H. Talal and R. Zagrouba, "MADS based on DL techniques on the Internet of Things (IoT): Survey," *Electronics (Basel)*, vol. 10, no. 21, p. 2598, 2021.
- [11] M. Che and M. F. Tuo, "Server Program Analysis Based on HTTP Protocol," *MATEC Web of Conferences*, vol. 63, 2016, doi: 10.1051/mateconf/20166305023.
- [12] A. Gerodimos, L. Maglaras, M. A. Ferrag, N. Ayres, and I. Kantzavelou, "IoT: Communication protocols and security threats," 2023. doi: 10.1016/j.iotcps.2022.12.003.
- [13] W. Zegeye, A. Jemal, and K. Kornegay, "Connected Smart Home over Matter Protocol," in *Digest of Technical Papers - IEEE International Conference on Consumer Electronics*, 2023. doi: 10.1109/ICCE56470.2023.10043520.
- [14] X. Liu, B. Cui, J. Fu, and J. Ma, "HFuzz: Towards automatic fuzzing testing of NB-IoT core network protocols implementations," *Future Generation Computer Systems*, vol. 108, 2020, doi: 10.1016/j.future.2019.12.032.
- [15] D. Soni and A. Makwana, "A survey on mqtt: a protocol of internet of things (iot)," in *International conference on telecommunication, power analysis and computing techniques (ICTPACT-2017)*, 2017, pp. 173–177.
- [16] I. Fette, Inc. Google, A. Melnikov, and I. Ltd., "RFC6455 - The WebSocket Protocol," *J Chem Inf Model*, 2011.
- [21] Arduino IDE Documentation: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2/>
- [22] <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>
- [23] MongoDB: <https://www.mongodb.com/resources/products/fundamentals/basics>
- [24] MongoDB Atlas: <https://www.mongodb.com/docs/atlas/>
- [25] ESP8266 WiFi library: <https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html>

- [26] PubSubClient library: <https://www.hivemq.com/blog/mqtt-client-library-encyclopedia-arduino-pubsubclient/>
- [27] ArduinoJSON library: <https://github.com/bblanchon/ArduinoJson>