



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΣΥΣΤΗΜΑ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΕΠΙΠΕΔΟΥ
ΑΙΩΡΟΥΜΕΝΩΝ ΣΩΜΑΤΙΔΙΩΝ $PM_{2.5}$ ΣΤΗΝ ΠΟΛΗ
ΤΟΥ ΒΟΛΟΥ

υπό

ΓΕΩΡΓΙΟΥ Ι. ΚΑΛΑΝΤΖΗ

Διπλωματική Εργασία

Υπεβλήθη για την εκπλήρωση των απαιτήσεων για την απόκτηση του
Διπλώματος Μηχανολόγου Μηχανικού

Βόλος 2021

Εγκρίθηκε από τα Μέλη της Τριμελούς Εξεταστικής Επιτροπής:

Πρώτος Εξεταστής Δρ. Γεώργιος Κ.Δ. Σαχαρίδης
(Επιβλέπων) Αναπληρωτής Καθηγητής, Τμήμα Μηχανολόγων
Μηχανικών, Πανεπιστήμιο Θεσσαλίας

Δεύτερος Εξεταστής Δρ. Δημήτριος Παντελής
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Τρίτος Εξεταστής Δρ. Νικόλαος Ανδρίτσος
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,
Πανεπιστήμιο Θεσσαλίας

Αφιερωμένη στους γονείς μου.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή κύριο Σαχαρίδη Κ.Δ. Γεώργιο για την σημαντική υποστήριξη του, την καθοδήγηση και τις παραγωγικές υποδείξεις του στην παρούσα εργασία. Επίσης θα ήθελα να ευχαριστήσω τον Καθηγητή Ανδρίτσο Νικόλαο για τις σημαντικές γνώσεις που μου έδωσε καθ' όλη την διάρκεια της εργασίας. Παράλληλα να ευχαριστήσω το Τμήμα Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας για την παραχώρηση των χώρων του τόσο για τις έρευνες όσο και για την παρουσίαση της εργασίας. Επιπλέον θα ήθελα να ευχαριστήσω τον Ιωάννη Κουτσαυτική για τις πολύτιμες οδηγίες και συμβουλές που μου έδωσε κατά την κατασκευή της ιστοσελίδας, καθώς και τον Κυριάκο Ζήσιο. Θα ήθελα να ευχαριστήσω τον Διευθυντή της Πνευμονολογικής Κλινικής του Πανεπιστημιακού Νοσοκομείου Λάρισας κ. Γουργουλιάνη Κωνσταντίνο, τον διευθυντή των Τμήματος Επειγόντων Περιστατικών του Νοσοκομείου Βόλου κ. Καραμέτο Ηλία, τον γιατρό κ. Κανελλόπουλο Νικόλαο και όλους τους γιατρούς που έλαβαν μέρος στην καταχώρηση των δεδομένων στην εφαρμογή για την εξαιρετική συνεργασία. Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για την συνεχή συμπαράσταση και την στήριξη σε όλη μου την πορεία μέχρι σήμερα.

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας εργασίας με τίτλο “GreenYourAir” είναι η δημιουργία ενός δικτύου μέτρησης της ατμοσφαιρικής ρύπανσης από 12 συσκευές μέτρησης (GreenYourAir/1178) στην πόλη του Βόλου και συγκεκριμένα του ρύπου $PM_{2.5}$, η δημιουργία μίας εφαρμογής καταγραφής περιστατικών του Νοσοκομείου του Βόλου που συνδέονται άμεσα με την ατμοσφαιρική ρύπανση, η δημιουργία της ιστοσελίδας <http://greenyourair.org/> στην οποία οι πολίτες μπορούν να ενημερώνονται σε πραγματικό χρόνο για τις τιμές των $PM_{2.5}$ στην πόλη του Βόλου και η ανάλυση των μετρήσεων που πραγματοποιήθηκαν από 01/03/2019 έως και 29/02/2020. Η παρούσα εργασία υλοποιήθηκε στο χρονικό διάστημα από τον Οκτώβριο του 2018 έως τον Οκτώβριο του 2020. Επίσης αναπτύχθηκε η συσκευή μέτρησης $PM_{2.5}$ GreenYourAir/1497. Τα κύρια μέρη αυτής της εργασίας τα οποία αναλύονται είναι η επιλογή του hardware της συσκευής μέτρησης, ο προγραμματισμός της, η σύνδεση της στο διαδίκτυο για την μεταφορά δεδομένων σε πραγματικό χρόνο και η δημιουργία μιας online βάσης δεδομένων για την αποθήκευση των μετρήσεων. Η εγκατάσταση των συσκευών GreenYourAir/1178 σε επλεγμένα μέρη στα οποία θα λειτουργούν 24 ώρες τη ημέρα. Το επόμενο μέρος της εργασίας αφορά την δημιουργία της εφαρμογής στην οποία θα καταγράφονται τα περιστατικά του Νοσοκομείου του Βόλου για το χρονικό διάστημα από 01/03/2019 έως 29/02/2020. Η δημιουργία της ιστοσελίδας στην οποία μπορούν να ενημερώνονται οι πολίτες για τις τιμές των $PM_{2.5}$ της πόλης του Βόλου σε πραγματικό χρόνο και τέλος η ανάλυση και η εξαγωγή δεδομένων για την συγκέντρωση των $PM_{2.5}$ για το χρονικό διάστημα από 01/03/2019 έως 29/02/2020 στην πόλη του Βόλου.

Στην συγκεκριμένη εργασία αναφέρονται στοιχεία τα οποία αφορούν γενικότερα την ατμοσφαιρική ρύπανση όπως και οι οδηγίες του Παγκόσμιου Οργανισμού Υγείας, του Ευρωπαϊκού Οργανισμού Περιβάλλοντος και του Υπουργείου Περιβάλλοντος και Ενέργειας.

Για να υποστηριχθούν και να υλοποιηθούν όλες οι απαιτήσεις της εργασίας χρησιμοποιήθηκαν εργαλεία όπως το Google Forms, Arduino IDE, Google Apps Script και άλλα. Επίσης χρησιμοποιήθηκαν διαφορετικές γλώσσες προγραμματισμού όπως η C++, η Java-Script, Visual Basic και άλλες.

Τέλος η εργασία υλοποιήθηκε λαμβάνοντας υπόψιν όλες τις συνθήκες που επικρατούν στην πόλη του Βόλου αλλά και στις απαιτήσεις μίας εργασίας τέτοιου μεγέθους. Αντιμετωπίστηκε με σοβαρότητα και η μελέτη των δεδομένων έγινε με μεγάλη προσοχή ώστε να εξαχθούν ασφαλή συμπεράσματα για την συγκέντρωση των $PM_{2.5}$ στην πόλη του Βόλου.

Περιεχόμενα

| | |
|--|----|
| Ευχαριστίες | 4 |
| ΠΕΡΙΛΗΨΗ | 5 |
| Κατάλογος Εικόνων | 9 |
| Κατάλογος Πινάκων | 10 |
| Κατάλογος για τους Κώδικες | 11 |
| Κατάλογος Διαγραμμάτων | 12 |
| 1 ΕΙΣΑΓΩΓΗ | 13 |
| 2 ΑΤΜΟΣΦΑΙΡΙΚΟΙ ΡΥΠΟΙ ΚΑΙ ΟΡΓΑΝΙΣΜΟΙ ΠΡΟΣΤΑΣΙΑΣ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ | 15 |
| 2.1 Κύριοι Ατμοσφαιρικοί Ρύποι | 15 |
| 2.2 Οργανισμοί προστασίας του περιβάλλοντος ανά τον κόσμο | 19 |
| 2.3 Αιωρούμενα Σωματίδια, κατηγορίες και χαρακτηριστικά | 20 |
| 3 ΕΠΙΛΟΓΗ HARDWARE, ΣΥΝΔΕΣΜΟΛΟΓΙΑ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΗΣ ΣΥΣΚΕΥΗΣ GreenYourAir/1497 | 23 |
| 3.1 Επιλογή μικροεπεξεργαστή, μερών της συσκευής και μετρητή | 23 |
| 3.2 Hardware της συσκευής | 24 |
| 3.2.1 Μικροεπεξεργαστής: NodeMCU V3 Lua WIFI | 24 |
| 3.2.2 Μετρητής: Particulate Matter Sensor PMS5003 by Plantower | 25 |
| 3.2.3 DHT11 Temperature and Humidity Sensor Module | 26 |
| 3.2.4 MTS-203 toggle switch | 27 |
| 3.2.5 Wire Jumper Cables for Arduino | 27 |
| 3.3 Συνδεσμολογία της συσκευής GreenYourAir/1497 | 28 |
| 3.3.1 Πρώτο στάδιο συνδεσμολογίας | 29 |
| 3.3.2 Δεύτερο στάδιο συνδεσμολογίας | 30 |
| 3.3.3 Τρίτο στάδιο συνδεσμολογίας | 32 |
| 3.3.4 Τέταρτο στάδιο συνδεσμολογίας | 33 |
| 3.4 Προγραμματισμός της συσκευής GreenYourAir/1497 και αποστολή δεδομένων στο cloud | 34 |
| 3.4.1 Προετοιμασία του cloud για την μεταφορά δεδομένων | 35 |
| 3.4.2 Κώδικας της συσκευής | 40 |
| 3.4.3 Προγραμματισμός της συσκευής | 47 |
| 4 ΕΓΚΑΤΑΣΤΑΣΗ ΣΥΚΕΥΩΝ ΣΕ ΣΗΜΕΙΑ ΤΗΣ ΠΟΛΗΣ ΤΟΥ ΒΟΛΟΥ | 50 |
| 4.1 Σημεία τοποθέτησης | 50 |
| 4.2 Κουτί της συσκευής για την εγκατάσταση | 51 |
| 4.3 Εγκατάσταση των συσκευών | 52 |

| | | |
|----------|--|-----|
| 5 | ΕΦΑΡΜΟΓΗ ΚΑΤΑΓΡΑΦΗΣ ΠΕΡΙΣΤΑΤΙΚΩΝ ΣΤΟ ΝΟΣΟΚΟΜΕΙΟ ΤΟΥ ΒΟΛΟΥ | 54 |
| 5.1 | Δεδομένα της εφαρμογής για τα συμπτώματα του περιστατικού..... | 54 |
| 5.2 | Δημιουργία της εφαρμογής..... | 56 |
| 5.2.1 | Δημιουργία του βασικού κορμού της εφαρμογής με την χρήση του Google Forms..... | 57 |
| 5.2.2 | Εκκίνηση του AppSheet και δημιουργία της εφαρμογής..... | 61 |
| 6 | ΙΣΤΟΣΕΛΙΔΑ ΤΟΥ GreenYourAir | 68 |
| 6.1 | Προγραμματισμός του API στο Apps script..... | 68 |
| 6.1.1 | Προετοιμασία εγγράφου στο Google Sheets και προγραμματισμός του πρώτου API..... | 68 |
| 6.1.2 | Προγραμματισμός του δεύτερου API για την μεταφορά των δεδομένων στην ιστοσελίδα..... | 76 |
| 6.2 | Προγραμματισμός και κώδικας ιστοσελίδας..... | 81 |
| 6.2.1 | Κώδικας για την «Αρχική»..... | 81 |
| 6.2.2 | Κώδικας για τον «Χάρτη»..... | 84 |
| 6.2.3 | Κώδικας για τα «Διαγράμματα»..... | 89 |
| 6.2.4 | Κώδικας για το «Ημερολόγιο»..... | 95 |
| 6.2.5 | Κώδικας για τα «Νέα»..... | 98 |
| 6.2.6 | Κώδικας για τις «Πληροφορίες»..... | 102 |
| 6.2.7 | Κώδικας του CSS..... | 105 |
| 6.2.8 | Κώδικας για την κλήση και την αποθήκευση των δεδομένων του API..... | 115 |
| 6.2.9 | Κώδικας διαχείρισης του Server..... | 116 |
| 7 | ΜΕΤΡΗΣΕΙΣ ΓΙΑ ΤΗΝ ΧΡΟΝΙΚΗ ΠΕΡΙΟΔΟ ΑΠΟ 01/03/2019 ΕΩΣ 29/02/2020 | 120 |
| 7.1 | Μέσος όρος δωδεκάμηνης περιόδου..... | 120 |
| 7.2 | Μηνιαίοι μέσοι όροι..... | 120 |
| 7.3 | Ημερήσιο μέσοι όροι..... | 124 |
| 7.3.1 | Ημερήσιοι μέσοι όροι για τον Μάρτιο 2019..... | 124 |
| 7.3.2 | Ημερήσιοι μέσοι όροι για τον Απρίλιο 2019..... | 126 |
| 7.3.3 | Ημερήσιοι μέσοι όροι για τον Μάιο 2019..... | 127 |
| 7.3.4 | Ημερήσιοι μέσοι όροι για τον Ιούνιο 2019..... | 129 |
| 7.3.5 | Ημερήσιοι μέσοι όροι για τον Ιούλιο 2019..... | 130 |
| 7.3.6 | Ημερήσιοι μέσοι όροι για τον Αύγουστο 2019..... | 132 |
| 7.3.7 | Ημερήσιοι μέσοι όροι για τον Σεπτέμβριο 2019..... | 133 |
| 7.3.8 | Ημερήσιοι μέσοι όροι για τον Οκτώβριο 2019..... | 135 |

| | | |
|--------|--|------------|
| 7.3.9 | Ημερήσιοι μέσοι όροι για τον Νοέμβριο 2019 | 136 |
| 7.3.10 | Ημερήσιοι μέσοι όροι για τον Δεκέμβριο 2019 | 138 |
| 7.3.11 | Ημερήσιοι μέσοι όροι για τον Ιανουάριο 2020 | 139 |
| 7.3.12 | Ημερήσιοι μέσοι όροι για τον Φεβρουάριο 2020 | 141 |
| 8 | Συμπεράσματα | 142 |
| 8.1 | Ποιότητα του αέρα στην πόλη του Βόλου για την χρονική περίοδο από 01/03/2019 έως 29/02/2020 | 143 |
| 8.2 | Τρόποι Αντιμετώπισης..... | 145 |
| 8.3 | Περιθώρια για περαιτέρω έρευνα..... | 146 |
| | ΒΙΒΛΙΟΓΡΑΦΙΑ..... | 148 |

Κατάλογος Εικόνων

| | |
|--|----|
| Εικόνα 2.1: Θερμική μονάδα παραγωγής ηλεκτρικής ισχύος στην Πτολεμαΐδα. . . | 16 |
| Εικόνα 2.2: Δημιουργία όξινης βροχής. | 17 |
| Εικόνα 2.3: Φωτογραφία από το Λονδίνο κατά τη διάρκεια του έντονου περιστατικού ρύπανσης τον Δεκέμβριο του 1952. | 17 |
| Εικόνα 2.4: Μείωση ορατότητας από την υψηλή συγκέντρωση PMx. | 18 |
| Εικόνα 2.5: Σύγκριση του μεγέθους των κύριων κατηγοριών των Α.Σ. | 22 |
| Εικόνα 3.1: NodeMCU V3 Lua WIFI..... | 24 |
| Εικόνα 3.2: Particulate Matter Sensor PMS5003 by Plantower..... | 25 |
| Εικόνα 3.3: Μέθοδος σκέδασης του μετρητή Particulate Matter Sensor PMS5003 by Plantower..... | 26 |
| Εικόνα 3.4: DHT11 Temperature and Humidity Sensor Module..... | 26 |
| Εικόνα 3.5: MTS-203 toggle switch. | 27 |
| Εικόνα 3.6: Wire jumper cables for Arduino. | 27 |
| Εικόνα 3.7: Συνδεσμολογία Συσκευής. | 28 |
| Εικόνα 3.8: PIN του NodeMCU για το πρώτο στάδιο της συνδεσμολογίας. | 29 |
| Εικόνα 3.9: PIN του DHT11 για το πρώτο στάδιο της συνδεσμολογίας..... | 29 |
| Εικόνα 3.10: PIN του NodeMCU για το δεύτερο στάδιο της συνδεσμολογίας..... | 30 |
| Εικόνα 3.11: PIN του PMS5003 για το δεύτερο στάδιο της συνδεσμολογίας..... | 31 |
| Εικόνα 3.12: PIN του NodeMCU για το τρίτο στάδιο της συνδεσμολογίας..... | 32 |
| Εικόνα 3.13: PIN του MTS-203 για το τρίτο στάδιο της συνδεσμολογίας. | 32 |
| Εικόνα 3.14: PIN του MTS-203 για το τέταρτο στάδιο της συνδεσμολογίας..... | 33 |
| Εικόνα 3.15: PIN του PMS5003 για το τέταρτο στάδιο της συνδεσμολογίας. | 34 |
| Εικόνα 3.16: Δημιουργία ενός κενού εγγράφου στο Google Sheets..... | 35 |
| Εικόνα 3.17: Μεταφορά στο Google Apps Script από το Google Sheets. | 35 |
| Εικόνα 3.18: Περιβάλλον του Google Apps Script..... | 36 |
| Εικόνα 3.19: Δημοσίευση του API για την αποστολή και την καταγραφή των δεδομένων στο Google Sheets..... | 39 |
| Εικόνα 3.20: Development ID για την αποστολή και την αποθήκευση των δεδομένων στο Google Sheets..... | 40 |
| Εικόνα 3.21: Επιλογή πλακέτας στο Arduino IDE. | 47 |
| Εικόνα 3.22: Επιλογή της θύρας στο Arduino IDE..... | 48 |
| Εικόνα 3.23: Ανέβασμα του κώδικα στον Arduino..... | 48 |
| Εικόνα 3.24: Μορφή αποθήκευσης των μετρήσεων στο Google Sheets..... | 49 |
| Εικόνα 4.1: Τοποθεσίες των συσκευών..... | 51 |
| Εικόνα 4.2: Μπροστά και πίσω όψη του κουτιού. | 51 |
| Εικόνα 4.3: Πλάγια όψη του κουτιού..... | 52 |
| Εικόνα 4.4: Συσκευή στο Πεδίον Άρεως στο Τμήμα Μηχανολόγων Μηχανικών. . | 52 |
| Εικόνα 4.5: Συσκευή στην Κοραή..... | 53 |
| Εικόνα 4.6: Συσκευή στη Βρυζάκη. | 53 |
| Εικόνα 5.1: Σχήμα εφαρμογής και ενώσεις..... | 57 |
| Εικόνα 5.2: Πρώτη ενότητα εφαρμογής. | 59 |
| Εικόνα 5.3: Δεύτερη ενότητα εφαρμογής..... | 60 |
| Εικόνα 5.4: Δημιουργία εγγράφου στο Google Sheets για την αποθήκευση των δεδομένων που καταχωρεί ο χρήστης..... | 60 |
| Εικόνα 5.5: Εκκίνηση του AppSheet από το Google Forms..... | 61 |
| Εικόνα 5.6: Launch button και μετάβαση στο Dashboard του AppSheet..... | 62 |

| | |
|--|-----|
| Εικόνα 5.7: Ορισμός ονόματος, περιγραφής και εικονιδίου της εφαρμογής..... | 62 |
| Εικόνα 5.8: Μετάβαση στις επιλογές για την εμφάνιση της ερώτησης "Διάγνωση Εξιτηρίου"..... | 63 |
| Εικόνα 5.9: Ορισμός συνθήκης για την εμφάνιση της ερώτησης "Διάγνωση Εξιτηρίου"..... | 64 |
| Εικόνα 5.10: Έλεγχος της καταχώρησης στην ερώτηση "Ηλικία"..... | 64 |
| Εικόνα 5.11: Συνθήκη για την εμφάνιση των περιστατικών κάθε κλινικής. | 66 |
| Εικόνα 5.12: Καταχώρηση γενικών δεδομένων του περιστατικού..... | 66 |
| Εικόνα 5.13: Καταχώρηση των στοιχείων του ασθενή. | 67 |
| Εικόνα 5.14: Καταχωρίσεις του χρήστη..... | 67 |
| Εικόνα 6.1: Δημιουργία Apps Script..... | 69 |
| Εικόνα 6.2: Πίνακας αποθήκευσης των πιο πρόσφατων δεδομένων των μετρητών. | 70 |
| Εικόνα 6.3: Ορισμός εκτέλεσης του κώδικα του API ανά ένα λεπτό..... | 76 |
| Εικόνα 6.4: Sheet για τους μέσους όρους των τιμών. | 77 |
| Εικόνα 6.5: Δημοσίευση του δεύτερου API..... | 80 |
| Εικόνα 6.6: "Αρχική" της ιστοσελίδας. | 84 |
| Εικόνα 6.7: "Χάρτης" της ιστοσελίδας..... | 89 |
| Εικόνα 6.8: "Διαγράμματα" της ιστοσελίδας. | 95 |
| Εικόνα 6.9: "Ημερολόγιο" της ιστοσελίδας. | 98 |
| Εικόνα 6.10: "Νέα" της ιστοσελίδας..... | 101 |
| Εικόνα 8.1: Θερμοκρασιακή αναστροφή. | 144 |

Κατάλογος Πινάκων

| | |
|--|----|
| Πίνακας 2.1: Επιτρεπτά όρια ρύπων με βάση τον Παγκόσμιο Οργανισμό Υγείας.19 | |
| Πίνακας 2.2: Επιτρεπτά όρια ρύπων με βάση τον Ευρωπαϊκό Οργανισμό Περιβάλλοντος..... | 20 |
| Πίνακας 2.3: Επιτρεπτά όρια ρύπων με βάση το Υπουργείο Περιβάλλοντος και Ενέργειας..... | 20 |
| Πίνακας 3.1: Τεχνικά χαρακτηριστικά Particulate Matter Sensor PMS5003 by Plantower..... | 25 |
| Πίνακας 3.2: Επεξήγηση των PIN του NODEMCU για το πρώτο στάδιο της συνδεσμολογίας..... | 29 |
| Πίνακας 3.3: Επεξήγηση των PIN του DHT11 για το πρώτο στάδιο της συνδεσμολογίας..... | 30 |
| Πίνακας 3.4: Σύνδεση των PIN του NodeMCU με τα PIN του DHT11..... | 30 |
| Πίνακας 3.5: Επεξήγηση των PIN του NodeMCU για το δεύτερο στάδιο της συνδεσμολογίας..... | 31 |
| Πίνακας 3.6: Επεξήγηση των PIN του PMS5003 για το δεύτερο στάδιο της συνδεσμολογίας..... | 31 |
| Πίνακας 3.7: Σύνδεση των PIN του NodeMCU με τα PIN του PMS5003. | 31 |
| Πίνακας 3.8: Επεξήγηση των PIN του NodeMCU για το τρίτο στάδιο της συνδεσμολογίας..... | 32 |
| Πίνακας 3.9: Επεξήγηση των PIN του MTS-203 για το τρίτο στάδιο της συνδεσμολογίας..... | 32 |
| Πίνακας 3.10: Σύνδεση των PIN του Arduino με τα PIN του Shield..... | 33 |

| | |
|---|-----|
| Πίνακας 3.11: Επεξήγηση των PIN του MTS-203 για το τέταρτο στάδιο της συνδεσμολογίας..... | 33 |
| Πίνακας 3.12: Επεξήγηση των PIN του PMS5003 για το τέταρτο στάδιο της συνδεσμολογίας..... | 34 |
| Πίνακας 3.13: Σύνδεση των PIN του MTS-203 με τα PIN του PMS5003..... | 34 |
| Πίνακας 7.1: Μέσος όρος για την δωδεκάμηνη περίοδο..... | 120 |
| Πίνακας 7.2: Μηνιαίοι Μέσοι Όροι..... | 121 |
| Πίνακας 7.3: Εύρη τιμών από Μάρτιο 2019 - Μάιο 2019..... | 122 |
| Πίνακας 7.4: Εύρη τιμών από Ιούνιο 2019 - Αύγουστο 2019..... | 122 |
| Πίνακας 7.5: Εύρη τιμών από Σεπτέμβριο 2019 - Νοέμβριο 2019..... | 122 |
| Πίνακας 7.6: Εύρη τιμών από Δεκέμβριο 2019 - Φεβρουάριο 2020..... | 123 |
| Πίνακας 7.7: Αριθμός Υπερβάσεων ανά μήνα..... | 124 |
| Πίνακας 7.8: Συγκεντρωτικός πίνακας Υπερβάσεων..... | 124 |
| Πίνακας 7.9: Ημερήσιοι μέσοι όροι για τον Μάρτιο 2019..... | 125 |
| Πίνακας 7.10: Ημερήσιοι μέσοι όροι για το Απρίλιο 2019..... | 127 |
| Πίνακας 7.11: Ημερήσιοι μέσοι όροι για τον Μάιο 2019..... | 128 |
| Πίνακας 7.12: Ημερήσιοι μέσοι όροι για τον Ιούνιο 2019..... | 129 |
| Πίνακας 7.13: Ημερήσιοι μέσοι όροι για τον Ιούλιο 2019..... | 131 |
| Πίνακας 7.14: Ημερήσιοι μέσοι όροι για τον Αύγουστο 2019..... | 133 |
| Πίνακας 7.15: Ημερήσιοι μέσοι όροι για τον Σεπτέμβριο 2019..... | 134 |
| Πίνακας 7.16: Ημερήσιοι μέσοι όροι για τον Οκτώβριο 2019..... | 136 |
| Πίνακας 7.17: Ημερήσιοι μέσοι όροι για τον Νοέμβριο 2019..... | 137 |
| Πίνακας 7.18: Ημερήσιοι μέσοι όροι για τον Δεκέμβριο 2019..... | 139 |
| Πίνακας 7.19: Ημερήσιοι μέσοι όροι για τον Ιανουάριο 2020..... | 140 |
| Πίνακας 7.20: Ημερήσιοι μέσοι όροι για τον Φεβρουάριο 2020..... | 141 |

Κατάλογος για τους Κώδικες

| | |
|--|----|
| Κώδικας 3.1: Κώδικας για την δημιουργία του API μέσω Google Apps Script για την μεταφορά και την αποθήκευση των δεδομένων στο Google Sheets..... | 37 |
| Κώδικας 3.2: Πρώτο μέρος του κώδικα για την μεταφορά και την αποθήκευση των δεδομένων στο Google Sheet..... | 38 |
| Κώδικας 3.3: Δεύτερο μέρος του κώδικα για την μεταφορά και την αποθήκευση των δεδομένων στο Google Sheet..... | 39 |
| Κώδικας 3.4: Κώδικας της συσκευής GreenYourAir/1497..... | 43 |
| Κώδικας 3.5: Μέρος Α του κώδικα της συσκευής..... | 44 |
| Κώδικας 3.6: Μέρος Β του κώδικα της συσκευής..... | 45 |
| Κώδικας 3.7: Μέρος Γ του κώδικα της συσκευής..... | 46 |
| Κώδικας 5.1: Συνθήκη για την εμφάνιση της ερώτησης "Διάγνωση Εξιτηρίου".... | 63 |
| Κώδικας 5.2: Συνθήκη για την αντιστοίχιση των συμπτωμάτων με τις κλινικές..... | 65 |
| Κώδικας 6.1: Κώδικας API για την αποθήκευση των δεδομένων των 12 μετρητών..... | 73 |
| Κώδικας 6.2: Πρώτο σκέλος του κώδικα του API για την αποθήκευση των δεδομένων των 12 μετρητών..... | 74 |
| Κώδικας 6.3: Δεύτερο σκέλος του κώδικα του API για την αποθήκευση των δεδομένων των 12 μετρητών..... | 75 |
| Κώδικας 6.4: Μακροεντολή για τον υπολογισμό του μέσου όρου της πρώτης γραμμής του ενιαίου αρχείου των μετρήσεων..... | 77 |

| | |
|---|-----|
| Κώδικας 6.5: Κώδικας API για την αποστολή των μέσων όρων των μετρήσεων στην ιστοσελίδα..... | 80 |
| Κώδικας 6.6: Κώδικας της ιστοσελίδας για την “Αρχική”..... | 83 |
| Κώδικας 6.7: Κώδικας της ιστοσελίδας για τον “Χάρτη”..... | 89 |
| Κώδικας 6.8: Κώδικας της ιστοσελίδας για τα “Διαγράμματα”..... | 94 |
| Κώδικας 6.9: Κώδικας της ιστοσελίδας για το “Ημερολόγιο”..... | 98 |
| Κώδικας 6.10: Κώδικας της ιστοσελίδας για τα “Νέα”..... | 101 |
| Κώδικας 6.11: Κώδικας της ιστοσελίδας για τις “Πληροφορίες”..... | 105 |
| Κώδικας 6.12: Κώδικας CSS της ιστοσελίδας..... | 115 |
| Κώδικας 6.13: Κώδικας για την κλήση στο API..... | 116 |
| Κώδικας 6.14: Κώδικας διαχείρισης του Server..... | 119 |

Κατάλογος Διαγραμμάτων

| | |
|--|-----|
| Διάγραμμα 7.1: Μηνιαίοι Μέσοι Όροι..... | 121 |
| Διάγραμμα 7.2: Ημερήσιοι μέσοι όροι για τον Μάρτιο 2019..... | 125 |
| Διάγραμμα 7.3: Ημερήσιοι μέσοι όροι για τον Απρίλιο 2019..... | 127 |
| Διάγραμμα 7.4: Ημερήσιοι μέσοι όροι για τον Μάιο 2019..... | 128 |
| Διάγραμμα 7.5: Ημερήσιοι μέσοι όροι για τον Ιούνιο 2019..... | 130 |
| Διάγραμμα 7.6: Ημερήσιοι μέσοι όροι για τον Ιούλιο 2019..... | 131 |
| Διάγραμμα 7.7: Ημερήσιοι μέσοι όροι για τον Αύγουστο 2019..... | 133 |
| Διάγραμμα 7.8: Ημερήσιοι μέσοι όροι για τον Σεπτέμβριο 2019..... | 134 |
| Διάγραμμα 7.9: Ημερήσιοι μέσοι όροι για τον Οκτώβριο 2019..... | 136 |
| Διάγραμμα 7.10: Ημερήσιοι μέσοι όροι για τον Νοέμβριο 2019..... | 137 |
| Διάγραμμα 7.11: Ημερήσιοι μέσοι όροι για τον Δεκέμβριο 2019..... | 139 |
| Διάγραμμα 7.12: Ημερήσιοι μέσοι όροι για τον Ιανουάριο 2020..... | 140 |
| Διάγραμμα 7.13: Ημερήσιοι μέσοι όροι για τον Φεβρουάριο 2020..... | 142 |
| Διάγραμμα 8.1: Ημερήσιοι μέσοι όροι από 01/03/2019 έως 29/02/2020..... | 143 |

1 ΕΙΣΑΓΩΓΗ

Η κακή ποιότητα του αέρα που αναπνέουμε επηρεάζει την υγεία μας και την καθημερινή μας ζωή. Τα τελευταία χρόνια η ατμοσφαιρική ρύπανση είναι ένα από τα προβλήματα που αντιμετωπίζουν οι σύγχρονες μεγαλουπόλεις σε όλο τον κόσμο. Σύμφωνα πρόσφατη έρευνα του Παγκόσμιου Οργανισμού Υγείας εκτιμάται ότι η ατμοσφαιρική ρύπανση είναι υπεύθυνη για τον πρόωρο θάνατο 4,2 εκατομμυρίων ατόμων το χρόνο [1]. Ένας από τους κυριότερους αέριους ρύπους είναι τα μικροσκοπικά αιωρούμενα σωματίδια, τα οποία, σύμφωνα με πρόσφατη ανακοίνωση του Ευρωπαϊκού Οργανισμού Περιβάλλοντος (ΕΕΑ) είναι υπεύθυνα μόνο για την Ελλάδα για περισσότερους από 11000 πρόωρους θανάτους [2].

Η ατμοσφαιρική ρύπανση είναι ένα σύγχρονο πρόβλημα το οποίο καλείται να αντιμετωπίσει ο άνθρωπος. Οι καθημερινές συνήθειες και ανάγκες του ανθρώπου είναι η κύρια πηγή παραγωγής ρύπων σε όλο τον κόσμο. Η μετακίνηση, η θέρμανση, η χρήση ηλεκτρικών και ηλεκτρονικών συσκευών και πολλά άλλα είναι οι αιτίες που προκαλούν την παραγωγή ρύπων είτε άμεσα είτε έμμεσα.

Σημαντικό ρόλο στην παραγωγή ρύπων έχει και η ραγδαία ανάπτυξη της βιομηχανίας, ως επακόλουθο της αύξησης του παγκόσμιου πληθυσμού και των αναγκών του. Τα αέρια απόβλητα της βιομηχανίας, έχει αποδειχθεί, πως είναι ιδιαίτερα βλαβερά για την υγεία του ανθρώπου. Τα τελευταία χρόνια οι ανεπτυγμένες χώρες επενδύουν σε διαδικασίες αντιμετώπισης των αέριων αποβλήτων από τις βιομηχανίες, παρέχοντας οικονομική υποστήριξη και τεχνολογική επάρκεια για την ανάπτυξη των διαδικασιών αυτών. Γενικά η αντιμετώπιση του φαινομένου αυτού είναι μία πολύ ακριβή διαδικασία και για τις βιομηχανίες αλλά και για το κάθε κράτος. Οι μη ανεπτυγμένες χώρες που δεν μπορούν να υποστηρίξουν οικονομικά και τεχνολογικά τις διαδικασίες αυτές δυστυχώς παράγουν και ελευθερώνουν στον αέρα απόβλητα τα οποία έχουν αντίκτυπο στην υγεία των πολιτών, όχι μόνο στην χώρα αλλά και στις γειτονικές, μιας και οι αέριοι ρύποι δεν έχουν σύνορα.

Το κάθε κράτος έχει ορίσει συγκεκριμένα όρια για την αέρια ρύπανση και για κάθε ρύπο ξεχωριστά. Ο Παγκόσμιος Οργανισμός Υγείας μετά από μελέτες έχει ορίσει συγκεκριμένα όρια για κάθε ρύπο ανάλογα τις ιδιαιτερότητες του. Οι ανεπτυγμένες χώρες ακολουθούν τα όρια που έχει θεσπίσει ο Π.Ο.Υ. προσαρμόζοντας την νομοθεσία του σε αυτά. Δυστυχώς όμως οι μη ανεπτυγμένες δεν μπορούν να ακολουθήσουν τις οδηγίες του Π.Ο.Υ. και το πρόβλημα της αέριας ρύπανσης σε αυτές είναι μεγαλύτερο.

Οι συνέπειες της κακής ποιότητας του αέρα επηρεάζουν την ανθρώπινη υγεία σε σημαντικό βαθμό. Οι ευπαθείς ομάδες επηρεάζονται πιο άμεσα και τα προβλήματα όπως το άσθμα, η αναπνευστική ανεπάρκεια, η βρογχίτιδα, τα καρδιακά και κυκλοφορικά νοσήματα και άλλα αναπνευστικά νοσήματα παρατηρείται να αυξάνονται όταν υπάρχει επεισόδιο αέριας ρύπανσης. Συνεπώς η κακή ποιότητα του αέρα είναι ένα πολύ σημαντικό πρόβλημα με σοβαρές επιπτώσεις στην υγεία του ανθρώπου που πρέπει να αντιμετωπιστεί όσο το δυνατόν πιο άμεσα.

Στόχος της παρούσας εργασίας είναι η δημιουργία ενός αξιόπιστου δικτύου παρακολούθησης των συγκεντρώσεων των αιωρούμενων σωματιδίων (PM_{2.5}) στην πόλη του Βόλου με την χρήση της συσκευής GreenYourAir/1178 και η εκτίμηση της συσχέτισής τους με αυξημένα περιστατικά με καρδιο-αναπνευστικά προβλήματα. Στο πλαίσιο της εργασίας μελετήθηκαν οι διαφορετικές τεχνικές μέτρησης των PM_{2.5} και αναπτύχθηκε η συσκευή GreenYourAir/1497 με δυνατότητα μέτρησης των PM_{1.0}, PM_{2.5} και των PM₁₀.

Η παρούσα εργασία διαρθρώνεται ως εξής:

Στο Κεφάλαιο 2 αναφέρονται οι βασικοί ατμοσφαιρικοί ρύποι, καθώς και οδηγίες από για την ατμοσφαιρική ρύπανση του Παγκόσμιου Οργανισμού Υγείας, του Ευρωπαϊκού Οργανισμού Περιβάλλοντος και του Υπουργείου Περιβάλλοντος και Ενέργειας.

Στο Κεφάλαιο 3 αναλύεται η μεθοδολογία για την δημιουργία της συσκευής GreenYourAir/1497 η οποία μπορεί να μετρήσει την Θερμοκρασία, την Υγρασία, Τα PM_{1.0}, τα PM_{2.5} και τα PM₁₀. Παρουσιάζεται η διαδικασία της συνδεσμολογίας, ο κώδικας της συσκευής και ο προγραμματισμός της συσκευής.

Στο Κεφάλαιο 4 περιγράφεται η μεθοδολογία της επιλογής των σημείων για την τοποθέτηση των μετρητών και την δημιουργία του δικτύου μέτρησης στην πόλη του Βόλου.

Στο Κεφάλαιο 5 αναφέρεται στην εφαρμογή καταγραφής περιστατικών από το Νοσοκομείο του Βόλου. Αναλύεται η διαδικασία και τα βήματα για την δημιουργία της εφαρμογής.

Στο Κεφάλαιο 6 παρουσιάζονται οι κώδικες και η διαδικασία για την δημιουργία της ιστοσελίδας <http://greenyourair.org/>. Στην ιστοσελίδα οι ενδιαφερόμενοι μπορούν να ενημερωθούν για τα επίπεδα των PM_{2.5} στην πόλη του Βόλου σε πραγματικό χρόνο, τα επίπεδα των PM_{2.5} των τελευταίων 24 ωρών καθώς και άλλες πληροφορίες.

Στο Κεφάλαιο 7 παρουσιάζονται τα δεδομένα για την χρονική περίοδο από 01/03/2019 έως και 29/02/2020 για τα επίπεδα των PM_{2.5} στην πόλη του Βόλου.

Στο Κεφάλαιο 8 αναφέρονται τα συμπεράσματα που προκύπτουν από τα δεδομένα για τα επίπεδα των PM_{2.5} για την χρονική περίοδο από 01/03/2019 έως και 29/02/2020. Επίσης παρουσιάζονται τρόποι αντιμετώπισης της ατμοσφαιρικής ρύπανσης σε συλλογικό και ατομικό επίπεδο.

2 ΑΤΜΟΣΦΑΙΡΙΚΟΙ ΡΥΠΟΙ ΚΑΙ ΟΡΓΑΝΙΣΜΟΙ ΠΡΟΣΤΑΣΙΑΣ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ

Στο κεφάλαιο αυτό αναφέρονται οι βασικοί ατμοσφαιρικοί ρύποι οι οποίοι προκαλούν προβλήματα στο περιβάλλον αλλά και στη υγεία του ανθρώπου. Επίσης αναφέρονται οι οδηγίες για την ατμοσφαιρική ρύπανση του Παγκόσμιου Οργανισμού Υγείας, του Ευρωπαϊκού Οργανισμού Περιβάλλοντος και του Υπουργείου Περιβάλλοντος και Ενέργειας.

2.1 Κύριοι Ατμοσφαιρικοί Ρύποι

Οι ατμοσφαιρικοί ρύποι χωρίζονται σε δύο κατηγορίες. Στους πρωτογενείς ρύπους, που εκπέμπονται άμεσα στην ατμόσφαιρα από τις πηγές ρύπανσης (για παράδειγμα το διοξείδιο του θείου και το μονοξείδιο του αζώτου) και στους δευτερογενείς ρύπους που σχηματίζονται στην ατμόσφαιρα μετά από χημικές αντιδράσεις (για παράδειγμα το όζον). Οι ρύποι αυτοί χωρίζονται στους αέριους ρύπους και στους σωματιδιακούς ρύπους που έχουν στερεή μορφή ή υγρή μορφή.

Μονοξείδιο του άνθρακα (CO)

Το μονοξείδιο του άνθρακα είναι ένα άοσμο, άχρωμο και άγευστο αέριο που είναι εξαιρετικά τοξικό και εκλύεται κατά την ατελή καύση ορυκτών καυσίμων και βιομάζας και από τις δασικές πυρκαγιές. Ο καπνός του τσιγάρου, η κυκλοφοριακή συμφόρηση και η βιομηχανία είναι βασικές πηγές παραγωγής μονοξειδίου του άνθρακα. Το μονοξείδιο του άνθρακα συμμετέχει στη δημιουργία της φωτοχημικής αιθαλομίχλης.

Διοξείδιο του άνθρακα (CO₂)

Το διοξείδιο του άνθρακα είναι άοσμο, άχρωμο και άγευστο, χωρίς να είναι τοξικό όπως το CO. Είναι υποπροϊόν της καύσης ορυκτών καυσίμων όπως του πετρελαίου, της βενζίνης και άλλων, αλλά και άλλων ανθρωπογενών δραστηριοτήτων όπως η αποψίλωση δασών και οι αλλαγές χρήσης γης. Αποτελεί το σημαντικότερο αέριο του θερμοκηπίου. Διοξείδιο του άνθρακα εκπέμπεται και από φυσικές πηγές όπως τα ηφαίστεια, την αναπνοή των ανθρώπων, των ζώων και των φυτών.

Οι θερμικές μονάδες παραγωγής ηλεκτρικής ισχύος που χρησιμοποιούν ορυκτά καύσιμα εκπέμπουν μεγάλες ποσότητες αέριων ρύπων όπως για παράδειγμα το διοξείδιο του άνθρακα. Στην Εικόνα 2.1 παρουσιάζεται η μονάδα παραγωγής ηλεκτρικής ισχύος στην Πτολεμαίδα.



Εικόνα 2.1: Θερμική μονάδα παραγωγής ηλεκτρικής ισχύος στην Πτολεμαΐδα.

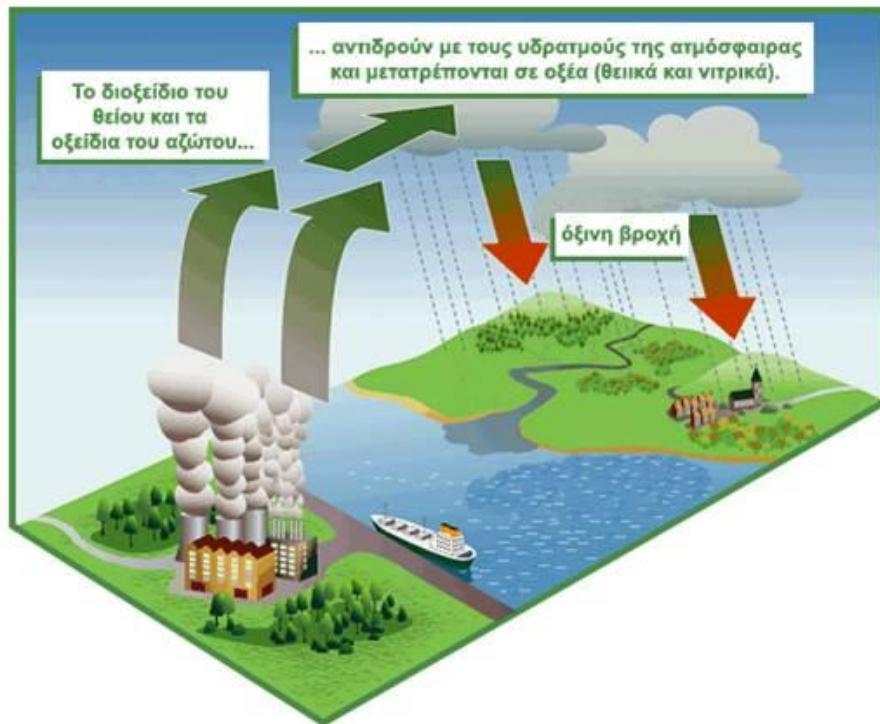
Οξειδία του Αζώτου (NO_x)

Τα δύο κύρια οξειδία του αζώτου είναι το μονοξείδιο του αζώτου (NO) και το διοξείδιο του αζώτου (NO₂). Παράγονται κυρίως κατά την καύση ορυκτών καυσίμων και κύριες πηγές ρύπανσης θεωρούνται οι κινητήρες οχημάτων, τα εργοστάσια, οι θερμικοί σταθμοί παραγωγής ηλεκτρικής ενέργειας και η θέρμανση κτιρίων. Τα οξειδία του αζώτου αποτελούν έναν από τους πιο σημαντικούς ρύπους που συμβάλουν άμεσα στην ποιότητα του αέρα και στην ανθρώπινη υγεία. Συμμετέχουν περίπου κατά το 1/3 στη δημιουργία της όξινης βροχής, η οποία προκαλεί σημαντικά οικολογικά προβλήματα.

Διοξείδιο του Θείου (SO₂)

Οι γαϊάνθρακες και το πετρέλαιο περιέχουν σημαντικές ποσότητες θείου, το μεγαλύτερο μέρος του οποίου μετατρέπεται σε διοξείδιο του θείου κατά την καύση τους. Το διοξείδιο του θείου αποτελεί έναν σημαντικό αέριο ρύπο με σημαντικές επιπτώσεις στην υγεία των ανθρώπων και κυρίως στα παιδιά και τους ηλικιωμένους. Στο διοξείδιο του θείου οφείλεται κατά κύριο λόγο η δημιουργία του φαινομένου της όξινης βροχής (Εικόνα 2.2). Με τα μέτρα που έχουν ληφθεί τα τελευταία 30 χρόνια έχουν μειωθεί έως και 90% οι εκπομπές των όξινων αερίων.

Τα αποτελέσματα της ρύπανσης από το SO₂ εμφανίζονται πολλές φορές ύστερα από πολύχρονη έκθεση με τη μορφή χρόνιας βρογχίτιδας, καρκίνου του λάρυγγα και άλλων αναπνευστικών προβλημάτων. Βέβαια δεν λείπουν και περιστατικά όπου από την ξαφνική και έντονη ρύπανση του SO₂ (σε συνδυασμό με θερμοκρασιακή αναστροφή) έχουν χαθεί ανθρώπινες ζωές. Ένα τέτοιο σημαντικό επεισόδιο ρύπανσης συνέβη στο Λονδίνο τον Δεκέμβριο του 1952 και είχε ως αποτέλεσμα τον θάνατο χιλιάδων ανθρώπων.



Εικόνα 2.2: Δημιουργία όξινης βροχής.

Όζον (O₃)

Το όζον στην στρατόσφαιρα είναι ένα απαραίτητο συστατικό για την προστασία από την υπεριώδη ακτινοβολία και την μείωση των επιπτώσεων αυτής της ακτινοβολίας στον άνθρωπο, τα ζώα, το φυτοπλαγκτόν και τα υλικά. Όταν όμως σχηματίζεται στην τροπόσφαιρα λόγω των φωτοχημικών αντιδράσεων, αποτελεί ένα σημαντικό ρύπο με βλαβερές συνέπειες στην ανθρώπινη υγεία, τα δάση και τις καλλιέργειες.



Εικόνα 2.3: Φωτογραφία από το Λονδίνο κατά τη διάρκεια του έντονου περιστατικού ρύπανσης τον Δεκέμβριο του 1952.

Αιωρούμενα σωματίδια (PMx)

Τα Αιωρούμενα σωματίδια τα οποία συνδέονται άμεσα με την ανθρώπινη υγεία είναι τα $PM_{1.0}$, τα $PM_{2.5}$ και τα PM_{10} . Τα αιωρούμενα σωματίδια προκαλούν σημαντικά προβλήματα στην υγεία του ανθρώπου, με τα σημαντικότερα να εντοπίζονται στο αναπνευστικό και καρδιαγγειακό σύστημα. Οι ευπαθείς ομάδες (ηλικιωμένοι, παιδιά, άτομα με χρόνια αναπνευστικά προβλήματα κ.α.) επηρεάζονται άμεσα από τις υψηλές τιμές συγκεντρώσεων αιωρούμενων σωματιδίων στην ατμόσφαιρα και οι επιπτώσεις στην υγεία τους και την καθημερινή τους ζωή είναι σημαντικές. Όσο πιο μικρά είναι τα σωματίδια τόσο πιο εύκολα εισέρχονται στους πνεύμονες και ανάλογα με την σύσταση τους προκαλούν ασθένειες όπως το άσθμα, η χρόνια βρογχίτιδα, ο καρκίνος στους πνεύμονες, ακόμη και ο πρόωρος θάνατος σε περιπτώσεις που οι συγκεντρώσεις είναι πολύ υψηλές. Εκτός από τα προβλήματα που δημιουργούν στην υγεία του ανθρώπου, τα αιωρούμενα σωματίδια προκαλούν προβλήματα και στο περιβάλλον. Ένα χαρακτηριστικό παράδειγμα των Α.Σ. είναι ότι μειώνουν την ορατότητα όταν βρίσκονται σε υψηλές συγκεντρώσεις στην ατμόσφαιρα (Εικόνα 2.4).



Εικόνα 2.4: Μείωση ορατότητας από την υψηλή συγκέντρωση PMx.

2.2 Οργανισμοί προστασίας του περιβάλλοντος ανά τον κόσμο

Στις μέρες μας η προστασία του περιβάλλοντος αποτελεί ένα από τα πιο σημαντικά προβλήματα που καλείται να αντιμετωπίσει η ανθρωπότητα. Για τον λόγο αυτό έχουν ιδρυθεί οργανισμοί που έχουν ως σκοπό την παρακολούθηση και την δημοσίευση αποτελεσμάτων που αφορούν την ποιότητα του αέρα, αλλά και την επιβολή ορίων για κάθε έναν από τους αέριους ρύπους ξεχωριστά. Οι βασικότεροι οργανισμοί που επενδύουν σε έρευνες για την καταπολέμηση και την παρακολούθηση της αέριας ρύπανσης είναι:

- Παγκόσμιος Οργανισμός Υγείας

Ο Παγκόσμιος Οργανισμός Υγείας είναι μία εξειδικευμένη υπηρεσία των Ηνωμένων Εθνών με αρμοδιότητα την δημόσια υγεία. Στόχος του Π.Ο.Υ. είναι η διατήρηση του υψηλότερου επιπέδου υγείας για όλους τους λαούς του κόσμου. Η αέρια ρύπανση σχετίζεται άμεσα με την υγεία του ανθρώπου, συνεπώς η παρακολούθηση της και η επιβολή ορίων είναι ένα από τα κύρια αντικείμενα ενασχόλησης του Π.Ο.Υ.. Η Ελλάδα έχει ενταχθεί στον Ο.Η.Ε. από το 1945, έτσι οφείλει να υπακούει και να ακολουθεί τις συμβουλές του Π.Ο.Υ. σχετικά με την αέρια ρύπανση και την αντιμετώπιση της [1]. Σύμφωνα με τις οδηγίες του Π.Ο.Υ. τα επιτρεπτά όρια της αέριας ρύπανσης παρουσιάζονται στον Πίνακα 2.1.

| Ρύπος | Χρονικό Περιθώριο | Μέση τιμή |
|-------------------|--|---|
| NO ₂ | Μέσος όρος 1 ώρας: Ετήσιος μέσος όρος: | 200 μg/m ³ 40 μg/m ³ |
| SO ₂ | Μέσος όρος 10 λεπτών: Μέσος όρος 24 ωρών: | 500 μg/m ³ 20 μg/m ³ |
| O ₃ | Μέσος όρος 8 ωρών: | 100 μg/m ³ |
| PM _{2.5} | Μέσος όρος 24 ωρών: Ετήσιος μέσος όρος: | 25 μg/m ³ 10 μg/m ³ |
| PM ₁₀ | Μέσος όρος 24 ωρών: Ετήσιος μέσος όρος: | 50 μg/m ³ 20 μg/m ³ |

Πίνακας 2.1: Επιτρεπτά όρια ρύπων με βάση τον Παγκόσμιο Οργανισμό Υγείας.

- Ευρωπαϊκός Οργανισμός Περιβάλλοντος

Ο Ευρωπαϊκός Οργανισμός Περιβάλλοντος είναι ένας οργανισμός της Ευρωπαϊκής Ένωσης που έχει ως σκοπό την παροχή έγκυρης και ανεξάρτητης πληροφόρησης για το περιβάλλον. Επίσης στοχεύει στην υποστήριξη της βιώσιμης ανάπτυξης συμβάλλοντας στην επίτευξη σημαντικών και μετρήσιμων βελτιώσεων λαμβάνοντας όλα τα αναγκαία μέτρα για την προστασία του περιβάλλοντος της Ευρώπης. Σύμφωνα με τις οδηγίες του Ε.Ο.Π. [3] τα επιτρεπτά όρια της αέριας ρύπανσης παρουσιάζονται στον Πίνακα 2.2.

| Ρύπος | Χρονικό Περιθώριο | Μέση τιμή |
|-------------------|----------------------------------|-----------------------|
| NO ₂ | Μέσος όρος 1 ώρας: | 200 µg/m ³ |
| | Ετήσιος μέσος όρος: | 40 µg/m ³ |
| SO ₂ | Μέσος όρος 1 ώρας: | 350 µg/m ³ |
| | Μέσος όρος 24 ωρών: | 125 µg/m ³ |
| CO | Μέγιστη ημερήσια 8ωρη μέση τιμή: | 10 mg/m ³ |
| PM _{2.5} | Ετήσιος μέσος όρος: | 25 µg/m ³ |
| PM ₁₀ | Μέσος όρος 24 ωρών: | 50 µg/m ³ |
| | Ετήσιος μέσος όρος: | 40 µg/m ³ |

Πίνακας 2.2: Επιτρεπτά όρια ρύπων με βάση τον Ευρωπαϊκό Οργανισμό Περιβάλλοντος.

- Υπουργείο Περιβάλλοντος και Ενέργειας

Στην χώρα μας το Υπουργείο Περιβάλλοντος και Ενέργειας είναι υπεύθυνο για την παρακολούθηση της ατμοσφαιρικής ρύπανσης. Με βάση τους Παγκόσμιους και Ευρωπαϊκούς φορείς και τις οδηγίες αυτών για την προστασία του περιβάλλοντος το Υπουργείο Περιβάλλοντος και Ενέργειας είναι υπεύθυνο για την θέσπιση συγκεκριμένων ορίων για την ατμοσφαιρική ρύπανση. Σύμφωνα με τις οδηγίες του Υ.Π.Ε. [4] τα επιτρεπτά όρια της αέριας ρύπανσης παρουσιάζονται στον Πίνακα 2.3.

| Ρύπος | Χρονικό Περιθώριο | Μέση τιμή |
|-------------------|----------------------------------|-----------------------|
| NO ₂ | Μέσος όρος 1 ώρας: | 200 µg/m ³ |
| | Ετήσιος μέσος όρος: | 40 µg/m ³ |
| SO ₂ | Μέσος όρος 1 ώρας: | 350 µg/m ³ |
| | Ετήσιος μέσος όρος: | 125 µg/m ³ |
| CO | Μέγιστη ημερήσια 8ωρη μέση τιμή: | 10 mg/m ³ |
| O ₃ | Μέγιστη ημερήσια 8ωρη μέση τιμή: | 120 µg/m ³ |
| PM _{2.5} | Ετήσιος μέσος όρος: | 25 µg/m ³ |
| PM ₁₀ | Μέσος όρος 24 ωρών: | 50 µg/m ³ |
| | Ετήσιος μέσος όρος: | 20 µg/m ³ |

Πίνακας 2.3: Επιτρεπτά όρια ρύπων με βάση το Υπουργείο Περιβάλλοντος και Ενέργειας .

2.3 Αιωρούμενα Σωματίδια, κατηγορίες και χαρακτηριστικά

Αιωρούμενα Σωματίδια είναι οποιουδήποτε μεγέθους υλικό στον αέρα σε στερεή ή υγρή μορφή από μερικά νανόμετρα έως και κάποιες δεκάδες μικρόμετρα. Τα σωματίδια με διάμετρο μικρότερη των 10 μικρομέτρων σχηματίζουν τα αερολύματα (καπνός, ομίχλη κ.α.). Επίσης συμμετέχουν σε φωτοχημικές αντιδράσεις με αποτέλεσμα να αλλάζει η σύσταση και το μέγεθος τους. Η ταξινόμηση των Α.Σ. γίνεται με βάση το μέγεθος, το σχήμα, την πυκνότητα, την τοξικότητα και την ικανότητα συμμετοχής σε αντιδράσεις.

Τα Α.Σ. χωρίζονται σε τρεις κύριες κατηγορίες με βάση την συμπεριφορά τους ως προς το αναπνευστικό σύστημα των ανθρώπων. Στην πρώτη κατηγορία ανήκουν τα σωματίδια με διάμετρο μικρότερη των 100 μικρομέτρων και μεγαλύτερη των 10 μικρομέτρων, τα οποία εισέρχονται μέχρι τον λάρυγγα και χαρακτηρίζονται ως εισπνεύσιμα. Στην δεύτερη κατηγορία ανήκουν τα σωματίδια με διάμετρο μικρότερη των 10 μικρομέτρων και μεγαλύτερη των 4 μικρομέτρων, τα οποία διασχίζουν τον λάρυγγα και τους πνεύμονες και χαρακτηρίζονται ως θωρακικό κλάσμα. Στην τρίτη και τελευταία κατηγορία ανήκουν τα σωματίδια με διάμετρο μικρότερη των 4 μικρομέτρων, τα οποία φτάνουν στο εσωτερικό των πνευμόνων και χαρακτηρίζονται ως αναπνεύσιμα. Επίσης με βάση την προέλευση τους τα Α.Σ. χωρίζονται σε λεπτομερή, έχουν διάμετρο μικρότερη από 1 μικρόμετρο και παραμένουν στην ατμόσφαιρα για μεγάλο χρονικό διάστημα και σε χονδροειδή τα οποία έχουν διάμετρο μεγαλύτερη από 2.5 μικρόμετρα.

Κατά κύριο λόγο τα αιωρούμενα σωματίδια προέρχονται από ανθρωπογενείς αλλά και φυσικές πηγές. Στις φυσικές πηγές κατατάσσονται τα ηφαίστεια, οι δασικές πυρκαγιές, η θάλασσα, η μεταφερόμενη σκόνη από τις ερήμους, η γύρη των φυτών και άλλα. Στις ανθρωπογενείς πηγές κατατάσσονται οι διεργασίες καύσης (κινητήρες αυτοκινήτων, θέρμανση, παραγωγή ηλεκτρικής ενέργειας με την χρήση θερμικών σταθμών, οι βιομηχανικές δραστηριότητες, οι μεταφορές, η γεωργία και άλλα.

Τα αιωρούμενα σωματίδια $PM_{1.0}$, $PM_{2.5}$ και PM_{10} (σχηματικά παρουσιάζονται στην Εικόνα 2.5) παραμένουν στην ατμόσφαιρα για συγκεκριμένο χρόνο. Ο χρόνος αυτός εξαρτάται από την ταχύτητα καθίζησης η οποία περιγράφεται από τον νόμο του Stokes.

$$\text{Νόμος του Stokes: } V_{ts} = \frac{P_p \cdot D_p^2 \cdot g}{18 \cdot \mu}$$

Όπου:

V_{ts} : οριακή ταχύτητα καθίζησης των σωματιδίων

g : επιτάχυνση της βαρύτητας

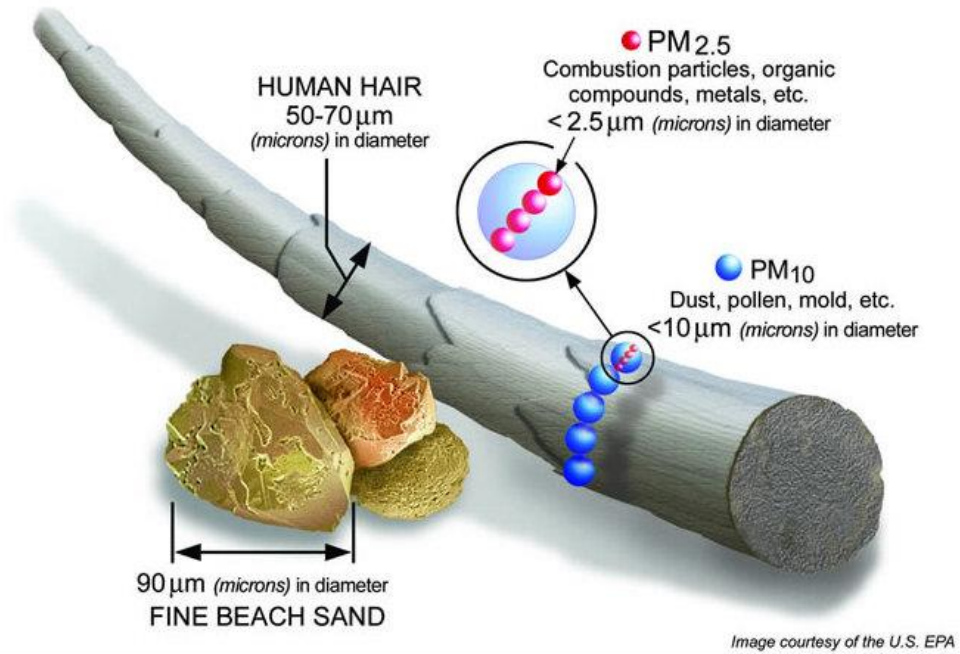
P_p : πυκνότητα του σωματιδίου σε

D_p : ακτίνα του σωματιδίου σε μm

μ : ιξώδες του μέσου καθίζησης

Συνεπώς με βάση το σχήμα, την διάμετρο και τα χαρακτηριστικά του σωματιδίου μπορεί να εκτιμηθεί ο χρόνος παραμονής του στην ατμόσφαιρα. Τα μεγάλα σωματίδια παραμένουν στην ατμόσφαιρα για λίγο χρονικό διάστημα. Για παράδειγμα τα σωματίδια της τάξης των δεκάδων μικρομέτρων παραμένουν στην ατμόσφαιρα για μερικές ώρες. Τα μικρότερα σωματίδια με διάμετρο από 2 έως 3 μικρόμετρα παραμένουν στην ατμόσφαιρα για μερικές ημέρες ανάλογα και τις μετεωρολογικές συνθήκες που επικρατούν. Τα πολύ μικρά σωματίδια τα οποία έχουν διάμετρο μικρότερη από 1 μικρόμετρο παραμένουν στην ατμόσφαιρα για μεγάλο χρονικό διάστημα. Επίσης τα σωματίδια λόγω του μικρού μεγέθους που έχουν μπορούν να μεταφερθούν από τον άνεμο από κάποιες δεκάδες έως και 1000 km προτού αποθεθούν

στην ξηρά ή σε υδάτινες επιφάνειες. Αυτό έχει ως αποτέλεσμα να υπάρχει ατμοσφαιρική ρύπανση σε περιοχές που δεν έχουν σημαντικές ανθρωπογενείς ή φυσικές πηγές ρύπανσης.



Εικόνα 2.5: Σύγκριση του μεγέθους των κύριων κατηγοριών των Α.Σ.

3 ΕΠΙΛΟΓΗ HARDWARE, ΣΥΝΔΕΣΜΟΛΟΓΙΑ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΗΣ ΣΥΣΚΕΥΗΣ GreenYourAir/1497

Στο κεφάλαιο αυτό παρουσιάζονται τα μέρη που χρειάζονται για την σωστή λειτουργία της συσκευής GreenYourAir/1497, η συνδεσμολογία και ο προγραμματισμός που απαιτείται. Στο παρελθόν έχουν γίνει αρκετές μελέτες ανάπτυξης συσκευών χαμηλού κόστους για την μέτρηση της ατμοσφαιρικής ρύπανσης όπως η μελέτη στην Σιάν της Κίνας [5], μελέτες του Οργανισμού Προστασίας του Περιβάλλοντος των Ηνωμένων Πολιτειών της Αμερικής [6] και άλλες μελέτες [7], [8], [9]. Λαμβάνοντας υπόψιν τα αποτελέσματα και την μεθοδολογία των μελετών αυτών έγινε και η επιλογή του μικροεπεξεργαστή, των μερών της συσκευής καθώς και του μετρητή για την κατασκευή της συσκευής GreenYourAir/1497.

3.1 Επιλογή μικροεπεξεργαστή, μερών της συσκευής και μετρητή

Για την δημιουργία μιας συσκευής μέτρησης σωματιδίων $PM_{2.5}$ μία από τις πιο σημαντικές επιλογές είναι η επιλογή του μετρητή. Η μέτρηση των σωματιδίων $PM_{2.5}$ αποτελεί μία πιο περίπλοκη διαδικασία σε σχέση με του υπόλοιπους ρύπους λόγω της ιδιομορφίας και της ποικιλίας των PM_x . Η μέτρηση της συγκέντρωσης των PM_x χωρίζεται σε δύο κατηγορίες οι οποίες διακρίνονται στις μεθόδους που χρησιμοποιούν μια μέθοδο σήμανσης και στην συνέχεια μετρούν την συγκέντρωση με την χρήση φίλτρου και σε αυτές που γίνονται σε πραγματικό χρόνο. Στην πρώτη κατηγορία μέτρησης της συγκέντρωσης ανήκει η μέθοδος των κωνικών στοιχείων, η μέθοδος της β-ακτινοβολίας και η μέθοδος του μαύρου καπνού. Και οι τρεις μέθοδοι λειτουργούν με ροή αέρα που στο τέλος τα σωματίδια προσκρούουν πάνω σε ένα φίλτρο, στο οποίο μετράτε η συγκέντρωση αυτών. Στην δεύτερη κατηγορία ανήκει η μέθοδος της σκέδασης του φωτός, η μέθοδος της απευθείας απεικόνισης και η μέθοδος της επισκόπησης του φωτός. Στις μέρες μας η πιο διαδεδομένη μέθοδος της δεύτερης κατηγορίας είναι αυτή της σκέδασης του φωτός. Η λειτουργία της βασίζεται στην ροή αέρα μέσω ενός θαλάμου στον οποίο υπάρχει μία ακτίνα laser. Η ποσότητα φωτός που διασκορπίζεται από τα σωματίδια ανιχνεύεται από την ακτίνα laser και μεταφράζεται σε ηλεκτρικούς παλμούς. Στην συνέχεια ο μικροεπεξεργαστής αναλύει τα σήματα και υπολογίζει την συγκέντρωση μάζας με βάση το πλάτος των παλμών. Ένα εξίσου σημαντικό κομμάτι της όλης διαδικασίας είναι η επιλογή του μικροεπεξεργαστή. Η επιλογή του θα πρέπει να γίνει με τέτοιο τρόπο ώστε να καλύπτει όλες τις απαιτήσεις του μετρητή και τις απαιτήσεις της συνολικής λειτουργικότητας της συσκευής. Οι πιο σημαντικές απαιτήσεις του μετρητή είναι οι παροχές ρεύματος που προσφέρει ο μικροεπεξεργαστής, ο αριθμός αναλογικών θυρών και ο αριθμός ψηφιακών θυρών. Οι απαιτήσεις της συνολικής λειτουργικότητας της συσκευής είναι κατά κύριο λόγο η συνδεσιμότητα αυτής στο διαδίκτυο είτε με ethernet είτε με Wi-fi και οι επιλογές προγραμματισμού. Τέλος σημαντικό κομμάτι είναι και οι διαθέσιμες βιβλιοθήκες που παρέχονται για κάθε μετρητή και για κάθε μικροεπεξεργαστή. Αυτό είναι πολύ

σημαντικό για την σωστή λειτουργία της συσκευής και για την κάλυψη όλων των αναγκών που προκύπτουν από τα διαφορετικά μέρη που χρησιμοποιούνται κατά την συναρμολόγηση και τον προγραμματισμό αυτής.

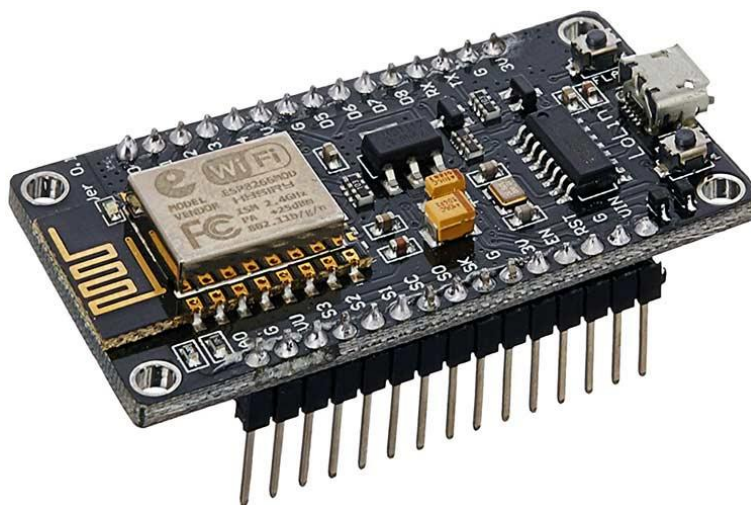
Συνοψίζοντας το hardware της συσκευής είναι το παρακάτω:

- Μικροεπεξεργαστής: NodeMCU V3 Lua WIFI.
- Μετρητής: Particulate Matter Sensor PMS5003 by Plantower.
- DHT11 Temperature and Humidity Sensor Module.
- MTS-203 toggle switch
- Wire Jumper Cables for Arduino

3.2 Hardware της συσκευής

Παρακάτω παρουσιάζονται τα διαφορετικά μέρη hardware της συσκευής που χρησιμοποιήθηκαν.

3.2.1 Μικροεπεξεργαστής: NodeMCU V3 Lua WIFI



Εικόνα 3.1: NodeMCU V3 Lua WIFI

Η επιλογή του συγκεκριμένου μικροεπεξεργαστή έγινε για τις δυνατότητες που παρέχει τόσο στις θήρες που διαθέτει όσο και στις διαθέσιμες βιβλιοθήκες που υπάρχουν. Βασίζεται στο Microcontroller Tensilica 32-bit RISC CPU Xtensa LX106. Είναι συμβατό με αρκετά open source projects όπως το `lua-cjson` και το `SPIFFS`. Διαθέτει μία θύρα `micro USB` που χρησιμοποιείται για την τροφοδοσία και τον προγραμματισμό του μικροεπεξεργαστή. Επίσης διαθέτει και ενσωματωμένο `Wi-fi` το οποίο είναι απαραίτητο για την μεταφορά των δεδομένων σε πραγματικό χρόνο. Το module του `Wi-fi` είναι το `ESP8266`.

3.2.2 Μετρητής: Particulate Matter Sensor PMS5003 by Plantower



Εικόνα 3.2: Particulate Matter Sensor PMS5003 by Plantower.

Η επιλογή του συγκεκριμένου μετρητή έγινε λαμβάνοντας υπόψιν δύο σημαντικά κριτήρια. Το ένα από αυτά είναι η ακρίβεια του και το δεύτερο η σωστή επικοινωνία και μεταφορά δεδομένων με τον επεξεργαστή. Ο μετρητής χρησιμοποιεί την μέθοδο σκέδασης για την μέτρηση της συγκέντρωσης των $PM_{2.5}$. Επίσης η ίδια μέθοδος χρησιμοποιείται και για την μέτρηση των $PM_{1.0}$ και των PM_{10} τα οποία μπορεί να ανιχνεύσει ο μετρητής.

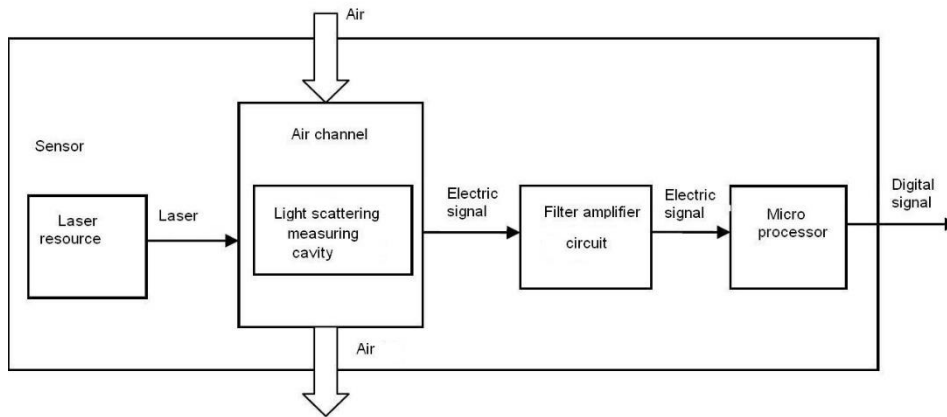
Στον παρακάτω πίνακα παρουσιάζονται τα τεχνικά χαρακτηριστικά του μετρητή όπως αυτά παρέχονται από τον κατασκευαστή.

| | |
|---|--|
| Operating Voltage: 5.0VDC | Total Response Time: ≤ 10 s |
| Range of measurement: 0.3~1.0 ; 1.0~2.5 ; 2.5~10 | Active Current: ≤ 100 μ A |
| Counting Efficiency: 50%@0.3 μ m 98%@ $\geq 0.5\mu$ m | Standby Current: ≤ 200 μ A |
| Effective Range (PM2.5 Standard): 0~500 μ g/ m^3 | Interface Level: L <0.8 @3.3 H >2.7@3.3 V |
| Maximum Range (PM2.5 Standard): ≥ 1000 μ g/ m^3 | Working Temperature Range: -10~+60 $^{\circ}$ C |
| Resolution: 1 μ g/ m^3 | Working Humidity Range: 0~99% |
| Maximum Consistency Error (PM2.5 standard data): $\pm 10\%$ @100~500 μ g/ m^3 $\pm 10\mu$ g/ m^3 @0~100 μ g/ m^3 | Storage Temperature Range: -40~+80 $^{\circ}$ C |
| Standard Volume: 0.1 L | MTTF: ≥ 3 Years |
| Single Response Time: < 1 s | Physical Size: 50×38×21 mm |

Πίνακας 3.1: Τεχνικά χαρακτηριστικά Particulate Matter Sensor PMS5003 by Plantower.

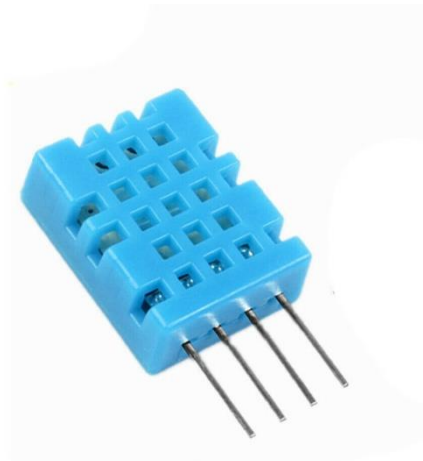
Να σημειωθεί ότι ο μετρητής είναι ψηφιακός και όχι αναλογικός. Αυτό έχει ως αποτέλεσμα όλοι οι υπολογισμοί για την συγκέντρωση των $PM_{1.0}$, $PM_{2.5}$ και PM_{10} στην ατμόσφαιρα γίνονται από τον ίδιο τον μετρητή.

Όπως αναφέρθηκε ο μετρητής χρησιμοποιεί την μέθοδο σκέδασης για τον υπολογισμό της συγκέντρωσης των $PM_{1.0}$, $PM_{2.5}$ και PM_{10} στην ατμόσφαιρα. Στο παρακάτω διάγραμμα παρουσιάζεται η μέθοδος σκέδασης και ο τρόπος χρήσης της από τον μετρητή.



Εικόνα 3.3: Μέθοδος σκέδασης του μετρητή Particulate Matter Sensor PMS5003 by Plantower.

3.2.3 DHT11 Temperature and Humidity Sensor Module



Εικόνα 3.4: DHT11 Temperature and Humidity Sensor Module.

Η επιλογή του συγκεκριμένου module έγινε για την ακρίβεια της μέτρησης της θερμοκρασίας και της υγρασίας που παρέχει και γιατί είναι συμβατό με τον μικροεπεξεργαστή. Η χρήση του είναι αναγκαία διότι ο μετρητής δεν μπορεί να μετρήσει την θερμοκρασία και την υγρασία, συνεπώς ένα εξωτερικό module είναι αναγκαίο. Λειτουργεί από τα 3.5 έως τα 5.5 Volts, το εύρος μέτρησης για την

θερμοκρασία είναι από 0 έως 50 °C, το εύρος μέτρησης για την υγρασία είναι από 20 έως 90% και η ακρίβεια του είναι ± 1 °C και ± 1 % αντίστοιχα.

3.2.4 MTS-203 toggle switch

Για την σωστή λειτουργία της συσκευής είναι απαραίτητη η χρήση ενός διακόπτη ON-OFF-ON. Ο σκοπός του διακόπτη στην συνδεσμολογία είναι για την μεταφορά από την λειτουργία προγραμματισμού στην κανονική λειτουργία της συσκευής.



Εικόνα 3.5: MTS-203 toggle switch.

3.2.5 Wire Jumper Cables for Arduino

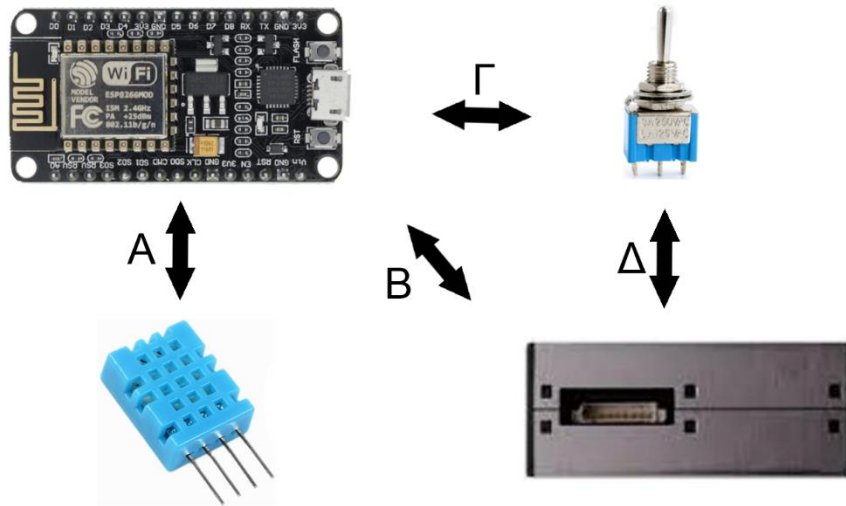
Για την σύνδεση όλων των μερών της συσκευής είναι απαραίτητη η χρήση wires jumper cables.



Εικόνα 3.6: Wire jumper cables for Arduino.

3.3 Συνδεσμολογία της συσκευής GreenYourAir/1497

Παρακάτω παρουσιάζεται ο τρόπος σύνδεσης όλων των μερών του hardware για την σωστή λειτουργία της συσκευής.



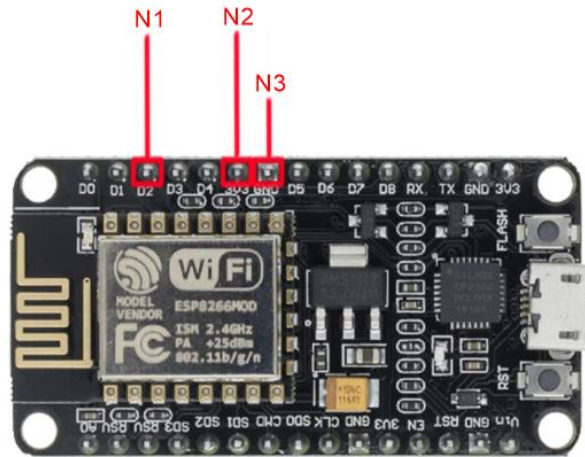
Εικόνα 3.7: Συνδεσμολογία Συσκευής.

Όπως φαίνεται και στην πάνω εικόνα υπάρχουν τέσσερα στάδια για την συνδεσμολογία της συσκευής. Το πρώτο στάδιο είναι η σύνδεση του μικροεπεξεργαστή NodeMCU με τον αισθητήρα θερμοκρασίας και υγρασίας DHT11. Το δεύτερο στάδιο είναι η σύνδεση του μικροεπεξεργαστή NodeMCU με τον μετρητή PMS5003. Το τρίτο στάδιο είναι η σύνδεση του μικροεπεξεργαστή NodeMCU με τον διακόπτη MTS-203. . Το τέταρτο και τελευταίο στάδιο είναι η σύνδεση του μικροεπεξεργαστή NodeMCU με τον διακόπτη MTS-203.

3.3.1 Πρώτο στάδιο συνδεσμολογίας

Στο στάδιο αυτό θα αναλυθεί ο τρόπος σύνδεσης του μικροεπεξεργαστή με τον αισθητήρα θερμοκρασίας και υγρασίας.

- PIN του NodeMCU

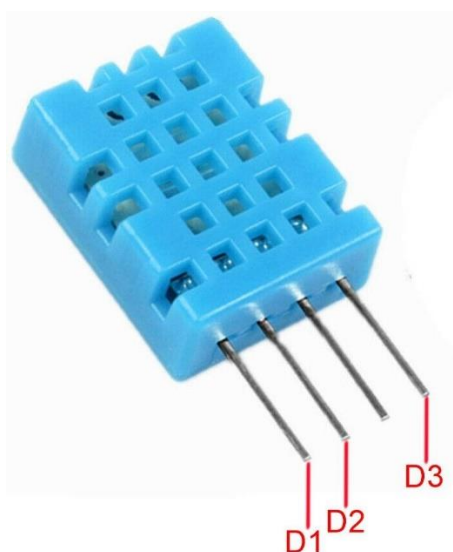


Εικόνα 3.8: PIN του NodeMCU για το πρώτο στάδιο της συνδεσμολογίας.

| Αναγνωριστικό | Επεξήγηση |
|---------------|--------------|
| N1 | D2 (Digital) |
| N2 | 3.3V (Power) |
| N3 | GND (Ground) |

Πίνακας 3.2: Επεξήγηση των PIN του NODEMCU για το πρώτο στάδιο της συνδεσμολογίας.

- PIN του DHT11



Εικόνα 3.9: PIN του DHT11 για το πρώτο στάδιο της συνδεσμολογίας.

| Αναγνωριστικό | Επεξήγηση |
|---------------|-----------|
| D1 | VCC |
| D2 | Data |
| D3 | GND |

Πίνακας 3.3: Επεξήγηση των PIN του DHT11 για το πρώτο στάδιο της συνδεσμολογίας.

Με βάση την Εικόνα 3.8, την Εικόνα 3.9, τον Πίνακα 3.2 και τον Πίνακα 3.3 η σύνδεση των PIN του NodeMCU με τα PIN του DHT11 είναι:

| PIN NodeMCU | PIN DHT11 |
|-------------|-----------|
| N1 | D2 |
| N2 | D1 |
| N3 | D3 |

Πίνακας 3.4: Σύνδεση των PIN του NodeMCU με τα PIN του DHT11.

Σημαντικό σημείο σε αυτήν την διαδικασία είναι η σωστή τοποθέτηση του 3.3V από το NodeMCU στο DHT11. Η λάθος σύνδεση αυτών των PIN μπορεί να προκαλέσει προβλήματα στο DHT11, ακόμα και να το κάψει.

Μετά το πέρας της σύνδεσης των PIN του NodeMCU με τα PIN του DHT11 ολοκληρώνεται το πρώτο στάδιο της συνδεσμολογίας.

3.3.2 Δεύτερο στάδιο συνδεσμολογίας

Στο στάδιο αυτό θα αναλυθεί ο τρόπος σύνδεσης του μικροεπεξεργαστή με τον μετρητή.

- PIN του NodeMCU

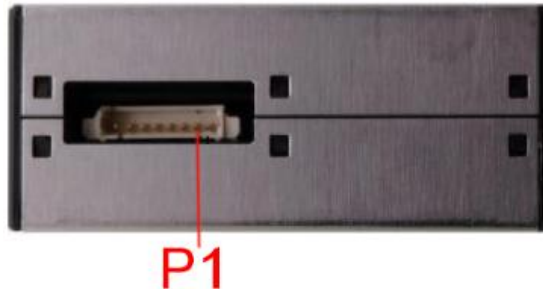


Εικόνα 3.10: PIN του NodeMCU για το δεύτερο στάδιο της συνδεσμολογίας.

| Αναγνωριστικό | Επεξήγηση |
|---------------|--------------|
| N1 | GND (Ground) |

Πίνακας 3.5: Επεξήγηση των PIN του NodeMCU για το δεύτερο στάδιο της συνδεσμολογίας.

- PIN του PMS5003



Εικόνα 3.11: PIN του PMS5003 για το δεύτερο στάδιο της συνδεσμολογίας.

| Αναγνωριστικό | Επεξήγηση |
|---------------|--------------|
| P1 | GND (Ground) |

Πίνακας 3.6: Επεξήγηση των PIN του PMS5003 για το δεύτερο στάδιο της συνδεσμολογίας.

Με βάση την Εικόνα 3.10, την Εικόνα 3.11, τον Πίνακα 3.5 και τον Πίνακα 3.6 η σύνδεση των PIN του μικροεπεξεργαστή με τα PIN του μετρητή είναι:

| PIN NodeMCU | PIN PMS5003 |
|-------------|-------------|
| N1 | P1 |

Πίνακας 3.7: Σύνδεση των PIN του NodeMCU με τα PIN του PMS5003.

Για την σωστή σύνδεση του μετρητή με οποιαδήποτε συσκευή χρησιμοποιείται το καλώδιο σύνδεσης το οποίο είναι συμβατό με τον μετρητή και παρέχεται από τον κατασκευαστή. Το καλώδιο σύνδεσης έρχεται μαζί με τον μετρητή στην ίδια συσκευασία, επομένως δεν χρειάζεται να γίνει αγορά αυτού ξεχωριστά.

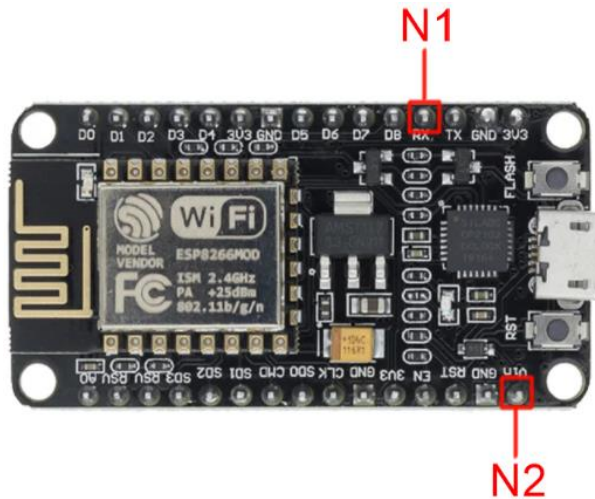
Επειδή τα PIN που διαθέτει ο μετρητής δεν είναι ίδιου μεγέθους με αυτά του μικροεπεξεργαστή δεν είναι δυνατή η χρήση των jumper cables απευθείας. Για να πραγματοποιηθεί σωστά η σύνδεση χρησιμοποιείται το καλώδιο σύνδεσης του μετρητή και ένα jumper cable. Κόβεται το θηλυκό του jumper cable και το θηλυκό του καλωδίου του μετρητή και ενώνετε το jumper cable με το καλώδιο του μετρητή που συνδέεται με το αντίστοιχο PIN που αναφέρεται παραπάνω.

Μετά το πέρας της σύνδεσης των PIN του μικροεπεξεργαστή με τα PIN του μετρητή ολοκληρώνεται το δεύτερο στάδιο της συνδεσμολογίας

3.3.3 Τρίτο στάδιο συνδεσμολογίας

Στο στάδιο αυτό θα αναλυθεί ο τρόπος σύνδεσης του μικροεπεξεργαστή με τον διακόπτη.

- PIN του NodeMCU



Εικόνα 3.12: PIN του NodeMCU για το τρίτο στάδιο της συνδεσμολογίας.

| Αναγνωριστικό | Επεξήγηση |
|---------------|----------------|
| N1 | RX (Digital 0) |
| N2 | Vin (Power) |

Πίνακας 3.8: Επεξήγηση των PIN του NodeMCU για το τρίτο στάδιο της συνδεσμολογίας.

- PIN του MTS-203



Εικόνα 3.13: PIN του MTS-203 για το τρίτο στάδιο της συνδεσμολογίας.

| Αναγνωριστικό | Επεξήγηση |
|---------------|-----------|
| M1 | PIN |
| M2 | PIN |

Πίνακας 3.9: Επεξήγηση των PIN του MTS-203 για το τρίτο στάδιο της συνδεσμολογίας.

Με βάση την Εικόνα 3.12, την Εικόνα 3.13, τον Πίνακα 3.8 και τον Πίνακα 3.9 η σύνδεση των PIN του μικροεπεξεργαστή με τα PIN του διακόπτη είναι:

| PIN NodeMCU | PIN MTS-203 |
|-------------|-------------|
| N1 | M2 |
| N2 | M1 |

Πίνακας 3.10: Σύνδεση των PIN του Arduino με τα PIN του Shield.

Μετά το πέρας της σύνδεσης των PIN του μικροεπεξεργαστή με τα PIN του διακόπτη ολοκληρώνεται το τρίτο στάδιο της συνδεσμολογίας.

3.3.4 Τέταρτο στάδιο συνδεσμολογίας

Στο στάδιο αυτό θα αναλυθεί ο τρόπος σύνδεσης του διακόπτη με τον μετρητή.

- PIN του MTS-203

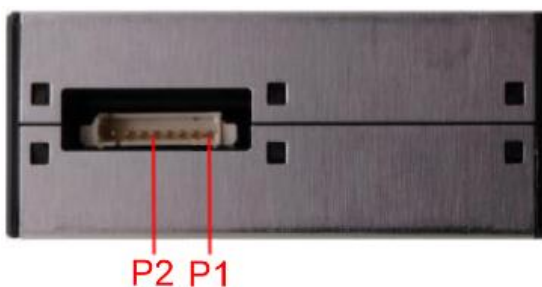


Εικόνα 3.14: PIN του MTS-203 για το τέταρτο στάδιο της συνδεσμολογίας.

| Αναγνωριστικό | Επεξήγηση |
|---------------|-----------|
| M1 | PIN |
| M2 | PIN |

Πίνακας 3.11: Επεξήγηση των PIN του MTS-203 για το τέταρτο στάδιο της συνδεσμολογίας.

- PIN του PMS5003



Εικόνα 3.15: PIN του PMS5003 για το τέταρτο στάδιο της συνδεσμολογίας.

| Αναγνωριστικό | Επεξήγηση |
|---------------|-----------------------|
| P1 | VCC (Power) |
| P2 | TXD/I2C_SDL (Digital) |

Πίνακας 3.12: Επεξήγηση των PIN του PMS5003 για το τέταρτο στάδιο της συνδεσμολογίας.

Με βάση την Εικόνα 3.14, την Εικόνα 3.15, τον Πίνακα 3.11 και τον Πίνακα 3.12 η σύνδεση των PIN του διακόπτη με τα PIN του μετρητή είναι:

| PIN MTS-203 | PIN PMS5003 |
|-------------|-------------|
| M1 | P1 |
| M2 | P2 |

Πίνακας 3.13: Σύνδεση των PIN του MTS-203 με τα PIN του PMS5003.

Μετά το πέρας της σύνδεσης των PIN του διακόπτη με τα PIN του μετρητή ολοκληρώνεται το τέταρτο και τελευταίο στάδιο της συνδεσμολογίας.

3.4 Προγραμματισμός της συσκευής GreenYourAir/1497 και αποστολή δεδομένων στο cloud

Ένα από τα πιο σημαντικά στάδια όλης της διαδικασίας της ανάπτυξης της συσκευής είναι ο προγραμματισμός της. Ο προγραμματισμός της συσκευής έχει μεγάλο αντίκτυπο στην ορθή λειτουργία της και στην ποιότητα των μετρήσεων. Η μεταφορά δεδομένων μέσω του διαδικτύου στο cloud είναι ένα από τα βασικά χαρακτηριστικά της συσκευής. Η διαδικασία αυτή πραγματοποιείται σε πραγματικό χρόνο. Παρακάτω παρουσιάζεται η διαδικασία προγραμματισμού της συσκευής και αναλύονται τα στάδια που απαιτούνται για την λειτουργία της σε πραγματικές συνθήκες και η αποστολή δεδομένων στο cloud.

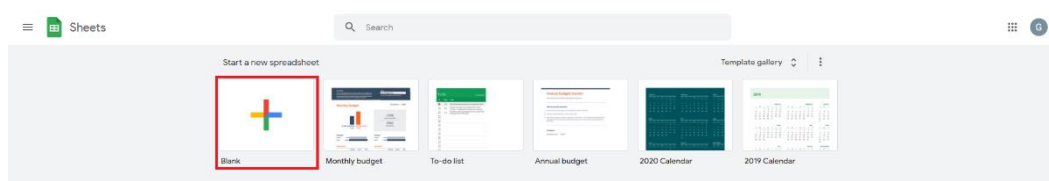
3.4.1 Προετοιμασία του cloud για την μεταφορά δεδομένων

Το στάδιο αυτό περιλαμβάνει την δημιουργία μιας βάσης δεδομένων στο cloud με την χρήση των εργαλείων της Google. Τα εργαλεία της Google που χρησιμοποιούνται είναι το Google Sheets και το Google Apps Script. Το Google Sheets παίζει μίας βάσης δεδομένων για την αποθήκευση και την εύκολη προβολή και επεξεργασία των μετρήσεων. Το Google Apps Script είναι το API που χρησιμοποιείται για την αποστολή των δεδομένων και την αποθήκευση αυτών στο Google Sheets.

Παρακάτω παρουσιάζεται η διαδικασία δημιουργίας ενός Google Sheet, ενός Google Apps Script και η σύνδεση αυτών των δύο για την αποστολή και την αποθήκευση των δεδομένων.

- Βήμα 1: Δημιουργία ενός κενού εγγράφου Google Sheets.

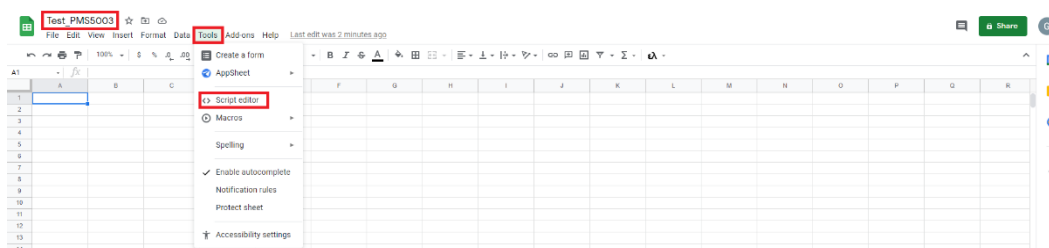
Σε έναν browser επισκεπτόμαστε την ιστοσελίδα: <https://docs.google.com/spreadsheets/> και συνδεόμαστε με τον λογαριασμό μας στην google ή δημιουργούμε έναν νέο. Στην συνέχεια επιλέγουμε την δημιουργία ενός κενού εγγράφου.



Εικόνα 3.16: Δημιουργία ενός κενού εγγράφου στο Google Sheets.

- Βήμα 2: Προγραμματισμός API για την μεταφορά και αποθήκευση των δεδομένων.

Μεταφερόμαστε στο κενό έγγραφο που δημιουργήσαμε. Δίνουμε ένα όνομα στο έγγραφο και στην συνέχεια πηγαίνουμε στο μενού στο "Tools" και στο dropdown menu που εμφανίζεται επιλέγουμε το "Script editor". Στην συνέχεια θα μεταφερθούμε στο Apps Script για να γράψουμε τον κώδικα που απαιτείται για την διαδικασία αποστολής και αποθήκευσης δεδομένων από την συσκευή GreenYourAir/1497.



Εικόνα 3.17: Μεταφορά στο Google Apps Script από το Google Sheets.

- Βήμα 3: Κώδικας για την μεταφορά και την αποθήκευση των δεδομένων από την συσκευή GreenYourAir/1497.

Στο Google Apps Script γράφουμε τον κώδικα που απαιτείται. Παρακάτω παρουσιάζεται το περιβάλλον του Apps Script και αναλύεται ο κώδικας που χρησιμοποιείται.

```

1 function doGet(e) {
2   Logger.log( JSON.stringify(e) ); // view parameters
3   var result = 'Ok'; // assume success
4   if (e.parameter == 'undefined') {
5     result = 'No Parameters';
6   }
7   else {
8     var sheet_id = '1Uhw4fDEcVZOxqsIB8VfswECdH066F-SbO3SOkzDVbcY'; // Spreadsheet ID
9     var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet(); // get Active sheet
10    var newRow = sheet.getLastRow() + 1;
11    var rowData = [];
12    rowData[0] = new Date(); // Timestamp in column A
13    for (var param in e.parameter) {
14      Logger.log('In for loop, param=' + param);
15      var value = stripQuotes(e.parameter[param]);
16      Logger.log(param + ':' + e.parameter[param]);
17      switch (param) {
18        case 'SensorName': //Parameter
19          rowData[1] = value; //Value in column B
20          result += 'Written on column B';
21          break;
22        case 'PM2.5': //Parameter
23          rowData[2] = value; //Value in column C
24          result += 'Written on column C';
25          break;
26        case 'temperature': //Parameter
27          rowData[3] = value; //Value in column D
28          result += 'Written on column D';
29          break;
30        case 'Humidity': //Parameter
31          rowData[4] = value; //Value in column E
32          result += 'Written on column E';
33          break;
34        case 'PM10': //Parameter
35          rowData[5] = value; //Value in column F
36          result += 'Written on column F';
37          break;
38        case 'PM10': //Parameter
39          rowData[6] = value; //Value in column G
40          result += 'Written on column G';
41          break;
42        default:
43          result = 'unsupported parameter';
44      }
45    }
46    Logger.log( JSON.stringify(rowData) );
47    // Write new row below

```

Εικόνα 3.18: Περιβάλλον του Google Apps Script.

Παρακάτω αναλύεται ο κώδικας που χρησιμοποιείται για τον προγραμματισμό του API. Η γλώσσα προγραμματισμού του κώδικα είναι η JavaScript και τα δεδομένα μεταφέρονται σε μορφή JSON. Στον κώδικα δηλώνουμε τις στήλες που απαιτούνται για την αποθήκευση των δεδομένων καθώς και τα ονόματα αυτών. Ο αριθμός των γραμμών είναι δυναμικός και αυξάνεται με κάθε νέα καταχώρηση. Επίσης δηλώνουμε και τον χρόνο στον οποίο καταχωρήθηκε κάθε γραμμή δεδομένων.

```

function doGet(e) {
  Logger.log( JSON.stringify(e) ); // view parameters
  var result = 'Ok'; // assume success
  if (e.parameter == 'undefined') {
    result = 'No Parameters';
  }
  else {
    var sheet_id = '1Uhw4fDEcVZOxqsIB8VfswECdH066F-SbO3SOkzDVbcY'; //
Spreadsheet ID
    var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet(); //
get Active sheet
    var newRow = sheet.getLastRow() + 1;
    var rowData = [];
    rowData[0] = new Date(); //
Timestamp in column A
    for (var param in e.parameter) {
      Logger.log('In for loop, param=' + param);
      var value = stripQuotes(e.parameter[param]);
      Logger.log(param + ':' + e.parameter[param]);
      switch (param) {
        case 'SensorName': //Parameter
          rowData[1] = value; //Value in column B

```

```

        result = 'Written on column B';
        break;
    case 'PM2.5': //Parameter
        rowData[2] = value; //Value in column C
        result += ' ,Written on column C';
        break;
    case 'Temperature': //Parameter
        rowData[3] = value; //Value in column D
        result += ' ,Written on column D';
        break;
    case 'Humidity': //Parameter
        rowData[4] = value; //Value in column E
        result += ' ,Written on column E';
        break;
    case 'PM1.0': //Parameter
        rowData[5] = value; //Value in column F
        result += ' ,Written on column F';
        break;
    case 'PM10': //Parameter
        rowData[6] = value; //Value in column G
        result += ' ,Written on column G';
        break;
    default:
        result = "unsupported parameter";
    }
}
Logger.log(JSON.stringify(rowData));
// Write new row below
var newRange = sheet.getRange(newRow, 1, 1, rowData.length);
newRange.setValues([rowData]);
}
// Return result of operation
return ContentService.createTextOutput(result);
}
/**
 * Remove leading and trailing single or double quotes
 */
function stripQuotes( value ) {
    return value.replace(/^[\'"]|[\']$/g, "");
}
}

```

Κώδικας 3.1: Κώδικας για την δημιουργία του API μέσω Google Apps Script για την μεταφορά και την αποθήκευση των δεδομένων στο Google Sheets.

1. Πρώτο μέρος του κώδικα.

Στο πρώτο μέρος του κώδικα γίνεται η σύνδεση μεταξύ του Google Sheet στο οποίο θα γίνεται η καταγραφή των δεδομένων. Η σύνδεση αυτή πραγματοποιείται ορίζοντας το ID του Google Sheet το οποίο το βρίσκουμε στο URL του εγγράφου Google Sheet. Στο σημείο αυτό βρίσκουμε και την τελευταία γραμμή ώστε να γίνει η καταγραφή των δεδομένων στην επόμενη. Επίσης στο σημείο αυτό δημιουργούμε τον πίνακα με όνομα "rowData" τον οποίο χρησιμοποιούμε στην συνέχεια για την καταγραφή των δεδομένων στην νέα γραμμή. Τέλος ορίζεται το timestamp της καταγραφής.

```

var sheet_id = '1Uhw4fDEcVZOxqsIB8VfswECdH066F-SbO3SOkzDVbcY';
// Spreadsheet ID
var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet();
// get Active sheet
var newRow = sheet.getLastRow() + 1;
var rowData = [];

```

```
rowData[0] = new Date(); // Timestamp in column A
```

Κώδικας 3.2: Πρώτο μέρος του κώδικα για την μεταφορά και την αποθήκευση των δεδομένων στο Google Sheet.

2. Δεύτερο μέρος του κώδικα.

Στο δεύτερο μέρος του κώδικα χρησιμοποιούμε το response του API, δηλαδή τα δεδομένα που έχει στείλει η συσκευή. Τα διαχωρίζουμε με βάση τις στήλες που ορίζουμε στον κώδικα και στην συνέχεια καταγράφουμε τα δεδομένα στο Google Sheets. Οι σύνδεση των δεδομένων με κάθε στήλη ορίζεται στατικά. Στον κώδικα η στήλη στην οποία θα καταγραφούν τα PM_{2.5} είναι η τρίτη στήλη και στον πίνακα "rowData" είναι αντίστοιχα η τρίτη στήλη. Στην διαδικασία αυτή είναι σημαντικό να υπάρχει πλήρης αντιστοίχιση στον τρόπο που χρησιμοποιεί η συσκευή το API για να στείλει δεδομένα και στον τρόπο που χρησιμοποιείται για την καταγραφή αυτών στο Google Sheet.

```
for (var param in e.parameter) {
    Logger.log('In for loop, param=' + param);
    var value = stripQuotes(e.parameter[param]);
    Logger.log(param + ':' + e.parameter[param]);
    switch (param) {
        case 'SensorName': //Parameter
            rowData[1] = value; //Value in column B
            result = 'Written on column B';
            break;
        case 'PM2.5': //Parameter
            rowData[2] = value; //Value in column C
            result += ' ,Written on column C';
            break;
        case 'Temperature': //Parameter
            rowData[3] = value; //Value in column D
            result += ' ,Written on column D';
            break;
        case 'Humidity': //Parameter
            rowData[4] = value; //Value in column E
            result += ' ,Written on column E';
            break;
        case 'PM1.0': //Parameter
            rowData[5] = value; //Value in column F
            result += ' ,Written on column F';
            break;
        case 'PM10': //Parameter
            rowData[6] = value; //Value in column G
            result += ' ,Written on column G';
            break;
        default:
            result = "unsupported parameter";
    }
}
Logger.log(JSON.stringify(rowData));
// Write new row below
var newRange = sheet.getRange(newRow, 1, 1, rowData.length);
newRange.setValues([rowData]);
}
// Return result of operation
return ContentService.createTextOutput(result);
}
/**
 * Remove leading and trailing single or double quotes
```

```

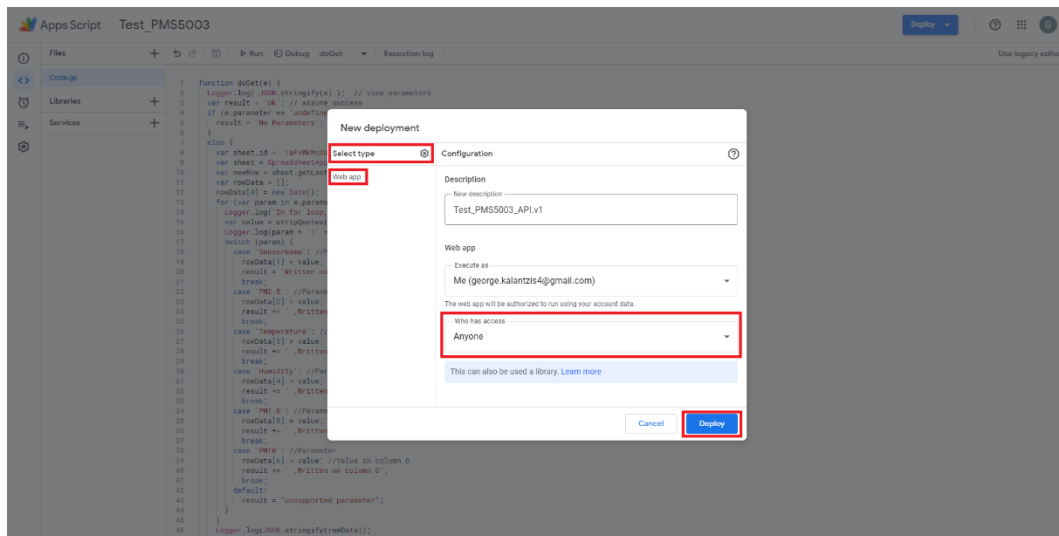
*/
function stripQuotes( value ) {
  return value.replace(/^[\'"]|[\']$/g, "");
}

```

Κώδικας 3.3: Δεύτερο μέρος του κώδικα για την μεταφορά και την αποθήκευση των δεδομένων στο Google Sheet.

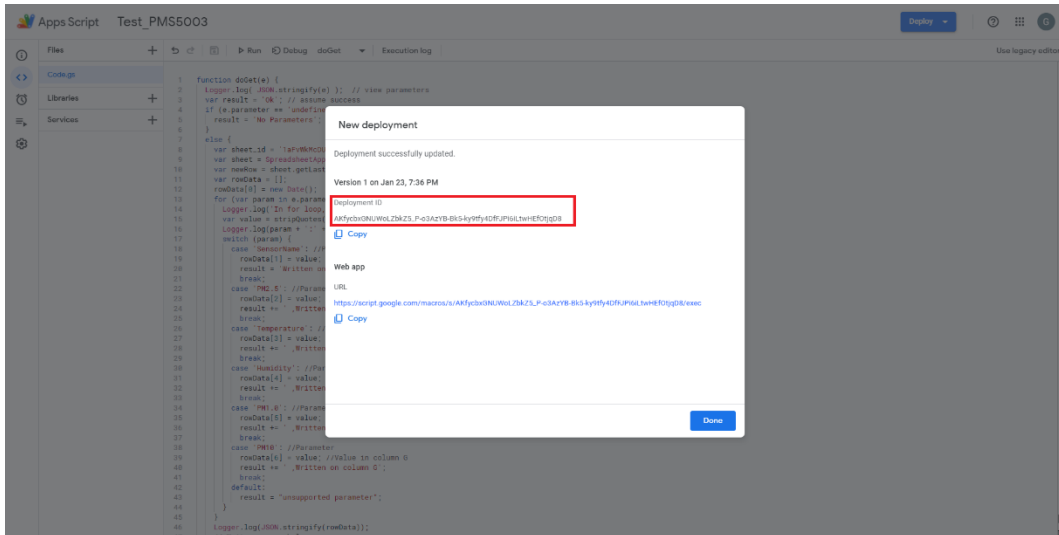
- Βήμα 4: Δημοσίευση του API.

Το τελευταίο βήμα της διαδικασίας είναι η δημοσίευση του API ώστε να μπορέσουμε να το χρησιμοποιήσουμε στον κώδικα της συσκευής για την αποστολή και την αποθήκευση των δεδομένων. Επιλέγουμε το “Deploy” που βρίσκεται πάνω δεξιά στο περιβάλλον του Apps Script. Στην συνέχεια επιλέγουμε να το δημοσιεύσουμε ως “Webapp”. Τέλος δίνουμε πρόσβαση σε όλους του πιθανούς χρήστες (στην περίπτωση αυτή είναι η συσκευή) επιλέγοντας το “Anyone” στο πεδίο “Who has access”. Αυτό το βήμα είναι σημαντικό διότι αν δεν δώσουμε πρόσβαση σε όλους τους πιθανούς χρήστες τότε η συσκευή δεν θα μπορέσει να στείλει δεδομένα. Το “Deployment ID” που δημιουργείται μπορεί να χρησιμοποιηθεί από οποιονδήποτε χωρίς να χρειάζεται ταυτοποίηση ο λογαριασμός του, αλλά μόνο εφόσον του το δώσουμε. Σε καμία περίπτωση δεν μπορεί κάποιος να έχει πρόσβαση στα δεδομένα ή να στείλει δεδομένα αν δεν του δώσουμε το “Deployment ID”, ασχέτως που επιλέξαμε το “Anyone”.



Εικόνα 3.19: Δημοσίευση του API για την αποστολή και την καταγραφή των δεδομένων στο Google Sheets.

Μετά το πέρας της διαδικασίας επιλέγουμε το “Deploy”. Με τον τρόπο αυτό δημιουργούμε το “Deployment ID” το οποίο θα το χρησιμοποιήσουμε στον κώδικα της συσκευής για την αποστολή και την αποθήκευση των δεδομένων στο Google Sheets.



Εικόνα 3.20: Development ID για την αποστολή και την αποθήκευση των δεδομένων στο Google Sheets.

3.4.2 Κώδικας της συσκευής

Το στάδιο αυτό περιλαμβάνει τον κώδικα του μικροεπεξεργαστή. Η γλώσσα προγραμματισμού που χρησιμοποιείται είναι η C++. Ο κώδικας χωρίζεται στο κομμάτι που πραγματοποιούνται οι μετρήσεις και στο κομμάτι στο οποίο στέλνονται οι μετρήσεις στο cloud. Για να τρέξει σωστά ο κώδικας απαιτείται η χρήση συγκεκριμένων βιβλιοθηκών. Επίσης ο κώδικας χωρίζεται σε δύο void. Η μία χρησιμοποιείται κατά την εκκίνηση του μικροεπεξεργαστή και η δεύτερη χρησιμοποιείται για τις μετρήσεις και την αποστολή αυτών στον cloud.

Παρακάτω παρουσιάζεται και αναλύεται ο κώδικας.

```
#include <ESP8266WiFi.h> //ESP8266 WiFi Library
#include "HTTPSRedirect.h"
#include "PMS.h" //PMS Library
#include "DHT.h" //DHT11 Library

#define DHTTYPE DHT11 // DHT 11

/*Put your SSID & Password*/
const char* ssid = "g_k_4"; // Enter SSID here
const char* password = "@123456@789@"; //Enter Password here

PMS pms(Serial);
PMS::DATA data;

// DHT Sensor
uint8_t DHTPin = D2; //DHT Pin Sensor

// Initialize DHT sensor.
DHT dht(DHTPin, DHTTYPE);

float Temperature;
float Humidity;

//String's for Google Docs
```



```

float sumpm25, sumtemp, sumhum, sumpm01, sumpm10, curpm25, curtemp,
curhum, curpm01, curpm10, avepm25, avetemp, avehum, avepm01, avepm10;
int counter;

String sheetpm25 = "";
String sheettemp = "";
String sheethum = "";
String sheetpm01 = "";
String sheetpm10 = "";

String devid = "Test_PMS5003"; //Device Name

const char* host = "script.google.com";
const int httpsPort = 443;
WiFiClientSecure client;
const char* fingerprint = "";
String Devscript_ID = "AKfycbxGNUWoLZbkZ5_P-o3AzYB-Bk5-
ky9tffy4DfFJPi6iLtwHEfOtjqD8"; // Replace development id

void setup()
{
  Serial.begin(9600); // GPIO1, GPIO3 (TX/RX pin on ESP-12E Development
Board)

//CODE FOR DHT SENSOR
  pinMode(DHTPin, INPUT);

  dht.begin();

//CODE FOR WIFI
  Serial.println("Connecting to ");
  Serial.println(ssid);

  //connect to your local wi-fi network
  WiFi.begin(ssid, password);

  //check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected..!");
  Serial.print("Got IP: "); Serial.println(WiFi.localIP());
}

void loop()
{
  if (pms.read(data)){

    counter = 0;
    sumpm25 = 0;
    sumtemp = 0;
    sumhum = 0;
    sumpm01 = 0;
    sumpm10 = 0;

    for (int i=0; i<60; i++){

      curpm25 = (data.PM_AE_UG_2_5);
      curtemp = dht.readTemperature();
      curhum = dht.readHumidity();
      curpm01 = (data.PM_AE_UG_1_0);
      curpm10 = (data.PM_AE_UG_10_0);

      sumpm25 += curpm25;
      sumtemp += curtemp;

```

```

    sumhum += curhum;
    sumpm01 += curpm01;
    sumpm10 += curpm10;

    counter ++;

    delay(3000);
}

avepm25 = sumpm25/counter;
avetemp = sumtemp/counter;
avehum = sumhum/counter;
avepm01 = sumpm01/counter;
avepm10 = sumpm10/counter;

//Insert metrics into "sheet" variables
sheetpm25 = String(avepm25);
sheettemp = String(avetemp);
sheethum = String(avehum);
sheetpm01 = String(avepm01);
sheetpm10 = String(avepm10);

client.setInsecure();
Serial.print("connecting to ");
Serial.println(host);
if (!client.connect(host, httpsPort)) {
    Serial.println("connection failed");
    return;
}

if (client.verify(fingerprint, host)) {
    Serial.println("certificate matches");
} else {
    Serial.println("certificate doesn't match");
}

String url = "/macros/s/" + Devscript_ID + "/exec?SensorName=" + devid
+ "&PM2.5=" + sheetpm25 + "&Temperature=" + sheettemp + "&Humidity=" +
sheethum + "&PM1.0=" + sheetpm01 + "&PM10=" + sheetpm10;
Serial.print("requesting URL: ");
Serial.println(url);

client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "User-Agent: BuildFailureDetectorESP8266\r\n" +
    "Connection: close\r\n\r\n");

Serial.println("request sent");
while (client.connected()) {
    String line = client.readStringUntil('\n');
    if (line == "\r") {
        Serial.println("headers received");
        break;
    }
}

String line = client.readStringUntil('\n');
if (line.startsWith("{\"state\":\"success\"}") {
    Serial.println("esp8266/Arduino CI successful!");
} else {
    Serial.println("esp8266/Arduino CI has failed");
}
Serial.println("reply was:");
Serial.println("=====");
Serial.println(line);
Serial.println("=====");
Serial.println("closing connection");

```

```
}  
}
```

Κώδικας 3.4: Κώδικας της συσκευής GreenYourAir/1497.

- Μέρος A του κώδικα της συσκευής.

Στο μέρος A του κώδικα γίνεται η εισαγωγή των απαραίτητων βιβλιοθηκών και ο ορισμός των μεταβλητών που απαιτούνται για τα υπόλοιπα μέρη του κώδικα.

Βιβλιοθήκες:

1. ESP8266Wifi.h
Χρησιμοποιείται για την σωστή λειτουργία του Wi-fi, για την σύνδεση της συσκευής στο Wi-fi καθώς και για την αποστολή των δεδομένων.
2. HTTPSRedirect.h
Χρησιμοποιείται για την αποστολή των δεδομένων.
3. PMS.h
Είναι απαραίτητη για την σωστή λειτουργία του μετρητή PMS5003.
4. DHT.h
Είναι απαραίτητη για την σωστή λειτουργία του αισθητήρα θερμοκρασίας και υγρασίας DHT11.

Μεταβλητές:

1. Οι τρεις κύριες μεταβλητές που χρησιμοποιούνται είναι:
 - a. Ssid = "WifiNetwork"
Προσδιορίζει όνομα του δικτύου στο οποίο θα συνδεθεί η συσκευή.
 - b. Password = "*****"
Προσδιορίζει τον κωδικό του.
 - c. DHTPin = D2;
Προσδιορίζει το Pin του NodeMCU στο οποίο έχει συνδεθεί ο αισθητήρας μέτρησης θερμοκρασίας και υγρασίας DHT11.
 - d. devid = "Test_PMS5003"
Προσδιορίζει το όνομα της συσκευής το οποίο θα εμφανίζεται στο Google Sheets.
 - e. Devscript_ID = "AKfycbxGNUWoLZbkZ5_P-o3AzYB-Bk5-ky9tfy4DfFJPi6iLtwHEfOtiqD8"
Προσδιορίζει το "Deployment ID" για την αποστολή των δεδομένων σε συγκεκριμένο Google Sheet.
2. Όλες οι υπόλοιπες μεταβλητές είναι βοηθητικές και χρησιμοποιούνται για την προσωρινή αποθήκευση των μετρήσεων και την εξαγωγή του μέσου όρου για την αποστολή στο cloud.

```
#include <ESP8266WiFi.h> //ESP8266 WiFi Library  
#include "HTTPSRedirect.h"  
#include "PMS.h" //PMS Library  
#include "DHT.h" //DHT11 Library
```

```

#define DHTTYPE DHT11 // DHT 11

/*Put your SSID & Password*/
const char* ssid = "g_k_4"; // Enter SSID here
const char* password = "@123456@789@"; //Enter Password here

PMS pms(Serial);
PMS::DATA data;

// DHT Sensor
uint8_t DHTPin = D2; //DHT Pin Sensor

// Initialize DHT sensor.
DHT dht(DHTPin, DHTTYPE);

float Temperature;
float Humidity;

//String's for Google Docs
float sumpm25, sumtemp, sumhum, sumpm01, sumpm10, curpm25, curtemp,
curhum, curpm01, curpm10, avepm25, avetemp, avehum, avepm01, avepm10;
int counter;

String sheetpm25 = "";
String sheettemp = "";
String sheethum = "";
String sheetpm01 = "";
String sheetpm10 = "";

String devid = "Test_PMS5003"; //Device Name

const char* host = "script.google.com";
const int httpsPort = 443;
WiFiClientSecure client;
const char* fingerprint = "";
String Devscript_ID = "AKfyxcbxGNUWoLZbkZ5_P-o3AzYB-Bk5-
ky9tfy4DfFJpi6iLtwHEfOtjqD8"; // Replace development id

```

Κώδικας 3.5: Μέρος Α του κώδικα της συσκευής.

- Μέρος Β του κώδικα της συσκευής.

Στο μέρος Β του κώδικα βρίσκεται το setup. Στο μέρος αυτό καλούνται οι συναρτήσεις που είναι απαραίτητες για την λειτουργία του κώδικα. Ορίζονται οι μεταβλητές για των διαφορετικών modules της συσκευής καθώς γίνεται και η σύνδεση στο δίκτυο Wi-fi που έχει οριστεί.

```

void setup()
{
  Serial.begin(9600); // GPIO1, GPIO3 (TX/RX pin on ESP-12E
  Development Board)

  //CODE FOR DHT SENSOR
  pinMode(DHTPin, INPUT);

  dht.begin();

  //CODE FOR WIFI
  Serial.println("Connecting to ");
  Serial.println(ssid);

```

```

//connect to your local wi-fi network
WiFi.begin(ssid, password);

//check wi-fi is connected to wi-fi network
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected..!");
Serial.print("Got IP: "); Serial.println(WiFi.localIP());
}

```

Κώδικας 3.6: Μέρος Β του κώδικα της συσκευής.

- Μέρος Γ του κώδικα της συσκευής.

Στο μέρος Γ του κώδικα βρίσκεται το loop. Στο μέρος αυτό πραγματοποιείται η διαδικασία των μετρήσεων και η αποστολή των δεδομένων στο cloud. Χρησιμοποιούνται όλες οι βοηθητικές μεταβλητές που ορίστηκαν στο μέρος Α για την προσωρινή αποθήκευση και εξαγωγή του μέσου όρου 60 μετρήσεων που κάθε μία από αυτές πραγματοποιείται σε διάστημα 3 δευτερολέπτων.

Στο μέρος αυτό γίνεται αποστολή των μετρήσεων στο cloud. Για κάθε έναν μετρητή ξεχωριστά πρέπει να δημιουργηθεί μία βάση δεδομένων με την χρήση των εργαλείων της Google όπως παρουσιάστηκε στην ενότητα «2.4.1) Προετοιμασία του cloud για την μεταφορά δεδομένων».

```

void loop()
{
  if (pms.read(data)){

    counter = 0;
    sumpm25 = 0;
    sumtemp = 0;
    sumhum = 0;
    sumpm01 = 0;
    sumpm10 = 0;

    for (int i=0; i<60; i++){

      curpm25 = (data.PM_AE_UG_2_5);
      curtemp = dht.readTemperature();
      curhum = dht.readHumidity();
      curpm01 = (data.PM_AE_UG_1_0);
      curpm10 = (data.PM_AE_UG_10_0);

      sumpm25 += curpm25;
      sumtemp += curtemp;
      sumhum += curhum;
      sumpm01 += curpm01;
      sumpm10 += curpm10;

      counter ++;

      delay(3000);
    }

    avepm25 = sumpm25/counter;
    avetemp = sumtemp/counter;
    avehum = sumhum/counter;
  }
}

```

```

avepm01 = sumpm01/counter;
avepm10 = sumpm10/counter;

//Insert metrics into "sheet" variables
sheetpm25 = String(avepm25);
sheettemp = String(avetemp);
sheethum = String(avehum);
sheetpm01 = String(avepm01);
sheetpm10 = String(avepm10);

client.setInsecure();
Serial.print("connecting to ");
Serial.println(host);
if (!client.connect(host, httpsPort)) {
    Serial.println("connection failed");
    return;
}

if (client.verify(fingerprint, host)) {
    Serial.println("certificate matches");
} else {
    Serial.println("certificate doesn't match");
}

String url = "/macros/s/" + Devscript_ID + "/exec?SensorName=" +
devid + "&PM2.5=" + sheetpm25 + "&Temperature=" + sheettemp +
"&Humidity=" + sheethum + "&PM1.0=" + sheetpm01 + "&PM10=" +
sheetpm10;
Serial.print("requesting URL: ");
Serial.println(url);

client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "User-Agent: BuildFailureDetectorESP8266\r\n" +
    "Connection: close\r\n\r\n");

Serial.println("request sent");
while (client.connected()) {
    String line = client.readStringUntil('\n');
    if (line == "\r") {
        Serial.println("headers received");
        break;
    }
}

String line = client.readStringUntil('\n');
if (line.startsWith("{\"state\":\"success\"}") {
    Serial.println("esp8266/Arduino CI successfull!");
} else {
    Serial.println("esp8266/Arduino CI has failed");
}
Serial.println("reply was:");
Serial.println("=====");
Serial.println(line);
Serial.println("=====");
Serial.println("closing connection");
}
}

```

Κώδικας 3.7: Μέρος Γ του κώδικα της συσκευής.

Η μονάδα μέτρησης των PM είναι $\mu\text{g}/\text{m}^3$, της θερμοκρασίας σε $^{\circ}\text{C}$ και της υγρασίας σε ποσοστό.

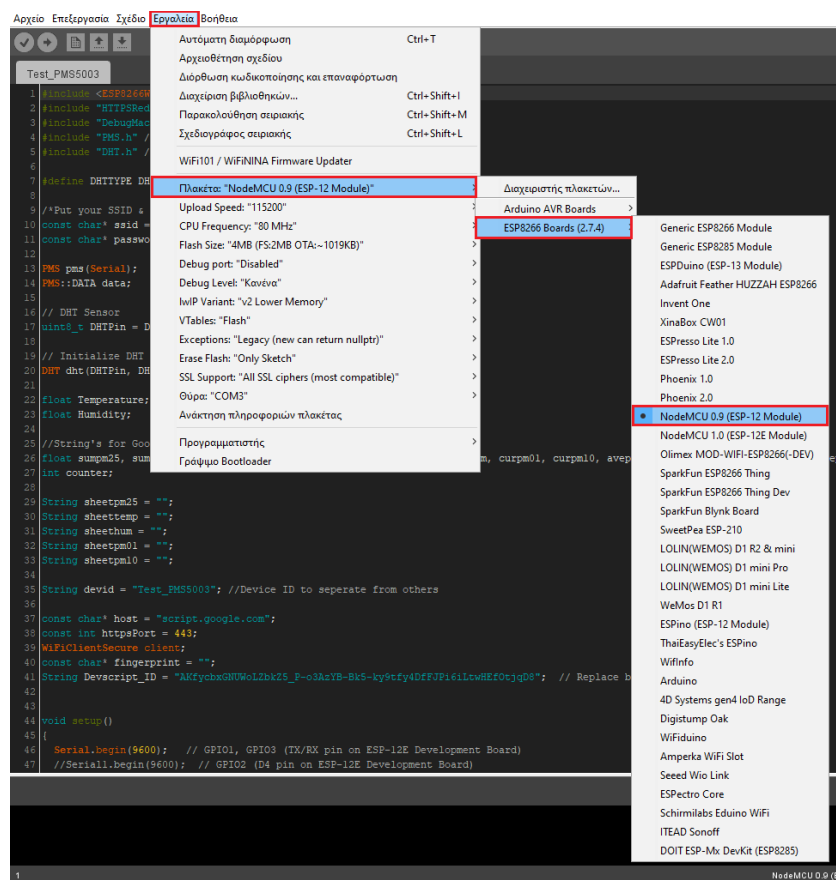
Στο σημείο αυτό ολοκληρώνεται η ανάλυση του κώδικα της συσκευής. Με τον παραπάνω κώδικα η συσκευή πραγματοποιεί μία μέτρηση κάθε 3 δευτερόλεπτα και κάθε 60 μετρήσεις υπολογίζεται ο μέσος όρος αυτών και γίνεται αποστολή των δεδομένων στο cloud. Συνεπώς η συσκευή στέλνει μία μέτρηση κάθε 3 λεπτά στο cloud. Σημαντικό κομμάτι του κώδικα είναι η σωστή αντιστοίχιση των μεταβλητών των τιμών των μετρήσεων με το API ώστε να γίνει σωστή αντιστοίχιση με τις στήλες που έχουν οριστεί στο API για κάθε μία από τις τιμές απάντησης.

3.4.3 Προγραμματισμός της συσκευής

Μετά το πέρας της συνδεσμολογίας του hardware της συσκευής και την ανάπτυξη του κώδικα ακολουθεί ο προγραμματισμός της συσκευής. Για τον προγραμματισμό της συσκευής χρησιμοποιείται η έκδοση 1.8.13 του Arduino IDE. Παλαιότερες εκδόσεις ενδέχεται να μην υποστηρίζουν τις βιβλιοθήκες που χρησιμοποιούνται από τον κώδικα όπως και τον μικροεπεξεργαστή NodeMCu. Ο τρόπος προγραμματισμού της συσκευής είναι με την σύνδεση της συσκευής στον υπολογιστή χρησιμοποιώντας ένα καλώδιο micro USB.

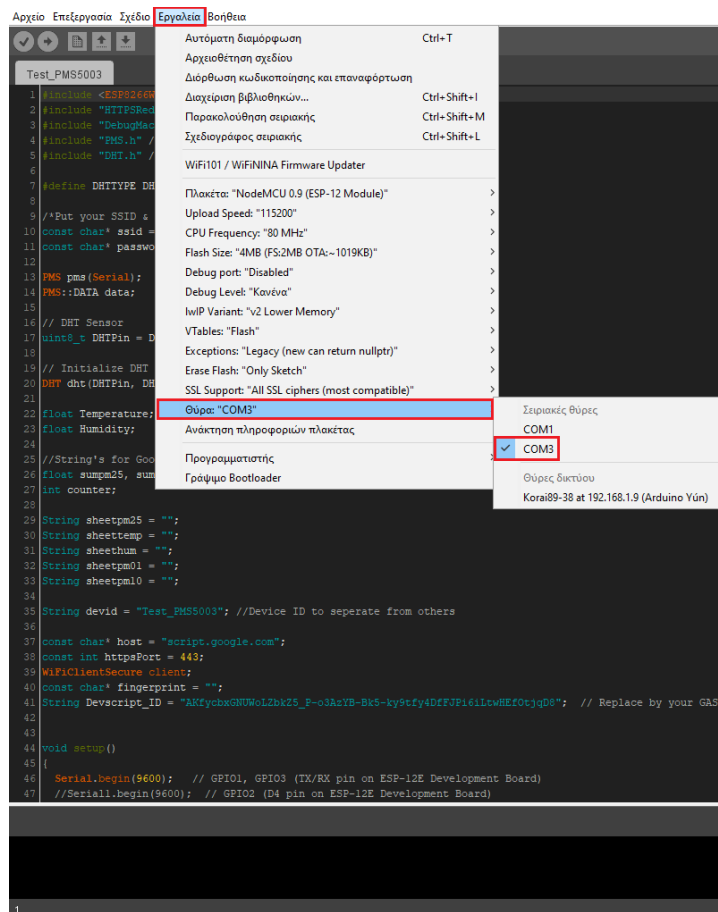
Προγραμματισμός της συσκευής GreenYourAir/1497:

1. Ανοίγουμε το Arduino IDE με τον κώδικα της συσκευής.
2. Στο μενού επιλέγουμε Εργαλεία → Πλακέτα → NodeMCU 0.9 (ESP-12)



Εικόνα 3.21: Επιλογή πλακέτας στο Arduino IDE.

3. Στο μενού επιλέγουμε Εργαλεία → Θύρα και επιλέγουμε την θύρα στην οποία είναι συνδεδεμένο ο μικροεπεξεργαστής NodeMCU. Στην περίπτωση αυτή είναι η θύρα “COM3”.



Εικόνα 3.22: Επιλογή της θύρας στο Arduino IDE.

4. Στην συνέχεια πατάμε το ανέβασμα ώστε να γίνει ο προγραμματισμός της συσκευής. Ταυτόχρονα με αυτό γίνεται και compile του κώδικα, συνεπώς σε περίπτωση που υπάρχουν συντακτικά λάθη το Arduino IDE θα μας ενημερώσει.



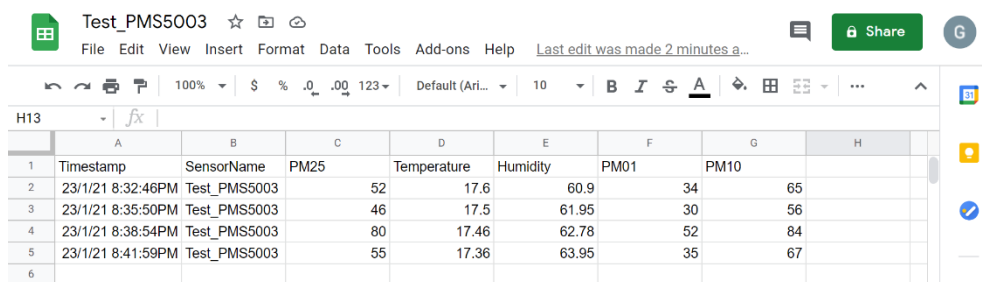
Εικόνα 3.23: Ανέβασμα του κώδικα στον Arduino.

Με την ολοκλήρωση των παραπάνω ο Arduino έχει προγραμματιστεί και συνδεθεί στο διαδίκτυο που επιθυμούμε. Πλέον οι μετρήσεις στέλνονται στο cloud και συγκεκριμένα στο έγγραφο Google Sheets που δημιουργήσαμε.

Κατά τον προγραμματισμό της συσκευής ο διακόπτης πρέπει να είναι στην τρίτη θέση. Κατά την λειτουργία της συσκευής ο διακόπτης πρέπει να είναι στην πρώτη θέση. Οι θέσεις του διακόπτη είναι “ON-OFF-ON”.

Η μορφή στην οποία αποθηκεύονται οι μετρήσεις είναι:

- 1^η στήλη: Χρονική Σήμανση.
- 2^η στήλη: Όνομα Συσκευής.
- 3^η στήλη: Συγκέντρωση των $PM_{2.5}$ σε $\mu g/m^3$.
- 4^η στήλη: Τιμή θερμοκρασίας σε $^{\circ}C$.
- 5^η στήλη: Τιμή υγρασίας %.
- 6^η στήλη: Συγκέντρωση των $PM_{1.0}$ σε $\mu g/m^3$.
- 7^η στήλη: Συγκέντρωση των PM_{10} σε $\mu g/m^3$.



| | A | B | C | D | E | F | G | H |
|---|-------------------|--------------|------|-------------|----------|------|------|---|
| 1 | Timestamp | SensorName | PM25 | Temperature | Humidity | PM01 | PM10 | |
| 2 | 23/1/21 8:32:46PM | Test_PMS5003 | 52 | 17.6 | 60.9 | 34 | 65 | |
| 3 | 23/1/21 8:35:50PM | Test_PMS5003 | 46 | 17.5 | 61.95 | 30 | 56 | |
| 4 | 23/1/21 8:38:54PM | Test_PMS5003 | 80 | 17.46 | 62.78 | 52 | 84 | |
| 5 | 23/1/21 8:41:59PM | Test_PMS5003 | 55 | 17.36 | 63.95 | 35 | 67 | |
| 6 | | | | | | | | |

Εικόνα 3.24: Μορφή αποθήκευσης των μετρήσεων στο Google Sheets.

4 ΕΓΚΑΤΑΣΤΑΣΗ ΣΥΚΕΥΩΝ ΣΕ ΣΗΜΕΙΑ ΤΗΣ ΠΟΛΗΣ ΤΟΥ ΒΟΛΟΥ

Στο κεφάλαιο αυτό παρουσιάζονται τα σημεία στα οποία τοποθετήθηκαν οι συσκευές GreenYourAir/1178 καθώς και η διαδικασία τοποθέτησης τους. Στο παρελθόν έχουν γίνει αρκετές μελέτες για την μεθοδολογία επιλογής σημείων τοποθέτησης των συσκευών με διαφορετικά κριτήρια, όπως η μελέτη που έγινε για την Κωνσταντινούπολη [10] καθώς και η μελέτη για το Όσλο [11]. Στην μελέτη της Κωνσταντινούπολης χρησιμοποιήθηκαν μέθοδοι ανάλυσης χαρακτηριστικών της πόλης με GIS συστήματα, διαχωρίστηκε η πόλη σε περιοχές με διαφορετικά χαρακτηριστικά όπως για παράδειγμα βιομηχανικές ζώνες, οικιστικές ζώνες κ.λπ. Στην μελέτη του Όσλο αναφέρονται πληροφορίες για τις συσκευές καθώς και για τις τοποθεσίες αυτών καθώς και ο τρόπος που επιλέχθηκαν. Οι τοποθεσίες των μετρητών είναι τα νηπαγωγεία της πόλης.

4.1 Σημεία τοποθέτησης

Η επιλογή των σημείων τοποθέτησης των συσκευών έγινε μετά από ανάλυση των γεωγραφικών και γεωμορφολογικών χαρακτηριστικών της πόλης, των συνθηκών που επικρατούν μέσα στην πόλη και των συνθηκών που επικρατούν έξω από αυτήν. Τα δεδομένα τα οποία λήφθηκαν υπόψιν για την επιλογή των σημείων των συσκευών είναι:

1. Γεωγραφικά και γεωμορφολογικά χαρακτηριστικά της πόλης.
2. Τοποθεσία των βιομηχανικών περιοχών της πόλης.
3. Κυκλοφοριακή συμφόρηση στην έκταση της πόλης η οποία χωρίστηκε σε:
 - a. Υψηλή κυκλοφοριακή συμφόρηση
 - b. Μεσαία κυκλοφοριακή συμφόρηση
 - c. Χαμηλή κυκλοφοριακή συμφόρηση
4. Ζώνες με διαφορετικά χαρακτηριστικά της πόλης. Χωρίστηκαν σε:
 - a. Οικιστική ζώνη υψηλής πυκνότητας
 - b. Οικιστική ζώνη μεσαίας πυκνότητας
 - c. Οικιστική ζώνη χαμηλής πυκνότητας
 - d. Εμπορική και Ψυχαγωγική ζώνη
 - e. Βιομηχανική ζώνη.
5. Μέθοδοι θέρμανσης που χρησιμοποιούνται στην έκταση της πόλης.
6. Τοποθεσίες των σταθμών λεωφορείων και τραίνων της πόλης.
7. Τοποθεσία του εμπορικού και του επιβατικού λιμανιού της πόλης.

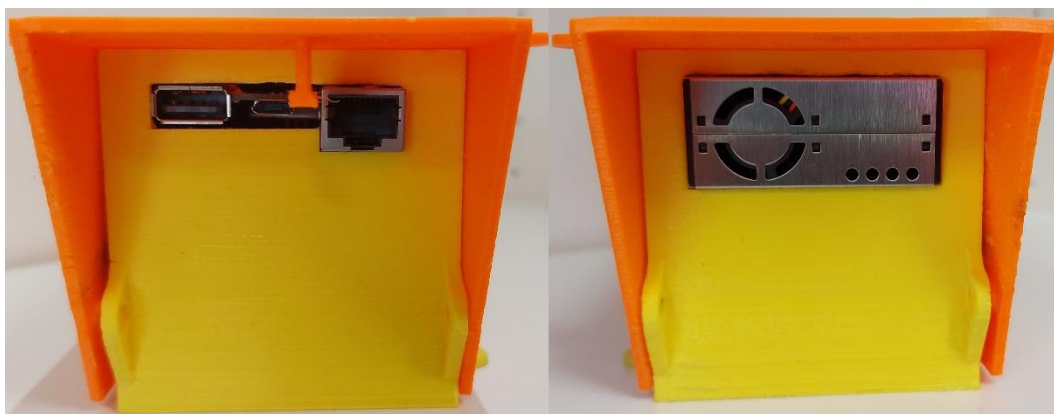
Τα παραπάνω δεδομένα λήφθηκαν κατά την χρήση ενός μαθηματικού μοντέλου για την βέλτιστη κατανομή των συσκευών στην πόλη του Βόλου έγινε ο προσδιορισμός των σημείων στα οποία πρέπει να εγκατασταθούν οι συσκευές. Το μαθηματικό μοντέλο δεν αποτελεί κομμάτι της παρούσας εργασίας και συνεπώς δεν θα παρουσιαστεί. Παρακάτω παρουσιάζονται οι τοποθεσίες των συσκευών μετά τον προσδιορισμό αυτών από το μαθηματικό μοντέλο.



Εικόνα 4.1: Τοποθεσίες των συσκευών.

4.2 Κουτί της συσκευής για την εγκατάσταση

Μετά την συναρμολόγηση των συσκευών είναι απαραίτητη η δημιουργία ενός κουτιού με συγκεκριμένες προδιαγραφές για την ορθή λειτουργία της συσκευής. Για τον λόγο αυτό σχεδιάστηκε και εκτυπώθηκε σε 3D εκτυπωτές το κουτί στο οποίο τοποθετείται το hardware της συσκευής. Η εκτύπωση του κουτιού έγινε από την TED3D στον Βόλο. Παρακάτω παρουσιάζονται φωτογραφίες του κουτιού.



Εικόνα 4.2: Μπροστά και πίσω όψη του κουτιού.



Εικόνα 4.3: Πλάγια όψη του κουτιού.

4.3 Εγκατάσταση των συσκευών

Μετά τον προσδιορισμό των σημείων τοποθέτησης των μετρητών ακολουθεί η διαδικασία εγκατάστασης. Συνεπώς δημιουργήθηκαν 12 συσκευές μέτρησης, προγραμματίστηκαν και στο τέλος έγινε η εγκατάσταση τους στα σημεία.

Κάθε σημείο καλύπτει όλες τις απαιτήσεις της συσκευής για την σωστή λειτουργία της. Οι απαιτήσεις αυτές είναι:

1. Συνεχόμενη σύνδεση στο διαδίκτυο.
2. Συνεχόμενη παροχή ρεύματος.
3. Εξωτερικός χώρος για την τοποθέτηση της συσκευής (π.χ. μπαλκόνι)
4. Εύκολη προσβασιμότητα στον χώρο της συσκευής για τυχόν επιδιορθώσεις βλαβών που μπορεί να προκύψουν κατά την λειτουργία της.

Μετά την προμήθεια των απαραίτητων υλικών για την εγκατάσταση των συσκευών (π.χ. καλώδιο ρεύματος, φως, βίδες κ.λπ.) πραγματοποιήθηκε η εγκατάσταση στο κάθε σημείο ξεχωριστά ανάλογα και με τις ιδιαιτερότητες του.



Εικόνα 4.4: Συσκευή στο Πεδιον Άρεως στο Τμήμα Μηχανολόγων Μηχανικών.



Εικόνα 4.5: Συσκευή στην Κοραή.



Εικόνα 4.6: Συσκευή στη Βρυζάκη.

5 ΕΦΑΡΜΟΓΗ ΚΑΤΑΓΡΑΦΗΣ ΠΕΡΙΣΤΑΤΙΚΩΝ ΣΤΟ ΝΟΣΟΚΟΜΕΙΟ ΤΟΥ ΒΟΛΟΥ

Στο κεφάλαιο αυτό παρουσιάζεται και αναλύεται η διαδικασία κατασκευής της εφαρμογής για την καταχώρηση περιστατικών του Νοσοκομείου του Βόλου που σχετίζονται με την ατμοσφαιρική ρύπανση. Για την κατασκευή της εφαρμογής χρησιμοποιήθηκε το online εργαλείο της Google, AppSheet.

Σκοπός της εφαρμογής είναι να παρέχει έναν εύκολο και γρήγορο τρόπο καταχώρησης περιστατικών που προσήλθαν στο Νοσοκομείο και η αποθήκευση των δεδομένων αυτών στο cloud. Η καταγραφή των δεδομένων αυτών έχει ως σκοπό την αξιοποίηση τους σε μία έρευνα με κύριο αντικείμενο την συσχέτιση των ιατρικών περιστατικών με την ατμοσφαιρική ρύπανση. Στο παρελθόν έχουν γίνει παρόμοιες έρευνες συσχέτισης των περιστατικών με ατμοσφαιρικούς ρύπους όπως

5.1 Δεδομένα της εφαρμογής για τα συμπτώματα του περιστατικού

Οι κλινικές οι οποίες επιλέχθηκαν για την έρευνα είναι:

1. Παθολογική Κλινική
2. Τμήμα Επειγόντων Περιστατικών Παθολογικής Κλινικής
3. Παιδιατρική Κλινική
4. Τμήμα Επειγόντων Περιστατικών Παιδιατρικής Κλινικής
5. Καρδιολογική Κλινική

Για κάθε μία από τις παραπάνω κλινικές επιλέχθηκαν συγκεκριμένα συμπτώματα τα οποία με βάση την βιβλιογραφία σχετίζονται άμεσα με την ατμοσφαιρική ρύπανση.

Τα συμπτώματα εισαγωγής και προσέλευσης για κάθε κλινική είναι:

- Παθολογική Κλινική
 - Βήχας/Απόχρεμψη
 - Δύσπνοια
 - Θωρακικό Άλγος
 - Λιποθυμικό Επεισόδιο
 - Λοίμωξη Αναπνευστικού
 - Παρόξυνση Άσθματος
 - Παρόξυνση Χρόνιας Αναπνευστικής Πνευμονοπάθειας
- Τμήμα Επειγόντων Περιστατικών Παθολογικής Κλινικής
 - Βήχας/Απόχρεμψη
 - Δύσπνοια
 - Θωρακικό Άλγος
 - Λιποθυμικό Επεισόδιο
 - Λοίμωξη Ανωτέρου Αναπνευστικού
 - Λοίμωξη Κατωτέρου Αναπνευστικού
 - Παρόξυνση Άσθματος

- Παρόξυνση Χρόνιας Αναπνευστικής Πνευμονοπάθειας
- Παιδιατρική Κλινική
 - Αμυγδαλίτιδα
 - Βήχας
 - Βήχας/Απόχρεμψη
 - Βρογχολίτιδα
 - Βρογχίτιδα
 - Δύσπνοια
 - Εμπόρετος
 - Λαρυγγίτιδα
 - Λοίμωξη Ανωτέρου Αναπνευστικού
 - Λοίμωξη Κατωτέρου Αναπνευστικού
 - Παρόξυνση Άσθματος
 - Ωταλγία
 - Ωτίτιδα
- Τμήμα Επειγόντων Περιστατικών Παιδιατρικής Κλινικής
 - Αμυγδαλίτιδα
 - Βήχας
 - Βήχας/Απόχρεμψη
 - Βρογχολίτιδα
 - Βρογχίτιδα
 - Δύσπνοια
 - Εμπόρετος
 - Λαρυγγίτιδα
 - Λοίμωξη Ανωτέρου Αναπνευστικού
 - Λοίμωξη Κατωτέρου Αναπνευστικού
 - Παρόξυνση Άσθματος
 - Ωταλγία
 - Ωτίτιδα
- Καρδιολογική Κλινική
 - Απορρύθμιση Χρόνιας Καρδιακής Ανεπάρκειας
 - Εκφόρτιση Απινιδωτή
 - Κοιλιακή Μαρμαρυγή
 - Κοιλιακή Ταχυκαρδία
 - Κολποκοιλιακός Αποκλεισμός
 - Οξύ Έμφραγμα
 - Οξύ Πνευμονικό Οίδημα
 - Οξύ Στεφανιαίο Σύνδρομο
 - Παροξυσμική Κολπική Μαρμαρυγή
 - Παροξυσμική Υπερκοιλιακή Ταχυκαρδία
 - Συκοπικό Επεισόδιο

Οι διαγνώσεις εξιτηρίου κάθε κλινικής είναι:

- Παθολογική Κλινική
 - Αγγειακό Εγκεφαλικό Επεισόδιο
 - Ανακοπή
 - Αναπνευστική Ανεπάρκεια

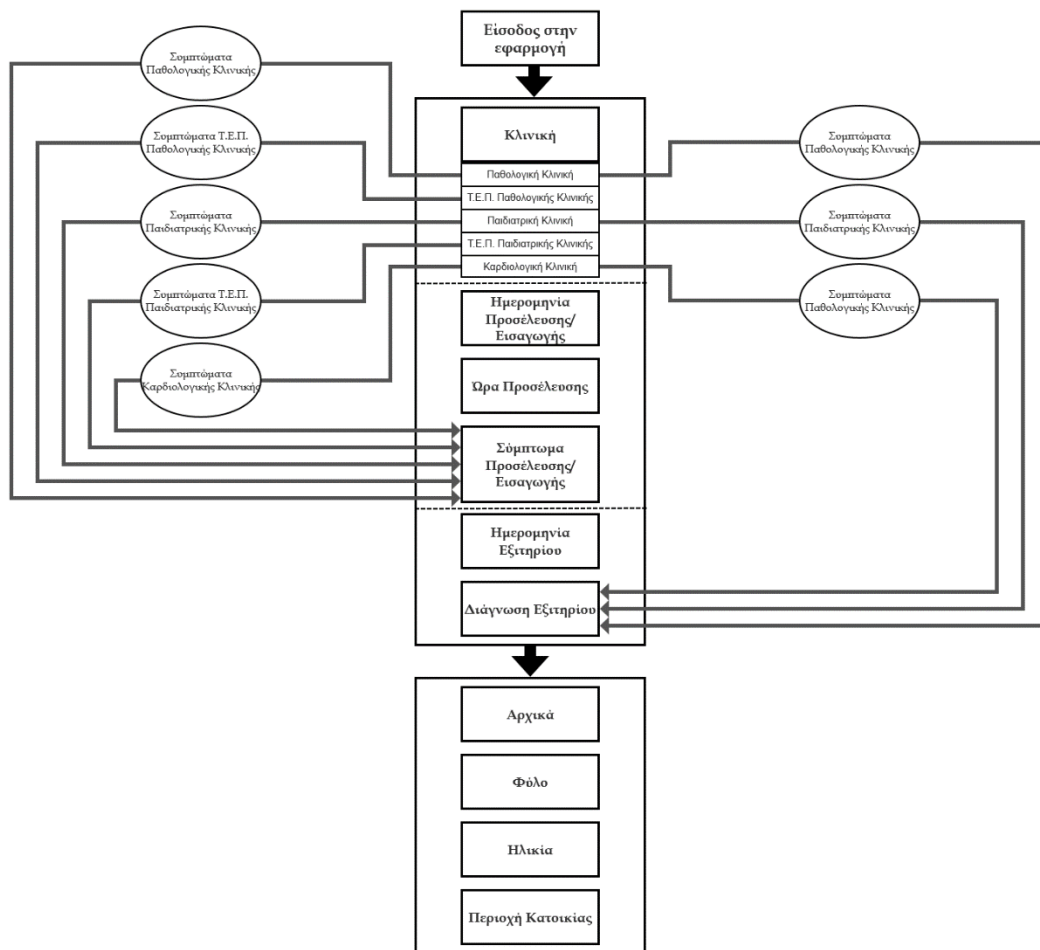
- Καρδιακή Ανεπάρκεια
- Κολπική Μαρμαρυγή
- Λοίμωξη Ανωτέρου Αναπνευστικού
- Λοίμωξη Κατωτέρου Αναπνευστικού
- Οξύ Πνευμονικό Οίδημα
- Παρόξυνση Άσθματος
- Παρόξυνση Χρόνιας Αποφρακτικής Πνευμονοπάθειας
- Συγκοπή και Λιποθυμία
- Παιδιατρική Κλινική
 - Αμυγδαλίτιδα
 - Άσθμα
 - Βρογχολίτιδα
 - Βρογχίτιδα
 - Ιογενής Λοίμωξη
 - Λαρυγγίτιδα
 - Λοίμωξη Ανωτέρου Αναπνευστικού
 - Λοίμωξη Κατωτέρου Αναπνευστικού
 - Πνευμονία
 - Ωτίτιδα
- Καρδιολογική Κλινική
 - Απορρύθμιση Χρόνιας Καρδιακής Ανεπάρκειας
 - Εκφόρτιση Απινιδωτή
 - Κοιλιακή Μαρμαρυγή
 - Κοιλιακή Ταχυκαρδία
 - Κολποκοιλιακός Αποκλεισμός
 - Οξύ Έμφραγμα
 - Οξύ Πνευμονικό Οίδημα
 - Οξύ Στεφανιαίο Σύνδρομο
 - Παροξυσμική Κολπική Μαρμαρυγή/Πτερυγισμός
 - Παροξυσμική Υπερκοιλιακή Ταχυκαρδία
 - Συκοπικό Επεισόδιο

5.2 Δημιουργία της εφαρμογής

Παρακάτω αναλύονται τα βήματα για την δημιουργία της εφαρμογής. Για την δημιουργία της χρησιμοποιούνται τα εργαλεία Google Forms, Google Sheets και AppSheet by Google.

Αρχικά πρέπει να προσδιοριστούν οι απαιτήσεις της εφαρμογής και τα δεδομένα τα οποία θα παρουσιάζονται και θα αποθηκεύονται από τον χρήστη. Το κομμάτι αυτό επηρεάζει και το front-end αλλά και το back-end της εφαρμογής. Το front-end και το back-end της εφαρμογής δημιουργείται αυτόματα από τα εργαλεία που χρησιμοποιούνται χωρίς να χρειάζεται να αναπτυχθεί κώδικας. Όμως η σωστή επικοινωνία αυτών των δύο, οι λειτουργικότητες και το user interface της εφαρμογής

εξαρτάται από τις επιλογές που κάνουμε κατά την δημιουργία όλων των μερών που απαιτούνται.



Εικόνα 5.1: Σχήμα εφαρμογής και ενώσεων.

5.2.1 Δημιουργία του βασικού κορμού της εφαρμογής με την χρήση του Google Forms

Το στάδιο αυτό περιλαμβάνει την δημιουργία μιας βάσης δεδομένων στο cloud με την χρήση των εργαλείων της Google. Τα εργαλεία της Google που χρησιμοποιούνται είναι το Google Forms και το Google Sheets. Το Google Forms παίζει τον ρόλο ενός API. Στην ουσία με τις ερωτήσεις του Google Forms δίνεται η δυνατότητα αποστολής απαντήσεων από την εφαρμογή. Το Google Sheets χρησιμοποιείται από το Google Forms για την αποθήκευση και την εύκολη προβολή και επεξεργασία των καταχωρήσεων.

Παρακάτω παρουσιάζεται η διαδικασία δημιουργίας μιας φόρμας στο Google Forms με την δομή που απαιτείται από το AppSheet και τα δεδομένα που θα εισάγει ο χρήστης στην εφαρμογή.

Στην συγκεκριμένη εφαρμογή υπάρχουν εξαρτήσεις δεδομένων και δεν θα εμφανίζονται όλα τα δεδομένα σε κάθε επιλογή του χρήστη, αλλά μόνο αυτά που απαιτούνται. Υπάρχουν αρκετοί τρόποι να δημιουργηθούν συσχετίσεις και εξαρτήσεις μεταξύ των δεδομένων. Στην περίπτωση αυτή η εξάρτηση των δεδομένων θα γίνει μέσα από το AppSheet. Σε όλα τα παρακάτω βήματα προετοιμάζουμε τις ερωτήσεις της φόρμας με τέτοιο τρόπο ώστε να μπορέσουμε να τα διαχειριστούμε στο AppSheet. Με τον τρόπο αυτό δημιουργούμε μία εφαρμογή που είναι φιλική προς το χρήστη και κάθε φορά εμφανίζονται τα δεδομένα που απαιτείται να καταχωρήσει ο χρήστης σε κάθε επιλογή που κάνει.

- Βήμα 1: Δημιουργία φόρμας στο Google Forms.

Σε έναν browser επισκεπτόμαστε την ιστοσελίδα: <https://docs.google.com/forms/> και συνδεόμαστε με τον λογαριασμό μας στην google ή δημιουργούμε έναν νέο. Στην συνέχεια επιλέγουμε την δημιουργία μιας νέας κενής φόρμας.

- Βήμα 2: Προετοιμασία ερωτήσεων της φόρμας.

1. Δεδομένα της πρώτης ενότητας της εφαρμογής.

Μεταφερόμαστε στην κενή φόρμα που δημιουργήσαμε με σκοπό να δημιουργήσουμε την πρώτη ενότητα της εφαρμογής. Σε αυτή την ενότητα ανήκουν τα γενικά δεδομένα που αφορούν το περιστατικό και είναι:

- Κλινική (format: Multiple choice)
- Ημερομηνία Προσέλευσης/Εισαγωγής (format: Date)
- Ώρα Προσέλευσης (format: Time)
- Διάγνωση Προσέλευσης/Εισαγωγής (format: Short answer)
- Ημερομηνία Εξιτηρίου (format: Date)
- Διάγνωση Εξιτηρίου (format: Short answer)

Σε κάθε μία από τις ερωτήσεις ορίζουμε και το format το οποίο θέλουμε. Η «Ημερομηνία Εξιτηρίου» και η «Διάγνωση Εξιτηρίου» εξαρτάται από το αν έγινε εισαγωγή ή όχι. Η επιλογή αυτή θα εμφανίζεται μόνο στις Κλινικές και όχι στο Τμήμα Επειγόντων Περιστατικών της κάθε κλινικής. Επίσης τα δεδομένα που εμφανίζονται στην ερώτηση «Διάγνωση Προσέλευσης/Εισαγωγής» και στην ερώτηση «Διάγνωση Εξιτηρίου» εξαρτώνται από την Κλινική, συνεπώς θα εμφανίζονται μόνο τα δεδομένα που αντιστοιχούν σε κάθε Κλινική και αναφέρονται αναλυτικά στην ενότητα 4.2.

Section 2 of 3

Γενικά δεδομένα περιστατικού

Description (optional)

Κλινική

- Παθολογική Κλινική
- Τμήμα Επειγόντων Περιστατικών Παθολογικής Κλινικής
- Παιδιατρική Κλινική
- Τμήμα Επειγόντων Περιστατικών Παιδιατρικής Κλινικής
- Καρδιολογική Κλινική

Ημερομηνία Προσέλευσης/Εισαγωγής

Month, day, year

Ωρα Προσέλευσης

Time

Σύμπτωμα/Διάγνωση Προσέλευσης

Short answer text

Ημερομηνία Εξιτηρίου

Month, day, year

Διάγνωση Εξιτηρίου

Short answer text

Εικόνα 5.2: Πρώτη ενότητα εφαρμογής.

2. Δεδομένα της δεύτερης ενότητας της εφαρμογής.

Στην ενότητα αυτή τα δεδομένα αφορούν τον ασθενή και τα προσωπικά του στοιχεία. Τα δεδομένα αυτής της ενότητας δεν εξαρτώνται από τις επιλογές του χρήστη και εμφανίζονται πάντα τα δεδομένα αυτής της ενότητας είναι:

- Αρχικά (format: Short answer)
- Φύλο (format: Multiple choice)
- Ηλικία (format: Short answer)
- Περιοχή Κατοικίας (format: Short answer)

Σε κάθε μία από τις ερωτήσεις ορίζουμε και το format το οποίο θέλουμε.

Section 3 of 3

Στοιχεία Ασθενή

Description (optional)

Αρχικά

Short answer text

Φύλο

Αρσενικό

Θηλυκό

Ηλικία

Short answer text

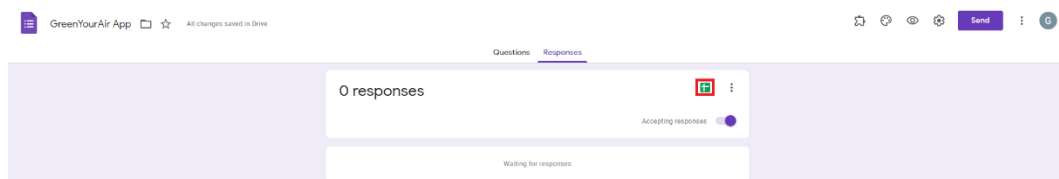
Περιοχή Κατοικίας

Short answer text

Εικόνα 5.3: Δεύτερη ενότητα εφαρμογής.

- Βήμα 3: Δημιουργία εγγράφου στο Google Sheets

Για εύκολη προβολή και επεξεργασία των δεδομένων που καταχωρούνται από τους χρήστες είναι απαραίτητη η δημιουργία ενός εγγράφου στο Google Sheets. Το έγγραφο αυτό δημιουργείται μέσα από την φόρμα και συνδέεται άμεσα με αυτήν και τις ερωτήσεις, καθώς και τις απαντήσεις που είναι τα δεδομένα που καταχωρούν οι χρήστες.



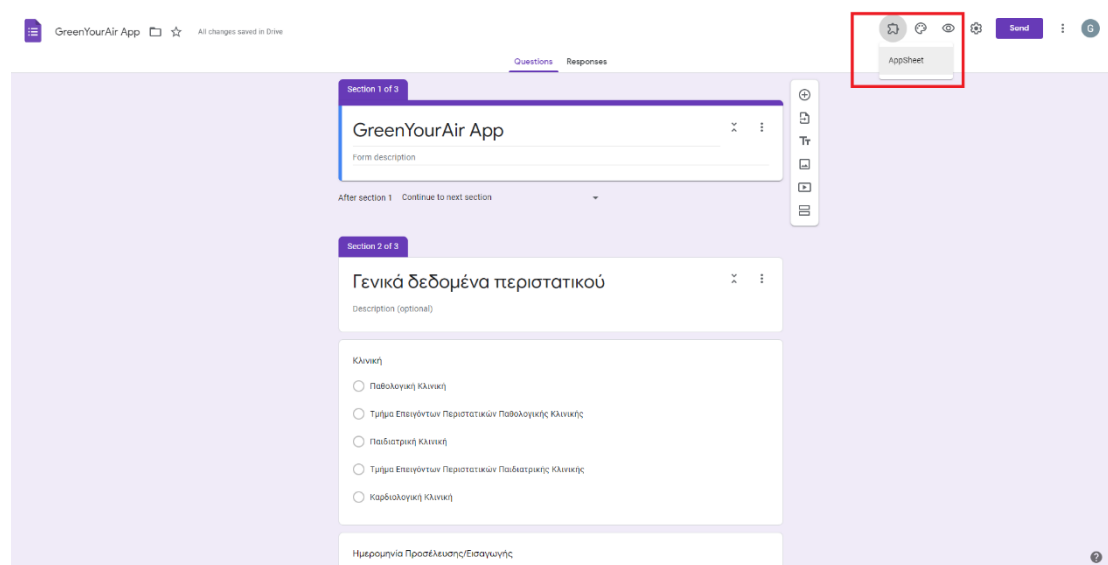
Εικόνα 5.4: Δημιουργία εγγράφου στο Google Sheets για την αποθήκευση των δεδομένων που καταχωρεί ο χρήστης.

5.2.2 Εκκίνηση του AppSheet και δημιουργία της εφαρμογής

Το στάδιο αυτό περιλαμβάνει την δημιουργία της εφαρμογής με τη χρήση του AppSheet. Για να γίνει εκκίνηση του AppSheet είναι απαραίτητο να έχει δημιουργηθεί το έγγραφο στο Google Sheets που συνδέεται με την φόρμα. Αυτό είναι υποχρεωτικό για την αντιστοίχιση των δεδομένων που θα βλέπει ο χρήστης και την αποθήκευση αυτών. Στο AppSheets οι δυνατότητες είναι πάρα πολλές και οι τρόποι υλοποίησης μιας εφαρμογής ποικίλουν. Για την συγκεκριμένη εφαρμογή θα αντιστοιχίσουμε τα συμπτώματα με τις κλινικές ώστε η διαδικασία καταχώρησης να είναι πιο φιλική προς τον χρήστη. Μέσω του Appsheet μπορούμε να κάνουμε και βελτιώσεις στο design της εφαρμογής. Παρακάτω παρουσιάζονται τα βήματα που απαιτούνται για την ολοκλήρωση και τη δημιουργία της εφαρμογής.

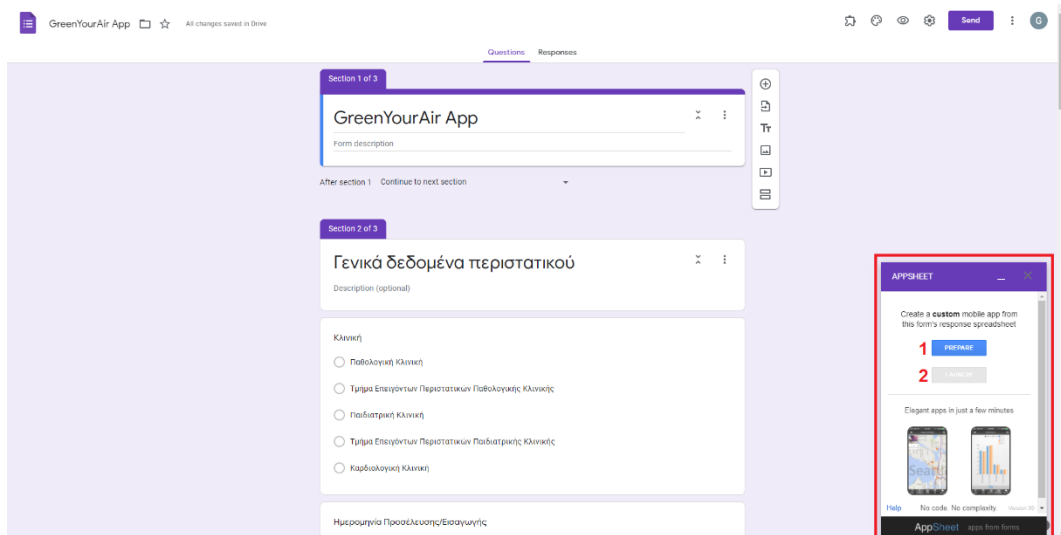
- Βήμα 1: Εκκίνηση του AppSheet μέσα από το Google Forms

Η εκκίνηση του AppSheet μέσα από το Google Forms μας εξασφαλίζει την σωστή σύνδεση του εγγράφου Google Sheets με την εφαρμογή. Με τον τρόπο αυτό διασφαλίζουμε την σωστή επικοινωνία των διαφορετικών πεδίων της εφαρμογής και την σωστή αποθήκευση των δεδομένων. Για την εκκίνηση του AppSheet πηγαίνουμε στο Add-ons και επιλέγουμε το AppSheet.



Εικόνα 5.5: Εκκίνηση του AppSheet από το Google Forms.

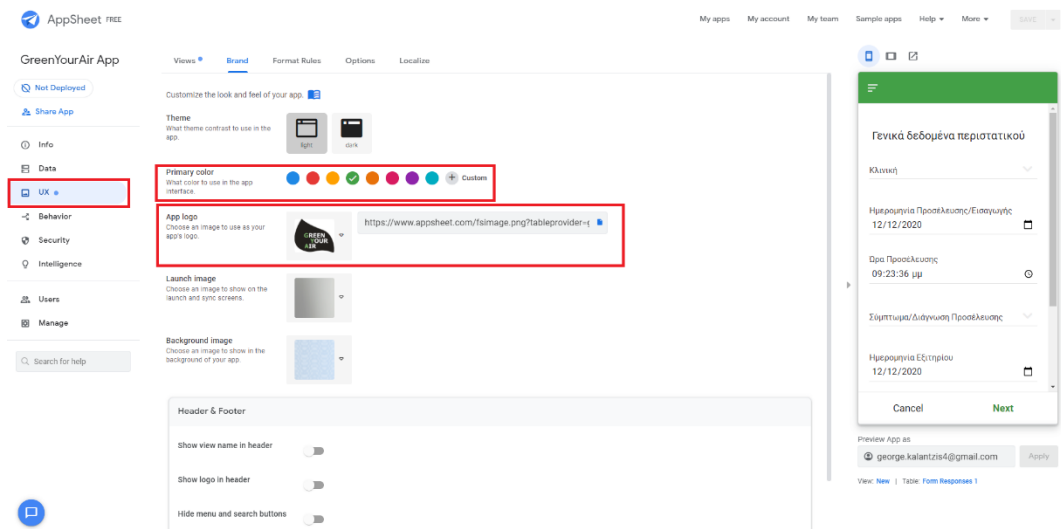
Στην συνέχεια επιλέγουμε “Launch”. Μέσα στο pop-up του AppSheet επιλέγουμε “Prepare” και στην συνέχεια “Launch”. Μετά το “Launch” μεταφερόμαστε στο Dashboard του AppSheet στο οποίο έχουμε όλες τις επιλογές και τις δυνατότητες για την ολοκλήρωση της εφαρμογής.



Εικόνα 5.6: Launch button και μετάβαση στο Dashboard του AppSheet.

- Βήμα 2: Ορισμός εικονιδίου και γενικών χρωμάτων της εφαρμογής

Στο menu αριστερά επιλέγουμε το UX και στην συνέχεια το Brand. Εκεί μπορούμε να ορίσουμε το εικονίδιο της εφαρμογής και τα γενικά χρώματα της εφαρμογής. Στο UX αλληλοεπιδράμε με την εφαρμογή στο κομμάτι του User Experience. Έτσι λοιπόν σε αυτή την κατηγορία χτίζουμε σιγά σιγά τις λειτουργικότητες της εφαρμογής και τα βασικά μέρη που αφορούν το User Interface σε γενικό επίπεδο.



Εικόνα 5.7: Ορισμός ονόματος, περιγραφής και εικονιδίου της εφαρμογής.

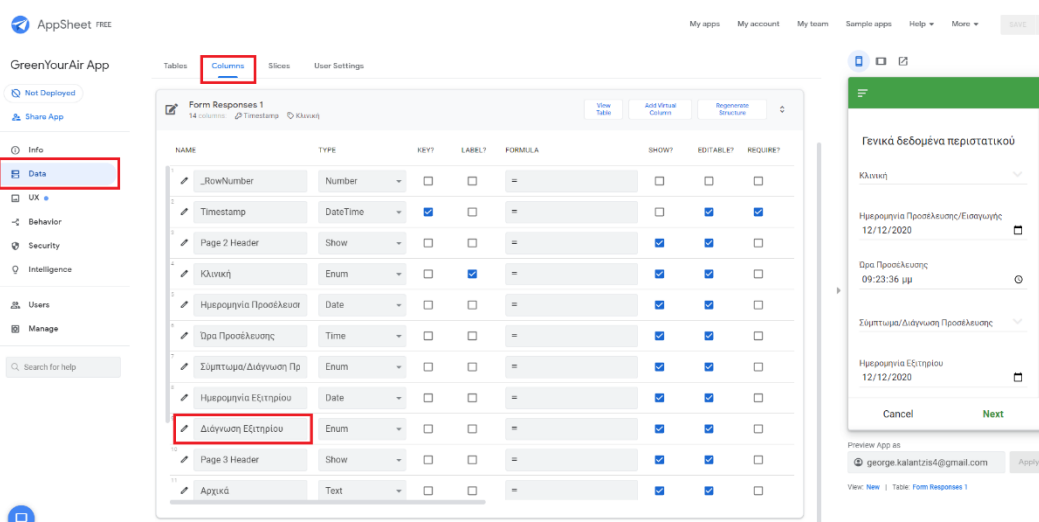
- Βήμα 3: Βελτιώσεις στο format και στην εμφάνιση των δεδομένων

Στο menu αριστερά επιλέγουμε το Data και στην συνέχεια το Columns. Στην ενότητα αυτή εμφανίζονται τα δεδομένα της εφαρμογής και το format αυτών, καθώς και τι

συνθήκες πρέπει να ικανοποιούνται ώστε να εμφανιστούν οι ερωτήσεις που εξαρτώνται από τις επιλογές του χρήστη. Εδώ επίσης μπορούμε να ορίσουμε συγκεκριμένους κανόνες για τα δεδομένα μας ώστε να ενημερώνουμε τον χρήστη σε περίπτωση που εισάγει λάθος δεδομένα σε κάποια ερώτηση. Σκοπός είναι η σωστή καταχώρηση των δεδομένων. Παρακάτω παρουσιάζεται η σύνδεση που πρέπει να γίνει ώστε να εμφανίζεται η ερώτηση «Διάγνωση Εξιτηρίου» που πρέπει να εμφανίζεται μόνο στην περίπτωση που ο χρήστης επιλέξει κάποια κλινική. Επίσης παρουσιάζεται και ένας έλεγχος για την «Ηλικία» ώστε να αποφύγουμε τυχόν λανθασμένες καταχωρήσεις δεδομένων.

1. Παράδειγμα 1: «Διάγνωση Εξιτηρίου»

Η «Διάγνωση Εξιτηρίου» πρέπει να εμφανίζεται όταν ο χρήστης επιλέξει κάποια κλινική. Συνεπώς θα δημιουργήσουμε μια συνθήκη που να ορίζει την εμφάνιση της ερώτησης αυτής όταν ο χρήστης επιλέξει «Παθολογική Κλινική», Παιδιατρική Κλινική ή Καρδιολογική Κλινική. Για να το κάνουμε αυτό επιλέγουμε το edit δίπλα από την γραμμή που αναφέρεται στην «Διάγνωση εξιτηρίου».



Εικόνα 5.8: Μετάβαση στις επιλογές για την εμφάνιση της ερώτησης "Διάγνωση Εξιτηρίου".

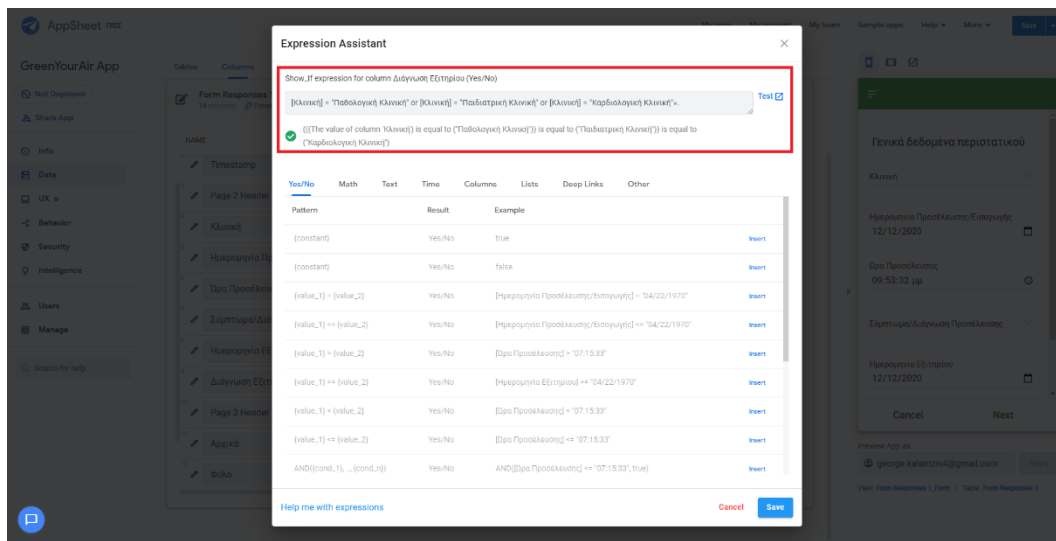
Στην συνέχεια στο μενού που εμφανίζεται επιλέγουμε το "Show?". Μετά την επιλογή του "Show?" μεταφερόμαστε σε ένα νέο μενού στο οποίο ορίζουμε την συνθήκη για την εμφάνιση της συγκεκριμένης ερώτησης στην εφαρμογή. Θέλουμε να εμφανίζεται αυτή η ερώτηση μόνο όταν ο χρήστης επιλέγει κλινική. Η συνθήκη που χρησιμοποιούμε είναι:

```
[Κλινική] = "Παθολογική Κλινική" or [Κλινική] = "Παιδιατρική Κλινική" or [Κλινική] = "Καρδιολογική Κλινική"
```

Κώδικας 5.1: Συνθήκη για την εμφάνιση της ερώτησης "Διάγνωση Εξιτηρίου".

Μεγάλη προσοχή στην αντιστοίχιση και στα ονόματα. Σε περίπτωση ορθογραφικού λάθους η συνθήκη δεν θα ισχύει και η εφαρμογή δεν θα είναι λειτουργική.

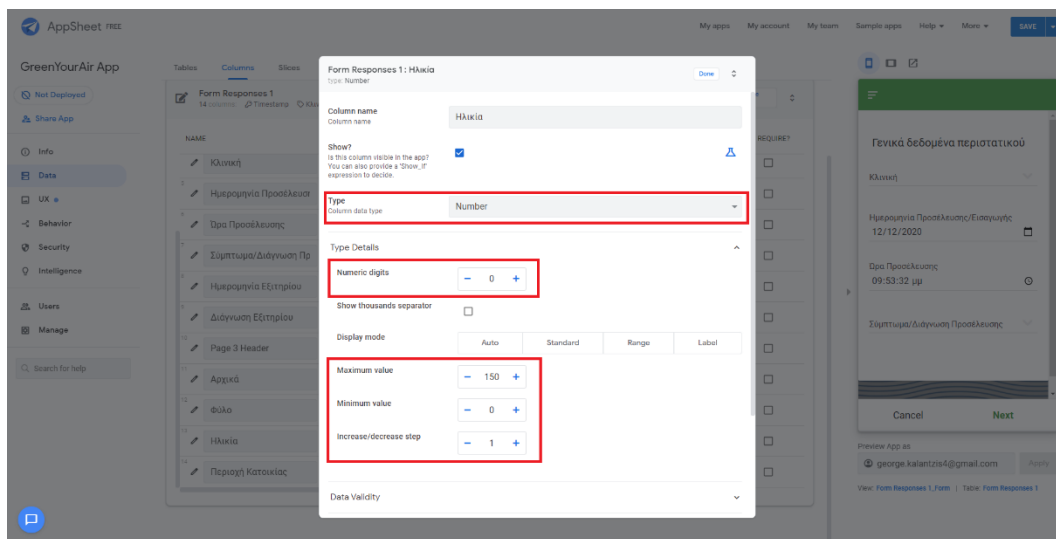
Η ίδια συνθήκη χρησιμοποιείται και για την «Ημερομηνία Εξιτηρίου».



Εικόνα 5.9: Ορισμός συνθήκης για την εμφάνιση της ερώτησης "Διάγνωση Εξιτηρίου".

2. Παράδειγμα 2: «Ηλικία»

Για τον έλεγχο της ποιότητας των δεδομένων μπορούμε να ορίσουμε κάποιες συνθήκες που να ελέγχουν τα δεδομένα που καταχωρεί ο χρήστης. Στην περίπτωση της «Ηλικίας» θέλουμε να είναι αριθμός μεγαλύτερος του μηδέν και μικρότερος από μία λογική μέγιστη τιμή. Για να το κάνουμε αυτό πηγαίνουμε στο "edit" στην γραμμή της «Ηλικίας». Στο μενού που εμφανίζεται επιλέγουμε στο "Type" το "Number". Στην συνέχεια ορίζουμε την ελάχιστη αποδεκτή τιμή, την μέγιστη αποδεκτή τιμή και τα δεκαδικά ψηφία.



Εικόνα 5.10: Έλεγχος της καταχώρησης στην ερώτηση "Ηλικία".

Μέσα από αυτό το μενού μπορούμε να ορίσουμε τις απαραίτητες συνθήκες και του απαραίτητους ελέγχους για την δημιουργία μια εφαρμογής φιλική προς τον χρήστη.

- Βήμα 4: Εμφάνιση των δεδομένων στις ερωτήσεις «Σύμπτωμα/Διάγνωση Προσέλευσης» και «Διάγνωση Εξιτηρίου»

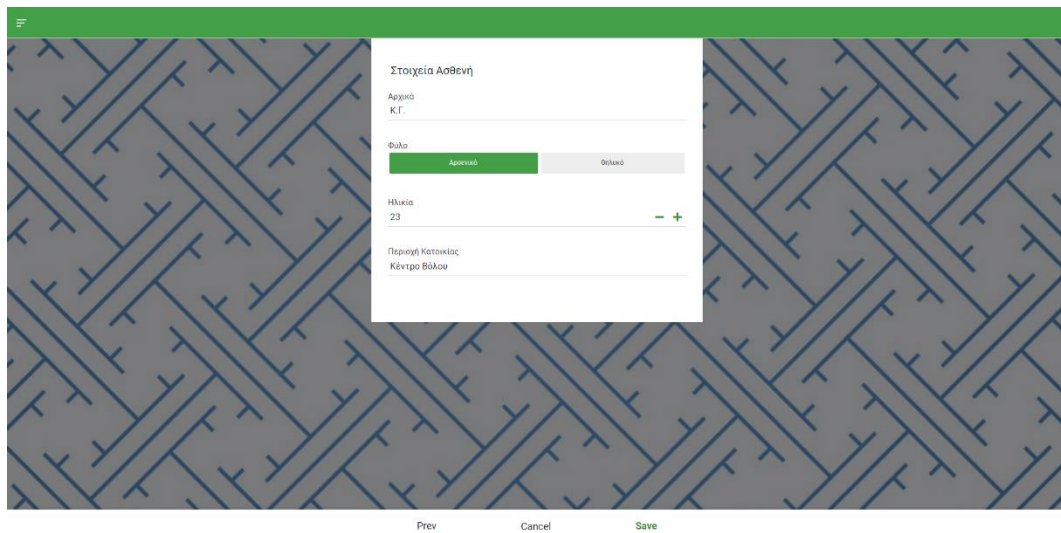
Τα συμπτώματα που εμφανίζονται στην ερώτηση «Σύμπτωμα/Διάγνωση Προσέλευσης» εξαρτώνται από την κλινική την οποία επιλέγει ο χρήστης. Συνεπώς είναι απαραίτητη η δημιουργία κάποιων συνθηκών με σκοπό την σωστή σύνδεση των συμπτωμάτων με τις κλινικές. Το ίδιο ισχύει και για την «Διάγνωση Εξιτηρίου». Η διαδικασία είναι η ίδια.

Για να ορίσουμε την συνθήκη πηγαίνουμε στο “Data” και επιλέγουμε το “Columns”. Εκεί εντοπίζουμε το «Σύμπτωμα/Διάγνωση Προσέλευσης» και επιλέγουμε το “edit”. Στο μενού που μας εμφανίζεται επιλέγουμε το “Data Validity” και στην συνέχεια το “Valid If”. Η συνθήκη που θα δημιουργήσουμε είναι:

```
IFS(
  [Κλινική] = "Παθολογική Κλινική",
  {"Βήχας/Απόχρεμψη", "Δύσπνοια", "Θωρακικό Άλγος", "Λιποθυμικό Επεισόδιο", "Λοίμωξη
  Αναπνευστικού", "Παρόξυνση
  Ασθματος", "Παρόξυνση Χρόνιας Αναπνευστικής Πνευμονοπάθειας"},
  [Κλινική] = "Τμήμα Επειγόντων Περιστατικών Παθολογικής Κλινικής",
  {"Βήχας/Απόχρεμψη", "Δύσπνοια", "Θωρακικό Άλγος", "Λιποθυμικό Επεισόδιο", "Λοίμωξη
  Ανωτέρου Αναπνευστικού",
  "Λοίμωξη Κατωτέρου Αναπνευστικού", "Παρόξυνση Άσθματος", "Παρόξυνση Χρόνιας
  Αναπνευστικής
  Πνευμονοπάθειας"},
  [Κλινική] = "Παιδιατρική Κλινική",
  {"Αμυγδαλίτιδα", "Βήχας", "Βήχας/Απόχρεμψη", "Βρογχιολίτιδα", "Βρογχίτιδα", "Δύσπνοια",
  "Εμπύρετος", "Λαρυγγίτιδα",
  "Λοίμωξη Ανωτέρου Αναπνευστικού", "Λοίμωξη Κατωτέρου Αναπνευστικού", "Παρόξυνση
  Άσθματος", "Ωταλγία",
  "Ωτίτιδα"},
  [Κλινική] = "Τμήμα Επειγόντων Περιστατικών Παιδιατρικής Κλινικής",
  {"Αμυγδαλίτιδα", "Βήχας", "Βήχας/Απόχρεμψη", "Βρογχιολίτιδα", "Βρογχίτιδα", "Δύσπνοια",
  "Εμπύρετος", "Λαρυγγίτιδα",
  "Λοίμωξη Ανωτέρου Αναπνευστικού", "Λοίμωξη Κατωτέρου Αναπνευστικού", "Παρόξυνση
  Άσθματος", "Ωταλγία",
  "Ωτίτιδα"},
  [Κλινική] = "Καρδιολογική Κλινική",
  {"Απορρυθμία Χρόνιας Καρδιακής Ανεπάρκειας", "Εκφόρτιση Απινιδωτή", "Κοιλιακή
  Μαρμαρυγή", "Κοιλιακή
  Ταχυκαρδία", "Κολποκοιλιακός Αποκλεισμός", "Οξύ Έμφραγμα", "Οξύ Πνευμονικό Οίδημα",
  "Οξύ Στεφανιαίο Σύνδρομο",
  "Παροξυσμική Κολπική Μαρμαρυγή", "Παροξυσμική Υπερκοιλιακή Ταχυκαρδία", "Συκοπικό
  Επεισόδιο"}
)
```

Κώδικας 5.2: Συνθήκη για την αντιστοίχιση των συμπτωμάτων με τις κλινικές.

Με τον τρόπο αυτό ορίσαμε τα συμπτώματα που θα εμφανίζονται ανάλογα με την κλινική την οποία έχει επιλέξει ο χρήστης. Μεγάλη προσοχή στην αντιστοίχιση και στα ονόματα. Σε περίπτωση ορθογραφικού λάθους η συνθήκη δεν θα ισχύει και η εφαρμογή δεν θα είναι λειτουργική.



Εικόνα 5.13: Καταχώρηση των στοιχείων του ασθενή.

| Κλινική | Ημερομηνία Προσέλευσης | Ώρα Προσέλευσης | Συμπτωματολόγηση Προσέλευσης | Ημερομηνία Εισαγωγής | Διάγνωση Εισαγωγής | Αρχικό Κ.Γ. | Φύλο | Ηλικία | Περιοχή Κατοικίας |
|--------------------|------------------------|-----------------|------------------------------|----------------------|--------------------|-------------|----------|--------|-------------------|
| Παθολογική Κλινική | 13/12/2020 | 1:59:00 AM | Βήχας/Απύρεση | | | Κ.Γ. | Αρσενικό | 23 | Κέντρο Βόλου |

Εικόνα 5.14: Καταχωρίσεις του χρήστη.

- Βήμα 5: Αποστολή εφαρμογής στους χρήστες

Η εφαρμογή είναι κλειδωμένη και απαιτείται να δοθεί πρόσβαση σε κάποιον χρήστη από τον διαχειριστή της εφαρμογής. Επίσης ο χρήστης συνδέεται με το email του το οποίο για λόγους συμβατότητας προτιμάται να είναι Gmail έτσι ώστε να γίνεται έλεγχος ταυτοποίησης του χρήστη. Πρόσβαση σε χρήστες δίνουμε στην ενότητα “Users” που βρίσκεται στο μενού αριστερά.

6 ΙΣΤΟΣΕΛΙΔΑ ΤΟΥ GreenYourAir

Στο κεφάλαιο αυτό παρουσιάζεται και αναλύεται η διαδικασία κατασκευής της ιστοσελίδας του GreenYourAir στην οποία οι πολίτες μπορούν να ενημερώνονται για τις τιμές των μικροσωματιδίων $PM_{2.5}$ στην πόλη του Βόλου σε πραγματικό χρόνο. Ο προγραμματισμός της ιστοσελίδας έγινε με την χρήση HTML, CSS και Java-script. Επίσης δημιουργήθηκε ένα API για την επικοινωνία της ιστοσελίδας με τα δεδομένα των μετρητών. Το API προγραμματίστηκε στο εργαλείο της Google, Apps script, καθώς τα δεδομένα στέλνονται από τους μετρητές στο cloud της Google.

6.1 Προγραμματισμός του API στο Apps script

Τα δεδομένα τα οποία χρειάζεται η ιστοσελίδα του GreenYourAir για να λειτουργήσει σωστά είναι οι τιμές των μετρήσεων του τελευταίου 24ώρου. Στην πόλη του Βόλου έχουν τοποθετηθεί δώδεκα μετρητές $PM_{2.5}$, συνεπώς υπάρχουν δώδεκα έγγραφα στο Google Sheets στα οποία αποθηκεύονται οι μετρήσεις κάθε 3 λεπτά. Για να υποστηριχθεί η λειτουργικότητα της ιστοσελίδας το πρώτο βήμα είναι η συλλογή των δεδομένων του τελευταίου 24ώρου σε ένα νέο έγγραφο Google Sheets, η υλοποίηση των υπολογισμών που απαιτούνται και η δημοσίευση των αποτελεσμάτων στην ιστοσελίδα.

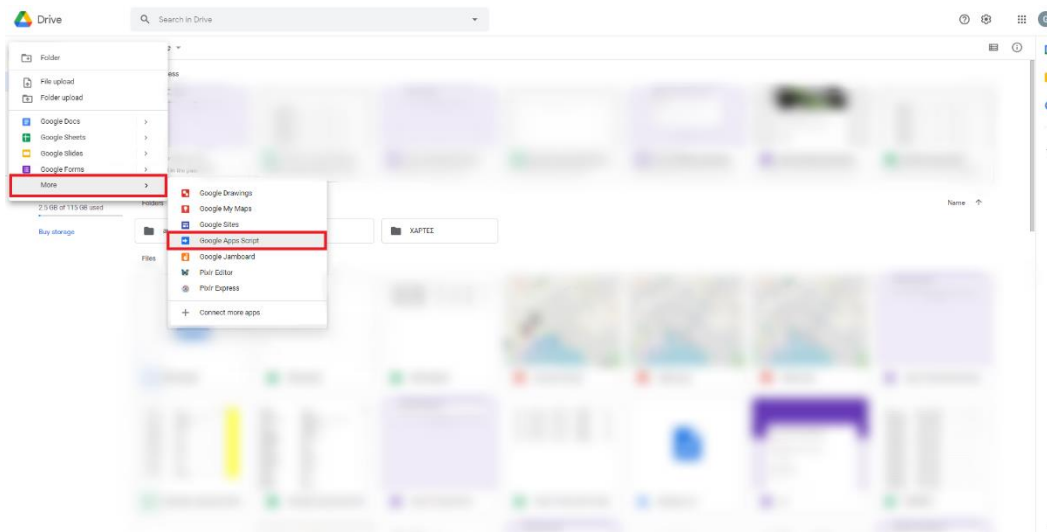
6.1.1 Προετοιμασία εγγράφου στο Google Sheets και προγραμματισμός του πρώτου API

Το πρώτο API το οποίο χρησιμοποιείται έχει ως σκοπό να συλλέξει τα δεδομένα του τελευταίου 24ώρου των δώδεκα μετρητών σε ένα κοινό έγγραφο Google Sheets. Όλοι οι μετρητές στέλνουν δεδομένα, κάθε 3 λεπτά, όμως η αποστολή αυτή δεν γίνεται την ίδια χρονική στιγμή. Κάθε ένα λεπτό το API συλλέγει τα πιο πρόσφατα δεδομένα κάθε μετρητή και τα ταξινομεί σε έναν πίνακα. Στον πίνακα αυτόν τα δεδομένα ανανεώνονται κάθε 1 λεπτό και οι πιο κοντινές, χρονικά, μεταξύ τους μετρήσεις ανήκουν στην ίδια γραμμή. Κατά μέσο όρο ένας μετρητής στέλνει 456 δεδομένα το 24ωρο. Συνεπώς ο πίνακας αποτελείται από τις 456 τιμές του τελευταίου 24ώρου όλων των μετρητών οι οποίες ταξινομούνται με βάση τον χρόνο αποστολής της κάθε τιμής. Στην συνέχεια υπολογίζεται ο μέσος όρος για κάθε γραμμή και έτσι οι τιμές του τελευταίου 24ώρου δημοσιεύονται στην ιστοσελίδα. Οι μετρητές ενδέχεται να μην στείλουν δεδομένα για κάποιο χρονικό διάστημα. Στην περίπτωση αυτή ο μέσος όρος κάθε γραμμής αποτελείται μόνο από τις τιμές των μετρητών που λειτουργούν.

Με το συγκεκριμένο API σε κάθε γραμμή του πίνακα επιτυγχάνεται η αποθήκευση των τιμών που είναι πιο κοντά μεταξύ τους. Στην ίδια γραμμή η μεγαλύτερη απόκλιση μεταξύ των μετρήσεων μπορεί να είναι έως 3 λεπτά, διότι οι μετρητές δεν στέλνουν ταυτόχρονα τα δεδομένα τους.

- Βήμα 1: Δημιουργία του script

Για την δημιουργία του script επισκεπτόμαστε το Google Drive. Στην συνέχεια επιλέγουμε το κουμπι “New”. Στην λίστα που εμφανίζεται επιλέγουμε “More” και τέλος επιλέγουμε το “Google Apps Script”.



Εικόνα 6.1: Δημιουργία Apps Script.

- Βήμα 2: Κώδικας του πρώτου API

Στο dashboard του Apps Script στο οποίο δημιουργήσαμε μπορούμε να γράψουμε τον κώδικα που θα κάνει την διαδικασία αποθήκευσης και αποστολής των δεδομένων στην ιστοσελίδα. Στο σημείο αυτό είναι απαραίτητο να υπάρχει το έγγραφο Google Sheet στο οποίο θα αποθηκεύονται οι τιμές αυτές. Πριν γράψουμε τον κώδικα δημιουργούμε ένα έγγραφο Google Sheets με τα παρακάτω στοιχεία:

Σύνολο στηλών: 24

Σύνολο γραμμών: 458 (456 μετρήσεις + 2 επικεφαλίδες)

Επίσης χρειαζόμαστε μια ακόμη στήλη ώστε να αποθηκεύουμε την τελευταία γραμμή του εγγράφου του Google sheet κάθε μετρητή. Την τιμή αυτή θα την χρησιμοποιήσουμε στο έλεγχο στον οποίο θα αναγνωρίζουμε αν η τιμή που υπάρχει στο sheet κάθε μετρητή είναι η πιο πρόσφατη. Το έγγραφο Google Sheet έχει την παρακάτω μορφή:

Ο πίνακας έχει σταθερό αριθμό γραμμών. Κάθε 1 λεπτό ελέγχεται αν κάποιος μετρητής έχει νέα τιμή. Αν ναι, τότε η τελευταία τιμή, η οποία είναι στην γραμμή 3 και είναι 24 ώρες πριν από την τιμή που θα εισαχθεί διαγράφεται. Οι υπόλοιπες 455 τιμές του μετρητή μετακινούνται 1 γραμμή πάνω και στην γραμμή 458 εισέρχεται η πιο πρόσφατη τιμή του μετρητή. Η διαδικασία αυτή γίνεται για κάθε μετρητή ξεχωριστά. Κάθε μετρητής αποτελείται από 2 στήλες. Στην πρώτη στήλη αποθηκεύεται το timestamp της κάθε μέτρησης και στην δεύτερη στήλη αποθηκεύεται η τιμή της μέτρησης. Με τον τρόπο αυτό σε κάθε γραμμή υπάρχουν οι τιμές όλων των μετρητών οι οποίες είναι πιο κοντά μεταξύ τους χρονικά. Η μέγιστη διαφορά μπορεί να είναι έως 3 λεπτά. Παρακάτω παρουσιάζεται και αναλύεται ο κώδικας.

```
function ValuesofSensors() {

    var ids = ["1k20dApuc-6QR3fszFYY_W4E44w_uv47habvEhYo",
        "1qoBGb5Kdwt96GI9ghkUQt2n_LawM9M6o8414vCxJo",
        "1b8mXx1nasf9JRA8OrgZjtKvzSSoiWTWmj45s6kPdhhP3g",
        "1odf6YRX12GZNYy4tTjXw28Ue1zrY54605o6msUOwg",
        "16I5l6pjJ_c6V7he3eVPqevDi9Z7891dsGTEBtC4f9E",
        "1YeGnbOMfdsaUmLJ2-6KyDbgho-ZtZZFj347RgfeCxNy0",
        "148QGwjfdaTd69SaW9D9Xcu1lUEe72YVZsQ1VP074",
        "17kUg7DBJKSzkDRN9E6Kv_RfRjNGaYtSt-qv1FT200",
        "1LQsrEfASWCwsc9hHUKg0yNYcJKEsaKdmqaoMqz3yc",
        "1_NlfeKWLD-Z-bKZ7eD4TcFy0ZMyf8Vt1458Z4",
        "14JYQ1BKrakfdWOLDk5dKtmGHoz8aVK_tiQLKbou0",
        "1ubgmSJKD459D3UqBSwfdyBaCkv5781sMZJTzk"
    ];

    var sheetname = ["1 απαντήσεις φόρμας",
        "1 απαντήσεις φόρμας",
        "1 απαντήσεις φόρμας",
        "1 απαντήσεις φόρμας",
        "1 απαντήσεις φόρμας",
        "1 απαντήσεις φόρμας",
        "1 απαντήσεις φόρμας",
        "1 απαντήσεις φόρμας",
        "1 απαντήσεις φόρμας",
        "1 απαντήσεις φόρμας",
        "1 απαντήσεις φόρμας",
        "1 απαντήσεις φόρμας"
    ];

    var sumsheet = SpreadsheetApp
        .openById('1o-HJsnvZCJdsamBTrjfN3QdopW0_sQoevkgik46yZE')
        .getSheetByName('Sensors');

    var columncounter = 1; //GYA_API_DATA file (first column of the sensor: timestamp,
    second column of the sensor: pm value)
    var sumsheetlastrow = sumsheet.getLastRow() //lastrow = 458 (456 values per day)

    var currenttime = new Date()

    if (sumsheetlastrow == 458) { //if lastrow != 458 then the something went wrong
        for (var i = 0; i < 12; i++) { //12 sensors

            var sheet = SpreadsheetApp
                .openById(ids[i])
                .getSheetByName(sheetname[i]);

            var sheettimestamp = new Date(sheet.getRange(sheet.getLastRow(), 1).getValue())
            //timestamp per sensor
            var sumsheettimestamp = new Date(sumsheet.getRange(458,
            columncounter).getValue()) //last timestamp per sensor
            var sensorlastrow = sheet.getLastRow() //per sensor
            var previouslastrow = sumsheet.getRange(i + 3, 26).getValue() // GYA_API_DATA
            per sensor
        }
    }
}
```

```

    if (sensorlastrow == previouslastrow) {
        var difference = ((currenttime.getTime() - sheettimestamp.getTime()) / (1000
* 3600 * 24)) * 1440
        if (difference > 3.25) { //if the difference between the current time and
the latest timestamp of the sensor is bigger than 3.5 minutes then the sensor
stopped working
            var difference_sumvscurrent = ((currenttime.getTime() -
sumsheettimestamp.getTime()) / (1000 * 3600 * 24)) * 1440
            if (difference_sumvscurrent >= 2.9999) { //if the sensor stopped working
then every 3 minutes write "no data" in the GYA_API_DATA

                sumsheets.getRange(3, columncounter, 1,
2).deleteCells(SpreadsheetApp.Dimension.ROWS) //delete the cells of the first row of
the sensor and shift up

                var nopm = "no data"

                sumsheets.getRange(458, columncounter).setValue(currenttime) //write the
current time in the timestamp and not the timestamp of the sensor because the sensor
stopped
                sumsheets.getRange(458, columncounter + 1).setValue(nopm)

                sumsheets.getRange(i + 3, 26).setValue(sensorlastrow) //write the last
row of the sensor

            }
        }
    }
    else { //if the sensor's last row is different of the previous one then write
the new value in the GYA_API_DATA

        sumsheets.getRange(3, columncounter, 1,
2).deleteCells(SpreadsheetApp.Dimension.ROWS)

        var pm = sheet.getRange(sheet.getLastRow(), 3).getValues()

        sumsheets.getRange(458, columncounter).setValue(sheettimestamp) //write the
timestamp of the sensor
        sumsheets.getRange(458, columncounter + 1).setValue(calibratedvalue)

        sumsheets.getRange(i + 3, 26).setValue(sensorlastrow) //write the last row of
the sensor for the next comparison

    }
    columncounter += 2
}
}
else { //if the lastrow of the GYA_API DATA is not 458 then reset the file (the
file will restore after 24 hours because the matrices will be empty)
    var range = sumsheets.getRange(3, 1, 997, 24) //delete all the content of the
file except of headers
    range.clearContent()

    for (var i = 0; i < 12; i++) { //12 sensors

        var sheet = SpreadsheetApp
            .openById(ids[i])
            .getSheetByName(sheetname[i]);

        var sheettimestamp = new Date(sheet.getSheetValues(sheet.getLastRow(), 1, 1,
1))
        var sensorlastrow = sheet.getLastRow()

        var difference = ((currenttime.getTime() - sheettimestamp.getTime()) / (1000 *
3600 * 24)) * 1440

        if (difference <= 3.5) { //if the difference between the current time and
latest timestamp of the sensor is <= 3.5 minutes then the sensor did not stopped
working

            var pm = sheet.getRange(sheet.getLastRow(), 3).getValues()

            sumsheets.getRange(458, columncounter).setValue(sheettimestamp) //first
timestamp (write at 458 row)

```



```

sumsheet.getRange(458, columncounter + 1).setValue(calibratedvalue) //first
value (write at 458 row)

sumsheet.getRange(i + 3, 26).setValue(sensorlastrow)

}
else { //if the difference is bigger than 3.5 minutes then the sensor stopped
working. Write "no data" at row 458

var nopm = "no data"

sumsheet.getRange(458, columncounter).setValue(currenttime)
sumsheet.getRange(458, columncounter + 1).setValue(nopm)

sumsheet.getRange(i + 3, 26).setValue(sensorlastrow)

}
columncounter += 2
}
}
}
}

```

Κώδικας 6.1: Κώδικας API για την αποθήκευση των δεδομένων των 12 μετρητών.

Κάθε έγγραφο στο Google Sheet έχει ένα μοναδικό ID. Το ID αυτό το βρίσκουμε στο link του Google Sheet. Τα IDs των δώδεκα μετρητών ορίζονται στον πίνακα με όνομα "ids". Εκτός από τα IDs κάθε Google Sheet χρειαζόμαστε και το όνομα του Sheet στο οποίο αποθηκεύονται τα δεδομένα που αποστέλλονται από τον μετρητή. Τα ονόματα από τα Sheets ορίζονται στον πίνακα "sheetname". Στην συνέχεια χρειαζόμαστε και το ID από το Sheet στο οποίο θα αποθηκεύονται και θα ανανεώνονται κάθε 1 λεπτό οι τιμές των δώδεκα μετρητών. Το ID και το Sheet ορίζονται στην μεταβλητή "sumsheet". Τα παραπάνω είναι απαραίτητα για να διαβαστούν τα δεδομένα κάθε μετρητή και να γραφτούν στο ενιαίο έγγραφο Google Sheet.

Στο πρώτο σκέλος του κώδικα αν η τελευταία γραμμή του αρχείου είναι η 458 τότε διαβάζεται το timestamp και η τελευταία γραμμή στο Sheet που είναι αποθηκευμένα τα δεδομένα του μετρητή καθώς και η τελευταία γραμμή και το τελευταίο timestamp το οποίο γράφτηκε στο ενιαίο έγγραφο των μετρητών. Στην συνέχεια συγκρίνεται η τελευταία γραμμή του Sheet του μετρητή με την τελευταία γραμμή που έχει γραφτεί στο ενιαίο έγγραφο. Αν αυτή η γραμμή είναι διαφορετική τότε οβήνεται η τιμή του μετρητή που βρίσκεται στην γραμμή 3 του ενιαίου εγγράφου, οι υπόλοιπες 455 γραμμές μετακινούνται μία γραμμή πάνω και η νέα τιμή γράφεται στην 458 γραμμή του ενιαίου αρχείου. Αν αυτή η γραμμή είναι ίδια τότε ελέγχεται η διαφορά της ώρας μεταξύ της τελευταία γραμμής του Sheet του μετρητή με την ώρα στην οποία τρέχει ο κώδικας. Αν η διαφορά είναι μεγαλύτερη από 3.25 λεπτά τότε σημαίνει ότι ο μετρητής δεν έχει στείλει νέα μέτρηση και έχει σταματήσει. Στην περίπτωση αυτή οβήνεται η τιμή του μετρητή που βρίσκεται στην γραμμή 3 του ενιαίου εγγράφου, οι υπόλοιπες 455 γραμμές μετακινούνται μία γραμμή πάνω και γράφεται η τιμή "no data" στην 458 γραμμή του ενιαίου αρχείου. Αν η διαφορά της ώρας είναι μικρότερη από 3.25 λεπτά τότε δεν γίνεται κάποια ενέργεια.

```

if (sumsheetlastrow == 458) { //if lastrow != 458 then the something went wrong
for (var i = 0; i < 12; i++) { //12 sensors

var sheet = SpreadsheetApp
.openById(ids[i])
.getSheetByName(sheetname[i]);

```



```

else { //if the lastrow of the GYA_API DATA is not 458 then reset the file (the
file will restore after 24 hours because the matrices will be empty)
var range = sumsheet.getRange(3, 1, 997, 24) //delete all the content of the
file except of headers
range.clearContent()

for (var i = 0; i < 12; i++) { //12 sensors

var sheet = SpreadsheetApp
.openById(ids[i])
.getSheetByName(sheetname[i]);

var sheettimestamp = new Date(sheet.getSheetValues(sheet.getLastRow(), 1, 1,
1))
var sensorlastrow = sheet.getLastRow()

var difference = ((currenttime.getTime() - sheettimestamp.getTime()) / (1000 *
3600 * 24)) * 1440

if (difference <= 3.5) { //if the difference between the current time and
latest timestamp of the sensor is <= 3.5 minutes then the sensor did not stopped
working

var pm = sheet.getRange(sheet.getLastRow(), 3).getValues()

sumsheet.getRange(458, columncounter).setValue(sheettimestamp) //first
timestamp (write at 458 row)
sumsheet.getRange(458, columncounter + 1).setValue(calibratedvalue) //first
value (write at 458 row)

sumsheet.getRange(i + 3, 26).setValue(sensorlastrow)

}
else { //if the difference is bigger than 3.5 minutes then the sensor stopped
working. Write "no data" at row 458

var nopm = "no data"

sumsheet.getRange(458, columncounter).setValue(currenttime)
sumsheet.getRange(458, columncounter + 1).setValue(nopm)

sumsheet.getRange(i + 3, 26).setValue(sensorlastrow)

}
columncounter += 2
}
}
}

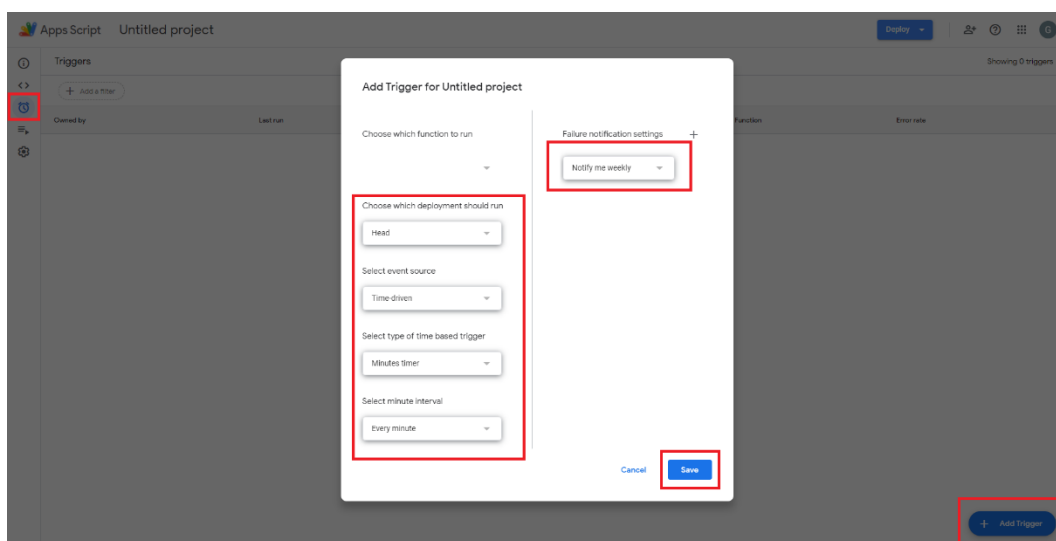
```

Κώδικας 6.3: Δεύτερο σκέλος του κώδικα του API για την αποθήκευση των δεδομένων των 12 μετρητών.

Με τον κώδικα αυτό επιτυγχάνεται η μεταφορά των δεδομένων από κάθε έγγραφο Google Sheet κάθε μετρητή σε ένα ενιαίο έγγραφο στο οποίο για κάθε γραμμή κάθε ένα λεπτό θα ανανεώνονται τα δεδομένα και θα υπολογίζονται οι μέσοι όροι κάθε γραμμής. Μετά τον υπολογισμό τα δεδομένα μεταφέρονται στην ιστοσελίδα και παρουσιάζονται στον χρήστη. Έτσι κάθε ένα λεπτό ο χρήστης μπορεί να δει τις ανανεωμένες τιμές των μικροσωματιδίων PM_{2.5} για την πόλη του Βόλου σε πραγματικό χρόνο.

- Βήμα 3: Ορισμός εκτέλεσης κώδικα ανά 1 λεπτό

Για να ορίσουμε την αυτόματη εκτέλεση του κώδικα ανά 1 λεπτό πηγαίνουμε στο μενού που βρίσκεται αριστερά στο dashboard του Apps Script. Επιλέγουμε το Trigger. Στην συνέχεια στο μενού του Trigger επιλέγουμε να εκτελείται ο κώδικας ανά ένα λεπτό. Επίσης προσθέτουμε και την επιλογή “Notify me weekly” με την οποία κάθε βδομάδα θα μας έρχεται ένα Report σε περίπτωση που ο κώδικας δεν εκτελέστηκε σωστά για οποιονδήποτε λόγο. Η επιλογή του λεπτού έγινε γιατί τα Requests στα έγγραφα Google Sheets είναι περιορισμένα στη δωρεάν έκδοση, σε περίπτωση που ο μέγιστος αριθμός των δωρεάν Requests ξεπεραστεί το API δεν θα λειτουργεί για ένα συγκεκριμένο αριθμό ημερών, ανάλογα με τον συνολικό αριθμό των Requests. Έτσι με την επιλογή του ενός λεπτού εξασφαλίζεται η ομαλή λειτουργία του API.



Εικόνα 6.3: Ορισμός εκτέλεσης του κώδικα του API ανά ένα λεπτό.

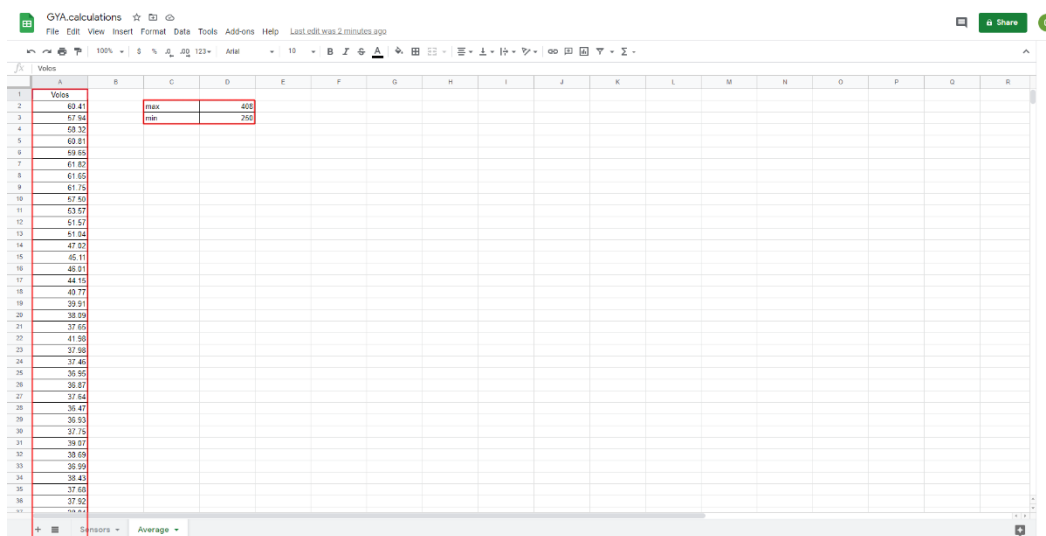
6.1.2 Προγραμματισμός του δεύτερου API για την μεταφορά των δεδομένων στην ιστοσελίδα

Το δεύτερο API το οποίο θα μεταφέρει τα δεδομένα από το ενιαίο έγγραφο Google Sheets στην ιστοσελίδα προγραμματίζεται και αυτό στο Apps Script. Πριν τον κώδικα και τον προγραμματισμό του API είναι απαραίτητο να υπάρχει στο ενιαίο έγγραφο ένα Sheet στο οποίο θα υπολογίζονται και θα αποθηκεύονται οι μέσοι όροι των τιμών των μετρήσεων των 456 γραμμών που αποθηκεύονται στο πρώτο Sheet.

- Βήμα 1: Υπολογισμός μέσων όρων και αποθήκευση των τιμών

Για την μεταφορά των μέσων όρων των τιμών των μετρήσεων των μετρητών για τις τελευταίες 24 ώρες δημιουργούμε ένα Sheet στο οποίο θα γίνονται οι υπολογισμοί. Οι υπολογισμοί θα γίνονται με την χρήση μακροεντολών γιατί ο χρόνος που απαιτείται είναι πολύ μικρότερος σε σχέση με τον χρόνο που πραγματοποιούνται οι

υπολογισμοί με την χρήση κώδικα. Παρακάτω παρουσιάζεται η μορφή του Sheet και οι μακροεντολές που χρησιμοποιούνται.



Εικόνα 6.4: Sheet για τους μέσους όρους των τιμών.

Σε κάθε μία γραμμή της στήλης A χρησιμοποιείται η μακροεντολή που υπολογίζει τον μέσο όρο της αντίστοιχης γραμμής που βρίσκεται στο ενιαίο έγγραφο των μετρήσεων των 12 μετρητών για τις τελευταίες 24 ώρες. Επίσης υπολογίζεται η μέγιστη και η ελάχιστη τιμή από τις 456 τιμές του τελευταίου 24ώρου. Με τον τρόπο αυτό κάθε φορά που ανανεώνεται μία τιμή στο ενιαίο έγγραφο των μετρήσεων ανανεώνονται αυτόματα και οι μέσοι όροι των γραμμών. Τα δεδομένα από το Sheet είναι τα δεδομένα που απεικονίζονται στη ιστοσελίδα.

```
=average(Sensors!A3:X3)
```

Κώδικας 6.4: Μακροεντολή για τον υπολογισμό του μέσου όρου της πρώτης γραμμής του ενιαίου αρχείου των μετρήσεων.

- Βήμα 2: Κώδικας του δεύτερου API

Ανοίγουμε ένα νέο dashboard στο Apps Script για να γράψουμε τον κώδικα της μεταφοράς των δεδομένων στην ιστοσελίδα. Αφού έχουμε δημιουργήσει το καινούριο Sheet στο ενιαίο έγγραφο των μετρήσεων ελέγχουμε τον συνολικό αριθμό των γραμμών. Οι γραμμές θα πρέπει να είναι 457. Η πρώτη γραμμή το Header και οι υπόλοιπες 456 είναι οι τιμές του μέσου όρου των μετρήσεων. Παρακάτω παρουσιάζεται και αναλύεται ο κώδικας του δεύτερου API.

```
function doGet(e) {
  var averagesheet = SpreadsheetApp
    .openById('1O-HJsnvZCJdsamBTrjfN3QdopW0_sQoenvkgik46yZE')
    .getSheetByName('Average') //GYA_API_DATA Volos

  var pmvalues = averagesheet.getSheetValues(2, 1, 456, 1)
  .reduce(function(a, b) {return a.concat(b)})
}
```

```

var maxrow = averagesheet.getRange(2, 4).getValue();
var minrow = averagesheet.getRange(3, 4).getValue();

var matrix = []

for (var i = 0; i < 456; i++){
  var value = parseFloat(pmvalues[i]) //pm values: first 456 values of matrix
  var fixedvalue = value.toFixed(2)
  matrix[i] = fixedvalue
}

var dt = new Date() //get current time

var h = dt.getHours()
var m = dt.getMinutes()
var s = dt.getSeconds()

var day = "" + dt.getDate()
var month = "" + (dt.getMonth() + 1)
var year = dt.getFullYear()

if (month.length < 2){
  month = "0" + month
}

if (day.length < 2){
  day = "0" + day
}

var date = day + "/" + month + "/" + year

var cm = m;
var cs = s;

if (m < 10){
  m = "0" + m
}

if (s < 10){
  s = "0" + s
}

var ltime = h + ":" + m + ":" + s

var counter = 1
var hval = 0

for (var i = 456; i < 480; i++){ //create timestamps
  if (h - counter < 0){

    hval = 24 - (counter - h);
    matrix[i] = hval + ":" + m + ":" + s
    counter += 1

  }else{

    hval = h - counter;
    matrix[i] = hval + ":" + m + ":" + s
    counter += 1

  }
}

matrix[480] = date

var maxh = Math.trunc(maxrow/19)
var maxm = (maxrow - maxh*19)*3
var maxs = (maxrow - maxh*19)*9

maxh += h
maxm += cm
maxs += cs

if(maxs == 60){
  maxm += 1
}

```

```

maxs = 0
}else if(maxs > 60){
maxm += Math.trunc(maxs/60)
maxs = maxs - (Math.trunc(maxs/60)*60)
}

if(maxm == 60){
maxh += 1
maxm = 0
}else if(maxm > 60){
maxh += Math.trunc(maxm/60)
maxm = maxm - (Math.trunc(maxm/60)*60)
}

if(maxh == 24 ){
maxh = 0
}else if (maxh > 24){
maxh = maxh - 24
}

if (maxm < 10){
maxm = "0" + maxm
};

if (maxs < 10){
maxs = "0" + maxs
}

var maxtime = maxh + ":" + maxm + ":" + maxs;

var minh = Math.trunc(minrow/19)
var minm = (minrow - minh*19)*3
var mins = (minrow - minh*19)*9

minh += h
minm += cm
mins +=cs

if(mins == 60){
minm += 1
mins = 0
}else if(mins > 60){
minm += Math.trunc(mins/60)
mins = mins - (Math.trunc(mins/60)*60)
}

if(minm == 60){
minh += 1
minm = 0
}else if(minm > 60){
minh += Math.trunc(minm/60)
minm = minm - (Math.trunc(minm/60)*60)
}

if(minh == 24 ){
minh = 0
}else if (minh > 24){
minh = minh - 24
}

if (minm < 10){
minm = "0" + minm
};

if (mins < 10){
mins = "0" + mins
};

var mintime = minh + ":" + minm + ":" + mins

matrix[481] = maxtime;
matrix[482] = mintime;

var results = matrix

return ContentService.createTextOutput( //return data

```

```
e.parameters.callback + JSON.stringify(results)
).setMimeType(ContentService.MimeType.JAVASCRIPT)
}
```

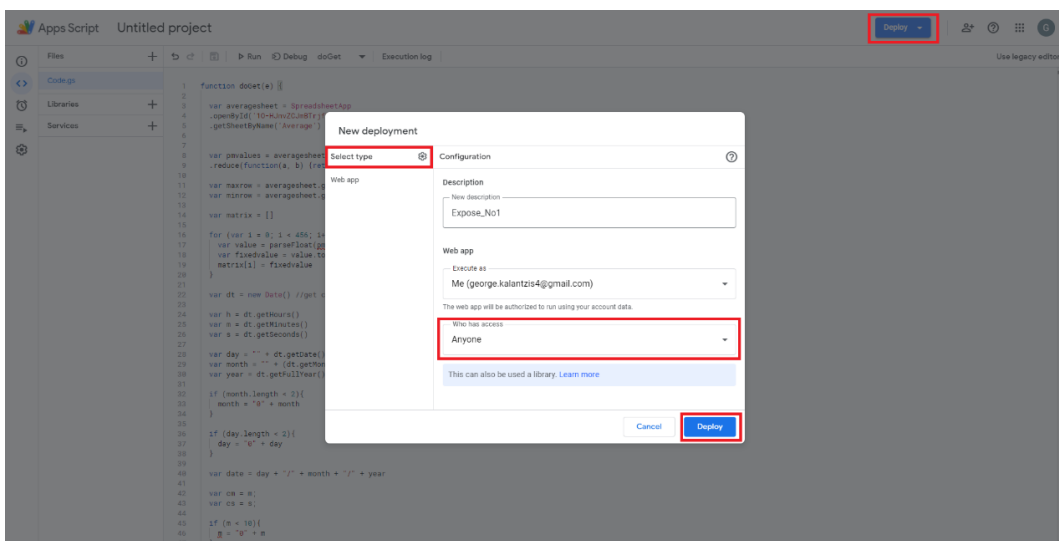
Κώδικας 6.5: Κώδικας API για την αποστολή των μέσων όρων των μετρήσεων στην ιστοσελίδα.

Στο πρώτο σκέλος του κώδικα διαβάζονται τα δεδομένα που υπάρχουν αποθηκευμένα στο Sheet. Στην συνέχεια διαβάζονται η μέγιστη και η ελάχιστη τιμή των 456 τιμών. Τέλος δημιουργούνται τα timestamps και οι ημερομηνία της κλήσης.

Για την αποστολή των δεδομένων σε μορφή συμβατή με τον τρόπο που διαβάζονται τα δεδομένα από την ιστοσελίδα, είναι απαραίτητο να χρησιμοποιηθεί το τελευταίο κομμάτι του κώδικα στο οποίο μετατρέπονται τα δεδομένα και δημοσιεύονται από το API. Με την μορφή αυτή είναι πιο εύκολη η διαχείριση των δεδομένων στην ιστοσελίδα.

- Βήμα 3: Δημοσίευση του δεύτερου API

Για την δημοσίευση του δεύτερου API στο “Deploy” πάνω δεξιά. Στην συνέχεια στο μενού που θα εμφανιστεί επιλέγουμε το “Web App” και επιτρέπουμε την πρόσβαση σε όλους τους χρήστες. Στην συνέχεια πατάμε “Deploy” και στο σημείο αυτό θα εμφανιστεί το API.



Εικόνα 6.5: Δημοσίευση του δεύτερου API.

Στο σημείο αυτό ολοκληρώνεται η υποδομή για την αποθήκευση, τον υπολογισμό των μέσων όρων και την μεταφορά των δεδομένων από τα Google Sheets κάθε μετρητή στην ιστοσελίδα.

6.2 Προγραμματισμός και κώδικας ιστοσελίδας

Η ιστοσελίδα του GreenYourAir αποτελείται από την «Αρχική» που περιέχει γενικές πληροφορίες για την δράση αυτή, από τον «Χάρτη» στον οποίο εμφανίζεται η τελευταία τιμή του μέσου όρου των μετρήσεων των 12 μετρητών PM_{2.5} του Βόλου, καθώς και δεδομένα καιρού, από τα «Διαγράμματα» στα οποία παρουσιάζονται οι μέσοι όροι των μετρήσεων των PM_{2.5} των τελευταίων 24 ωρών, από το «Ημερολόγιο» στο οποίο παρουσιάζονται οι ημερήσιοι μέσοι όροι των PM_{2.5} από 01/03/2019 έως και σήμερα, από τα «Νέα» στα οποία δημοσιεύονται διάφορα νέα για την δράση καθώς και τα εβδομαδιαία δελτία τύπου με τις ημερήσιες τιμές των PM_{2.5} και από τις «Πληροφορίες» στις οποίες αναφέρονται γενικές πληροφορίες για την δράση. Κάθε μία κατηγορία χρησιμοποιεί τον δικό της κώδικα. Χρησιμοποιήθηκε HTML, CSS και Java-script για τον προγραμματισμό της ιστοσελίδας. Οι κώδικες έχουν αρκετές λεπτομέρειες και για τον λόγο αυτό παρακάτω θα αναλυθούν τα βασικά μέρη κάθε κώδικα. Επίσης η ιστοσελίδα παρέχεται στα ελληνικά και στα αγγλικά.

6.2.1 Κώδικας για την «Αρχική»

Στην «Αρχική» ο κώδικας αποτελείται από το “head” στο οποίο ορίζεται το όνομα της ιστοσελίδας, η σύνδεση με το CSS, η σύνδεση με το Google Analytics στο οποίο λαμβάνουμε δεδομένα για την επισκεψιμότητα της ιστοσελίδας καθώς και όλα τα απαραίτητα στοιχεία για την σωστή λειτουργία της ιστοσελίδας. Στο “body” ορίζεται το navigation bar της ιστοσελίδας και τα dropdown menu όπου είναι απαραίτητο καθώς και οι πληροφορίες που αναγράφονται στην «Αρχική».

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta property="og:image" content="../GYA_icon.png">
  <meta property="og:image:type" content="image/png">
  <meta property="og:image:width" content="1024">
  <meta property="og:image:height" content="1024">
  <link rel="icon" type="image/png" href="../GYA_icon.png">
  <title>GreenYourAir</title>
  <link rel="stylesheet" href="../style.css">
  <link rel="stylesheet" href="../navbar.css">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
  <link
    rel="stylesheet"
    href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
  />
  <link
    rel="stylesheet"
    href="../fullcalendar/jquery-ui.css"
  />
  <!-- Global site tag (gtag.js) - Google Analytics -->
  <script async src="https://www.googletagmanager.com/gtag/js?id=UA-168804910-
1"></script>
  <script>
    window.dataLayer = window.dataLayer || [];
    function gtag(){dataLayer.push(arguments);}
  </script>
```

```

    gtag('js', new Date());

    gtag('config', 'UA-168804910-1');
  </script>

  <script src='../fullcalendar/jquery.min.js'></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  <script src='../fullcalendar/jquery-ui.js'></script>
</head>

<body>
  <header>
    <nav class="navbar navbar-default" >
      <div class="container-fluid">

        <div class="navbar-header">
          <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
            <i class="fa fa-th-large"></i>
            Μενού
          </button>
          <a class="navbar-brand" href="/">GreenYourAir</a>
        </div>

        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
          <ul class="nav navbar-nav">
            <li class="nav-item">
              <a class="nav-link" href="/">
                <i class="fa fa-home"></i>
                Αρχική
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/gr/mapgr">
                <i class="fa fa-globe"></i>
                Χάρτης
              </a>
            </li>
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
                <i class="fa fa-align-left"></i>
                Διαγράμματα
                <span class="caret"></span>
              </a>
              <ul class="dropdown-menu">
                <li><a href="/gr/chartsgr/voles">
                  <i class="fa fa-map-marker"></i>
                  Βόλος
                </a></li>
              </ul>
            </li>
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
                <i class="fa fa-calendar"></i>
                Ημερολόγιο
                <span class="caret"></span>
              </a>
              <ul class="dropdown-menu">
                <li><a href="/gr/calendar/voles">
                  <i class="fa fa-map-marker"></i>
                  Βόλος
                </a></li>
              </ul>
            </li>
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
                <i class="fa fa-file"></i>
                Νέα
                <span class="caret"></span>
              </a>
              <ul class="dropdown-menu">
                <li><a href="/gr/newsfeedgr/volesnewsfeed">

```

```

        <i class="fa fa-map-marker"></i>
        Βόλος
    </a></li>
</ul>
</li>
<li class="nav-item">
    <a class="nav-link" href="/gr/about">
        <i class="fa fa-info-circle"></i>
        Πληροφορίες
    </a>
</li>
</ul>
</div>
<div class="flag">
    <a href="/"></a>
    <a href="/en"></a>
</div>
</div>
</nav>
</header>

<h1 align="center"></h1>

<p class="first">
    Το κύριο αντικείμενο του GreenYourAir είναι η παρακολούθηση της ποιότητας
του αέρα στην πόλη του Βόλου και ειδικότερα η παρακολούθηση των μικροσωματιδίων με
διάμετρο μικρότερη των
    2.5 μικρομέτρων (PM2.5) .
</p>

<p class="second">
    Για πρώτη φορά στην ιστορία, οι πολίτες μπορούν να ενημερώνονται σε
πραγματικό χρόνο για την ποιότητα του αέρα στην πόλη του Βόλου και να προστατευτούν
(ειδικά οι ευάλωτες ομάδες πολιτών)
    σε περίπτωση που τα επίπεδα PM2.5 υπερβαίνουν τα επίσημα όρια του Παγκόσμιου
Οργανισμού Υγείας (ημερήσιο όριο: 25μg/m3 και ετήσιο όριο: 10μg/m3). Σημειώνεται ότι
η Ευρωπαϊκή Ένωση έχει ορίσει
    μόνο ετήσιο όριο για τα PM2.5 το οποίο είναι 25μg/m3.
</p>

<p class="third">
    Το πρόβλημα της ποιότητας του Αέρα απαιτεί μία διεπιστημονική προσέγγιση από
επιστήμονες με κατάλληλες και εξειδικευμένες θεωρητικές και τεχνικές γνώσεις. Η
ομάδα του GreenYourAir σχηματίστηκε
    το Σεπτέμβριο του 2017 στο Τμήμα Μηχανολόγων Μηχανικών του Πανεπιστημίου
Θεσσαλίας υπό την επίβλεψη του Αναπληρωτή Καθηγητή Δρ. Γεώργιου Κ.Δ. Σαχαρίδη, στο
πλαίσιο δύο ευρωπαϊκών ερευνητικών
    έργων, του GreenYourMove και του GreenYourRoute.
</p>

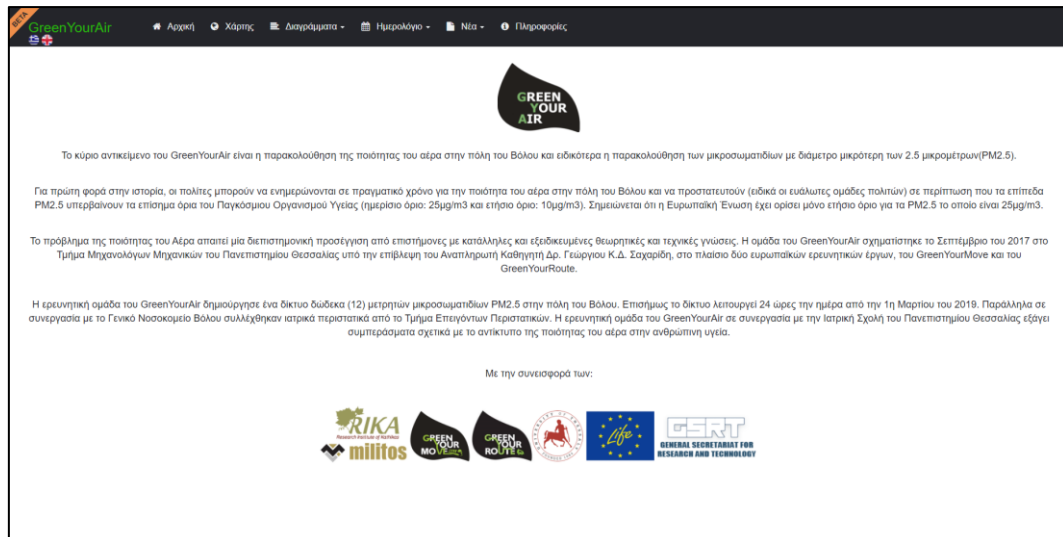
<p class="fourth">
    Η ερευνητική ομάδα του GreenYourAir δημιούργησε ένα δίκτυο δώδεκα (12)
μετρητών μικροσωματιδίων PM2.5 στην πόλη του Βόλου. Επισήμως το δίκτυο λειτουργεί 24
ώρες την ημέρα από την 1η Μαρτίου
    του 2019. Παράλληλα σε συνεργασία με το Γενικό Νοσοκομείο Βόλου συλλέχθηκαν
ιατρικά περιστατικά από το Τμήμα Επειγόντων Περιστατικών. Η ερευνητική ομάδα του
GreenYourAir σε συνεργασία με την
    Ιατρική Σχολή του Πανεπιστημίου Θεσσαλίας εξάγει συμπεράσματα σχετικά με το
αντίκτυπο της ποιότητας του αέρα στην ανθρώπινη υγεία.
</p>

<p class="logostexthome" align = "middle">
    Με την συνεισφορά των:
</p>

<p class="logoshome" align = "middle"></p>
</body>
</html>

```

Κώδικας 6.6: Κώδικας της ιστοσελίδας για την “Αρχική”.



Εικόνα 6.6: "Αρχική" της ιστοσελίδας.

6.2.2 Κώδικας για τον «Χάρτη»

Στον «Χάρτη» ο κώδικας αποτελείται από το “head” στο οποίο ορίζεται το όνομα της ιστοσελίδας, η σύνδεση με το CSS, η σύνδεση με το Google Analytics στο οποίο λαμβάνουμε δεδομένα για την επισκεψιμότητα της ιστοσελίδας καθώς και όλα τα απαραίτητα στοιχεία για την σωστή λειτουργία της ιστοσελίδας. Στο “body” ορίζεται το navigation bar της ιστοσελίδας και τα dropdown menu όπου είναι απαραίτητο. Επίσης ορίζεται ο χάρτης και ο τρόπος με τον οποίο διαβάζονται τα δεδομένα που είναι αποθηκευμένα στον sever. Στο κομμάτι αυτό δεν εισάγουμε το API αμέσως στην ιστοσελίδα αλλά αποθηκεύουμε τα δεδομένα στο server και από εκεί τα διαβάζουμε. Στο “body” ορίζονται επίσης και τα χρώματα του κύκλου που εξαρτώνται από την τιμή των $PM_{2.5}$, καθώς και τα δεδομένα καιρού.

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta property="og:image" content="../GYA_icon.png">
  <meta property="og:image:type" content="image/png">
  <meta property="og:image:width" content="1024">
  <meta property="og:image:height" content="1024">
  <link rel="icon" href="../GYA_icon.png">
  <title>GreenYourAir</title>
  <link rel="stylesheet" href="../style.css">
  <link rel="stylesheet" href="../navbar.css">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
  <link
    rel="stylesheet"
    href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
  />
  <link
    rel="stylesheet"
    href="../fullcalendar/jquery-ui.css"
  />

```

```

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.6.0/dist/leaflet.css"
/>

<!-- Global site tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?id=UA-168804910-
1"></script>
<script>
    window.dataLayer = window.dataLayer || [];
    function gtag(){dataLayer.push(arguments);}
    gtag('js', new Date());

    gtag('config', 'UA-168804910-1');
</script>

<script src="https://unpkg.com/leaflet@1.6.0/dist/leaflet.js"></script>
<script type="text/javascript" src="https://code.jquery.com/jquery-
1.12.0.min.js"></script>
<style>
    #map {
        position: absolute;
        top: 60px;
        bottom: 0;
        left: 0;
        right: 0;
        z-index: -1;
    }
</style>

<script src='../fullcalendar/jquery.min.js'></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<script src='../fullcalendar/jquery-ui.js'></script>
</head>

<body>
<header>
<nav class="navbar navbar-default" >
<div class="container-fluid">

<div class="navbar-header">
<button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
<i class="fa fa-th-large"></i>
Mevoú
</button>
<a class="navbar-brand" href="/">GreenYourAir</a>
</div>

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
<ul class="nav navbar-nav">
<li class="nav-item">
<a class="nav-link" href="/">
<i class="fa fa-home"></i>
Αρχική
</a>
</li>
<li class="nav-item">
<a class="nav-link" href="/gr/mapgr">
<i class="fa fa-globe"></i>
Χάρτης
</a>
</li>
<li class="dropdown">
<a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
<i class="fa fa-align-left"></i>
Διαγράμματα
<span class="caret"></span>
</a>
<ul class="dropdown-menu">
<li><a href="/gr/chartsgr/volos">
<i class="fa fa-map-marker"></i>
Βόλος
</a></li>

```

```

    </ul>
  </li>
  <li class="dropdown">
    <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
      <i class="fa fa-calendar"></i>
      Ημερολόγιο
      <span class="caret"></span>
    </a>
    <ul class="dropdown-menu">
      <li><a href="/gr/calendar/voles">
        <i class="fa fa-map-marker"></i>
        Βόλος
      </a></li>
    </ul>
  </li>
  <li class="dropdown">
    <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
      <i class="fa fa-file"></i>
      Νέα
      <span class="caret"></span>
    </a>
    <ul class="dropdown-menu">
      <li><a href="/gr/newsfeedgr/volesnewsfeed">
        <i class="fa fa-map-marker"></i>
        Βόλος
      </a></li>
    </ul>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="/gr/about">
      <i class="fa fa-info-circle"></i>
      Πληροφορίες
    </a>
  </li>
</ul>
</div>
<div class="flag">
  <a href="/gr/mapgr"></a>
  <a href="/en/mapen"></a>
</div>
</div>
</nav>
</header>

<div id="map"></div>
<script type="text/javascript" src="../../exportTable.js"></script>
<script type="text/javascript" async>
  var lat = 39.368540;
  var lon = 22.947291;
  var zoom = 12;

  var map = new L.map('map').setView([lat, lon], zoom);

  const tilelayer_url = `/maps/positron/{z}/{x}/{y}.png`;
  const tilelayer_options = {
    attribution: '<a href="https://www.maptiler.com/copyright/"
target="_blank">&copy; MapTiler</a> <a
href="https://www.openstreetmap.org/copyright" target="_blank">&copy; OpenStreetMap
contributors</a>',
  }
  const tilelayer = L.tileLayer(tilelayer_url , tilelayer_options)
  tilelayer.addTo(map);

  var circle1 = L.circle([lat, lon],{
    radius: 6000
  }).addTo(map);

  async function exportMap() {

  var pm25 = await exportTable()

```

```

var new_popup1 = L.popup({"autoClose": false, "closeOnClick": null})
    .setContent(
        "<b>Τελευταία ενημερωμένη τιμή του μέσου όρου των 12
μετρητών που καλύπτουν την πόλη του Βόλου.</b><br>PM2.5: " + pm25[455] + "
&mu;g/m3.</br><br>Τελευταία ενημέρωση: " + pm25[479] + " " + pm25[480] +
"<br>Πατήστε στον κύκλο για να δείτε πληροφορίες για τον καιρό."
    );

if (pm25[455] <=25){
    circle1.setStyle({fillColor: 'green', color:'green'});
}else if (pm25[455] <=50) {
    circle1.setStyle({fillColor: 'gold', color:'gold'});
}else if (pm25[455] <= 75){
    circle1.setStyle({fillColor: 'darkorange', color:'darkorange'});
}else if(pm25[455] <=100){
    circle1.setStyle({fillColor: 'red', color:'red'});
}else if (pm25[455] <=150){
    circle1.setStyle({fillColor: 'purple', color:'purple'});
}else{
    circle1.setStyle({fillColor: 'darkred', color:'darkred'});
}
circle1.bindPopup(new_popup1).openPopup();
}
exportMap();

var weatherpopup = L.popup();

async function onMapClick(e) {
    weatherpopup.setLatLng(e.latlng)
        .setContent("You clicked the map at " + e.latlng.toString())
        .openOn(map);

    const weather_api_url = `/weather/${e.latlng.lat}/${e.latlng.lng}`
    const weather_response = await fetch(weather_api_url);
    const weather_data = await weather_response.json();

    weatherlocation_lon = weather_data.coord.lon;
    weatherlocation_lat = weather_data.coord.lat;
    weatherstationname = weather_data.name
    weatherstationid = weather_data.id
    weathertime = weather_data.dt
    temperature = weather_data.main.temp;
    airpressure = weather_data.main.pressure;
    airhumidity = weather_data.main.humidity;
    temperature_min = weather_data.main.temp_min;
    temperature_max = weather_data.main.temp_max;
    windspeed = weather_data.wind.speed;
    winddirection = weather_data.wind.deg;
    cloudcoverage = weather_data.clouds.all;
    weatherconditionid = weather_data.weather[0].id
    weatherconditionstring = weather_data.weather[0].main
    weatherconditiondescription = weather_data.weather[0].description
    weatherconditionicon = weather_data.weather[0].icon

    var utctimecalc = new Date(weathertime * 1000);
    var months =
['01','02','03','04','05','06','07','08','09','10','11','12'];
    var year = utctimecalc.getFullYear();
    var month = months[utctimecalc.getMonth()];
    var date = utctimecalc.getDate();
    var hour = utctimecalc.getHours();
    var min = utctimecalc.getMinutes();
    var sec = utctimecalc.getSeconds();
    var time = date + '.' + month + '.' + year + ' ' + hour + ':' + min + '
Uhr';

    var weatherconditioniconhtml = "http://openweathermap.org/img/w/" +
weatherconditionicon + ".png";
    var weathertimenormal = time;
    var temperaturecelsius = Math.round((temperature - 273) * 100) / 100;
    var windspeedknots = Math.round((windspeed * 1.94) * 100) / 100;
    var windspeedkmh = Math.round((windspeed * 3.6) * 100) / 100;
    var winddirectionstring = "Im the wind from direction";
    if (winddirection > 348.75 && winddirection <= 11.25) {

```

```

        winddirectionstring = "North";
    } else if (winddirection > 11.25 && winddirection <= 33.75) {
        winddirectionstring = "Northnortheast";
    } else if (winddirection > 33.75 && winddirection <= 56.25) {
        winddirectionstring = "Northeast";
    } else if (winddirection > 56.25 && winddirection <= 78.75) {
        winddirectionstring = "Eastnortheast";
    } else if (winddirection > 78.75 && winddirection <= 101.25) {
        winddirectionstring = "East";
    } else if (winddirection > 101.25 && winddirection <= 123.75) {
        winddirectionstring = "Eastsoutheast";
    } else if (winddirection > 123.75 && winddirection <= 146.25) {
        winddirectionstring = "Southeast";
    } else if (winddirection > 146.25 && winddirection <= 168.75) {
        winddirectionstring = "Southsoutheast";
    } else if (winddirection > 168.75 && winddirection <= 191.25) {
        winddirectionstring = "South";
    } else if (winddirection > 191.25 && winddirection <= 213.75) {
        winddirectionstring = "Southsouthwest";
    } else if (winddirection > 213.75 && winddirection <= 236.25) {
        winddirectionstring = "Southwest";
    } else if (winddirection > 236.25 && winddirection <= 258.75) {
        winddirectionstring = "Westsouthwest";
    } else if (winddirection > 258.75 && winddirection <= 281.25) {
        winddirectionstring = "West";
    } else if (winddirection > 281.25 && winddirection <= 303.75) {
        winddirectionstring = "Westnorthwest";
    } else if (winddirection > 303.75 && winddirection <= 326.25) {
        winddirectionstring = "Northwest";
    } else if (winddirection > 326.25 && winddirection <= 348.75) {
        winddirectionstring = "Northnorthwest";
    } else {
        winddirectionstring = " - currently no winddata available - ";
    }
};

weatherpopup.setContent("Δεδομένα καιρού:<br>" + "<br>" + "<br><br>Θερμοκρασία: " + temperaturecelsius +
"\xBAC<br>Πίεση Αέρα: " + airpressure + " hPa<br>Υγρασία: " + airhumidity + "%" +
"<br>Νεφοκάλυψη: " + cloudcoverage + "%<br><br>Ταχύτητα Αέρα: " + windspeedkmh + "
km/h<br>Κατεύθυνση Αέρα: " + winddirectionstring + " (" + winddirection + "\xBE)");
    }
    circle1.on('click', onMapClick);
</script>

<table id = "mapmatrixcolors" align="right">
<tr>
<th id = "matrixheaders" >Όρια PM2.5 και Χρώματα</th>
</tr>
<tr>
<td id = "pm0_25">0-25 μg/m3</td>
</tr>
<tr>
<td id = "pm25_50">25-50 μg/m3</td>
</tr>
<tr>
<td id="pm50_75">50-75 μg/m3</td>
</tr>
<tr>
<td id="pm75_100">75-100 μg/m3</td>
</tr>
<tr>
<td id="pm100_150">100-150 μg/m3</td>
</tr>
<tr>
<td id="pm150">>150 μg/m3</td>
</tr>
</table>

<p class="logostext2" align="left">
Με την συνεισφορά των:
</p>

<p class="logos2" align="left">

</p>

```

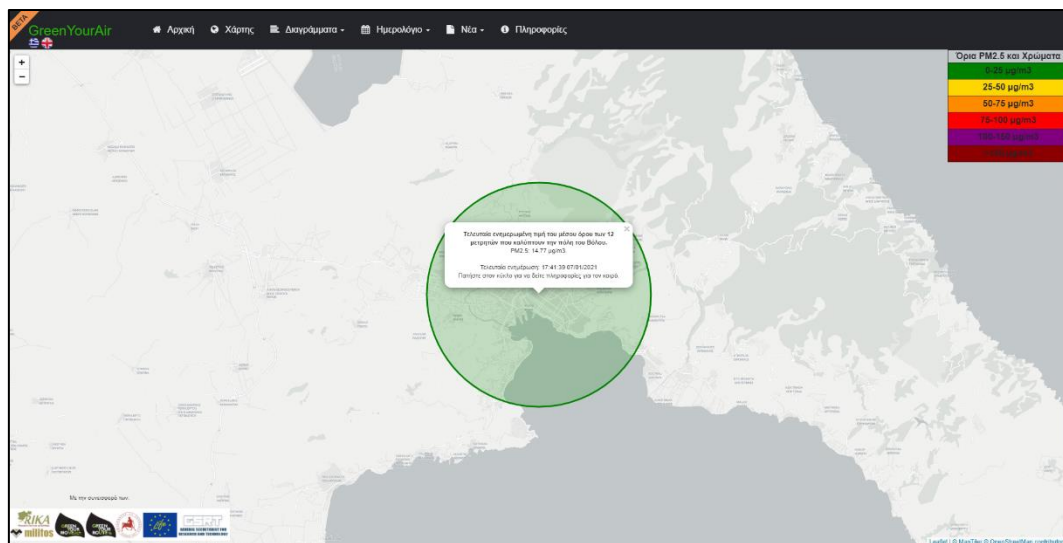


```

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
crossorigin="anonymous"></script>
</body>
</html>

```

Κώδικας 6.7: Κώδικας της ιστοσελίδας για τον "Χάρτη".



Εικόνα 6.7: "Χάρτης" της ιστοσελίδας.

6.2.3 Κώδικας για τα «Διαγράμματα»

Στα «Διαγράμματα» ο κώδικας αποτελείται από το "head" στο οποίο ορίζεται το όνομα της ιστοσελίδας, η σύνδεση με το CSS, η σύνδεση με το Google Analytics στο οποίο λαμβάνουμε δεδομένα για την επισκεψιμότητα της ιστοσελίδας καθώς και όλα τα απαραίτητα στοιχεία για την σωστή λειτουργία της ιστοσελίδας. Στο "body" ορίζεται το navigation bar της ιστοσελίδας και τα dropdown menu όπου είναι απαραίτητα. Επίσης ορίζονται τα διαγράμματα και ο τρόπος με τον οποίο διαβάζονται τα δεδομένα που είναι αποθηκευμένα στον sever. Στο κομμάτι αυτό δεν εισάγουμε το API αμέσως στην ιστοσελίδα αλλά αποθηκεύουμε τα δεδομένα στο server και από εκεί τα διαβάζουμε.

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

<link rel="icon" href="../../GYA_icon.png">
<title>GreenYourAir</title>
<link rel="stylesheet" href="../../style.css">
<link rel="stylesheet" href="../../navbar.css">
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
<link
  rel="stylesheet"
  href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
/>
<link
  rel="stylesheet"
  href="../../fullcalendar/jquery-ui.css"
/>

<!-- Global site tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?id=UA-168804910-
1"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());

  gtag('config', 'UA-168804910-1');
</script>

<script src="../../fullcalendar/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<script src="../../fullcalendar/jquery-ui.js"></script>
<script src="http://static.fusioncharts.com/code/latest/fusioncharts.js"></script>
</head>

<body>
  <header>
    <nav class="navbar navbar-default" >
      <div class="container-fluid">

        <div class="navbar-header">
          <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
            <i class="fa fa-th-large"></i>
            Μενού
          </button>
          <a class="navbar-brand" href="/">GreenYourAir</a>
        </div>

        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
          <ul class="nav navbar-nav">
            <li class="nav-item">
              <a class="nav-link" href="/">
                <i class="fa fa-home"></i>
                Αρχική
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/gr/mapgr">
                <i class="fa fa-globe"></i>
                Χάρτης
              </a>
            </li>
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
                <i class="fa fa-align-left"></i>
                Διαγράμματα
                <span class="caret"></span>
              </a>
              <ul class="dropdown-menu">
                <li><a href="/gr/chartsgr/volos">
                  <i class="fa fa-map-marker"></i>
                  Βόλος
                </a></li>
              </ul>
            </li>
            <li class="dropdown">

```

```

<a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
  <i class="fa fa-calendar"></i>
  Ημερολόγιο
  <span class="caret"></span>
</a>
<ul class="dropdown-menu">
  <li><a href="/gr/calendar/voles">
    <i class="fa fa-map-marker"></i>
    Βόλος
  </a></li>
</ul>
</li>
<li class="dropdown">
  <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
    <i class="fa fa-file"></i>
    Νέα
    <span class="caret"></span>
  </a>
  <ul class="dropdown-menu">
    <li><a href="/gr/newsfeedgr/volesnewsfeed">
      <i class="fa fa-map-marker"></i>
      Βόλος
    </a></li>
  </ul>
</li>
<li class="nav-item">
  <a class="nav-link" href="/gr/about">
    <i class="fa fa-info-circle"></i>
    Πληροφορίες
  </a>
</li>
</ul>
</div>
<div class="flag">
  <a href="/gr/chartsggr/voles"></a>
  <a href="/en/chartsen/voles"></a>
</div>
</div>
</nav>
</header>

<script type="text/javascript" src="../../exportTable.js"></script>
<script type="text/javascript">

```

```

  async function measurements() {

```

```

    var pm25 = await exportTable();

```

```

    var counter = 0;
    var input = [];
    for (var i = 0; i < 456; i++) {
      if (Number.isInteger(i / 19)) {
        input.push({ "label": pm25[479 - counter], "value": pm25[i] });
        counter += 1;
      } else {
        input.push({ "value": pm25[i] });
      }
    };

```

```

    var sum = 0;

```

```

    for (var i = 0; i < 456; i++) {
      sum += parseFloat(pm25[i]);
    };

```

```

    var meanpm25 = (sum / 456).toFixed(2);

```

```

    var min = parseFloat(pm25[455]);
    var max = parseFloat(pm25[455]);

```

```

    for (let i = 0; i < 456; i++) {

```

```

        if (parseFloat(pm25[i]) < min) {
            min =parseFloat(pm25[i])
        }

        if (parseFloat(pm25[i]) > max) {
            max = parseFloat(pm25[i]);
        };
    };

    document.getElementById("min").innerHTML = min + " \u03BCg/m3";
    document.getElementById("max").innerHTML = max + " \u03BCg/m3";
    document.getElementById("tmin").innerHTML = pm25[482];
    document.getElementById("tmax").innerHTML = pm25[481];

    var h1 = 0;
    var h2 = 0;
    var h6 = 0;
    var h12 = 0;
    var h18 = 0;

    for (var i = 455; i > 436; i--) {
        h1 += parseFloat(pm25[i]);
    };

    for (var i = 455; i > 417; i--) {
        h2 += parseFloat(pm25[i]);
    };

    for (var i = 455; i > 341; i--) {
        h6 += parseFloat(pm25[i]);
    };

    for (var i = 455; i > 227; i--) {
        h12 += parseFloat(pm25[i]);
    };

    for (var i = 455; i > 113; i--) {
        h18 += parseFloat(pm25[i]);
    };

    h1 = (h1 / 19).toFixed(2);
    h2 = (h2 / 38).toFixed(2);
    h6 = (h6 / 114).toFixed(2);
    h12 = (h12 / 228).toFixed(2);
    h18 = (h18 / 342).toFixed(2);

    document.getElementById("h1").innerHTML = h1 + " \u03BCg/m3";
    document.getElementById("h2").innerHTML = h2 + " \u03BCg/m3";
    document.getElementById("h6").innerHTML = h6 + " \u03BCg/m3";
    document.getElementById("h12").innerHTML = h12 + " \u03BCg/m3";
    document.getElementById("h18").innerHTML = h18 + " \u03BCg/m3";
    document.getElementById("h24").innerHTML = meanpm25 + " \u03BCg/m3";

    const dataSource = {
        chart: {
            caption: "Βόλος (Τιμές PM2.5 για τις τελευταίες 24 ώρες).",
            subcaption: "Μέσος όρος των 12 μετρητών που καλύπτουν την πόλη του Βόλου για τις τελευταίες 24 ώρες: " + meanpm25 + " \u03BCg/m3. (Ημερίσιο όριο Π.Ο.Υ.: 25.00 \u03BCg/m3).",
            yaxismname: "PM2.5",
            xaxisname: "Τελευταία Ενημέρωση: " + pm25[479] + " " + pm25[480] + "<br>Τελευταία ενημερωμένη τιμή του μέσου όρου των 12 μετρητών που καλύπτουν την πόλη του Βόλου: " + pm25[455] + " \u03BCg/m3",
            anchorradius: "5",
            plottoolttext: "PM2.5: <b>$value</b> \u03BCg/m3",
            showhovereffect: "10",
            showvalues: "0",
            labelstep: "19",
            labelDisplay: "rotate",
            numVDivLines: "23",
            vDivLineColor: "#24252A",
            vDivLineDashed: "1",
            vDivLineDashGap: "5",

```

```

        numbersuffix: "\u03BCg/m3",
        theme: "fusion",
        anchorbgcolor: "#24252A",
        palettecolors: "#24252A",
        captionFontSize: "18",
        captionFont: "Montserrat, sans-serif",
        xaxisNameFontSize: "16",
        xaxisNameFont: "Montserrat, sans-serif",
        yaxisNameFontSize: "15",
        yaxisNameFont: "Montserrat, sans-serif",
        drawAnchors : "0",
        showtooltip: "0"
    },
    data: input
};

FusionCharts.ready(function () {
    var myChart = new FusionCharts({
        type: "spline",
        renderAt: "chart_container",
        width: "100%",
        height: "280%",
        dataFormat: "json",
        dataSource
    }).render();
});

messurements();
</script>

<div class="popup1">
    <button type="button" class="btn1" data-toggle="modal" data-target="#myModal1"
style="font-family: Arial, sans-serif; color:#edf0f1;
background-color: #24252A; border-color:#24252A"><i class="fa fa-th-
large"></i>Min/Max Values</button>
    <div class="modal fade" id="myModal1" role="dialog">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <button type="button" class="close" data-
dismiss="modal">&times;</button>
                    <h4 class="modal-title">Min/Max Values</h4>
                </div>
                <div class="popup1-body">
                    <table class="voloscharts">
                        <tr>
                            <th class="thvoloscharts">Τελευταίες 24 ώρες</th>
                            <th class="thvoloscharts">Τιμές των PM2.5</th>
                            <th class="thvoloscharts" width="100px">Ωρα</th>
                        </tr>
                        <tr>
                            <th class="thvoloscharts">Ελάχιστη Τιμή </th>
                            <th class="thvoloscharts" id = "min"></th>
                            <th class="thvoloscharts" id = "tmin"></th>
                        </tr>
                        <tr>
                            <th class="thvoloscharts">Μέγιστη Τιμή</th>
                            <th class="thvoloscharts" id = "max"></th>
                            <th class="thvoloscharts" id = "tmax"></th>
                        </tr>
                    </table>
                </div>
            </div>
        </div>
    </div>

    <div class="popup2">
        <button type="button" class="btn3" data-toggle="modal" data-target="#myModal"
style="font-family: Montserrat, sans-serif; color:#edf0f1;
background-color: #24252A; border-color:#24252A"><i
class="fa fa-history"></i>Time Table</button>
        <div class="modal fade" id="myModal" role="dialog">
            <div class="modal-dialog">
                <div class="modal-content">
                    <div class="modal-header">

```

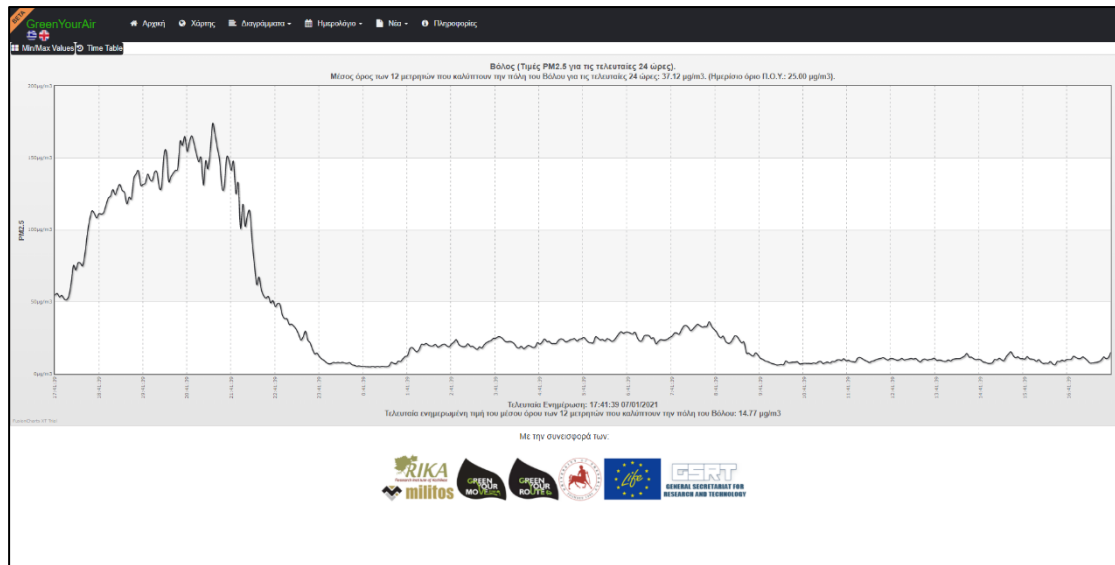
```

<button type="button" class="close" data-
dismiss="modal">&times;</button>
<h4 class="modal-title">Time Table</h4>
</div>
<div class="popup2-body">
<table class="voloscharts">
<tr>
<th class="thvoloscharts" width= "200px">Χρονικά Όρια</th>
<th class="thvoloscharts" width= "200px">Μέση τιμή PM2.5</th>
</tr>
<tr>
<th class="thvoloscharts">Τελευταία ώρα:</th>
<th class="thvoloscharts" id="h1"></th>
</tr>
<tr>
<th class="thvoloscharts">Τελευταίες 2 ώρες:</th>
<th class="thvoloscharts" id="h2"></th>
</tr>
<tr>
<th class="thvoloscharts">Τελευταίες 6 ώρες:</th>
<th class="thvoloscharts" id="h6"></th>
</tr>
<tr>
<th class="thvoloscharts">Τελευταίες 12 ώρες:</th>
<th class="thvoloscharts" id="h12"></th>
</tr>
<tr>
<th class="thvoloscharts">Τελευταίες 18 ώρες:</th>
<th class="thvoloscharts" id="h18"></th>
</tr>
<tr>
<th class="thvoloscharts">Τελευταίες 24 ώρες:</th>
<th class="thvoloscharts" id="h24"></th>
</tr>
</table>
</div>
</div>
</div>
</div>
<div id="chart_container">
<p>Chart</p>
</div>
<p class="logostext1" align="middle">
Με την συνεισφορά των:
</p>
<p class="logos1" align="middle">

</p>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVbN+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
crossorigin="anonymous"></script>
</body>
</html>

```

Κώδικας 6.8: Κώδικας της ιστοσελίδας για τα “Διαγράμματα”.



Εικόνα 6.8: "Διαγράμματα" της ιστοσελίδας.

6.2.4 Κώδικας για το «Ημερολόγιο»

Στο «Ημερολόγιο» ο κώδικας αποτελείται από το “head” στο οποίο ορίζεται το όνομα της ιστοσελίδας, η σύνδεση με το CSS, η σύνδεση με το Google Analytics στο οποίο λαμβάνουμε δεδομένα για την επισκεψιμότητα της ιστοσελίδας καθώς και όλα τα απαραίτητα στοιχεία για την σωστή λειτουργία της ιστοσελίδας. Στο “body” ορίζεται το navigation bar της ιστοσελίδας και τα dropdown menu όπου είναι απαραίτητο. Επίσης ορίζεται με τον οποίο διαβάζονται τα δεδομένα που είναι αποθηκευμένα στον sever για την δημιουργία του ημερολογίου. Στο κομμάτι αυτό δεν εισάγουμε το API αμέσως στην ιστοσελίδα αλλά αποθηκεύουμε τα δεδομένα στο server και από εκεί τα διαβάζουμε.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta property="og:image" content="../../GYA_icon.png">
    <meta property="og:image:type" content="image/png">
    <meta property="og:image:width" content="1024">
    <meta property="og:image:height" content="1024">
    <link rel="icon" type="image/png" href="../../GYA_icon.png">
    <title>GreenYourAir</title>
    <link
      href="../../fullcalendar/fullcalendar.min.css"
      rel="stylesheet"
    />
    <link
      rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
    />
    <link
      href="../../fullcalendar/fullcalendar.print.min.css"
      rel="stylesheet"
      media="print"
    />
    <link
  
```

```

    rel="stylesheet"
    href="../../fullcalendar/jquery-ui.css"
  />
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">

  <!-- Global site tag (gtag.js) - Google Analytics -->
  <script async src="https://www.googletagmanager.com/gtag/js?id=UA-168804910-
1"></script>
  <script>
    window.dataLayer = window.dataLayer || [];
    function gtag(){dataLayer.push(arguments);}
    gtag('js', new Date());

    gtag('config', 'UA-168804910-1');
  </script>

  <script src="../../fullcalendar/jquery.min.js"></script>
  <script src="../../fullcalendar/moment.min.js"></script>
  <script src="../../fullcalendar/fullcalendar.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  <script src="../../fullcalendar/sweetalert.min.js"></script>
  <script src="../../fullcalendar/jquery-ui.js"></script>

  <script type="module" src="../../fullcalendar/fullcalendar.js"></script>
  <link rel="stylesheet" href="../../fullcalendar/fullcalendar.css" />
</head>
<body>
  <header>
    <nav class="navbar navbar-default" >
      <div class="container-fluid">

        <div class="navbar-header">
          <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
            <i class="fa fa-th-large"></i>
            Μενού
          </button>
          <a class="navbar-brand" href="/">GreenYourAir</a>
        </div>

        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
          <ul class="nav navbar-nav">
            <li class="nav-item">
              <a class="nav-link" href="/">
                <i class="fa fa-home"></i>
                Αρχική
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/gr/mapgr">
                <i class="fa fa-globe"></i>
                Χάρτης
              </a>
            </li>
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
                <i class="fa fa-align-left"></i>
                Διαγράμματα
                <span class="caret"></span>
              </a>
              <ul class="dropdown-menu">
                <li><a href="/gr/chartsg/volos">
                  <i class="fa fa-map-marker"></i>
                  Βόλος
                </a></li>
              </ul>
            </li>
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
                <i class="fa fa-calendar"></i>
                Ημερολόγιο
                <span class="caret"></span>
              </a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </header>

```



```

        </a>
        <ul class="dropdown-menu">
          <li><a href="/gr/calendar/voles">
            <i class="fa fa-map-marker"></i>
            Βόλος
          </a></li>
        </ul>
      </li>
      <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
          <i class="fa fa-file"></i>
          Νέα
          <span class="caret"></span>
        </a>
        <ul class="dropdown-menu">
          <li><a href="/gr/newsfeedgr/volesnewsfeed">
            <i class="fa fa-map-marker"></i>
            Βόλος
          </a></li>
        </ul>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="/gr/about">
          <i class="fa fa-info-circle"></i>
          Πληροφορίες
        </a>
      </li>
    </ul>
  </div>
  <div class="flag">
    <a href="/gr/calendar/voles"></a>
    <a href="/en/calendar/voles"></a>
  </div>
</div>
</nav>
</header>

<h1 align="center"></h1>

<div id="calendar"></div>

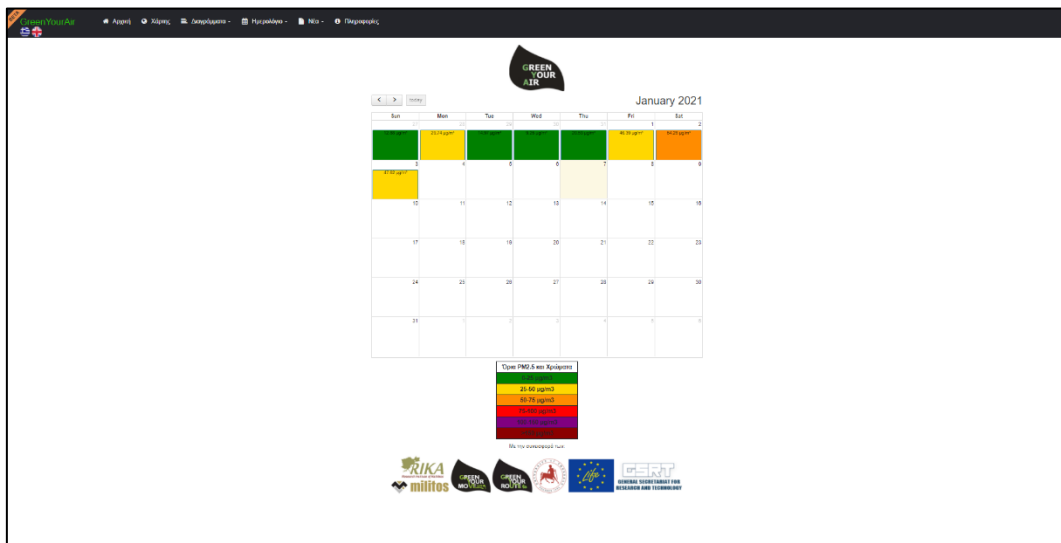
<table id = "calmatrixcolors">
  <tr>
    <th id = "matrixheaders" >Όρια PM2.5 και Χρώματα</th>
  </tr>
  <tr>
    <td id = "pm0_25">0-25 µg/m3</td>
  </tr>
  <tr>
    <td id = "pm25_50">25-50 µg/m3</td>
  </tr>
  <tr>
    <td id="pm50_75">50-75 µg/m3</td>
  </tr>
  <tr>
    <td id="pm75_100">75-100 µg/m3</td>
  </tr>
  <tr>
    <td id="pm100_150">100-150 µg/m3</td>
  </tr>
  <tr>
    <td id="pm150">>150 µg/m3</td>
  </tr>
</table>
<br>
<p class="logostext" align = "middle">
  Με την συνεισφορά των:
</p>

<p class="logos" align = "middle"></p>
</body>

```

</html>

Κώδικας 6.9: Κώδικας της ιστοσελίδας για το "Ημερολόγιο".



Εικόνα 6.9: "Ημερολόγιο" της ιστοσελίδας.

6.2.5 Κώδικας για τα «Νέα»

Στο «Ημερολόγιο» ο κώδικας αποτελείται από το "head" στο οποίο ορίζεται το όνομα της ιστοσελίδας, η σύνδεση με το CSS, η σύνδεση με το Google Analytics στο οποίο λαμβάνουμε δεδομένα για την επισκεψιμότητα της ιστοσελίδας καθώς και όλα τα απαραίτητα στοιχεία για την σωστή λειτουργία της ιστοσελίδας. Στο "body" ορίζεται το navigation bar της ιστοσελίδας και τα dropdown menu όπου είναι απαραίτητο. Επίσης ορίζονται τα δεδομένα του πίνακα που περιέχουν τα νέα που δημοσιεύονται, καθώς και τα Pdf αρχεία τα οποία δημοσιεύονται.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta property="og:image" content="../../../GYA_icon.png">
  <meta property="og:image:type" content="image/png">
  <meta property="og:image:width" content="1024">
  <meta property="og:image:height" content="1024">
  <link rel="icon" type="image/png" href="../../../GYA_icon.png">
  <title>GreenYourAir</title>
  <link rel="stylesheet" href="../../../style.css">
  <link rel="stylesheet" href="../../../navbar.css">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
  <link rel="stylesheet" href="../../../fullcalendar/jquery-ui.css" />

  <!-- Global site tag (gtag.js) - Google Analytics -->
  <script async src="https://www.googletagmanager.com/gtag/js?id=UA-168804910-
1"></script>
```

```

<script>
  window.dataLayer = window.dataLayer || [];
  function gtag() { dataLayer.push(arguments); }
  gtag('js', new Date());

  gtag('config', 'UA-168804910-1');
</script>

<script src='../././fullcalendar/jquery.min.js'></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<script src='../././fullcalendar/jquery-ui.js'></script>
<link rel="stylesheet"
href="https://cdn.datatables.net/1.10.19/css/jquery.dataTables.min.css">
<script
src="https://cdn.datatables.net/1.10.19/js/jquery.dataTables.min.js"></script>
</head>

<body>
  <header>
    <nav class="navbar navbar-default">
      <div class="container-fluid">

        <div class="navbar-header">
          <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse"
          data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
            <i class="fa fa-th-large"></i>
            Μενού
          </button>
          <a class="navbar-brand" href="/">GreenYourAir</a>
        </div>

        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
          <ul class="nav navbar-nav">
            <li class="nav-item">
              <a class="nav-link" href="/">
                <i class="fa fa-home"></i>
                Αρχική
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/gr/mapgr">
                <i class="fa fa-globe"></i>
                Χάρτης
              </a>
            </li>
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
aria-expanded="false">
                <i class="fa fa-align-left"></i>
                Διαγράμματα
                <span class="caret"></span>
              </a>
              <ul class="dropdown-menu">
                <li><a href="/gr/chartsgr/voles">
                  <i class="fa fa-map-marker"></i>
                  Βόλος
                </a></li>
              </ul>
            </li>
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
aria-expanded="false">
                <i class="fa fa-calendar"></i>
                Ημερολόγιο
                <span class="caret"></span>
              </a>
              <ul class="dropdown-menu">
                <li><a href="/gr/calendar/voles">
                  <i class="fa fa-map-marker"></i>
                  Βόλος
                </a></li>
              </ul>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </header>

```

```

</li>
<li class="dropdown">
  <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
  aria-expanded="false">
    <i class="fa fa-file"></i>
    Νέα
    <span class="caret"></span>
  </a>
  <ul class="dropdown-menu">
    <li><a href="/gr/newsfeedgr/volosnewsfeed">
      <i class="fa fa-map-marker"></i>
      Βόλος
    </a></li>
  </ul>
</li>
<li class="nav-item">
  <a class="nav-link" href="/gr/about">
    <i class="fa fa-info-circle"></i>
    Πληροφορίες
  </a>
</li>
</ul>
</div>
<div class="flag">
  <a href="/gr/newsfeedgr/volosnewsfeed"></a>
  <a href="/en/newsfeeden/volosnewsfeed"></a>
</div>
</div>
</nav>
</header>

<h1 align="center">
  
</h1>
<p class="news" align="middle">Τελευταία Νέα και Δημοσιεύσεις</p>

<div class="container" id="pagingnews" style="max-width: 90%">
  <table class="table table-fluid" id="myTable">
    <thead>
      <tr class="headernews">
        <th class="no-sort">Πληροφορίες</th>
        <th>Ημερομηνία Δημοσίευσης</th>
        <th class="no-sort">Αρχεία</th>
      </tr>
    </thead>
    <tbody>
      <tr class="entries">
        <td>Εβδομαδιαίο Δελτίο Τύπου: Ημερήσιοι μέσοι όροι των PM2.5 για την
περίοδο από 06/07/2020 έως 12/07/2020 για
την πόλη του Βόλου. Περισσότερες πληροφορίες στο συνημμένο αρχείο.</td>
        <td>13/07/2020</td>
        <td>
          <a href="NewsFeedFilesGR/volos/GYA_Volos_06_07_2020-12_07_2020.gr.pdf"
target="_blank" class="card-link"><i
            class="fa fa-file"></i>Gr</a>
        </td>
      </tr>
      <tr class="entries">
        <td>Εβδομαδιαίο Δελτίο Τύπου: Ημερήσιοι μέσοι όροι των PM2.5 για την
περίοδο από 29/06/2020 έως 05/07/2020 για
την πόλη του Βόλου. Περισσότερες πληροφορίες στο συνημμένο αρχείο.</td>
        <td>06/07/2020</td>
        <td>
          <a href="NewsFeedFilesGR/volos/GYA_Volos_29_06_2020-05_07_2020.gr.pdf"
target="_blank" class="card-link"><i
            class="fa fa-file"></i>Gr</a>
        </td>
      </tr>
    </tbody>
  </table>

```

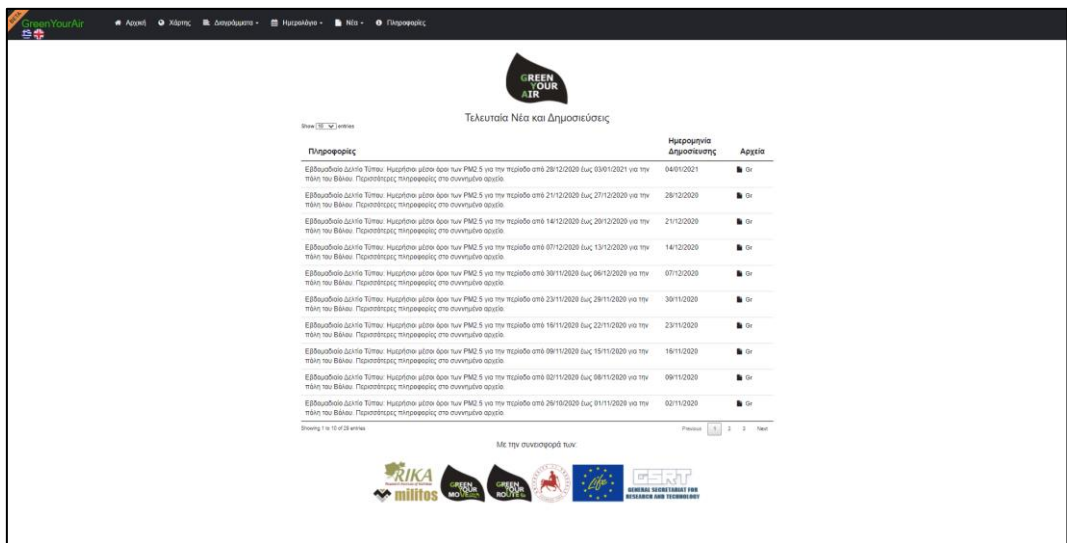
```

<td>Εβδομαδιαίο Δελτίο Τύπου: Ημερήσιοι μέσοι όροι των PM2.5 για την
περίοδο από 22/06/2020 έως 28/06/2020 για
την πόλη του Βόλου. Περισσότερες πληροφορίες στο συνημμένο αρχείο.</td>
<td>29/06/2020</td>
<td>
<a href="NewsFeedFilesGR/volos/GYA_Volos_22_06_2020-28_06_2020.gr.pdf"
target="_blank" class="card-link"><i
class="fa fa-file"></i>Gr</a>
</td>
</tr>
<tr class="entries">
<td>Δελτίο Τύπου: Ετήσιος και μηνιαίοι μέσοι όροι των PM2.5 για την
περίοδο από 01/03/2019 έως 29/02/2020 για
την πόλη του Βόλου. Περισσότερες πληροφορίες στο συνημμένο αρχείο.</td>
<td>15/06/2020</td>
<td>
<a href="NewsFeedFilesGR/volos/GYA_Volos_Press_Release_June_2020.pdf"
target="_blank" class="card-link">
<i class="fa fa-file"></i>
Gr
</a>
</td>
</tr>
</tbody>
</table>
</div>
<script>
$.fn.DataTable.ext.pager.numbers_length = 4;
$(document).ready(function () {
$('#myTable').DataTable({
"bSort": false, "bFilter": false, "lengthMenu": [[10, 25, 50, 100, -1], [10,
25, 50, 100, "All"]]
});
</script>
<p class="logostext" align="middle">
Με την συνεισφορά των:
</p>
<p class="logos" align="middle">

</p>
</body>
</html>

```

Κώδικας 6.10: Κώδικας της ιστοσελίδας για τα "Νέα".



Εικόνα 6.10: "Νέα" της ιστοσελίδας

6.2.6 Κώδικας για τις «Πληροφορίες»

Στο «Ημερολόγιο» ο κώδικας αποτελείται από το “head” στο οποίο ορίζεται το όνομα της ιστοσελίδας, η σύνδεση με το CSS, η σύνδεση με το Google Analytics στο οποίο λαμβάνουμε δεδομένα για την επισκεψιμότητα της ιστοσελίδας καθώς και όλα τα απαραίτητα στοιχεία για την σωστή λειτουργία της ιστοσελίδας. Στο “body” ορίζεται το navigation bar της ιστοσελίδας και τα dropdown menu όπου είναι απαραίτητο. Επίσης ορίζονται οι πληροφορίες που απεικονίζονται καθώς και οι συνδεσιμότητες με άλλα site τα οποία αναφέρονται σε αυτή την κατηγορία.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta property="og:image" content="../GYA_icon.png">
  <meta property="og:image:type" content="image/png">
  <meta property="og:image:width" content="1024">
  <meta property="og:image:height" content="1024">
  <link rel="icon" href="../GYA_icon.png">
  <title>GreenYourAir</title>
  <link rel="stylesheet" href="../style.css">
  <link rel="stylesheet" href="../navbar.css">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
  <link
    rel="stylesheet"
    href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
  />
  <link
    rel="stylesheet"
    href="../fullcalendar/jquery-ui.css"
  />

  <!-- Global site tag (gtag.js) - Google Analytics -->
  <script async src="https://www.googletagmanager.com/gtag/js?id=UA-168804910-
1"></script>
  <script>
    window.dataLayer = window.dataLayer || [];
    function gtag(){dataLayer.push(arguments);}
    gtag('js', new Date());

    gtag('config', 'UA-168804910-1');
  </script>

  <script src='../fullcalendar/jquery.min.js'></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  <script src="../fullcalendar/jquery-ui.js"></script>
</head>

<body>
  <header>
    <nav class="navbar navbar-default" >
      <div class="container-fluid">

        <div class="navbar-header">
          <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
            <i class="fa fa-th-large"></i>
            Μεvoú
          </button>
          <a class="navbar-brand" href="/">GreenYourAir</a>
        </div>

        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
          <ul class="nav navbar-nav">
```

```

<li class="nav-item">
  <a class="nav-link" href="/">
    <i class="fa fa-home"></i>
    Αρχική
  </a>
</li>
<li class="nav-item">
  <a class="nav-link" href="/gr/mapgr">
    <i class="fa fa-globe"></i>
    Χάρτης
  </a>
</li>
<li class="dropdown">
  <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
    <i class="fa fa-align-left"></i>
    Διαγράμματα
    <span class="caret"></span>
  </a>
  <ul class="dropdown-menu">
    <li><a href="/gr/chartsgr/volos">
      <i class="fa fa-map-marker"></i>
      Βόλος
    </a></li>
  </ul>
</li>
<li class="dropdown">
  <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
    <i class="fa fa-calendar"></i>
    Ημερολόγιο
    <span class="caret"></span>
  </a>
  <ul class="dropdown-menu">
    <li><a href="/gr/calendar/volos">
      <i class="fa fa-map-marker"></i>
      Βόλος
    </a></li>
  </ul>
</li>
<li class="dropdown">
  <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
    <i class="fa fa-file"></i>
    Νέα
    <span class="caret"></span>
  </a>
  <ul class="dropdown-menu">
    <li><a href="/gr/newsfeedgr/volosnewsfeed">
      <i class="fa fa-map-marker"></i>
      Βόλος
    </a></li>
  </ul>
</li>
<li class="nav-item">
  <a class="nav-link" href="/gr/about">
    <i class="fa fa-info-circle"></i>
    Πληροφορίες
  </a>
</li>
</ul>
</div>
<div class="flag">
  <a href="/gr/about"></a>
  <a href="/en/about"></a>
</div>
</div>
</nav>
</header>

<h1 align="center"></h1>

<p class = "mie" align = "middle">

```

```

<u>Πληροφορίες:</u>
</p>
<p class = "mail" align = "middle">
  Email: greenyourair.uth@gmail.com
</p>
<p class = "pj" align = "middle">
  <u>
    Διαχειριστές Έργου:
  </u>
  <br>
  </img>
  <br>
  Δρ. Γεώργιος Κ.Δ. Σαχαρίδης,</br>
  <i><small>Τμήμα Μηχανολόγων Μηχανικών</small></i>
  Πανεπιστήμιο Θεσσαλίας</small></i>
  <br>
  </img>
  <br>
  Δρ. Κωνσταντίνος Γουργουλιάνης,</br>
  <i><small>Διευθυντής Πνευμονολογικής Κλινικής</small></i>
  του Πανεπιστημιακού Νοσοκομείου Λάρισας</small></i>
  <br>
  </img>
  <br>
  Γεώργιος Ι. Καλαντζής,</br>
  <i><small>Ερευνητικό Ινστιτούτο Κάθικα</small></i>
  <br>
  </img>
  <br>
  Φίλιππος Καραμανλής,</br>
  <i><small>Μίλητος Συμβουλευτική Α.Ε.</small></i>
  <br>
</p>
<p class = "team" align = "middle">
  <u>
    Ερευνητική Ομάδα:
  </u>
  <br>
  </img>
  <br>
  Νεΐλος Ψαθάς</br>
  <i><small>Πανεπιστήμιο Θεσσαλίας</small></i>
  <br>
  </img>
  <br>
  Κυριάκος Ζήσιος</br>
  <i><small>Πανεπιστήμιο Θεσσαλίας</small></i>
</p>
<p class = "fb" align = "middle">
  <a id = "fbp" href = "https://www.facebook.com/GreenYourAirAction/" target =
  "_blank">
    Ακολουθήστε μας στο Facebook
  </a>
</p>

<p class = "sites" align = "middle"><u>Ιστοσελίδες Συνεργατών:</u></p>
<p align = "middle"><a id = "csites" href="http://mie.uth.gr/" target =
  "_blank">Τμήμα Μηχανολόγων Μηχανικών, Π.Θ.</a></p>
<p align = "middle"><a id = "csitesgreen" href="http://www.greenyourmove.eu/"
  target = "_blank">GreenYourMove Platform</a></p>
<p align = "middle"><a id = "csitesgreen" href="https://www.greenyourmove.org/"
  target = "_blank">GreenYourMove Site</a></p>
<p align = "middle"><a id = "csitesgreen" href="https://www.greenyourroute.com/"
  target = "_blank">GreenYourRoute</a></p>
<p align = "middle"><a id = "csites" href="https://www.rikalabs.com/" target =
  "_blank">Ερευνητικό Ινστιτούτο Κάθικα</a></p>
<p align = "middle"><a id = "csites" href="http://www.militos.org/en/" target =
  "_blank">Μίλητος Συμβουλευτική Α.Ε.</a></p>
<p align = "middle"><a id = "csites" href="https://www.uth.gr/" target =
  "_blank">Πανεπιστήμιο Θεσσαλίας</a></p>
<p align = "middle"><a id = "csites" href="https://ec.europa.eu/easme/en/life"
  target = "_blank">LIFE</a></p>
<p align = "middle"><a id = "csites"
  href="http://www.gsrt.gr/central.aspx?sId=119I428I1089I646I488772" target =
  "_blank">ΓΓΕΤ</a></p>

```



```

<p class="logostexthome" align = "middle">
  Με την συνεισφορά των:
</p>

<p class="logoshome" align = "middle"></p>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBN+IbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kr0JKI"
crossorigin="anonymous"></script>
</body>
</html>

```

Κώδικας 6.11: Κώδικας της ιστοσελίδας για τις “Πληροφορίες”.

6.2.7 Κώδικας του CSS

Ο κώδικας του CSS είναι απαραίτητος για την εμφάνιση της ιστοσελίδας. Στον κώδικα ορίζεται το πως θα εμφανίζεται κάθε διαφορετικό στοιχείο της ιστοσελίδας ανάλογα την συσκευή που χρησιμοποιεί ο χρήστης.

```

@import url("https://fonts.googleapis.com/css?family=Arial:500&display=swap");

* {
  margin: 0px;
  padding: 0px;
}

body {
  background-size: cover;
}

i {
  display: inline-block;
  margin-right: 0.5em;
}

h1 {
  position: relative;
  font-family: "Arial", sans-serif;
  font-weight: 500;
  font-size: 70px;
  color: #24252a;
  text-decoration: none;
  top: 10px;
}

.first {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 18px;
  color: #24252a;
  text-decoration: none;
  position: relative;
  top: 30px;
  padding: 0px 30px;
  text-align: center;
}

```

```

}

.second {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 18px;
  color: #24252a;
  text-decoration: none;
  position: relative;
  top: 60px;
  padding: 0px 30px;
  text-align: center;
}

.third {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 18px;
  color: #24252a;
  text-decoration: none;
  position: relative;
  top: 90px;
  padding: 0px 30px;
  text-align: center;
}

.fourth {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 18px;
  color: #24252a;
  text-decoration: none;
  position: relative;
  top: 120px;
  padding: 0px 30px;
  text-align: center;
}

.mie {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 25px;
  color: #24252a;
  text-decoration: none;
  text-align: center;
  position: relative;
  top: 25px;
}

.mail {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 18px;
  color: #24252a;
  text-decoration: none;
  position: relative;
  top: 25px;
  padding: 10px 60px;
}

.pj {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 18px;
  color: #24252a;
  text-decoration: none;
  position: relative;
  top: 25px;
  padding: 10px 60px;
}

.Managers {
  padding-top: 20px;
  width: 200px;
  height: auto;
}

```

```

.ResearchTeam {
  padding-top: 20px;
  width: 150px;
  height: auto;
}

.team {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 18px;
  color: #24252a;
  text-decoration: none;
  position: relative;
  top: 25px;
  padding: 10px 60px;
}

.sites {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 18px;
  color: #24252a;
  text-decoration: none;
  position: relative;
  top: 80px;
  padding: 10px 60px;
}

#fbp {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 18px;
  color: #0c5cb8;
  text-decoration: none;
  position: relative;
  top: 40px;
  padding: 10px 60px;
}

#fbp:hover {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 18px;
  color: #0088a9;
  position: relative;
  top: 40px;
  padding: 10px 60px;
  text-decoration: underline;
}

#csitesgreen {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 18px;
  color: #29b80c;
  text-decoration: none;
  position: relative;
  top: 80px;
  padding: 10px 60px;
}

#csitesgreen:hover {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 18px;
  color: #0088a9;
  position: relative;
  top: 80px;
  padding: 10px 60px;
  text-decoration: underline;
}

#csites {
  font-family: "Arial", sans-serif;
  font-weight: 200;

```

```

font-size: 18px;
color: #24252a;
text-decoration: none;
position: relative;
top: 80px;
padding: 10px 60px;
}

#csites:hover {
font-family: "Arial", sans-serif;
font-weight: 200;
font-size: 18px;
color: #0088a9;
position: relative;
top: 80px;
padding: 10px 60px;
text-decoration: underline;
}

.logostext {
font-family: "Arial", sans-serif;
font-weight: 200;
font-size: 18px;
color: #24252a;
text-decoration: none;
position: relative;
padding: 10px 60px;
}

.logos {
position: relative;
}

.logostexthome {
font-family: "Arial", sans-serif;
font-weight: 200;
font-size: 18px;
color: #24252a;
text-decoration: none;
position: relative;
top: 150px;
padding: 10px 60px;
}

.logoshome {
position: relative;
top: 170px;
}

.logos1 {
position: relative;
}

.logostext1 {
font-family: "Arial", sans-serif;
font-weight: 200;
font-size: 18px;
color: #24252a;
text-decoration: none;
position: relative;
padding: 10px 60px;
}

.logos2 {
position: fixed;
bottom: 0px;
}

.logostext2 {
font-family: "Arial", sans-serif;
font-weight: 200;
font-size: 10px;
color: #24252a;
text-decoration: none;
position: fixed;
bottom: 70px;
}

```

```

padding: 0px 110px;
}

.leaflet-popup-content {
text-align: center;
font-size: 11px;
font-family: "Arial", sans-serif;
width: 400px;
}

.leaflet-top .leaflet-control {
margin-top: 20px !important;
}

.btn1 {
float: left;
position: relative;
top: 1px;
left: 1px;
z-index: 5;
font-size: 16px;
border-radius: 5px;
cursor: pointer;
}

.btn3 {
position: relative;
top: 1px;
left: 3px;
z-index: 5;
font-size: 16px;
border-radius: 5px;
cursor: pointer;
}

.modal-content {
top: 100px;
width: 100% !important;
max-width: 400px;
margin: auto;
}

.modal-header {
border: 1px solid #edf0f1;
background-color: #24252a;
border-collapse: collapse;
font-family: "Arial", sans-serif;
font-weight: 500;
font-size: 16px;
color: #edf0f1;
text-decoration: none;
text-align: center;
padding: 5px 10px;
border-radius: 5px 5px 0px 0px;
}

.close {
color: #edf0f1 !important;
}

.voloscharts {
position: relative;
}

.voloscharts,
.thvoloscharts {
border: 1px solid #edf0f1;
background-color: #24252a;
border-collapse: collapse;
font-family: "Arial", sans-serif;
font-weight: 500;
font-size: 16px;
color: #edf0f1;
text-decoration: none;
text-align: center;
padding: 5px 10px;
}

```

```

}

#logos {
  width: 100%;
  max-width: 800px;
  height: auto;
}

#maplogo {
  width: 100%;
  max-width: 400px;
  height: auto;
}

.news {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 25px;
  color: #24252a;
  text-decoration: none;
  text-align: center;
  position: relative;
  top: 15px;
}

.card-link {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 15px;
  color: #24252a;
  text-decoration: none;
}

.card-link:hover {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 15px;
  color: #0088a9;
  position: relative;
  text-decoration: underline !important;
}

.table {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 16px;
  color: #24252a;
  text-decoration: none;
}

.headernews {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 18px;
  color: #24252a;
  text-decoration: none;
}

#pagingnews {
  font-family: "Arial", sans-serif;
  font-weight: 200;
  font-size: 12px;
  color: #24252a;
  text-decoration: none;
}

.entries:hover {
  background-color: #f2f2f2;
}

.calendar-day {
  width: 100px;
  min-width: 100px;
  max-width: 100px;
  height: 80px;
}

```

```

.calendar-table {
  margin: 0 auto;
  width: 700px;
}
.selected {
  background-color: #eee;
}
.outside .date {
  color: #ccc;
}
.timetitle {
  white-space: nowrap;
  text-align: right;
}

.event {
  border-top: 1px solid #000000;
  border-bottom: 1px solid #000000;
  background-image: linear-gradient(to bottom, #dff0d8 0px, #c8e5bc 100%);
  background-repeat: repeat-x;
  color: #3c763d;
  border-width: 1px;
  font-size: 0.75em;
  padding: 0 0.75em;
  line-height: 2em;
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
  margin-bottom: 1px;
}
.event.begin {
  border-left: 1px solid #000000;
  border-top-left-radius: 4px;
  border-bottom-left-radius: 4px;
}
.event.end {
  border-right: 1px solid #000000;
  border-top-right-radius: 4px;
  border-bottom-right-radius: 4px;
}
.event.all-day {
  border-top: 1px solid #9acfea;
  border-bottom: 1px solid #9acfea;
  background-image: linear-gradient(to bottom, #d9edf7 0px, #b9def0 100%);
  background-repeat: repeat-x;
  color: #31708f;
  border-width: 1px;
}
.event.all-day.begin {
  border-left: 1px solid #9acfea;
  border-top-left-radius: 4px;
  border-bottom-left-radius: 4px;
}
.event.all-day.end {
  border-right: 1px solid #9acfea;
  border-top-right-radius: 4px;
  border-bottom-right-radius: 4px;
}
.event.clear {
  background: none;
  border: 1px solid transparent;
}
.table-tight > thead > tr > th,
.table-tight > tbody > tr > th,
.table-tight > tfoot > tr > th,
.table-tight > thead > tr > td,
.table-tight > tbody > tr > td,
.table-tight > tfoot > tr > td {
  padding-left: 0;
  padding-right: 0;
}
.table-tight-vert > thead > tr > th,
.table-tight-vert > tbody > tr > th,
.table-tight-vert > tfoot > tr > th,
.table-tight-vert > thead > tr > td,
.table-tight-vert > tbody > tr > td,

```

```

.table-tight-vert > tfoot > tr > td {
  padding-top: 0;
  padding-bottom: 0;
}

@media screen and (max-width: 450px) {
  #matrixheaders {
    font-size: 11px;
    text-align: center;
    border: 1px solid #24252a;
    text-align: center;
    font-weight: bold;
  }
  #mapmatrixcolors {
    font-family: "Arial", sans-serif;
    font-weight: 200;
    font-size: 11px;
    position: relative;
    color: #24252a;
    text-align: center;
    margin-left: auto;
    margin-right: auto;
    width: 100px;
  }
  #pm0_25 {
    color: #24252a;
    background-color: green;
    border: 1px solid #24252a;
    text-align: center;
    padding: 3px !important;
    font-weight: bold;
  }
  #pm25_50 {
    color: #24252a;
    background-color: gold;
    border: 1px solid #24252a;
    text-align: center;
    padding: 3px !important;
    font-weight: bold;
  }
  #pm50_75 {
    color: #24252a;
    background-color: darkorange;
    border: 1px solid #24252a;
    text-align: center;
    padding: 3px !important;
    font-weight: bold;
  }
  #pm75_100 {
    color: #24252a;
    background-color: red;
    border: 1px solid #24252a;
    text-align: center;
    padding: 3px !important;
    font-weight: bold;
  }
  #pm100_150 {
    color: #24252a;
    background-color: purple;
    border: 1px solid #24252a;
    text-align: center;
    padding: 3px !important;
    font-weight: bold;
  }
  #pm150 {
    color: #24252a;
    background-color: darkred;
    border: 1px solid #24252a;
    text-align: center;
    padding: 3px !important;
  }
}

```



```

    font-weight: bold;
  }
}

@media screen and (min-width: 451px) {
  #matrixheaders {
    font-size: 16px;
    text-align: center;
    border: 1px solid #24252a;
    text-align: center;
    font-weight: bold;
  }
  #mapmatrixcolors {
    font-family: "Arial", sans-serif;
    font-weight: 200;
    font-size: 16px;
    position: relative;
    color: #24252a;
    text-align: center;
    margin-left: auto;
    margin-right: auto;
    width: 220px;
  }

  #pm0_25 {
    color: #24252a;
    background-color: green;
    height: 30px;
    border: 1px solid #24252a;
    text-align: center;
    padding: 3px !important;
    font-weight: bold;
  }

  #pm25_50 {
    color: #24252a;
    background-color: gold;
    height: 30px;
    border: 1px solid #24252a;
    text-align: center;
    padding: 3px !important;
    font-weight: bold;
  }

  #pm50_75 {
    color: #24252a;
    background-color: darkorange;
    height: 30px;
    border: 1px solid #24252a;
    text-align: center;
    padding: 3px !important;
    font-weight: bold;
  }

  #pm75_100 {
    color: #24252a;
    background-color: red;
    height: 30px;
    border: 1px solid #24252a;
    text-align: center;
    padding: 3px !important;
    font-weight: bold;
  }

  #pm100_150 {
    color: #24252a;
    background-color: purple;
    height: 30px;
    border: 1px solid #24252a;
    text-align: center;
    padding: 3px !important;
    font-weight: bold;
  }

  #pm150 {
    color: #24252a;
  }
}

```

```

background-color: darkred;
height: 30px;
border: 1px solid #24252a;
text-align: center;
padding: 3px !important;
font-weight: bold;
}
}

body:after {
content: "beta";
position: absolute;
width: 92px;
height: 24px;
background: #ee8e4a;
top: 8px;
left: -27px;
text-align: center;
font-size: 13px;
font-family: sans-serif;
text-transform: uppercase;
font-weight: bold;
color: #24252a;
line-height: 26px;
transform: rotate(-46deg);
z-index: 105;
}

.navbar {
border: none !important;
border-radius: 0px !important;
background-color: #24252a !important;
margin-bottom: 0px !important;
line-height: 0px;
}

a.navbar-brand {
font-family: "Arial", sans-serif;
font-weight: 500;
font-size: 25px;
color: #29b80c !important;
padding: 25px 35px !important;
padding-right: 60px !important;
text-decoration: none;
float: left;
}

.navbar > .container .navbar-brand,
.navbar > .container-fluid .navbar-brand {
margin-left: 0px;
}

.navbar-default .navbar-collapse,
.navbar-default .navbar-form {
border-color: #24252a !important;
}

.container-fluid > .navbar-collapse,
.container-fluid > .navbar-header,
.container > .navbar-collapse,
.container > .navbar-header {
margin-right: 25px;
margin-left: -15px;
}

.navbar-default .navbar-nav > li > a {
font-family: "Arial", sans-serif;
font-size: 16px;
color: #edf0f1 !important;
padding-top: 24px !important;
padding-bottom: 0px;
}

.navbar-default .navbar-nav > li > a:focus {
color: #edf0f1 !important;
}

```

```

.navbar-default .navbar-nav > li > a:hover {
  color: #0088a9 !important;
}
.navbar-default .navbar-nav > .open > a,
.navbar-default .navbar-nav > .open > a:focus,
.navbar-default .navbar-nav > .open > a:hover {
  color: #0088a9 !important;
  background-color: #24252a !important;
}

.navbar-default .navbar-nav .open .dropdown-menu > li > a {
  color: #edf0f1 !important;
}

.navbar-default .navbar-nav .open .dropdown-menu > li > a:hover {
  background-color: #24252a !important;
  color: #0088a9 !important;
}
.navbar-default .navbar-toggle {
  position: relative;
  font-size: 2rem;
  top: 9px;
  color: #edf0f1 !important;
  background-color: #24252a !important;
  border-color: #24252a !important;
}

.navbar-default .navbar-toggle:focus {
  background-color: #24252a !important;
}

.open > .dropdown-menu {
  background-color: #24252a;
  display: block;
}

.dropdown-menu > li > a {
  color: #edf0f1;
}

.flag {
  position: relative;
  top: -2px;
  padding: 0px 20px;
}

```

Κώδικας 6.12: Κώδικας CSS της ιστοσελίδας.

6.2.8 Κώδικας για την κλήση και την αποθήκευση των δεδομένων του API

Όλοι οι μέσοι όροι των μετρήσεων του τελευταίου 24ώρου αποθηκεύονται πρώτα στον server και μετά τροφοδοτούν την ιστοσελίδα. Αυτό γίνεται για να μην είναι ορατό το API στον χρήστη ώστε να προστατεύονται τα δεδομένα και για να μην γίνονται άσκοπες κλήσεις το API αν δεν υπάρχει κάποιος χρήστης στην πλατφόρμα διότι υπάρχει μέγιστο όριο κλήσεων στην δωρεάν έκδοση του Apps script. Έτσι κάθε φορά που εισέρχεται ένας χρήστης στην ιστοσελίδα ελέγχεται το timestamp της προηγούμενης κλήσης στο API. Αν έχει περάσει πάνω από ένα λεπτό από την τελευταία κλήση στο API τότε γίνεται νέα κλήση στο API και τα δεδομένα ανανεώνονται. Η διαδικασία αυτή γίνεται για όλους τους χρήστες συνολικά και όχι για κάθε έναν ξεχωριστά. Με τον τρόπο αυτό διασφαλίζεται ότι θα γίνει συγκεκριμένος αριθμός κλήσεων στο API χωρίς να υπάρχει πρόβλημα στην λειτουργία του και επίσης εξασφαλίζεται ότι πάντα ο χρήστης θα παίρνει τα πιο πρόσφατα δεδομένα.

```

async function exportTable() {
  const checkdb = await fetch("/measures");
  const data = await checkdb.json();
  var pm25 = [];

  if (data.length == 0) {
    const googleresp_ins = await fetch(
      "https://script.google.com/macros/s/AKfycbw1mJVj8H1PeAdasdsQW13daA1jY2Fadasasdsa9G1E6H/exec?callback="
    );
    const googledata_ins = await googleresp_ins.json();

    const options_ins = {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify(googledata_ins),
    };
    const dbresp_ins = await fetch("/insert", options_ins);
    const dbdata_ins = await dbresp_ins.json();
    pm25 = JSON.parse(dbdata_ins.sensordata);
  } else {
    const db_date = new Date(data[0].updatedAt);
    const db_date_plus1 = new Date(data[0].updatedAt);

    const now_date = new Date();
    const uid = data[0].id;

    db_date_plus1.setTime(db_date.getTime() + 1 * 60 * 1000);

    if (now_date > db_date_plus1) {
      const googleresp_updt = await fetch(
        "https://script.google.com/macros/s/AKfycbw1mJVj8H1PeAdasdsQW13daA1jY2Fadasasdsa9G1E6H/exec?callback="
      );
      const googledata_updt = await googleresp_updt.json();

      const options_updt = {
        method: "PUT",
        headers: {
          "Content-Type": "application/json",
        },
        body: JSON.stringify(googledata_updt),
      };
      const dbresp_updt = await fetch(`/update/${uid}`, options_updt);
      const dbdata_updt = await dbresp_updt.json();
      pm25 = JSON.parse(dbdata_updt.sensordata);
    } else {
      pm25 = JSON.parse(data[0].sensordata);
    }
  }
  console.log(pm25)
  return pm25;
}

```

Κώδικας 6.13: Κώδικας για την κλήση στο API.

6.2.9 Κώδικας διαχείρισης του Server

Ο κώδικας για την διαχείριση του server περιέχει όλες τις πληροφορίες οι οποίες αφορούν την ιστοσελίδα. Στον κώδικα αυτό ορίζεται ο τρόπος επικοινωνίας των

διαφορετικών html αρχείων καθώς και η επικοινωνία με την βάση δεδομένων στην οποία υπάρχουν όλα τα δεδομένα τα οποία εμφανίζονται στην ιστοσελίδα.

```
const express = require("express");
const app = express();
const port = 80;
const Sequelize = require("sequelize");
const bodyParser = require("body-parser");
const fetch = require("node-fetch");
const fs = require("fs");

app.use(bodyParser.json());
app.use(express.static("public"));

const MAP_API_KEY = "xJlQ2dsaFqGWAeGlz5wYn";
const WEATHER_API_KEY = "6b75fdas545fsg7cc8c7612f851cfafa08e1d";

const sequelize = new Sequelize({
  dialect: "sqlite",
  storage: "./database.sqlite",
});

//endpoints
//gr version
app.get("/", (req, res) => res.sendFile("public/gr/index.html"));
app.get("/gr/mapgr", (req, res) => res.sendFile("public/gr/map.html"));
app.get("/gr/chartsgr/volos", (req, res) =>
  res.sendFile("public/gr/chartsgr/volos.html")
);
app.get("/gr/calendar/volos", async (req, res) => {
  let url =

  "https://script.google.com/macros/s/AKfycbwLaZzQadnKFC0jtgfb75SDF65I1cxvKKgTDOhs-
  b9RukHXowV8/exec?callback=";

  if (!fs.existsSync("public/en/calendar/mean_values_volos.json")) {
    const events_resp = await fetch(url);
    const events_data = await events_resp.json();
    console.log(events_data);
    fs.writeFileSync(
      "public/en/calendar/mean_values_volos.json",
      JSON.stringify(events_data)
    );
  }

  if (!fs.existsSync("public/gr/calendar/mean_values_volos.json")) {
    const events_resp = await fetch(url);
    const events_data = await events_resp.json();
    console.log(events_data);
    fs.writeFileSync(
      "public/gr/calendar/mean_values_volos.json",
      JSON.stringify(events_data)
    );
  }

  res.sendFile("public/gr/calendar/volosCal.html");
});
app.get("/gr/newsfeedgr/volosnewsfeed", (req, res) =>
  res.sendFile("public/gr/newsfeedgr/volosNF.html")
);
app.get("/gr/about", (req, res) => res.sendFile("public/gr/about.html"));

//en version
app.get("/en", (req, res) => res.sendFile("public/en/indexen.html"));
app.get("/en/mapen", (req, res) => res.sendFile("public/en/mapen.html"));
app.get("/en/chartsen/volos", (req, res) =>
  res.sendFile("public/en/chartsen/volosen.html")
);

app.get("/en/calendar/volos", async (req, res) => {
  let url =
```

```

"https://script.google.com/macros/s/AKfycbwLaZzQdsa445d6sa8rmHJ6LPWximwIlcxvKKgTDOhs
-b9RudsaXowV8/exec?callback=";

if (!fs.existsSync("public/en/calendar/mean_values_volos.json")) {
  const events_resp = await fetch(url);
  const events_data = await events_resp.json();
  console.log(events_data);
  fs.writeFileSync(
    "public/en/calendar/mean_values_volos.json",
    JSON.stringify(events_data)
  );
}

if (!fs.existsSync("public/gr/calendar/mean_values_volos.json")) {
  const events_resp = await fetch(url);
  const events_data = await events_resp.json();
  console.log(events_data);
  fs.writeFileSync(
    "public/gr/calendar/mean_values_volos.json",
    JSON.stringify(events_data)
  );
}

res.sendFile("public/en/calendar/volosCalen.html");
});

app.get("/en/newsfeeden/volosnewsfeed", (req, res) =>
  res.sendFile("public/en/newsfeeden/volosNFen.html")
);
app.get("/en/about", (req, res) => res.sendFile("public/en/abouten.html"));

// api hidden links
app.get("/maps/positron/:z/:x/:y.png", async (req, res) => {
  const z = req.params.z;
  const x = req.params.x;
  const y = req.params.y;

  if (!fs.existsSync(`maptiles/${x}-${y}-${z}.png`)) {
    const map_url =
`https://api.maptiler.com/maps/positron/${z}/${x}/${y}.png?key=${MAP_API_KEY}`;
    const map_response = await fetch(map_url);
    const map_result = await map_response.blob();

    fs.writeFileSync(
      `maptiles/${x}-${y}-${z}.png`,
      new Buffer(await map_result.arrayBuffer())
    );
  }

  res.sendFile(`maptiles/${x}-${y}-${z}.png`);
} else {
  const png_created_date = fs.statSync(`maptiles/${x}-${y}-${z}.png`);

  const current_date = new Date();
  const tile_date = new Date(png_created_date.birthtime);
  const tile_delete_date = new Date(png_created_date.birthtime);

  tile_delete_date.setMonth(tile_date.getMonth() + 1);

  if (current_date > tile_delete_date) {
    fs.unlinkSync(`maptiles/${x}-${y}-${z}.png`);

    const map_url =
`https://api.maptiler.com/maps/positron/${z}/${x}/${y}.png?key=${MAP_API_KEY}`;
    const map_response = await fetch(map_url);
    const map_result = await map_response.blob();

    fs.writeFileSync(
      `maptiles/${x}-${y}-${z}.png`,
      new Buffer(await map_result.arrayBuffer())
    );
  }

  res.sendFile(`maptiles/${x}-${y}-${z}.png`);
} else {
  res.sendFile(`maptiles/${x}-${y}-${z}.png`);
}
}

```

```

    }
  });

  app.get("/weather/:lat/:lon", async (req, res) => {
    const lat = req.params.lat;
    const lon = req.params.lon;

    const weather_api_url =
    `https://api.openweathermap.org/data/2.5/weather?lat=${lat}&lon=${lon}&appid=${WEATHER_API_KEY}`;
    const weather_response = await fetch(weather_api_url);
    const weather_data = await weather_response.json();
    res.json(weather_data);
  });

  app.listen(port, "0.0.0.0", () => console.log(`App listening on port ${port}`));

  //database connection and endpoints
  sequelize
    .authenticate()
    .then(() => {
      console.log("Connection has been established successfully.");
    })
    .catch((err) => {
      console.log("unable to connect to the database:", err);
    });

  const Measures = sequelize.define("measures", {
    sensordata: Sequelize.TEXT,
  });

  sequelize.sync({ force: true }).then(() => {
    console.log(`Database and tables created!`);
  });

  app.get("/measures", (req, res) => {
    Measures.findAll().then((measures) => res.json(measures));
  });

  app.post("/insert", (req, res) => {
    let given_data = JSON.stringify(req.body);

    Measures.create({ sensordata: given_data }).then((measures) => {
      res.json(measures);
    });
  });

  app.put("/update/:id", (req, res) => {
    let given_dataupdt = JSON.stringify(req.body);

    Measures.findByPk(req.params.id).then((measures) => {
      measures.update({ sensordata: given_dataupdt });
      res.json(measures);
    });
  });
};

```

Κώδικας 6.14: Κώδικας διαχείρισης του Server.

Οι παραπάνω κώδικες χρησιμοποιήθηκαν για την κατασκευή της ιστοσελίδας GreenYourAir. Δυστυχώς λόγω του μεγάλου μεγέθους η ανάλυση τους είναι δύσκολη και πολλά σημεία στους κώδικες είναι αρκετά εξειδικευμένα για να αναλυθούν.

7 ΜΕΤΡΗΣΕΙΣ ΓΙΑ ΤΗΝ ΧΡΟΝΙΚΗ ΠΕΡΙΟΔΟ ΑΠΟ 01/03/2019 ΕΩΣ 29/02/2020

Στην ενότητα αυτή παρουσιάζονται οι μετρήσεις που πραγματοποιήθηκαν από 01/03/2019 έως 29/02/2020. Η τιμή των μετρήσεων αφορά την συγκέντρωση των μικροσωματιδίων PM_{2.5} των δώδεκα μετρητών του δικτύου GreenYourAir στην πόλη του Βόλου σε ημερήσια βάση και η μονάδα μέτρησης είναι σε μg/m³. Για την εξαγωγή όλων των τιμών χρησιμοποιήθηκαν συνολικά πάνω από 10 εκατομμύρια τιμές.

7.1 Μέσος όρος δωδεκάμηνης περιόδου

Ο μέσος όρος της δωδεκάμηνης περιόδου από 01/03/2019 έως 29/02/2020 των μικροσωματιδίων PM_{2.5} στην πόλη του Βόλου ανήλθε στα 30.03 μg/m³. Με βάση το ετήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας, το οποίο είναι 10 μg/m³, παρατηρείται υπέρβαση της τάξης του 200.30%. Με βάση το ετήσιο όριο που έχει θεσπίσει το Υπουργείο Περιβάλλοντος και Ενέργειας, το οποίο είναι 25 μg/m³, παρατηρείται υπέρβαση της τάξης του 20.12%.

| Μέσος όρος Μετρήσεων από 01/03/2019 έως 29/02/2020 | Υπέρβαση με βάση το ετήσιο όριο του Παγκόσμιου Οργανισμού Υγείας | Υπέρβαση με βάση το Υπουργείο Περιβάλλοντος και Ενέργειας |
|--|--|---|
| 30.03 μg/m ³ | 200.30% (10.00 μg/m ³) | 20.12% (25.00 μg/m ³) |

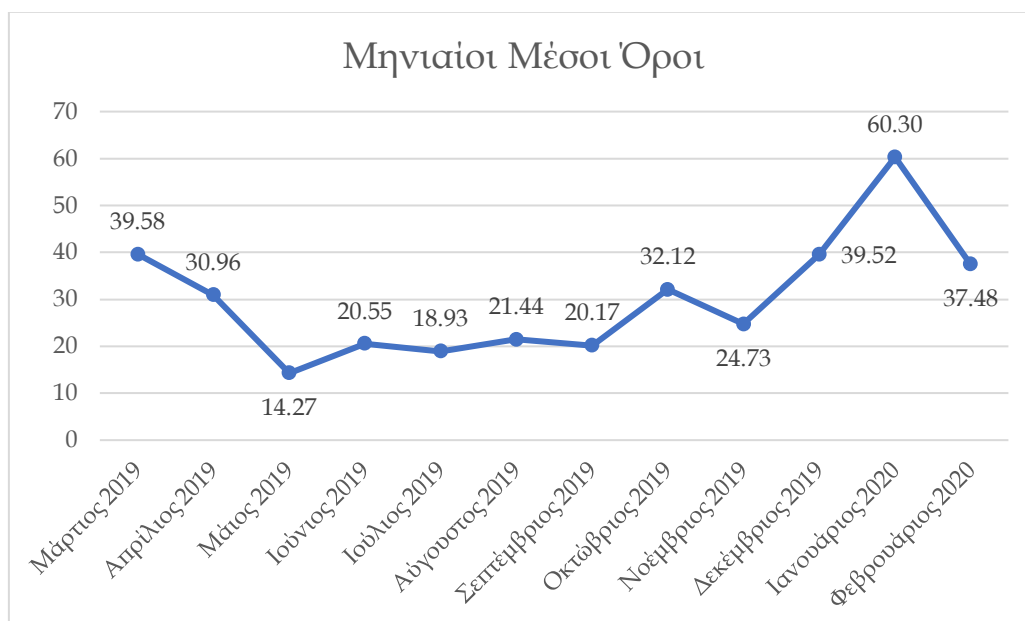
Πίνακας 7.1: Μέσος όρος για την δωδεκάμηνη περίοδο.

7.2 Μηνιαίοι μέσοι όροι

Παρακάτω παρουσιάζονται οι μηνιαίοι μέσοι όροι των μετρήσεων των PM_{2.5} για την δωδεκάμηνη περίοδο από 01/03/2019 έως 29/02/2020. Στον πίνακα αναγράφονται επίσης η μέγιστη και η ελάχιστη ημερήσια τιμή για κάθε μήνα των μετρήσεων καθώς και η τυπική απόκλιση των μετρήσεων. Επίσης παρουσιάζονται για κάθε μήνα ο αριθμός ημερών με το αντίστοιχο εύρος μέσων ημερήσιων τιμών καθώς και ο αριθμός υπερβάσεων κάθε μήνα με βάση το ημερήσιο όριο του Παγκόσμιου Οργανισμού Υγείας το οποίο είναι 25.00 μg/m³.

| Μήνας | Μέσος Όρος (μg/m ³) | Μέγιστη Τιμή (μg/m ³) | Ελάχιστη Τιμή (μg/m ³) | Τυπική Απόκλιση (μg/m ³) |
|------------------|---------------------------------|-----------------------------------|------------------------------------|--------------------------------------|
| Μάρτιος 2019 | 39.58 | 59.57 | 12.32 | 11.78 |
| Απρίλιος 2019 | 30.96 | 49.78 | 6.46 | 10.60 |
| Μάιος 2019 | 14.27 | 32.20 | 3.75 | 6.50 |
| Ιούνιος 2019 | 20.55 | 28.84 | 8.25 | 5.07 |
| Ιούλιος 2019 | 18.93 | 33.66 | 7.01 | 5.99 |
| Αύγουστος 2019 | 21.44 | 36.16 | 12.02 | 6.38 |
| Σεπτέμβριος 2019 | 20.17 | 34.07 | 11.64 | 5.96 |
| Οκτώβριος 2019 | 32.12 | 60.31 | 7.25 | 14.52 |
| Νοέμβριος 2019 | 24.73 | 50.80 | 11.00 | 10.07 |
| Δεκέμβριος 2019 | 39.52 | 69.54 | 12.66 | 18.21 |
| Ιανουάριος 2020 | 60.30 | 97.62 | 15.37 | 21.95 |
| Φεβρουάριος 2020 | 37.48 | 68.01 | 8.99 | 15.95 |

Πίνακας 7.2: Μηνιαίοι Μέσοι Όροι.



Διάγραμμα 7.1: Μηνιαίοι Μέσοι Όροι.

Παρακάτω παρουσιάζονται τα εύρη στα οποία ανήκουν οι ημερήσιες μετρήσεις των μικροσωματιδίων PM_{2.5} για κάθε μήνα.

| Εύρος Τιμών ($\mu\text{g}/\text{m}^3$) | Μάρτιος 2019 | Απρίλιος 2019 | Μάιος 2019 |
|---|-----------------|------------------|---------------|
| 0.00 – 25.00 | 4 | 9 | 29 |
| 25.01 – 30.00 | 1 | 1 | 0 |
| 30.01 – 35.00 | 7 | 10 | 2 |
| 35.01 – 40.00 | 1 | 5 | 0 |
| 40.01 – 45.00 | 8 | 3 | 0 |
| 45.01 – 50.00 | 3 | 2 | 0 |
| 50.01 – 55.00 | 5 | 0 | 0 |
| 55.01 – 60.00 | 2 | 0 | 0 |

Πίνακας 7.3: Εύρη τιμών από Μάρτιο 2019 - Μάιο 2019.

| Εύρος Τιμών ($\mu\text{g}/\text{m}^3$) | Ιούνιος 2019 | Ιούλιος 2019 | Αύγουστος 2019 |
|---|-----------------|-----------------|-------------------|
| 0.00 – 25.00 | 23 | 27 | 24 |
| 25.01 – 30.00 | 7 | 3 | 3 |
| 30.01 – 35.00 | 0 | 1 | 2 |
| 35.01 – 40.00 | 0 | 0 | 2 |

Πίνακας 7.4: Εύρη τιμών από Ιούνιο 2019 - Αύγουστο 2019.

| Εύρος Τιμών ($\mu\text{g}/\text{m}^3$) | Σεπτέμβριος 2019 | Οκτώβριος 2019 | Νοέμβριος 2019 |
|---|---------------------|-------------------|-------------------|
| 0.00 – 25.00 | 24 | 11 | 19 |
| 25.01 – 30.00 | 3 | 5 | 2 |
| 30.01 – 35.00 | 3 | 2 | 4 |
| 35.01 – 40.00 | 0 | 2 | 3 |
| 40.01 – 45.00 | 0 | 3 | 1 |
| 45.01 – 50.00 | 0 | 4 | 0 |
| 50.01 – 55.00 | 0 | 3 | 1 |
| 55.01 – 60.00 | 0 | 0 | 0 |
| 60.01 – 65.00 | 0 | 1 | 0 |

Πίνακας 7.5: Εύρη τιμών από Σεπτέμβριο 2019 - Νοέμβριο 2019.

| Εύρος Τιμών ($\mu\text{g}/\text{m}^3$) | Δεκέμβριος 2019 | Ιανουάριος 2020 | Φεβρουάριος 2020 |
|---|--------------------|--------------------|---------------------|
| 0.00 - 25.00 | 10 | 2 | 6 |
| 25.01 - 30.00 | 2 | 0 | 4 |
| 30.01 - 35.00 | 2 | 2 | 2 |
| 35.01 - 40.00 | 3 | 1 | 7 |
| 40.01 - 45.00 | 2 | 4 | 1 |
| 45.01 - 50.00 | 1 | 2 | 1 |
| 50.01 - 55.00 | 3 | 5 | 4 |
| 55.01 - 60.00 | 3 | 0 | 2 |
| 60.01 - 65.00 | 1 | 2 | 0 |
| 65.01 - 70.00 | 4 | 1 | 2 |
| 70.01 - 75.00 | 0 | 1 | 0 |
| 75.01 - 80.00 | 0 | 3 | 0 |
| 80.01 - 85.00 | 0 | 3 | 0 |
| 85.01 - 90.00 | 0 | 4 | 0 |
| 90.01 - 95.00 | 0 | 0 | 0 |
| 95.01 - 100.00 | 0 | 1 | 0 |

Πίνακας 7.6: Εύρη τιμών από Δεκέμβριο 2019 - Φεβρουάριο 2020.

Παρακάτω παρουσιάζεται ο αριθμός υπερβάσεων για κάθε μήνα με βάση το ημερήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας. Το ημερήσιο όριο είναι $25.00 \mu\text{g}/\text{m}^3$.

| Μήνας | Αριθμός Υπερβάσεων (ημέρες) | Αριθμός Υπερβάσεων (ποσοστό %) |
|------------------|-----------------------------------|--------------------------------------|
| Μάρτιος 2019 | 27 | 87.10 |
| Απρίλιος 2019 | 21 | 70.00 |
| Μάιος 2019 | 2 | 6.45 |
| Ιούνιος 2019 | 7 | 23.33 |
| Ιούλιος 2019 | 4 | 12.90 |
| Αύγουστος 2019 | 7 | 22.58 |
| Σεπτέμβριος 2019 | 6 | 20.00 |
| Οκτώβριος 2019 | 20 | 64.52 |
| Νοέμβριος 2019 | 11 | 36.67 |

| | | |
|------------------|----|-------|
| Δεκέμβριος 2019 | 21 | 67.74 |
| Ιανουάριος 2020 | 29 | 93.55 |
| Φεβρουάριος 2020 | 23 | 79.31 |

Πίνακας 7.7: Αριθμός Υπερβάσεων ανά μήνα.

Συνολικά 178 ημέρες από τις 366 παρατηρήθηκε υπέρβαση του ημερήσιου ορίου που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας. Κατά μέσο όρο παρατηρείται υπέρβαση 15 ημέρες τον μήνα για τα μικροσωματίδια PM_{2.5} στην πόλη του Βόλου.

| Συγκεντρωτικός πίνακας Υπερβάσεων για την δωδεκάμηνη περίοδο από 01/03/2019 έως 29/02/2020. | |
|--|-------|
| Συνολικό Αριθμός Υπερβάσεων (ημέρες) | 178 |
| Μέσος όρος Υπερβάσεων ανά μήνα (ημέρες) | 15 |
| Μέσος όρος Υπερβάσεων ανά μήνα (ποσοστό %) | 48.68 |

Πίνακας 7.8: Συγκεντρωτικός πίνακας Υπερβάσεων.

7.3 Ημερήσιο μέσοι όροι

Παρακάτω παρουσιάζονται οι ημερήσιοι μέσοι όροι των μικροσωματιδίων PM_{2.5} ανά μήνα για την πόλη του Βόλου.

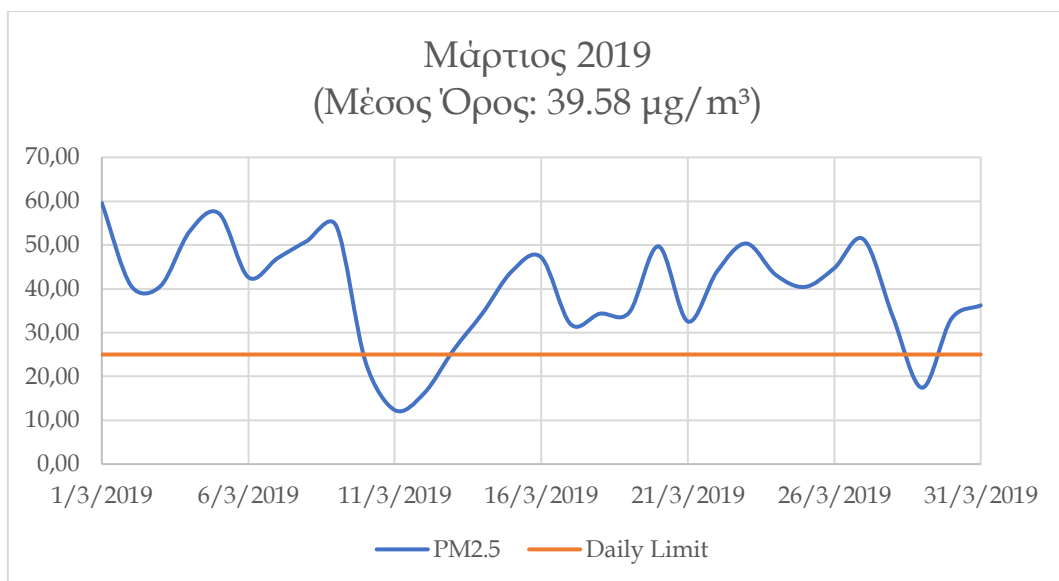
7.3.1 Ημερήσιοι μέσοι όροι για τον Μάρτιο 2019

Ο μέσος όρος των ημερήσιων τιμών των μικροσωματιδίων PM_{2.5} τον Μάρτιο του 2019 με βάση τους 12 μετρητές του GreenYourAir ανήλθε στα 39.58 µg/m³. Η μέγιστη τιμή που παρατηρήθηκε είναι 59.57 µg/m³ και η ελάχιστη 12.32 µg/m³. Το ημερήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας για τα PM_{2.5} είναι 25.00 µg/m³.

| Ημερομηνία | Μέσος όρος PM_{2.5} (µg/m³) |
|-------------------|---|
| 1/3/2019 | 59.57 |
| 2/3/2019 | 40.67 |
| 3/3/2019 | 40.59 |
| 4/3/2019 | 53.12 |
| 5/3/2019 | 57.19 |
| 6/3/2019 | 42.64 |

| | |
|-----------|-------|
| 7/3/2019 | 46.99 |
| 8/3/2019 | 50.90 |
| 9/3/2019 | 54.37 |
| 10/3/2019 | 23.34 |
| 11/3/2019 | 12.32 |
| 12/3/2019 | 16.14 |
| 13/3/2019 | 25.98 |
| 14/3/2019 | 34.46 |
| 15/3/2019 | 44.02 |
| 16/3/2019 | 47.16 |
| 17/3/2019 | 31.94 |
| 18/3/2019 | 34.31 |
| 19/3/2019 | 34.57 |
| 20/3/2019 | 49.69 |
| 21/3/2019 | 32.54 |
| 22/3/2019 | 43.96 |
| 23/3/2019 | 50.33 |
| 24/3/2019 | 43.21 |
| 25/3/2019 | 40.43 |
| 26/3/2019 | 44.70 |
| 27/3/2019 | 51.29 |
| 28/3/2019 | 33.66 |
| 29/3/2019 | 17.44 |
| 30/3/2019 | 33.13 |
| 31/3/2019 | 36.22 |

Πίνακας 7.9: Ημερήσιοι μέσοι όροι για τον Μάρτιο 2019.



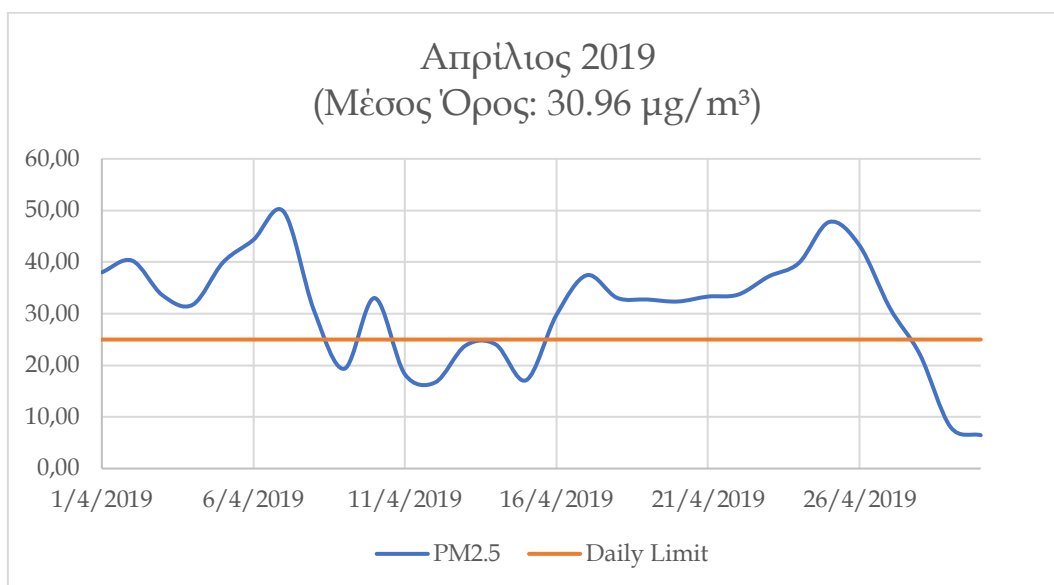
Διάγραμμα 7.2: Ημερήσιοι μέσοι όροι για τον Μάρτιο 2019.

7.3.2 Ημερήσιοι μέσοι όροι για τον Απρίλιο 2019

Ο μέσος όρος των ημερήσιων τιμών των μικροσωματιδίων PM_{2.5} τον Απρίλιο του 2019 με βάση τους 12 μετρητές του GreenYourAir ανήλθε στα 30.96 µg/m³. Η μέγιστη τιμή που παρατηρήθηκε είναι 49.78 µg/m³ και η ελάχιστη 6.46 µg/m³. Το ημερήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας για τα PM_{2.5} είναι 25.00 µg/m³.

| Ημερομηνία | Μέσος όρος PM _{2.5} (µg/m ³) |
|------------|--|
| 1/4/2019 | 38.02 |
| 2/4/2019 | 40.28 |
| 3/4/2019 | 33.58 |
| 4/4/2019 | 31.74 |
| 5/4/2019 | 39.94 |
| 6/4/2019 | 44.30 |
| 7/4/2019 | 49.78 |
| 8/4/2019 | 30.64 |
| 9/4/2019 | 19.34 |
| 10/4/2019 | 33.04 |
| 11/4/2019 | 18.26 |
| 12/4/2019 | 16.68 |
| 13/4/2019 | 23.83 |
| 14/4/2019 | 24.03 |
| 15/4/2019 | 17.11 |
| 16/4/2019 | 29.75 |
| 17/4/2019 | 37.46 |
| 18/4/2019 | 33.08 |
| 19/4/2019 | 32.76 |
| 20/4/2019 | 32.36 |
| 21/4/2019 | 33.32 |
| 22/4/2019 | 33.71 |
| 23/4/2019 | 37.21 |
| 24/4/2019 | 39.83 |
| 25/4/2019 | 47.77 |
| 26/4/2019 | 43.22 |
| 27/4/2019 | 31.11 |
| 28/4/2019 | 22.02 |
| 29/4/2019 | 8.07 |
| 30/4/2019 | 6.46 |

Πίνακας 7.10: Ημερήσιοι μέσοι όροι για το Απρίλιο 2019.



Διάγραμμα 7.3: Ημερήσιοι μέσοι όροι για τον Απρίλιο 2019.

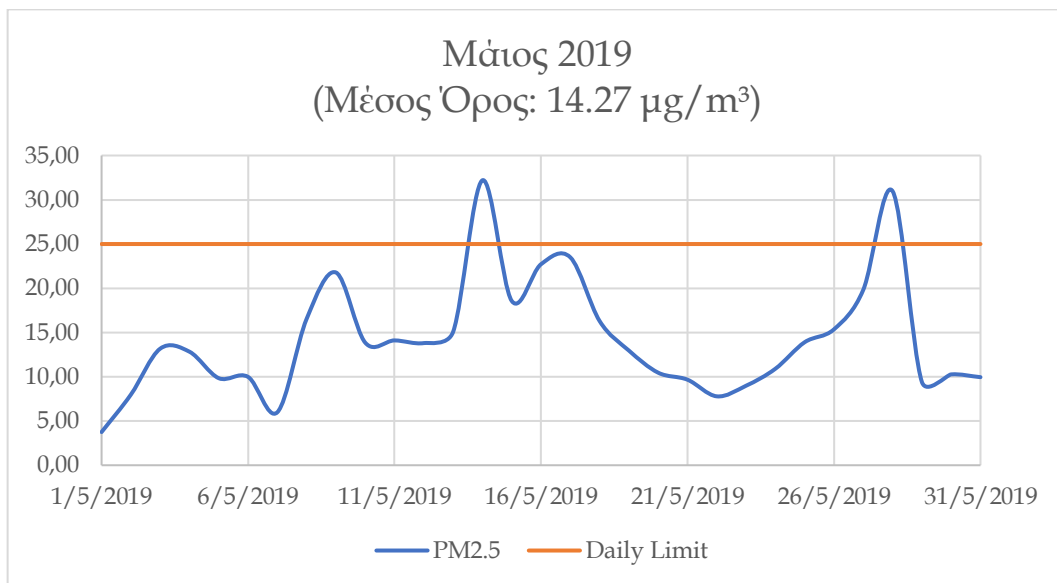
7.3.3 Ημερήσιοι μέσοι όροι για τον Μάιο 2019

Ο μέσος όρος των ημερήσιων τιμών των μικροσωματιδίων PM_{2.5} τον Μάιο του 2019 με βάση τους 12 μετρητές του GreenYourAir ανήλθε στα 14.27 µg/m³. Η μέγιστη τιμή που παρατηρήθηκε είναι 32.30 µg/m³ και η ελάχιστη 3.75 µg/m³. Το ημερήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας για τα PM_{2.5} είναι 25.00 µg/m³.

| Ημερομηνία | Μέσος όρος PM _{2.5} (µg/m ³) |
|------------|--|
| 1/5/2019 | 3.75 |
| 2/5/2019 | 7.98 |
| 3/5/2019 | 13.17 |
| 4/5/2019 | 12.82 |
| 5/5/2019 | 9.84 |
| 6/5/2019 | 9.98 |
| 7/5/2019 | 5.99 |
| 8/5/2019 | 16.55 |
| 9/5/2019 | 21.78 |
| 10/5/2019 | 13.85 |
| 11/5/2019 | 14.11 |
| 12/5/2019 | 13.81 |
| 13/5/2019 | 15.08 |

| | |
|-----------|-------|
| 14/5/2019 | 32.20 |
| 15/5/2019 | 18.57 |
| 16/5/2019 | 22.72 |
| 17/5/2019 | 23.49 |
| 18/5/2019 | 16.29 |
| 19/5/2019 | 12.96 |
| 20/5/2019 | 10.45 |
| 21/5/2019 | 9.66 |
| 22/5/2019 | 7.79 |
| 23/5/2019 | 8.98 |
| 24/5/2019 | 10.90 |
| 25/5/2019 | 13.90 |
| 26/5/2019 | 15.36 |
| 27/5/2019 | 19.82 |
| 28/5/2019 | 30.97 |
| 29/5/2019 | 9.42 |
| 30/5/2019 | 10.26 |
| 31/5/2019 | 9.95 |

Πίνακας 7.11: Ημερήσιοι μέσοι όροι για τον Μάιο 2019.



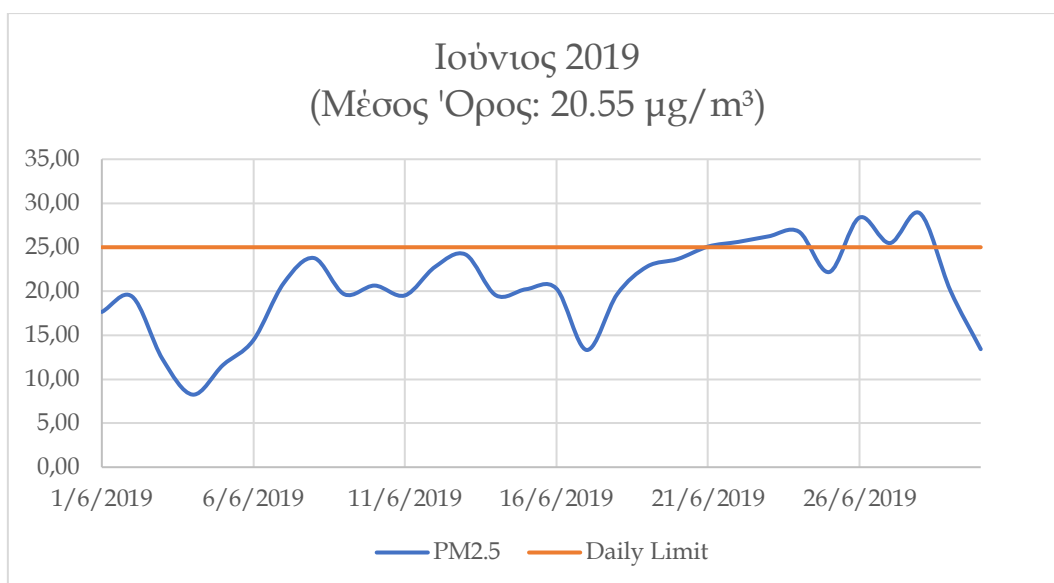
Διάγραμμα 7.4: Ημερήσιοι μέσοι όροι για τον Μάιο 2019.

7.3.4 Ημερήσιοι μέσοι όροι για τον Ιούνιο 2019

Ο μέσος όρος των ημερήσιων τιμών των μικροσωματιδίων PM_{2.5} τον Ιούνιο του 2019 με βάση τους 12 μετρητές του GreenYourAir ανήλθε στα 20.55 µg/m³. Η μέγιστη τιμή που παρατηρήθηκε είναι 28.84 µg/m³ και η ελάχιστη 8.25 µg/m³. Το ημερήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας για τα PM_{2.5} είναι 25.00 µg/m³.

| Ημερομηνία | Μέσος όρος PM _{2.5} (µg/m ³) |
|------------|--|
| 1/6/2019 | 17.65 |
| 2/6/2019 | 19.40 |
| 3/6/2019 | 12.30 |
| 4/6/2019 | 8.25 |
| 5/6/2019 | 11.61 |
| 6/6/2019 | 14.43 |
| 7/6/2019 | 20.91 |
| 8/6/2019 | 23.77 |
| 9/6/2019 | 19.65 |
| 10/6/2019 | 20.64 |
| 11/6/2019 | 19.52 |
| 12/6/2019 | 22.79 |
| 13/6/2019 | 24.17 |
| 14/6/2019 | 19.54 |
| 15/6/2019 | 20.23 |
| 16/6/2019 | 20.30 |
| 17/6/2019 | 13.32 |
| 18/6/2019 | 19.64 |
| 19/6/2019 | 22.81 |
| 20/6/2019 | 23.65 |
| 21/6/2019 | 25.07 |
| 22/6/2019 | 25.61 |
| 23/6/2019 | 26.24 |
| 24/6/2019 | 26.77 |
| 25/6/2019 | 22.18 |
| 26/6/2019 | 28.37 |
| 27/6/2019 | 25.47 |
| 28/6/2019 | 28.84 |
| 29/6/2019 | 20.06 |
| 30/6/2019 | 13.41 |

Πίνακας 7.12: Ημερήσιοι μέσοι όροι για τον Ιούνιο 2019.



Διάγραμμα 7.5: Ημερήσιοι μέσοι όροι για τον Ιούνιο 2019.

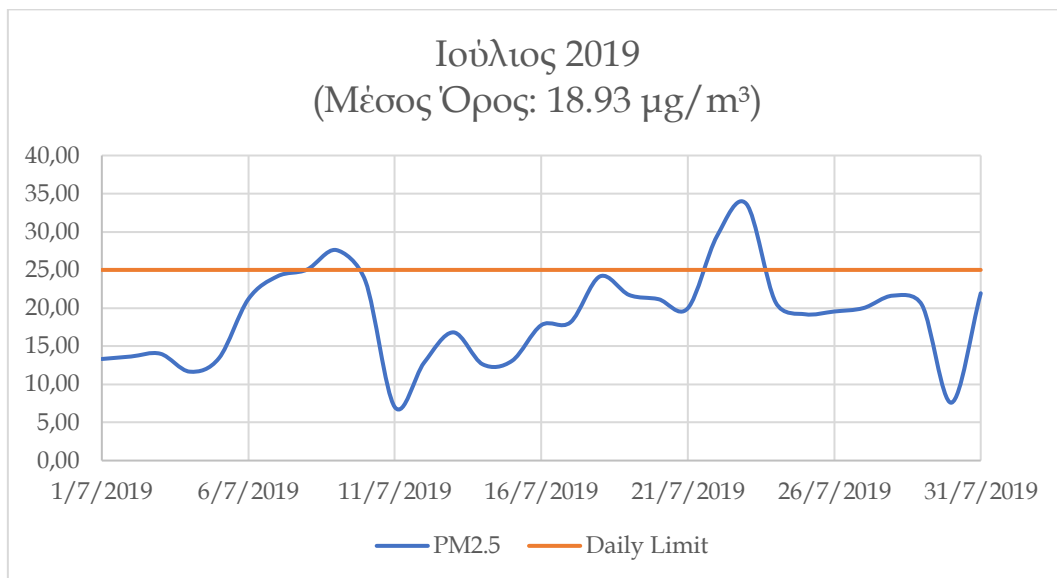
7.3.5 Ημερήσιοι μέσοι όροι για τον Ιούλιο 2019

Ο μέσος όρος των ημερήσιων τιμών των μικροσωματιδίων PM_{2.5} τον Ιούλιο του 2019 με βάση τους 12 μετρητές του GreenYourAir ανήλθε στα 18.93 $\mu\text{g}/\text{m}^3$. Η μέγιστη τιμή που παρατηρήθηκε είναι 33.66 $\mu\text{g}/\text{m}^3$ και η ελάχιστη 7.01 $\mu\text{g}/\text{m}^3$. Το ημερήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας για τα PM_{2.5} είναι 25.00 $\mu\text{g}/\text{m}^3$.

| Ημερομηνία | Μέσος όρος PM _{2.5} ($\mu\text{g}/\text{m}^3$) |
|------------|---|
| 1/7/2019 | 13.32 |
| 2/7/2019 | 13.65 |
| 3/7/2019 | 14.01 |
| 4/7/2019 | 11.66 |
| 5/7/2019 | 13.44 |
| 6/7/2019 | 21.16 |
| 7/7/2019 | 24.17 |
| 8/7/2019 | 25.04 |
| 9/7/2019 | 27.61 |
| 10/7/2019 | 23.51 |
| 11/7/2019 | 7.01 |
| 12/7/2019 | 12.84 |
| 13/7/2019 | 16.81 |

| | |
|-----------|-------|
| 14/7/2019 | 12.62 |
| 15/7/2019 | 13.07 |
| 16/7/2019 | 17.75 |
| 17/7/2019 | 18.14 |
| 18/7/2019 | 24.15 |
| 19/7/2019 | 21.71 |
| 20/7/2019 | 21.16 |
| 21/7/2019 | 19.97 |
| 22/7/2019 | 29.46 |
| 23/7/2019 | 33.66 |
| 24/7/2019 | 20.75 |
| 25/7/2019 | 19.19 |
| 26/7/2019 | 19.55 |
| 27/7/2019 | 20.00 |
| 28/7/2019 | 21.62 |
| 29/7/2019 | 20.37 |
| 30/7/2019 | 7.59 |
| 31/7/2019 | 21.94 |

Πίνακας 7.13: Ημερήσιοι μέσοι όροι για τον Ιούλιο 2019.



Διάγραμμα 7.6: Ημερήσιοι μέσοι όροι για τον Ιούλιο 2019.

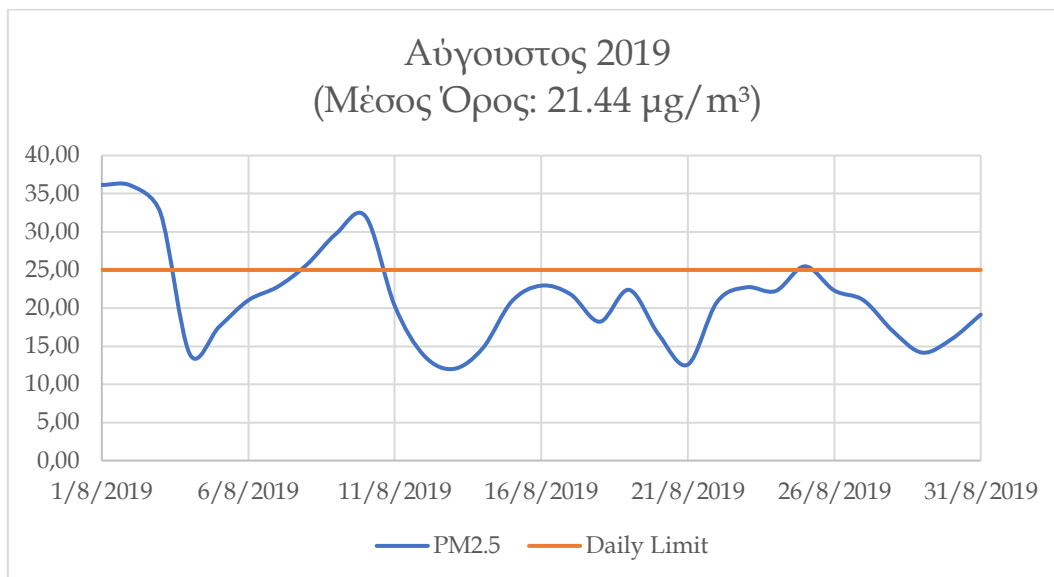
7.3.6 Ημερήσιοι μέσοι όροι για τον Αύγουστο 2019

Ο μέσος όρος των ημερήσιων τιμών των μικροσωματιδίων PM_{2.5} τον Αύγουστο του 2019 με βάση τους 12 μετρητές του GreenYourAir ανήλθε στα 21.44 µg/m³. Η μέγιστη τιμή που παρατηρήθηκε είναι 36.16 µg/m³ και η ελάχιστη 12.02 µg/m³. Το ημερήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας για τα PM_{2.5} είναι 25.00 µg/m³.

| Ημερομηνία | Μέσος όρος PM _{2.5} (µg/m ³) |
|------------|--|
| 1/8/2019 | 36.16 |
| 2/8/2019 | 36.05 |
| 3/8/2019 | 32.49 |
| 4/8/2019 | 13.98 |
| 5/8/2019 | 17.53 |
| 6/8/2019 | 21.02 |
| 7/8/2019 | 22.80 |
| 8/8/2019 | 25.70 |
| 9/8/2019 | 29.76 |
| 10/8/2019 | 32.04 |
| 11/8/2019 | 20.27 |
| 12/8/2019 | 13.77 |
| 13/8/2019 | 12.02 |
| 14/8/2019 | 14.72 |
| 15/8/2019 | 20.90 |
| 16/8/2019 | 22.95 |
| 17/8/2019 | 21.81 |
| 18/8/2019 | 18.21 |
| 19/8/2019 | 22.39 |
| 20/8/2019 | 16.57 |
| 21/8/2019 | 12.59 |
| 22/8/2019 | 20.80 |
| 23/8/2019 | 22.72 |
| 24/8/2019 | 22.24 |
| 25/8/2019 | 25.47 |
| 26/8/2019 | 22.31 |
| 27/8/2019 | 21.02 |
| 28/8/2019 | 16.98 |
| 29/8/2019 | 14.16 |

| | |
|-----------|-------|
| 30/8/2019 | 15.91 |
| 31/8/2019 | 19.14 |

Πίνακας 7.14: Ημερήσιοι μέσοι όροι για τον Αύγουστο 2019.



Διάγραμμα 7.7: Ημερήσιοι μέσοι όροι για τον Αύγουστο 2019.

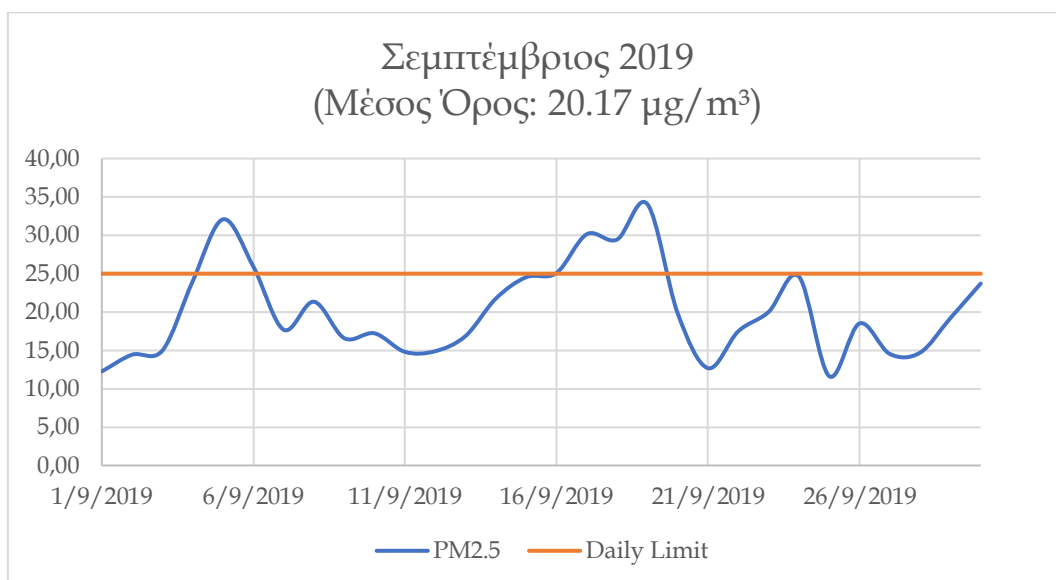
7.3.7 Ημερήσιοι μέσοι όροι για τον Σεπτέμβριο 2019

Ο μέσος όρος των ημερήσιων τιμών των μικροσωματιδίων PM_{2.5} τον Σεπτέμβριο του 2019 με βάση τους 12 μετρητές του GreenYourAir ανήλθε στα 20.17 µg/m³. Η μέγιστη τιμή που παρατηρήθηκε είναι 34.07 µg/m³ και η ελάχιστη 11.64 µg/m³. Το ημερήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας για τα PM_{2.5} είναι 25.00 µg/m³.

| Ημερομηνία | Μέσος όρος PM _{2.5} (µg/m ³) |
|------------|---|
| 1/9/2019 | 12.26 |
| 2/9/2019 | 14.44 |
| 3/9/2019 | 14.98 |
| 4/9/2019 | 24.03 |
| 5/9/2019 | 32.08 |
| 6/9/2019 | 25.92 |
| 7/9/2019 | 17.70 |
| 8/9/2019 | 21.35 |
| 9/9/2019 | 16.59 |

| | |
|-----------|-------|
| 10/9/2019 | 17.23 |
| 11/9/2019 | 14.81 |
| 12/9/2019 | 14.89 |
| 13/9/2019 | 16.88 |
| 14/9/2019 | 21.75 |
| 15/9/2019 | 24.54 |
| 16/9/2019 | 25.08 |
| 17/9/2019 | 30.14 |
| 18/9/2019 | 29.45 |
| 19/9/2019 | 34.07 |
| 20/9/2019 | 19.84 |
| 21/9/2019 | 12.69 |
| 22/9/2019 | 17.49 |
| 23/9/2019 | 20.01 |
| 24/9/2019 | 24.63 |
| 25/9/2019 | 11.64 |
| 26/9/2019 | 18.52 |
| 27/9/2019 | 14.54 |
| 28/9/2019 | 14.71 |
| 29/9/2019 | 19.21 |
| 30/9/2019 | 23.71 |

Πίνακας 7.15: Ημερήσιοι μέσοι όροι για τον Σεπτέμβριο 2019.



Διάγραμμα 7.8: Ημερήσιοι μέσοι όροι για τον Σεπτέμβριο 2019.

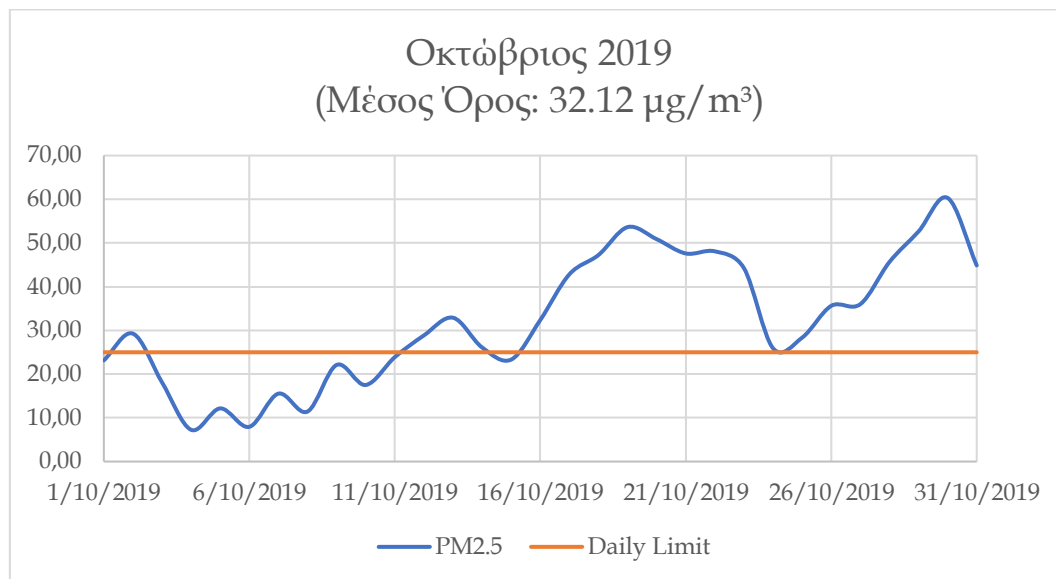
7.3.8 Ημερήσιοι μέσοι όροι για τον Οκτώβριο 2019

Ο μέσος όρος των ημερήσιων τιμών των μικροσωματιδίων PM_{2.5} τον Οκτώβριο του 2019 με βάση τους 12 μετρητές του GreenYourAir ανήλθε στα 32.12 μg/m³. Η μέγιστη τιμή που παρατηρήθηκε είναι 60.31 μg/m³ και η ελάχιστη 7.25 μg/m³. Το ημερήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας για τα PM_{2.5} είναι 25.00 μg/m³.

| Ημερομηνία | Μέσος όρος PM _{2.5} (μg/m ³) |
|------------|--|
| 1/10/2019 | 23.14 |
| 2/10/2019 | 29.26 |
| 3/10/2019 | 18.11 |
| 4/10/2019 | 7.25 |
| 5/10/2019 | 12.15 |
| 6/10/2019 | 7.95 |
| 7/10/2019 | 15.53 |
| 8/10/2019 | 11.44 |
| 9/10/2019 | 22.10 |
| 10/10/2019 | 17.52 |
| 11/10/2019 | 23.88 |
| 12/10/2019 | 28.85 |
| 13/10/2019 | 32.90 |
| 14/10/2019 | 26.11 |
| 15/10/2019 | 23.32 |
| 16/10/2019 | 32.38 |
| 17/10/2019 | 42.83 |
| 18/10/2019 | 47.24 |
| 19/10/2019 | 53.63 |
| 20/10/2019 | 50.87 |
| 21/10/2019 | 47.60 |
| 22/10/2019 | 48.09 |
| 23/10/2019 | 44.24 |
| 24/10/2019 | 25.91 |
| 25/10/2019 | 28.34 |
| 26/10/2019 | 35.62 |
| 27/10/2019 | 36.00 |
| 28/10/2019 | 45.64 |
| 29/10/2019 | 52.63 |

| | |
|------------|-------|
| 30/10/2019 | 60.31 |
| 31/10/2019 | 44.86 |

Πίνακας 7.16: Ημερήσιοι μέσοι όροι για τον Οκτώβριο 2019.



Διάγραμμα 7.9: Ημερήσιοι μέσοι όροι για τον Οκτώβριο 2019.

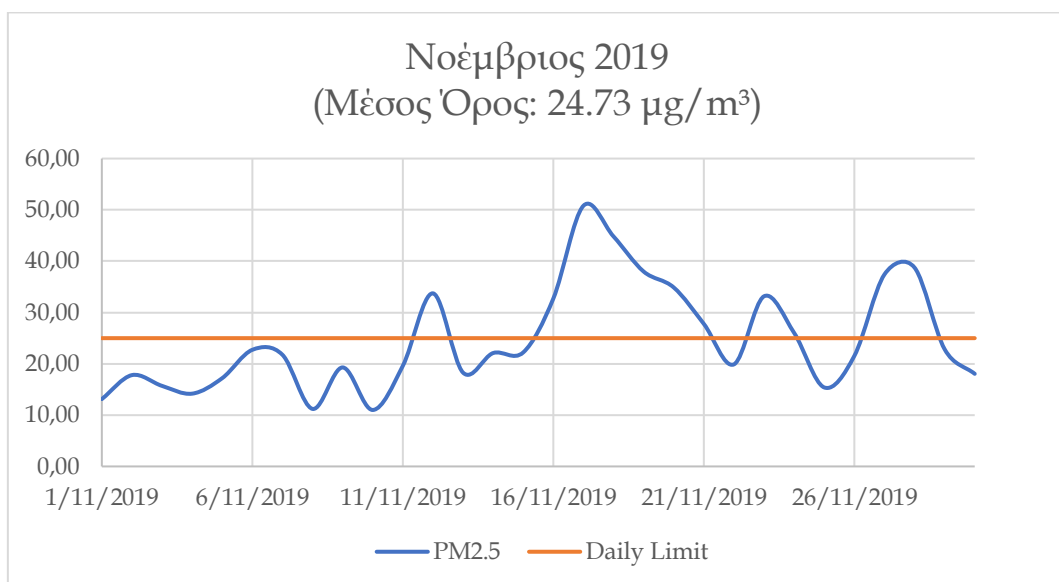
7.3.9 Ημερήσιοι μέσοι όροι για τον Νοέμβριο 2019

Ο μέσος όρος των ημερήσιων τιμών των μικροσωματιδίων PM_{2.5} τον Νοέμβριο του 2019 με βάση τους 12 μετρητές του GreenYourAir ανήλθε στα 24.73 µg/m³. Η μέγιστη τιμή που παρατηρήθηκε είναι 50.80 µg/m³ και η ελάχιστη 11.00 µg/m³. Το ημερήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας για τα PM_{2.5} είναι 25.00 µg/m³.

| Ημερομηνία | Μέσος όρος PM _{2.5} (µg/m ³) |
|------------|---|
| 1/11/2019 | 13.12 |
| 2/11/2019 | 17.79 |
| 3/11/2019 | 15.70 |
| 4/11/2019 | 14.18 |
| 5/11/2019 | 17.21 |
| 6/11/2019 | 22.72 |
| 7/11/2019 | 21.77 |
| 8/11/2019 | 11.21 |
| 9/11/2019 | 19.29 |

| | |
|------------|-------|
| 10/11/2019 | 11.00 |
| 11/11/2019 | 19.60 |
| 12/11/2019 | 33.73 |
| 13/11/2019 | 18.28 |
| 14/11/2019 | 22.10 |
| 15/11/2019 | 22.22 |
| 16/11/2019 | 32.71 |
| 17/11/2019 | 50.80 |
| 18/11/2019 | 44.82 |
| 19/11/2019 | 37.98 |
| 20/11/2019 | 34.89 |
| 21/11/2019 | 27.78 |
| 22/11/2019 | 19.89 |
| 23/11/2019 | 33.15 |
| 24/11/2019 | 26.03 |
| 25/11/2019 | 15.40 |
| 26/11/2019 | 21.54 |
| 27/11/2019 | 37.41 |
| 28/11/2019 | 38.75 |
| 29/11/2019 | 22.84 |
| 30/11/2019 | 18.04 |

Πίνακας 7.17: Ημερήσιοι μέσοι όροι για τον Νοέμβριο 2019.



Διάγραμμα 7.10: Ημερήσιοι μέσοι όροι για τον Νοέμβριο 2019.

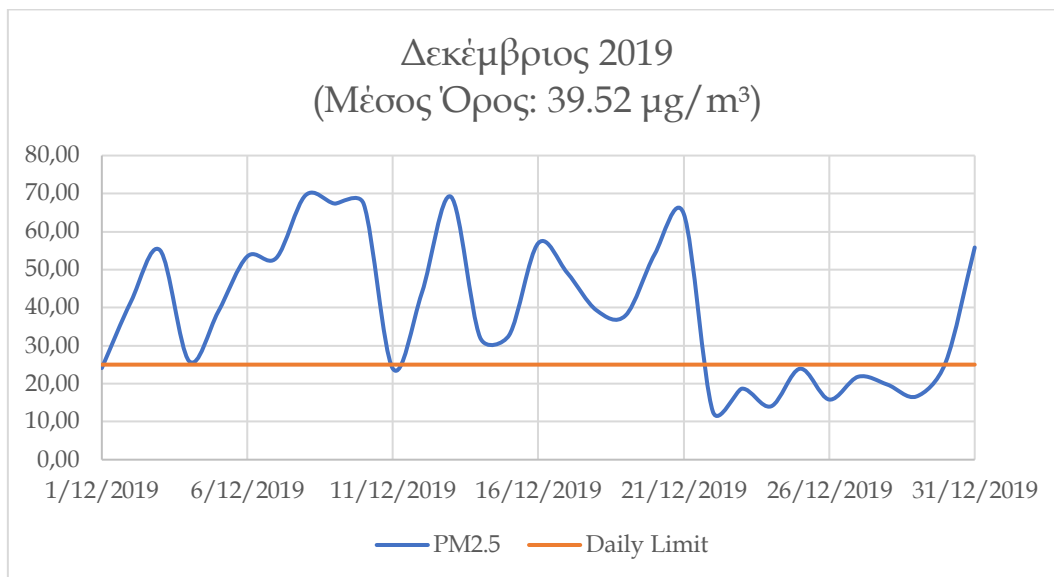
7.3.10 Ημερήσιοι μέσοι όροι για τον Δεκέμβριο 2019

Ο μέσος όρος των ημερήσιων τιμών των μικροσωματιδίων PM_{2.5} τον Δεκέμβριο του 2019 με βάση τους 12 μετρητές του GreenYourAir ανήλθε στα 39.52 μg/m³. Η μέγιστη τιμή που παρατηρήθηκε είναι 69.54 μg/m³ και η ελάχιστη 12.66 μg/m³. Το ημερήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας για τα PM_{2.5} είναι 25.00 μg/m³.

| Ημερομηνία | Μέσος όρος PM _{2.5} (μg/m ³) |
|------------|--|
| 1/12/2019 | 24.09 |
| 2/12/2019 | 41.55 |
| 3/12/2019 | 55.10 |
| 4/12/2019 | 25.91 |
| 5/12/2019 | 38.93 |
| 6/12/2019 | 53.46 |
| 7/12/2019 | 53.08 |
| 8/12/2019 | 69.54 |
| 9/12/2019 | 67.37 |
| 10/12/2019 | 67.27 |
| 11/12/2019 | 23.97 |
| 12/12/2019 | 43.91 |
| 13/12/2019 | 69.17 |
| 14/12/2019 | 32.15 |
| 15/12/2019 | 32.75 |
| 16/12/2019 | 56.96 |
| 17/12/2019 | 49.11 |
| 18/12/2019 | 39.31 |
| 19/12/2019 | 37.94 |
| 20/12/2019 | 54.03 |
| 21/12/2019 | 64.67 |
| 22/12/2019 | 12.66 |
| 23/12/2019 | 18.67 |
| 24/12/2019 | 14.04 |
| 25/12/2019 | 23.93 |
| 26/12/2019 | 15.81 |
| 27/12/2019 | 21.82 |
| 28/12/2019 | 19.74 |
| 29/12/2019 | 16.63 |

| | |
|------------|-------|
| 30/12/2019 | 25.55 |
| 31/12/2019 | 55.82 |

Πίνακας 7.18: Ημερήσιοι μέσοι όροι για τον Δεκέμβριο 2019.



Διάγραμμα 7.11: Ημερήσιοι μέσοι όροι για τον Δεκέμβριο 2019.

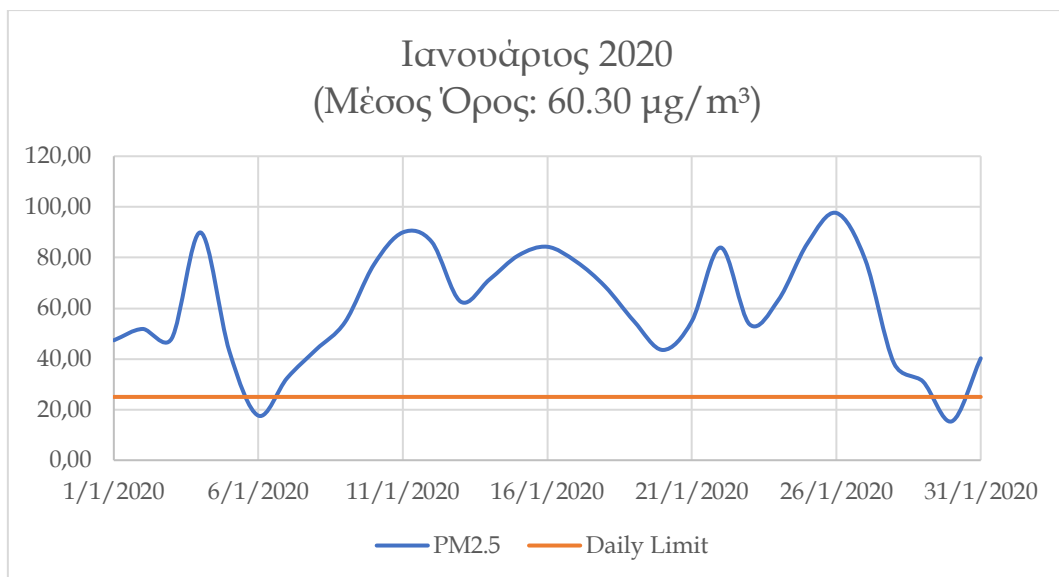
7.3.11 Ημερήσιοι μέσοι όροι για τον Ιανουάριο 2020

Ο μέσος όρος των ημερήσιων τιμών των μικροσωματιδίων PM_{2.5} τον Ιανουάριο του 2020 με βάση τους 12 μετρητές του GreenYourAir ανήλθε στα 60.30 µg/m³. Η μέγιστη τιμή που παρατηρήθηκε είναι 97.62 µg/m³ και η ελάχιστη 15.37 µg/m³. Το ημερήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας για τα PM_{2.5} είναι 25.00 µg/m³.

| Ημερομηνία | Μέσος όρος PM _{2.5} (µg/m ³) |
|------------|---|
| 1/1/2020 | 47.35 |
| 2/1/2020 | 51.83 |
| 3/1/2020 | 48.00 |
| 4/1/2020 | 89.91 |
| 5/1/2020 | 42.93 |
| 6/1/2020 | 17.61 |
| 7/1/2020 | 32.48 |
| 8/1/2020 | 43.67 |
| 9/1/2020 | 54.52 |

| | |
|-----------|-------|
| 10/1/2020 | 77.24 |
| 11/1/2020 | 89.93 |
| 12/1/2020 | 86.23 |
| 13/1/2020 | 62.63 |
| 14/1/2020 | 71.33 |
| 15/1/2020 | 80.85 |
| 16/1/2020 | 84.26 |
| 17/1/2020 | 78.41 |
| 18/1/2020 | 68.60 |
| 19/1/2020 | 54.89 |
| 20/1/2020 | 43.52 |
| 21/1/2020 | 54.83 |
| 22/1/2020 | 83.96 |
| 23/1/2020 | 53.67 |
| 24/1/2020 | 63.31 |
| 25/1/2020 | 85.54 |
| 26/1/2020 | 97.62 |
| 27/1/2020 | 79.26 |
| 28/1/2020 | 38.26 |
| 29/1/2020 | 31.10 |
| 30/1/2020 | 15.37 |
| 31/1/2020 | 40.23 |

Πίνακας 7.19: Ημερήσιοι μέσοι όροι για τον Ιανουάριο 2020.



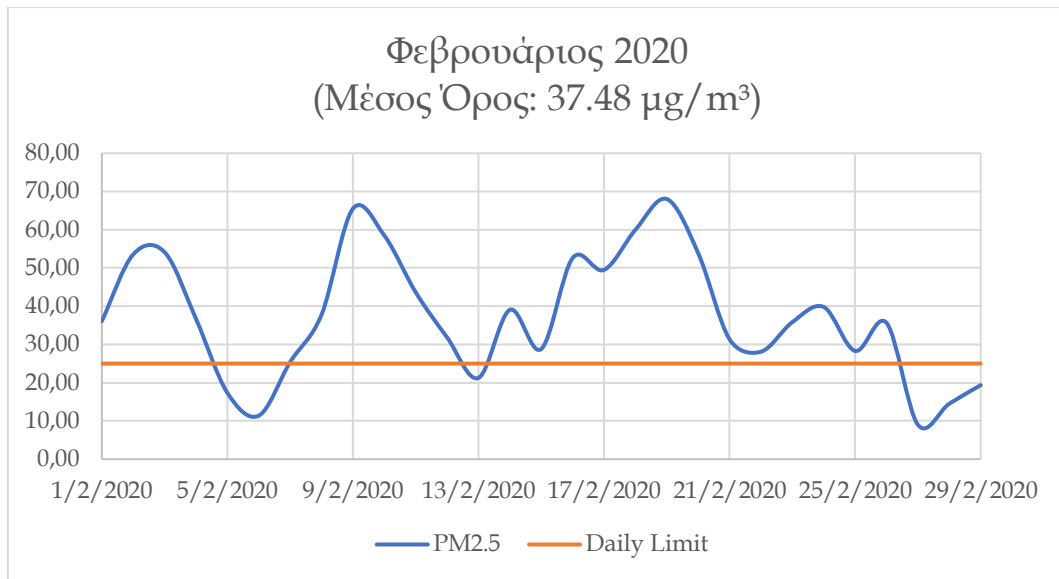
Διάγραμμα 7.12: Ημερήσιοι μέσοι όροι για τον Ιανουάριο 2020.

7.3.12 Ημερήσιοι μέσοι όροι για τον Φεβρουάριο 2020

Ο μέσος όρος των ημερήσιων τιμών των μικροσωματιδίων PM_{2.5} τον Φεβρουάριο του 2020 με βάση τους 12 μετρητές του GreenYourAir ανήλθε στα 37.48 μg/m³. Η μέγιστη τιμή που παρατηρήθηκε είναι 68.01 μg/m³ και η ελάχιστη 8.99 μg/m³. Το ημερήσιο όριο που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας για τα PM_{2.5} είναι 25.00 μg/m³.

| Ημερομηνία | Μέσος όρος PM _{2.5} (μg/m ³) |
|------------|--|
| 1/2/2020 | 36.10 |
| 2/2/2020 | 53.54 |
| 3/2/2020 | 54.12 |
| 4/2/2020 | 36.70 |
| 5/2/2020 | 17.34 |
| 6/2/2020 | 11.38 |
| 7/2/2020 | 25.41 |
| 8/2/2020 | 37.79 |
| 9/2/2020 | 65.51 |
| 10/2/2020 | 58.50 |
| 11/2/2020 | 43.63 |
| 12/2/2020 | 31.84 |
| 13/2/2020 | 21.26 |
| 14/2/2020 | 39.07 |
| 15/2/2020 | 28.80 |
| 16/2/2020 | 52.62 |
| 17/2/2020 | 49.50 |
| 18/2/2020 | 60.00 |
| 19/2/2020 | 68.01 |
| 20/2/2020 | 53.87 |
| 21/2/2020 | 31.34 |
| 22/2/2020 | 28.10 |
| 23/2/2020 | 35.86 |
| 24/2/2020 | 39.79 |
| 25/2/2020 | 28.31 |
| 26/2/2020 | 35.63 |
| 27/2/2020 | 8.99 |
| 28/2/2020 | 14.52 |
| 29/2/2020 | 19.37 |

Πίνακας 7.20: Ημερήσιοι μέσοι όροι για τον Φεβρουάριο 2020.



Διάγραμμα 7.13: Ημερήσιοι μέσοι όροι για τον Φεβρουάριο 2020.

8 Συμπεράσματα

Στην ενότητα αυτή αναλύονται τα συμπεράσματα που μπορούν να εξαχθούν από την μελέτη. Αυτά αφορούν την συγκέντρωση των μικροσωματιδίων $\text{PM}_{2.5}$ στην πόλη του Βόλου με βάση τις μετρήσεις του δικτύου GreenYourAir το οποίο αποτελείται από 12 μετρητές χαμηλού κόστους. Λόγω της μεγάλης χρονικής περιόδου της μελέτης (01/03/2019-29/02/2020) και του μεγάλου όγκου των δεδομένων (οι μετρητές στέλνουν δεδομένα κάθε 3 λεπτά) μπορούμε να καταλήξουμε σε πιο ασφαλή συμπεράσματα για την ποιότητα του αέρα της πόλης.

Οι κύριες πηγές ρύπανσης της πόλης μπορούν να διαχωριστούν στις παρακάτω βασικές κατηγορίες:

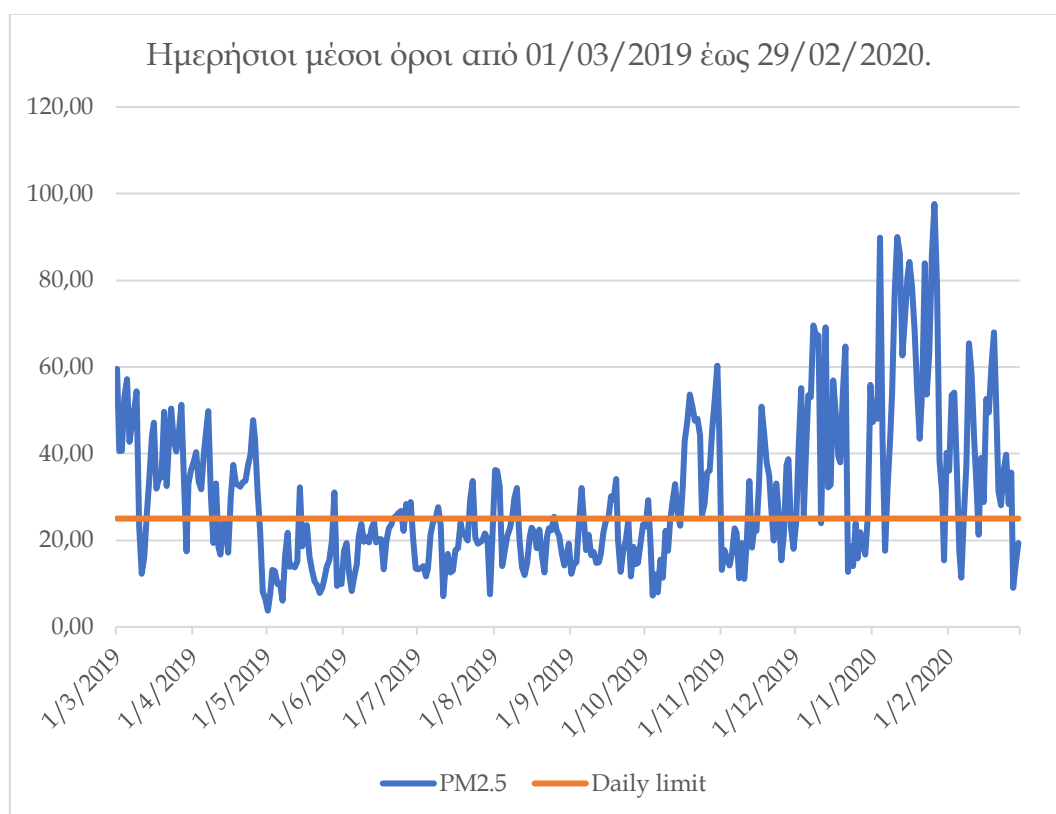
- Βιομηχανική Δραστηριότητα
- Θέρμανση οικιών, καταστημάτων κ.λπ.
- Κινητήρες αυτοκινήτων, φορτηγών κ.λπ.
- Επιβατικό και Εμπορικό Λιμάνι
- Επιβατικοί σταθμοί Λεωφορείων

Φυσικά οι παραπάνω δεν είναι οι μοναδικές πηγές ρύπανσης. Το φαινόμενο της ατμοσφαιρικής ρύπανσης αποτελεί ένα μεγάλο πρόβλημα παγκοσμίως και η αντιμετώπιση του είναι αρκετά δύσκολη. Επίσης οι ρύποι δεν δημιουργούνται μόνο μέσα στην πόλη αλλά μπορούν να μεταφερθούν από άλλες πόλεις οι οποίες βρίσκονται δεκάδες ή ακόμα εκατοντάδες χιλιόμετρα μακριά. Εκτός από τις ανθρωπογενείς πηγές ρύπανσης υπάρχουν ακόμα και οι φυσικές πηγές ρύπανσης όπως η σκόνη που μεταφέρεται από τις ερήμους (το φαινόμενο αυτό παρατηρείται στην πόλη του Βόλου

τις περιόδους όπου υπάρχουν Νότιοι άνεμοι και μεταφέρουν σκόνη από την έρημο Σαχάρα), οι δασικές πυρκαγιές, η ηφαιστειακή δραστηριότητα και πολλές άλλες.

8.1 Ποιότητα του αέρα στην πόλη του Βόλου για την χρονική περίοδο από 01/03/2019 έως 29/02/2020

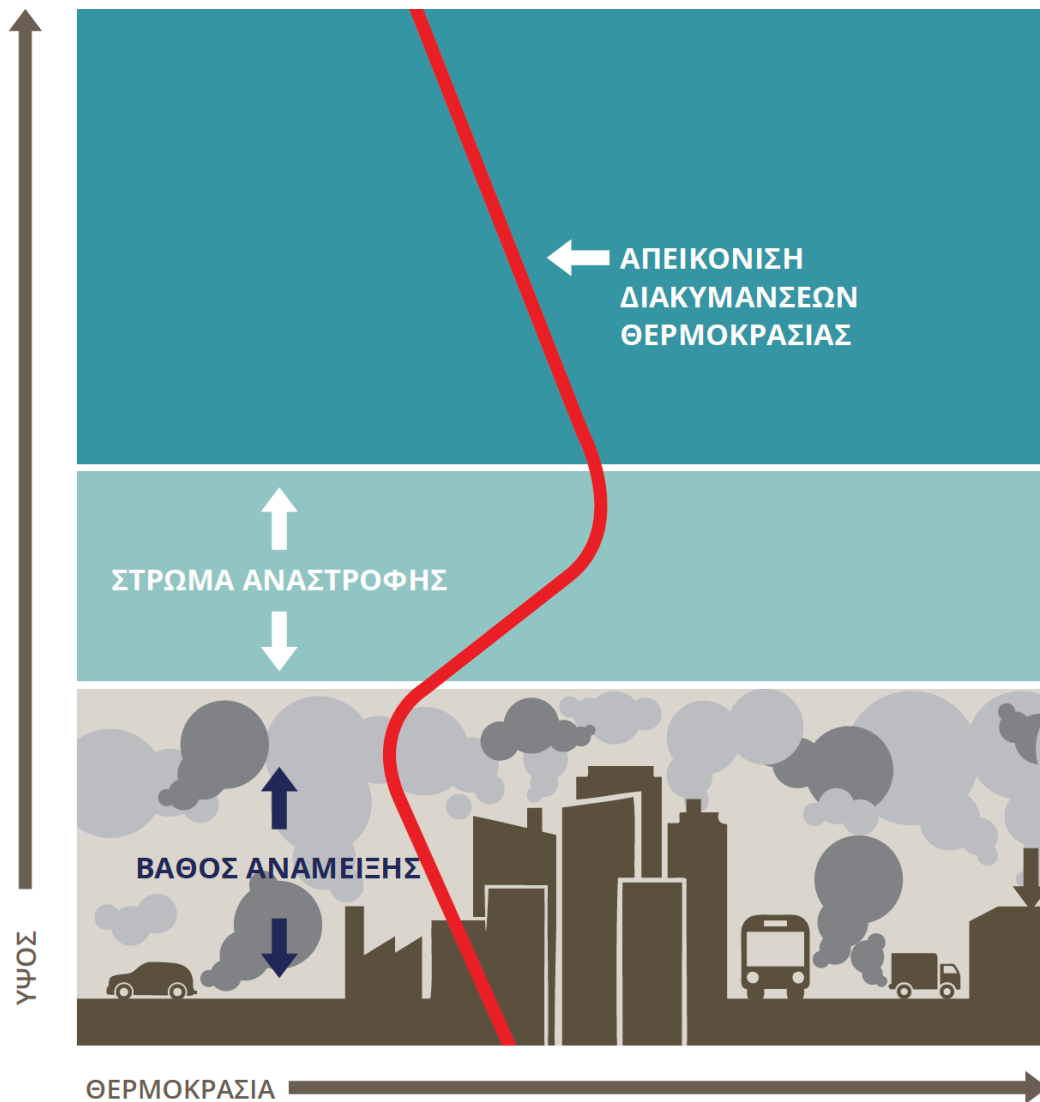
Με βάση το παρακάτω διάγραμμα το οποίο περιέχει τους ημερήσιους μέσους όρους της συγκέντρωσης των μικροσωματιδίων $PM_{2.5}$ για την χρονική περίοδο από 01/03/2019 έως 29/02/2020 παρατηρείται ότι οι μεγαλύτερες υπερβάσεις εντοπίζονται τους χειμερινούς μήνες. Οι ημερήσιοι μέσοι όροι των θερινών μηνών παρατηρείται να είναι κατά κύριο λόγο κάτω από το ημερήσιο όριο των $25.00 \mu g/m^3$ που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Υγείας. Συνολικά 178 ημέρες από τις 366 οι τιμές είναι μεγαλύτερες από το ημερήσιο όριο. Φυσικά η ανθρώπινη δραστηριότητα είναι μία από τις μεγαλύτερες πηγές ρύπανσης.



Διάγραμμα 8.1: Ημερήσιοι μέσοι όροι από 01/03/2019 έως 29/02/2020.

Οι μετεωρολογικές συνθήκες καθώς και τα γεωμορφολογικά χαρακτηριστικά της πόλης επηρεάζουν σημαντικά την ποιότητα του αέρα καθώς και την συγκέντρωση και την παραμονή των μικροσωματιδίων $PM_{2.5}$ στην ατμόσφαιρα για μεγαλύτερο διάστημα. Ένα συχνό φαινόμενο που παρατηρείται στην πόλη τους χειμερινούς μήνες και είναι υπεύθυνο για τον εγκλωβισμό των μικροσωματιδίων $PM_{2.5}$ είναι η

θερμοκρασιακή αναστροφή [12]. Το φαινόμενο της θερμοκρασιακής αναστροφής παρατηρείται του χειμερινούς μήνες και οφείλεται στο ότι τη νύχτα η απουσία της νεφοκάλυψης έχει ως αποτέλεσμα το έδαφος να χάνει θερμότητα με γρηγορότερους ρυθμούς και ο αέρας που έρχεται σε επαφή με το έδαφος να ψύχεται. Ο θερμότερος αέρας ανυψώνεται και ενεργεί σαν καπάκι παγιδεύοντας τον ψυχρότερο αέρα κοντά στο έδαφος. Με τον τρόπο αυτό η ρύπανση παγιδεύεται και έτσι το στρώμα αέρα που βρίσκεται κοντά στο έδαφος να μολύνεται όλο και περισσότερο μέχρι να αλλάξουν οι μετεωρολογικές συνθήκες της περιοχής.



Εικόνα 8.1: Θερμοκρασιακή αναστροφή.

8.2 Τρόποι Αντιμετώπισης

Η ποιότητα του αέρα που αναπνέουμε επηρεάζει την καθημερινότητα μας και την υγεία μας σε μεγάλο βαθμό. Συνεπώς η αντιμετώπιση της ατμοσφαιρικής ρύπανσης είναι ένα σημαντικό πρόβλημα στο οποίο πρέπει να βρεθούν λύσεις άμεσα. Οι λύσεις και τα μέτρα που προτείνονται μπορούν να κατηγοριοποιηθούν σε συλλογικό και ατομικό επίπεδο.

Συλλογικό επίπεδο:

- Περιορισμός της βιομηχανικής δραστηριότητας και συγκεκριμένα των βιομηχανιών που λειτουργούν με ορυκτά καύσιμα όπως το πετρέλαιο. Ο περιορισμός μπορεί να αποδεικνύεται με στοιχεία μείωσης της παραγωγής ή κατανάλωσης καυσίμου σε σχέση με το μέσο όρο λειτουργίας των προηγούμενων επτά ημερών.
- Μέτρηση και ποσοτικοποίηση των ρύπων που παράγονται από την βιομηχανική δραστηριότητα με σκοπό την ελάττωση τους με την χρήση τεχνολογιών όπως τα φίλτρα.
- Ημέρα χωρίς αυτοκίνητο. Οι ημέρες χωρίς αυτοκίνητο είναι ένα πολύ καλός τρόπος για να ενθαρρύνει κανείς του ανθρώπους να διερευνήσουν εναλλακτικούς τρόπους κυκλοφορίας στην πόλη. Πρόκειται για μία δράση που λαμβάνει χώρα σε πολλές πόλεις κατά τη διάρκεια της Ευρωπαϊκής Κινητικότητας, εκστρατεία που υποστηρίζεται από την Ευρωπαϊκή επιτροπή και διοργανώνεται κάθε Σεπτέμβριο. Το 2018 συμμετείχαν περισσότερες από 2400 πόλεις σε περισσότερες από 50 χώρες. Χάρη στην εκστρατεία έχουν υιοθετηθεί περισσότερα από 8000 μόνιμα μέτρα, όπως η από κοινού χρήση οχημάτων και προγράμματα κοινής χρήσης ποδηλάτων.
- Ενθάρρυνση αγοράς ηλεκτρικών αυτοκινήτων. Πέρα από την επιχορήγηση στις αγορές ηλεκτρικών αυτοκινήτων, ιδανική θα ήταν και η εισαγωγή πράσινων πινακίδων, με την χρήση των οποίων οι ιδιοκτήτες θα έχουν προνόμια στα τέλη κυκλοφορίας, στην χρήση πάρκινγκ κ.λπ.
- Απαγόρευση κυκλοφορίας οχημάτων που δεν διαθέτουν καταλύτη, των Ι.Χ. πετρελαιοκίνητων οχημάτων στο κέντρο της πόλης καθώς και των πετρελαιοκίνητων επαγγελματικών οχημάτων.
- Εκσυγχρονισμός του στόλου των αυτοκινήτων του δήμου, όπως τα απορριμματοφόρα, τα αυτοκίνητα που χρησιμοποιούν τα συνεργεία του δήμου κ.λπ.
- Αναβάθμιση των σπιτιών και των δημόσιων κτιρίων και της ενεργειακής τους κλάσης με σκοπό την μείωση της ενέργειας που απαιτείται για την θέρμανση του σπιτιού, καθώς και αλλαγή του τρόπου θέρμανσης σε έναν πιο φιλικό προς το περιβάλλον (π.χ. λέβητες φυσικού αερίου).
- Αναβάθμιση του στόλου των μέσων μαζικής μεταφοράς καθώς και του δικτύου με σκοπό την χρήση και την εξυπηρέτηση περισσότερων πολιτών.
- Έλεγχος στις καύσεις στα τζάκια των σπιτιών τους χειμερινούς μήνες για την ελάττωση της παραγωγής ρύπων.

- Μέτρα που έχουν ως σκοπό την μείωση της κυκλοφοριακής συμφόρησης στους δρόμους της πόλης και ειδικότερα στους κεντρικούς δρόμους. Πρόστιμα για παράνομο παρκάρισμα σε κεντρικούς δρόμους που δυσκολεύουν την κυκλοφορία των οχημάτων.
- Δακτύλιος στο κέντρο της πόλης με σκοπό την μείωση της κυκλοφοριακής συμφόρησης.
- Κατασκευή ποδηλατοδρόμων με σκοπό την χρήση ποδηλάτων από τους πολίτες με περισσότερη ασφάλεια.
- Κατασκευή δικτύων καταγραφής της ποιότητας του αέρα κάθε πόλης στην οποία οι πολίτες θα μπορούν να έχουν πρόσβαση στα δεδομένα σε πραγματικό χρόνο.

Σε ατομικό επίπεδο:

- Καθαρότερη μεταφορά όπως για παράδειγμα η χρήση ποδηλάτων, μέσω μαζικής μεταφοράς και το περπάτημα.
- Βελτιστοποίηση στον τρόπο χρήσης ενέργειας που καταναλώνεται από το σπίτι, για παράδειγμα αλλαγή των λαμπτήρων σε ενεργειακά αποδοτικούς λαμπτήρες, χρήση έξυπνων θερμοστατών που ορίζουν την θερμοκρασία του χώρου με αποδοτικότερο τρόπο κ.λπ.
- Παραγωγή λιγότερων πλαστικών απορριμμάτων καθώς η ανακύκλωση τους και η παραγωγή τους απαιτεί μεγαλύτερες ποσότητες ενέργειας.
- Ελαχιστοποίηση της χρήσης του αυτοκινήτου για κοντινές αποστάσεις μέσα στο κέντρο διότι η εύρεση πάρκινγκ έχει ως αποτέλεσμα την κατανάλωση περισσότερου καυσίμου και αύξηση της κυκλοφοριακής συμφόρησης.

Οι συλλογικοί και οι ατομικοί μέθοδοι για την ελαχιστοποίηση του προβλήματος της ρύπανσης είναι σημαντικό να ληφθούν από όλους χωρίς εξαιρέσεις. Φυσικά για να γίνει αυτό πρέπει πρώτα να συνειδητοποιήσουμε ως πολίτες το πρόβλημα και στην συνέχεια να προσπαθήσουμε να το λύσουμε είτε σε συλλογικό είτε σε ατομικό επίπεδο.

8.3 Περιθώρια για περαιτέρω έρευνα

Η παρούσα εργασία αποτελεί μία προσπάθεια ανάπτυξης ενός δικτύου μέτρησης των μικροσωματιδίων PM_{2.5} στην πόλη του Βόλου. Σκοπός είναι η ενημέρωση των πολιτών και η πρόσβαση αυτών στα δεδομένα σε πραγματικό χρόνο. Παρακάτω παρουσιάζονται κάποιες ιδέες για ανάπτυξη και περαιτέρω έρευνα.

- Βελτιστοποίηση της κατανομής των μετρητών στην πόλη για την καλύτερη καταγραφή της ρύπανσης.
- Βελτίωση και ανάπτυξη των μεθόδων που χρησιμοποιούνται για την καταγραφή και την μεταφορά δεδομένων.

- Ανάπτυξη συσκευών που μπορούν να μετρήσουν και άλλους ρύπους.
- Δημιουργία δικτύου και σε άλλες πόλεις της Ελλάδας.
- Καλύτερη ενημέρωση των πολιτών χρησιμοποιώντας και άλλες μεθόδους ενημέρωσης εκτός από την ιστοσελίδα του GreenYourAir.
- Μοντέλα για τον επιμερισμό των πηγών και τον εντοπισμό αυτών με σκοπό την αντιμετώπιση του προβλήματος της ρύπανσης πιο στενευμένα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. WHO, Ambient (outdoor) air pollution, 2018 [Πρόσβαση 10/01/2021, [https://www.who.int/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health)].
2. EEA, Premature deaths attributable to air pollution, 2021 [Πρόσβαση 15/01/2021, <https://www.eea.europa.eu/media/newsreleases/many-europeans-still-exposed-to-air-pollution-2015/premature-deaths-attributable-to-air-pollution>].
3. EEA, Air quality standards [Πρόσβαση 11/05/2020, <https://www.eea.europa.eu/themes/air/air-quality-concentrations/air-quality-standards>].
4. ΚΥΑ-Μέτρα για τη βελτίωση της ποιότητας της ατμόσφαιρας. [<https://www.moh.gov.gr/articles/health/dieythynsh-dhmosias-ygieinh/ygieinh-%20periballontos/prostasia-poiothtas-aera/3425-nomothesia-armodiothtwn-ths-d-nshs-%20ygeionomikhs-mhxanikhs-kai-ygieinh-periballontos>].
5. M. Gao, J. Cao, E. Seto, “A distributed network of low-cost continuous reading sensors to measure spatiotemporal variations of PM2.5 in Xi'an, China”, Environmental Pollution, 199, 2015, 56-65.
6. W. Jiao, et al., “Community Air Sensor Network (CAIRSENSE) Project: Lower Cost, Continuous Ambient Monitoring Methods”, Presented at 108th AWMA Conference, RTP, NC, June 23 - 25, 2015.
7. C. Borrego, et al., “Assessment of air quality microsensors versus reference methods: The EuNetAir joint exercise”, Atmospheric Environment, 147, 2016, 246-263.
8. P. Schneider, et al., “Mapping urban air quality in near real-time using observations from low-cost sensors and model information”, Environment International, 106, 2017, 234-247.
9. W. Ying Yi, et al., “A Survey of Wireless Sensor Network Based Air Pollution Monitoring Systems”, DOI: 10.3390/s151229859, 2015.
10. E. Bozyazi, et al., “Analysis and mapping of air pollution using a GIS approach: A case study of Istanbul”, DOI: 10.2495/AIR000431, 2000.
11. N. Castell, et.al, “Localized real-time information on outdoor air quality at kindergartens in Oslo, Norway using low-cost sensor nodes”, 165, 2019, 410-419.
12. ΕΟΠ, Η θερμοκρασιακή αναστροφή παγιδεύει τη ρύπανση στο επίπεδο του εδάφους [Πρόσβαση 08/09/2016, <https://www.eea.europa.eu/el/pressroom/grafikes-plirofories/i-thermokrasiaki-anastrofi-pagideyei-ti/view>]

Χρήσιμοι σύνδεσμοι:

- <https://airly.org/map/en/>
- <https://waqi.info/>
- <https://www.epa.gov/>
- <https://www.who.int/>

- <https://www.wikipedia.org/>
- <https://stackoverflow.com/>
- <https://www.w3schools.com/>
- <https://www.arduino.cc/>
- <https://www.google.com/sheets/about/>
- <https://www.google.gr/drive/about.html>
- <https://www.google.com/script/start/>
- <https://developers.google.com/apps-script>
- <https://leafletjs.com/reference-1.7.1.html>
- <https://getbootstrap.com/>
- <https://fontawesome.com/>
- <https://www.openstreetmap.org>
- <https://openweathermap.org/>
- <https://analytics.google.com/analytics/web/provision/#/provision>
- <https://www.dfrobot.com/>
- <http://www.plantower.com/en/>
- http://www.nodemcu.com/index_cn.html

Πηγές Εικόνων:

- Εικόνα 2.1 Πηγή: <https://www.tovima.gr/2019/06/07/society/tropologia-gia-ti-viosimotita-tis-lignitikis-monadas-ptolemaida-v/>
- Εικόνα 2.2 Πηγή: <https://pergamos.lib.uoa.gr/uoa/dl/frontend/file/lib/default/data/2814386/theFile>
- Εικόνα 2.3 Πηγή: <https://www.iefimerida.gr/news/82110/%CF%80%CF%8E%CF%82-%CE%BF-%CE%BA%CE%B1%CF%80%CE%BD%CF%8C%CF%82-%CE%B1%CF%80%CF%8C-%CF%84%CE%B1-%CF%84%CE%B6%CE%AC%CE%BA%CE%B9%CE%B1-%CF%83%CE%BA%CF%8C%CF%84%CF%89%CF%83%CE%B5-4000-%CE%BB%CE%BF%CE%BD%CE%B4%CF%81%CE%AD%CE%B6%CE%BF%CF%85%CF%82-%CF%84%CE%BF-1952>
- Εικόνα 2.4, Πηγή: <https://thairesidents.com/local/government-established-14-regulations-dealing-pm2-5-crisis/>
- Εικόνα 2.5, Πηγή: <https://www.epa.gov/pm-pollution/particulate-matter-pm-basics>
- Εικόνα 3.1, Πηγή: https://www.nodemcu.com/index_en.html#fr_54747661d775ef1a3600009e
- Εικόνα 3.2, Πηγή: http://www.aqmd.gov/docs/default-source/aq-spec/resources-page/plantower-pms5003-manual_v2-3.pdf?sfvrsn=2
- Εικόνα 3.3, Πηγή: http://www.aqmd.gov/docs/default-source/aq-spec/resources-page/plantower-pms5003-manual_v2-3.pdf?sfvrsn=2
- Εικόνα 3.4, Πηγή: <https://www.ebay.com/itm/Digital-DHT11-DHT-11-Sensitivity-Temperature-and-Humidity-Sensor-For-Arduino-/253376104535>
- Εικόνα 3.5, Πηγή: <http://www.massuse-relay.com/details/MTSSeries.pdf>

- Εικόνα 3.6, Πηγή: <https://www.indiamart.com/proddetail/jumper-wire-m-m-f-f-f-15359553673.html>
- Εικόνα 8.1, Πηγή: <https://www.eea.europa.eu/el/pressroom/grafikes-plirofories/i-thermokrasiaki-anastrofi-pagideyei-ti/view>