



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΥΛΟΠΟΙΗΣΗ ΜΙΑΣ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΤΗΝ  
ΕΥΡΕΣΗ ΠΛΗΡΟΦΟΡΙΩΝ ΔΗΜΩΝ ΤΟΥ  
ΝΟΜΟΥ ΘΕΣΣΑΛΟΝΙΚΗΣ.

Πτυχιακή Εργασία

Θεσσαλονίκη 2025

Τίτλος Π.Ε. Εφαρμογή για την εύρεση πληροφοριών που αφορούν το περιβάλλον των  
Δήμων του νομού Θεσσαλονίκης.

Κωδικός Π.Ε.25100

Όνοματεπώνυμο φοιτητή : Ευστράτιος Κουντουράς

Όνοματεπώνυμο εισηγητή καθηγητή : Χαράλαμπος Μπράτσας

Ημερομηνία ανάληψης Π.Ε 08-01-2025

**Υπεύθυνη Δήλωση**: Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών του Διεθνούς Πανεπιστημίου της Ελλάδας.

# Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Μπράτσα για την καθοδήγηση που μου προσέφερε κατά την εκπόνηση αυτή της πτυχιακής εργασίας και την βοήθειά του για την συλλογή των πληροφοριών, την χρήση αλλά και παρουσίαση αυτών.

Επιπλέον, θα ήθελα να ευχαριστήσω την οικογένειά μου που με στήριξαν καθόλη την διάρκεια φοίτησης μου.

# Περιεχόμενα

<b>Ευχαριστίες</b>	<b>4</b>
<b>Περιεχόμενα</b>	<b>5</b>
<b>Εισαγωγή</b>	<b>6</b>
Σημασία Πτυχιακής Εργασίας	8
Στόχοι Πτυχιακής Εργασίας	8
Δομή Πτυχιακής Εργασίας	9
<b>Θεωρητικό Υπόβαθρο</b>	<b>10</b>
Θεωρία	10
Σχετικές Ενέργειες	11
Υπάρχουσες Τεχνολογίες	12
Βάση Δεδομένων	13
Υπόβαθρο για τον αναγνώστη	13
Γεωχωρικές Συντεταγμένες (Latitude / Longitude)	13
Λειτουργία API και RESTful Αρχιτεκτονική	13
Αισθητήρας Περιβάλλοντος (π.χ. PM2.5 Sensor)	14
Προβαλλόμενα και Γεωγραφικά Συστήματα Αναφοράς	14
Δομή και Οργάνωση Βάσεων Δεδομένων (Συσχετίσεις)	14
Η δική μας λύση	15
Γεωγραφική Περιγραφή & Δεδομένα	15
Γεωγραφική κατανομή και ιδιαιτερότητες	15
Επισκόπηση των δεδομένων βλάστησης	16
Επισκόπηση των δεδομένων ατμοσφαιρικής ρύπανσης	16
Τεχνολογική Υλοποίηση	17
Λειτουργικότητα	17
Πρόσθετες Λειτουργικότητες	18
Τεχνολογίες	22
<b>Απαιτήσεις &amp; Σχεδίαση Συστήματος</b>	<b>23</b>
Απαιτήσεις Συστήματος	23
Λειτουργικές Απαιτήσεις	23
Μη Λειτουργικές Απαιτήσεις	23
Σχεδίαση Συστήματος	24
Αρχιτεκτονική Συστήματος	25
Front-end	25
Back-end	26
Εξωτερικά Εργαλεία	26
Ρόλοι Χρηστών και Δυνατότητες	26
Χρήστης	26
Διαχειριστής	27
Τεχνολογικές Επιλογές	28
<b>Υλοποίηση Συστήματος</b>	<b>29</b>
Front-end	29
Back-end	29

Υλοποίηση Βάσης Δεδομένων	37
Εξωτερικά Εργαλεία	38
Python	39
Βήματα Παραγωγής Κώδικα Μετεωρολογικών Σταθμών	39
Βήματα Παραγωγής Κώδικα Ρύπανσης	40
Βήματα Παραγωγής Κώδικα Τύπου Δένδρων	43
Βήματα Παραγωγής Κώδικα Οδών Δένδρων	45
Βήματα Παραγωγής Κώδικα Δένδρων	46
Εισαγωγή στην Βάση	47
APIs	48
Open Street Maps	48
Weather API	49
<b>Λειτουργικότητα &amp; Αξιολόγηση</b>	<b>51</b>
Εγγραφή & Είσοδος Χρήστη	51
Πλοήγηση Χρήστη	57
Οθόνη Προφίλ	57
Οθόνη Χάρτη Δένδρων	59
Φίλτρα Αναζήτησης	60
Αιτήματα Δέντρων	62
Οθόνη Βάσης Δέντρων	63
Οθόνη Κοινότητας	66
Οθόνη Ρύπανσης	66
Οθόνη Διαχειριστή	68
Ενεργοί Χρήστες	68
Αιτήματα	69
Οθόνη Βάσης Δέντρων	71
Δομή Παραδοτέων	72
<b>Για την ευκολία χρήσης έχουν κατασκευαστεί οι ακόλουθοι λογαριασμοί:</b>	<b>73</b>
<b>Συμπεράσματα &amp; Μελλοντικές Επεκτάσεις</b>	<b>73</b>
Συμπεράσματα	73
Προκλήσεις & Περιορισμοί	74
Μελλοντικές Επεκτάσεις	75
Υποστήριξη ανεβάσματος αρχείων (CSV/Excel)	75
Επέκταση σε περισσότερους Δήμους	75
Εμπλουτισμός πληροφορίας και κάλυψη ελλείψεων	76
Συστήματα ειδοποίησης και ανάλυσης	76
Χάρτης με live δεδομένα και χρονοδιάγραμμα μεταβολών	76
Εφαρμογή για κινητά	76
<b>Βιβλιογραφία</b>	<b>78</b>

# Εισαγωγή

Η ταχύτατη ανάπτυξη των αστικών κέντρων τα τελευταία χρόνια έχει επιφέρει σημαντικές και, πολλές φορές, μη αναστρέψιμες επιπτώσεις στο φυσικό περιβάλλον. Η συνεχής ανοικοδόμηση, η αύξηση της κυκλοφορίας οχημάτων και των βιομηχανικών δραστηριοτήτων, καθώς και η έντονη εκμετάλλευση των φυσικών πόρων, έχουν οδηγήσει σε ραγδαία αύξηση της ατμοσφαιρικής ρύπανσης, υποβάθμιση των πράσινων χώρων και απομάκρυνση του ανθρώπου από τη φύση (Κεφαλά #) (Χαραλαμπίδου) (SES).

Πλέον, σε πολλές μεγαλουπόλεις, το φαινόμενο της θερμικής νησίδας (urban heat island) είναι εμφανές (Κοντονίκας), τα επίπεδα διοξειδίου του αζώτου (NO<sub>2</sub>) και αιωρούμενων σωματιδίων (PM10, PM2.5) ξεπερνούν κατά πολύ τα όρια ασφαλείας, ενώ η πρόσβαση των πολιτών σε φυσικό πράσινο είναι περιορισμένη (Πεταλάς) (Κωλέτση). Η ποιότητα ζωής των κατοίκων υποβαθμίζεται, με επιπτώσεις στην υγεία, τη ψυχολογική ευεξία και τη δυνατότητα για βιώσιμη αστική ανάπτυξη (LAMDA).

Οι σύγχρονες πόλεις καλούνται, πλέον, να αντιμετωπίσουν ένα πλήθος σύνθετων προκλήσεων (EPA) (Gajare):

- Πώς μπορούν να μειώσουν τα επίπεδα ρύπανσης;
- Πού και πώς μπορούν να φυτευτούν νέα δέντρα ώστε να έχουν πραγματικό αντίκτυπο;
- Πώς μπορεί να ενημερώνεται ο πολίτης άμεσα και υπεύθυνα για τις περιβαλλοντικές συνθήκες της περιοχής του;

Σε αυτό το πλαίσιο, η ανάγκη για παρακολούθηση, καταγραφή, ανάλυση και οπτικοποίηση περιβαλλοντικών δεδομένων καθίσταται πιο επιτακτική από ποτέ. Η τεχνολογία, όταν αξιοποιείται σωστά, δεν είναι απλώς ένα εργαλείο καταγραφής — μπορεί να μετατραπεί σε πολλαπλασιαστή γνώσης και δράσης, τόσο για τους πολίτες όσο και για τους φορείς διακυβέρνησης (CityGreen) (White Oak).

Η ανάπτυξη κατάλληλων πληροφοριακών συστημάτων επιτρέπει (myEnv) (Mittermaier) (EPA) :

- Τη διαφάνεια και την ανοικτή διάχυση πληροφορίας,
- Την ενεργή συμμετοχή των πολιτών,
- Την υποστήριξη τεκμηριωμένων αποφάσεων σε επίπεδο δημοτικής και περιφερειακής αυτοδιοίκησης.

Ο συνδυασμός δεδομένων ρύπανσης με πληροφορίες για την αστική βλάστηση, όπως η παρουσία δέντρων και οι βοτανικές κατηγορίες τους, μπορεί να αποκαλύψει κρίσιμες συσχετίσεις: Ποιες περιοχές με περισσότερα δέντρα παρουσιάζουν βελτιωμένη ποιότητα αέρα; Πού χρειάζεται άμεσα δεντροφύτευση; Υπάρχουν κοινότητες που είναι περισσότερο εκτεθειμένες σε ρύπανση λόγω ελλιπούς πράσινου;

Η παρούσα πτυχιακή εργασία αναπτύσσει μία σύγχρονη διαδικτυακή εφαρμογή που στοχεύει στην απεικόνιση, διαχείριση και ανάλυση περιβαλλοντικών δεδομένων σχετικών με την ατμοσφαιρική ρύπανση και τις θέσεις δέντρων στον αστικό ιστό. Η αρχική εστίαση είναι στον

Δήμο Θεσσαλονίκης, ωστόσο η αρχιτεκτονική της εφαρμογής είναι σχεδιασμένη ώστε να υποστηρίζει μελλοντικά επεκτάσεις σε πολλούς δήμους.

Η εφαρμογή προσφέρει ένα εύχρηστο γραφικό περιβάλλον με διαδραστικό χάρτη, παρουσιάζοντας τις μετρήσεις ατμοσφαιρικής ρύπανσης ανά περιοχή και εμφανίζοντας τη θέση των δέντρων και την κατηγοριοποίησή τους. Παράλληλα, επιτρέπει τη διαχείριση και εισαγωγή νέων δεδομένων από τους χρήστες, λειτουργώντας ως μία βάση για "συλλογική καταγραφή" (collective perception).

Επομένως, η εφαρμογή αντλεί δεδομένα από σταθμούς μέτρησης ρύπανσης, συσχετίζοντάς τα με γεωχωρική πληροφορία για τοποθεσίες δέντρων και τα είδη τους. Μέσω ενός διαδραστικού χάρτη, ο χρήστης μπορεί να:

- Δει σε πραγματικό χρόνο τις περιβαλλοντικές συνθήκες της περιοχής του,
- Εξερευνήσει περιοχές με δεντροστοιχίες ή έλλειψη πρασίνου,
- Αναλύσει τα δεδομένα ανά περιοχή,
- Συμμετέχει στην καταγραφή πληροφοριών προσθέτοντας νέα στοιχεία.

Πέραν της επιστημονικής και τεχνολογικής της διάστασης, η εφαρμογή ενδυναμώνει τους πολίτες, τους καθιστά μέρος της λύσης και ενισχύει τη συλλογική περιβαλλοντική συνείδηση.

## Σημασία Πτυχιακής Εργασίας

Η σημασία μίας τέτοιας εφαρμογής είναι πολυεπίπεδη. Από τη μία, παρέχει στους πολίτες και στους ερευνητές εύκολη πρόσβαση σε δεδομένα που συνήθως βρίσκονται διάσπαρτα ή δύσκολα προσβάσιμα, και από την άλλη, υποστηρίζει τις δημοτικές αρχές και φορείς περιβάλλοντος στη λήψη τεκμηριωμένων αποφάσεων. Η σύνδεση των δεδομένων ρύπανσης με τις περιοχές πρασίνου μπορεί να οδηγήσει σε σημαντικά συμπεράσματα: για παράδειγμα, μπορεί να διαπιστωθεί εάν και κατά πόσο περιοχές με περισσότερα δέντρα παρουσιάζουν χαμηλότερα επίπεδα ρύπανσης, ή ποιες περιοχές χρήζουν περιβαλλοντικής ενίσχυσης.

Καθώς οι γρήγοροι ρυθμοί της καθημερινότητας και η πολύπλοκη και η πληθώρα πληροφορίας που λαμβάνουμε καθημερινά αποτελεί δύσκολη στην διαχείριση της, η σημασία μίας πλατφόρμας που συγκεντρώνει την πληροφορία από διαφορετικά άτομα και την παρουσιάζει μπορεί να συμβάλει σημαντικά στην σωστή διαχείρισή της. Επομένως, η ενεργή συμμετοχή των χρηστών με την προσθήκη νέων δεδομένων συμβάλλει στη δυναμική εξέλιξη της εφαρμογής, καθιστώντας την όχι απλώς ένα στατικό εργαλείο απεικόνισης, αλλά ένα διαδραστικό σύστημα περιβαλλοντικής παρακολούθησης με δυναμική ενημέρωση.

## Στόχοι Πτυχιακής Εργασίας

Οι κύριοι στόχοι της παρούσας εργασίας είναι οι εξής:

- Η ανάπτυξη μίας πλήρους λειτουργικής διαδικτυακής εφαρμογής που συλλέγει, αποθηκεύει και παρουσιάζει δεδομένα ρύπανσης και δέντρων.
- Η δημιουργία κατάλληλων εργαλείων για την εισαγωγή, καθαρισμό και μορφοποίηση

δεδομένων (data processing), κυρίως μέσω Python.

- Η σχεδίαση και υλοποίηση μίας βάσης δεδομένων SQL η οποία να υποστηρίζει τα παραπάνω δεδομένα με τρόπο αποδοτικό και επεκτάσιμο.
- Η ανάπτυξη μίας διπλής διεπαφής χρήστη, με διαχωρισμό μεταξύ βασικού χρήστη και διαχειριστή.
- Η οπτικοποίηση των δεδομένων μέσω διαδραστικού χάρτη και η ενσωμάτωση λειτουργιών που ενισχύουν την εμπειρία του χρήστη.
- Η διερεύνηση των προβλημάτων και προκλήσεων που σχετίζονται με την άντληση, τον καθαρισμό και την ομογενοποίηση των δεδομένων από ετερογενείς πηγές.

Στο τέλος αυτής της εργασίας θα έχει υλοποιηθεί ένα πλήρως λειτουργικό περιβάλλον ιστοσελίδας, στο οποίο οι χρήστες θα μπορούν να παρατηρούν και να ανεβάζουν στοιχεία που αφορά δέντρα. Το περιβάλλον θα έχει εξ αρχής βασικά στοιχεία δέντρων και ρύπανσης όπως αυτά έχουν συλλεχθεί από αρμόδιους φορείς, καθώς και τα τμήματα της Python που επιτρέπουν στην διαχείριση των ακατέργαστων δεδομένων και την εισαγωγή τους στην βάση δεδομένων.

## Δομή Πτυχιακής Εργασίας

Η εργασία έχει δομηθεί ως ακολούθως. Στο κεφάλαιο 1 βρίσκεται η εισαγωγή στο θέμα. Στο κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο σχετικά με παρόμοιες τεχνολογίες, τον αστικό ιστό της Θεσσαλονίκης και την βλάστηση σε αυτόν, εισάγοντας τον αναγνώστη στο ζήτημα και το θέμα της πτυχιακής. Στο κεφάλαιο 3 αναφέρονται οι σχεδιαστικές απαιτήσεις του συστήματος ενώ στο κεφάλαιο 4 η υλοποίηση αυτών στα διαφορετικά συνθετικά στοιχεία, αναφερόμενο τόσο στο back-end όσο και στο front-end, αλλά και στην τεχνική υλοποίηση κάθε ενός από αυτά. Στο κεφάλαιο 5 αναφέρεται η λειτουργικότητα και η αξιολόγηση των συστατικών στοιχείων της ιστοσελίδας, προσφέροντας μία περιγραφή του συνόλου του συστήματος και κάθε ξεχωριστής οθόνης περιήγησης. Τέλος, στο κεφάλαιο 6 παρουσιάζονται τα συμπεράσματα και οι μελλοντικές επεκτάσεις της εφαρμογής.

# Θεωρητικό Υπόβαθρο

Η ενότητα αυτή παρέχει μια σφαιρική επισκόπηση των βασικών θεωρητικών, τεχνολογικών και εφαρμοσμένων πτυχών του πεδίου, με στόχο την κατανόηση του πλαισίου εντός του οποίου εντάσσεται η παρούσα εργασία. Περιλαμβάνονται επίσης συγκρίσεις με άλλες λύσεις και ανάλυση της επιλεγμένης μεθοδολογίας.

## Θεωρία

Η σχέση μεταξύ περιβαλλοντικών δεδομένων και γεωχωρικής ανάλυσης έχει αποτελέσει αντικείμενο πολυάριθμων μελετών τα τελευταία χρόνια, καθώς η ανάγκη για καλύτερη κατανόηση και διαχείριση των φυσικών και αστικών περιβαλλόντων γίνεται ολοένα και πιο επιτακτική. Η ενσωμάτωση γεωχωρικών πληροφοριών με περιβαλλοντικούς δείκτες, όπως η ποιότητα του αέρα, η θερμοκρασία, η υγρασία και η κατανομή της βλάστησης, δημιουργεί νέες δυνατότητες για επιστημονική έρευνα, δημόσια πολιτική, αλλά και καθημερινή χρήση από πολίτες.

Η αξιοποίηση Πληροφοριακών Συστημάτων Γεωγραφικών Δεδομένων (GIS) επιτρέπει την αποθήκευση, ανάλυση και απεικόνιση πολυδιάστατων δεδομένων σε γεωγραφικά συμφραζόμενα (Sowjanya) (Παγωνάκης) (Ποζουκίδου and Τασοπούλου). Μέσω αυτών, οι χρήστες μπορούν να μελετήσουν χωρικά μοτίβα, να εντοπίσουν περιοχές υψηλής ρύπανσης, να εκτιμήσουν το επίπεδο αστικής θερμικής νησίδας ή να αξιολογήσουν τη συμβολή του πρασίνου στη βελτίωση της ποιότητας ζωής (Τσίγκου). Οι διαδραστικοί χάρτες, βασισμένοι σε βιβλιοθήκες όπως το Leaflet.js ή το Mapbox GL, δίνουν τη δυνατότητα για real-time αναπαράσταση δεδομένων με απλό και κατανοητό τρόπο (Sowjanya) (ΟΙΚΟΑΝΑΠΤΥΞΗ).

Τα τελευταία χρόνια, η ραγδαία εξάπλωση του φαινομένου των Έξυπνων Πόλεων (Smart Cities) έχει επηρεάσει σε βάθος τη στρατηγική πολλών Δήμων και Περιφερειών. Η ψηφιοποίηση των υπηρεσιών και η εγκατάσταση αισθητήρων IoT (Internet of Things) σε αστικές υποδομές (Sowjanya) (ΟΙΚΟΑΝΑΠΤΥΞΗ), όπως σύλολι φωτισμού, στάσεις λεωφορείων και δημόσια κτίρια, επιτρέπουν την αυτόματη συλλογή περιβαλλοντικών δεδομένων σε πραγματικό χρόνο (Παγωνάκης) (GeoForest). Αυτά τα δεδομένα μπορούν να περιλαμβάνουν από επίπεδα PM2.5 και PM10, έως CO<sub>2</sub>, NO<sub>2</sub>, ή μετρήσεις ηχορύπανσης και θερμικής ακτινοβολίας.

Επιπλέον, τεχνολογίες crowdsourcing, όπου οι ίδιοι οι πολίτες συμμετέχουν στην παροχή δεδομένων μέσω κινητών συσκευών ή εφαρμογών, ενισχύουν την ακρίβεια και την κάλυψη των δεδομένων, ενώ ταυτόχρονα ενδυναμώνουν τη συμμετοχή του κοινού στη διαμόρφωση πολιτικών (ΟΙΚΟΑΝΑΠΤΥΞΗ).

Στο πλαίσιο αυτό, οι τεχνολογίες Web Mapping και οι δυναμικές βάσεις δεδομένων διαδραματίζουν κρίσιμο ρόλο. Επιτρέπουν τη σύνθεση μεγάλου όγκου δεδομένων, την προσωποποιημένη παρουσίαση τους με βάση τη γεωγραφική τοποθεσία του χρήστη, και τη δημιουργία πλατφορμών παρακολούθησης και λήψης αποφάσεων για τις τοπικές αρχές. Χρησιμοποιώντας RESTful APIs και τεχνικές front-end rendering, οι εφαρμογές πλέον μπορούν να παρουσιάζουν δεδομένα περιβαλλοντικής κατάστασης ανά δήμο ή ακόμα και

ανά γειτονιά, υποστηρίζοντας λειτουργίες όπως φίλτρα, χρονοσειρές, συγκρίσεις και διαγράμματα.

Ιδιαίτερη σημασία δίνεται και στη διαλειτουργικότητα μεταξύ διαφορετικών συστημάτων, με τη χρήση προτύπων δεδομένων όπως GeoJSON, WMS/WFS, και την υποστήριξη προβαλλόμενων συστημάτων συντεταγμένων (όπως WGS 84, ΕΓΣΑ87 κ.ά.), εξασφαλίζοντας τη σωστή απεικόνιση και συνδυασμό δεδομένων από ετερογενείς πηγές (Sowjanya) (Ποζουκίδου and Τασοπούλου).

Η ενσωμάτωση όλων αυτών των τεχνολογιών όχι μόνο ενισχύει τη διαφάνεια και την πληροφόρηση του πολίτη, αλλά συνεισφέρει και στη λήψη τεκμηριωμένων αποφάσεων από τις δημοτικές αρχές σε θέματα πολεοδομικού σχεδιασμού, περιβαλλοντικής διαχείρισης και βιώσιμης ανάπτυξης.

## Σχετικές Ενέργειες

Υπάρχουν πολυάριθμες εφαρμογές σε παγκόσμιο επίπεδο που σχετίζονται με την απεικόνιση περιβαλλοντικών παραμέτρων, οι οποίες αξιοποιούν τεχνολογίες γεωχωρικής ανάλυσης, αισθητήρων IoT, τεχνητής νοημοσύνης και crowdsourcing για την παροχή χρήσιμων πληροφοριών προς πολίτες, επιστήμονες και αρμόδιες αρχές (Sowjanya) (GeoForest) (Τζωρτζάκης #) (Geomeletiki). Ορισμένες από τις πιο χαρακτηριστικές περιπτώσεις περιλαμβάνουν τα ακόλουθα.

**AirVisual (IQAir) και Plume Labs (Flow):** Πρόκειται για εφαρμογές που συλλέγουν σε πραγματικό χρόνο δεδομένα ατμοσφαιρικής ρύπανσης (PM2.5, PM10, CO, NO<sub>2</sub>, O<sub>3</sub>, κ.λπ.) από ένα συνδυασμό επίσημων σταθμών μέτρησης, crowdsourced συσκευών από χρήστες και φορητούς αισθητήρες (IQAir) (PlumeLabs). Οι εφαρμογές παρέχουν δυναμικούς χάρτες ποιότητας αέρα σε παγκόσμια κλίμακα, ειδοποιήσεις κινδύνου, καθώς και προβλέψεις βασισμένες σε μοντέλα πρόγνωσης (Sowjanya) (GeoForest) (Κόκκινος).

**TreeMap NYC:** Η εφαρμογή του Δήμου της Νέας Υόρκης καταγράφει κάθε δέντρο εντός της πόλης, παρέχοντας πληροφορίες όπως το είδος, το ύψος, η ηλικία, η θέση του, καθώς και οικολογικές παραμέτρους όπως το ποσό CO<sub>2</sub> που απορροφά ή η εξοικονόμηση νερού και ενέργειας που παρέχει μέσω σκίασης (Sowjanya) (Χαλκιάς) (NYC Trees)

**Treepedia / Green View Index (MIT & Google):** Βασιζόμενο σε εικόνες Google Street View, το Treepedia αναλύει πόσο “πράσινο” βλέπει ο μέσος πεζός σε διάφορες πόλεις του κόσμου (Treepedia) (Χiaojiang et al. #). Η μέτρηση γίνεται με υπολογισμό του ποσοστού πρασίνου που φαίνεται στις εικόνες μέσω αλγορίθμων επεξεργασίας εικόνας και AI, δημιουργώντας έναν δείκτη αστικής βλάστησης που διαφέρει από τις παραδοσιακές δορυφορικές μεθόδους.

**Barcelona Smart City Platform:** Η Βαρκελώνη έχει υλοποιήσει ένα ολοκληρωμένο σύστημα το οποίο ενοποιεί δεδομένα ρύπανσης, θερμοκρασίας, ηχορύπανσης και βλάστησης από αισθητήρες σε πραγματικό χρόνο (Barcelona). Μέσω της πλατφόρμας Sentilo, οι πολίτες μπορούν να δουν δεδομένα ανά δρόμο, να ενημερωθούν για κρίσεις ρύπανσης ή καύσιμα,

και να συμμετέχουν ενεργά στην καταγραφή προβλημάτων (Sowjanya) (Barcelona).

**Amsterdam Smart Citizens Lab:** Η Άμστερνταμ προωθεί συμμετοχικές δράσεις μέσω του Smart Citizens Lab, όπου πολίτες εκπαιδεύονται να κατασκευάζουν δικούς τους αισθητήρες και να συμμετέχουν στη συλλογή και ανάλυση περιβαλλοντικών δεδομένων, δημιουργώντας ένα δίκτυο ανοικτών δεδομένων για την πόλη (WaagFutureLab) (AmsterdamCity) (Henriquez #).

Παρά την ποικιλία και την τεχνολογική καινοτομία των παραπάνω παραδειγμάτων, λίγες εφαρμογές επικεντρώνονται σε επίπεδο Δήμου με στόχο τη συνδυαστική και διαδραστική απεικόνιση βλάστησης, ρύπανσης, μικροκλίματος και τοπικών διοικητικών ορίων σε ενιαίο, προσβάσιμο περιβάλλον (Sowjanya) (GeoForest).

Επιπλέον, πολλές εφαρμογές υστερούν στη διαλειτουργικότητα ή στην επαναχρησιμοποίηση δεδομένων από τοπικές αρχές, καθώς βασίζονται κυρίως σε κεντροποιημένες ή εμπορικές πηγές. Σε ελάχιστες περιπτώσεις γίνεται αξιοποίηση των δεδομένων για την ενίσχυση τοπικών πολιτικών ή για προσωποποιημένες ειδοποιήσεις προς τους δημότες.

Το υπάρχον κενό αφορά κυρίως (1) την έλλειψη εργαλείων που να απευθύνονται αποκλειστικά σε Δήμους μικρότερης κλίμακας, όπου δεν υπάρχει η τεχνογνωσία ή οι πόροι για μεγάλα έργα, (2) τη δυσκολία παραμετροποίησης για τοπικές ανάγκες, καθώς οι περισσότερες λύσεις είναι είτε γενικές είτε υπερτοπικές και (3) την απουσία σύνδεσης με τις γεωγραφικές ιδιαιτερότητες του εκάστοτε τόπου, όπως π.χ. ορεινότητα, επίπεδα βλάστησης ανά περιοχή, μικροκλιματικές διαφοροποιήσεις.

Η λύση που προτείνεται στην παρούσα εργασία επιχειρεί να καλύψει το παραπάνω κενό, εστιάζοντας:

- Στην προσαρμογή σε τοπικό επίπεδο Δήμου
- Στη συνδυαστική χαρτογράφηση περιβαλλοντικών και διοικητικών δεδομένων
- Στην εύχρηστη διαδραστική διεπαφή για τον απλό χρήστη, με τεχνολογίες όπως HTML/CSS, JS, PHP και SQL.
- Στη δυνατότητα επέκτασης για αξιοποίηση από άλλους Δήμους στο μέλλον.

Τέλος η λύση μας προσπαθεί να συνδυάσει καλές πρακτικές σε ένα σύγχρονο περιβάλλον ιστότοπου.

## Υπάρχουσες Τεχνολογίες

Οι βασικές τεχνολογίες που σχετίζονται με την ανάπτυξη τέτοιων εφαρμογών περιλαμβάνουν:

- Γλώσσες Web Development: HTML5, CSS3, JavaScript (Vanilla ή frameworks όπως React, Vue).
- Χαρτογραφικές βιβλιοθήκες: Leaflet.js, Mapbox GL JS, Google Maps API.
- Βάσεις δεδομένων: MySQL, PostgreSQL + PostGIS, MongoDB.
- Πίσω-μέρος (Back-end): PHP, Node.js, Python (Flask/Django), με RESTful APIs.
- Ανάκτηση δεδομένων: APIs (π.χ. OpenWeather, Copernicus), web scraping, IoT Gateways.

## Βάση Δεδομένων

Η βάση δεδομένων αποτελεί το κεντρικό αποθετήριο δεδομένων της εφαρμογής. Υπάρχουν δύο βασικοί τύποι (MongoDB) (Jain):

- Σχεσιακές (Relational Databases): Όπως MySQL, PostgreSQL, βασίζονται σε πίνακες και χρησιμοποιούν SQL για την ανάκτηση δεδομένων. Κατάλληλες για δομημένα δεδομένα με σαφείς σχέσεις.
- Μη Σχεσιακές (NoSQL): Όπως MongoDB, Firebase, ιδανικές για δεδομένα που μεταβάλλονται συχνά ή έχουν ημιδομημένο χαρακτήρα.

Στην παρούσα εργασία, αξιοποιείται σχεσιακή βάση δεδομένων (π.χ. MySQL), με δομή που επιτρέπει την αποθήκευση (Heaney) (Voigt #) (Caterino) (Singh):

- Γεωγραφικών οντοτήτων (π.χ. Δήμοι, θέσεις δέντρων),
- Περιβαλλοντικών μετρήσεων (ρύπανση, υγρασία, θερμοκρασία),
- Δημογραφικών ή διοικητικών δεδομένων.

## Υπόβαθρο για τον αναγνώστη

Για την πλήρη κατανόηση των επιμέρους τεχνολογιών και εργαλείων που χρησιμοποιούνται στην παρούσα εργασία, είναι απαραίτητο ο αναγνώστης να έχει εξοικείωση με ορισμένες βασικές έννοιες, οι οποίες αποτελούν θεμέλιο για την ανάπτυξη εφαρμογών βασισμένων σε περιβαλλοντικά και γεωχωρικά δεδομένα. Παρακάτω παρουσιάζονται οι πιο κρίσιμες από αυτές, συνοδευόμενες από παραδείγματα και επεξηγήσεις.

### Γεωχωρικές Συντεταγμένες (Latitude / Longitude)

Οι γεωχωρικές συντεταγμένες είναι το βασικότερο σύστημα προσδιορισμού θέσης στην επιφάνεια της Γης. Το σύστημα αυτό χρησιμοποιεί δύο παραμέτρους (LatLong) (GPS-Coordinates) (GISGeography) (WikiPedia):

- **Γεωγραφικό πλάτος (Latitude):** Μετράται σε μοίρες βόρεια ή νότια του Ισημερινού (0°), φτάνοντας έως και 90° στα γεωγραφικά πλάτη των Πόλων.
- **Γεωγραφικό μήκος (Longitude):** Μετράται σε μοίρες ανατολικά ή δυτικά του πρώτου Μεσημβρινού (0°), ο οποίος διέρχεται από το Γκρίνουιτς, και φτάνει έως και 180°.

Οι συντεταγμένες εκφράζονται συχνά ως (lat, lon) σε δεκαδική μορφή. Για παράδειγμα, η θέση της Αθήνας μπορεί να εκφραστεί ως **(37.9838, 23.7275)**.

### Λειτουργία API και RESTful Αρχιτεκτονική

Ένα **API (Application Programming Interface)** είναι ένα σύνολο από κανόνες και διεπαφές που επιτρέπουν σε διαφορετικά λογισμικά να "επικοινωνούν" μεταξύ τους. Σε περιβάλλοντα web, χρησιμοποιούνται ευρέως **RESTful APIs**, τα οποία βασίζονται στην αρχιτεκτονική REST (Representational State Transfer) (Jain) (Redhat).

Σε μια RESTful αρχιτεκτονική:

- Τα δεδομένα εκτίθενται μέσω URLs.
- Οι βασικές ενέργειες (ανάκτηση, εισαγωγή, τροποποίηση, διαγραφή) αντιστοιχούν στις HTTP μεθόδους GET, POST, PUT και DELETE.
- Η ανταλλαγή δεδομένων γίνεται συνήθως σε μορφή JSON ή XML.

Παράδειγμα: Ένα API που επιστρέφει την ποιότητα αέρα σε πραγματικό χρόνο για μια τοποθεσία μπορεί να καλείται ως εξής:

*GET https://api.airquality.com/data?lat=37.98&lon=23.73*

## Αισθητήρας Περιβάλλοντος (π.χ. PM2.5 Sensor)

Ένας **περιβαλλοντικός αισθητήρας** είναι μια συσκευή που συλλέγει δεδομένα από το φυσικό περιβάλλον. Συχνά χρησιμοποιούνται αισθητήρες (CurieJet) (Infineon):

- **PM2.5:** Μετρούν τα αιωρούμενα σωματίδια με διάμετρο μικρότερη των 2.5 μικρομέτρων στον αέρα, τα οποία σχετίζονται με προβλήματα αναπνοής και υγείας.
- **Θερμοκρασίας / Υγρασίας:** Καταγράφουν βασικές μεταβλητές μικροκλίματος.
- **CO<sub>2</sub> / NO<sub>2</sub>:** Για ανάλυση ατμοσφαιρικής ρύπανσης.

Οι αισθητήρες συνήθως συνδέονται με IoT πλατφόρμες ή APIs, αποστέλλοντας δεδομένα σε τακτά χρονικά διαστήματα.

## Προβαλλόμενα και Γεωγραφικά Συστήματα Αναφοράς

Ένα **Γεωγραφικό Σύστημα Αναφοράς (π.χ. WGS84)** βασίζεται σε σφαιρική αναπαράσταση της Γης και χρησιμοποιείται κυρίως για παγκόσμια συστήματα GPS (Google) (GISGeography) (WikiPedia). Εκφράζει συντεταγμένες με γεωγραφικό πλάτος και μήκος.

Αντίθετα, ένα **Προβαλλόμενο Σύστημα Αναφοράς** μετατρέπει την καμπυλωτή επιφάνεια της Γης σε επίπεδη (χαρτογραφική) επιφάνεια, χρησιμοποιώντας γεωμετρικούς μετασχηματισμούς. Είναι πιο κατάλληλο για τοπικές ή περιφερειακές αναπαραστάσεις, όπως ο **ΕΓΣΑ87** που χρησιμοποιείται στην Ελλάδα (GISGeography) (WikiPedia).

*Παράδειγμα: Μια εφαρμογή GIS μπορεί να απεικονίζει σωστά δεδομένα βλάστησης μόνο αν τα δεδομένα προβάλλονται στο κατάλληλο τοπικό σύστημα και όχι απλώς στο WGS84.*

## Δομή και Οργάνωση Βάσεων Δεδομένων (Συσχετίσεις)

Οι βάσεις δεδομένων οργανώνουν την πληροφορία σε **πίνακες**, όπου κάθε γραμμή αντιστοιχεί σε μια εγγραφή και κάθε στήλη σε ένα πεδίο. Η σωστή σχεδίαση περιλαμβάνει τη χρήση **σχέσεων** μεταξύ των πινάκων, όπως (Microsoft) (Virgo) (Jain):

- **One-to-One:** Κάθε εγγραφή ενός πίνακα συνδέεται με ακριβώς μία εγγραφή άλλου.

- **One-to-Many:** Μια εγγραφή του πρώτου πίνακα αντιστοιχεί σε πολλές εγγραφές του δεύτερου.
- **Many-to-Many:** Πολλαπλές εγγραφές συνδέονται αμοιβαία μέσω ενός τρίτου ενδιάμεσου πίνακα.

Παράδειγμα: Ένας πίνακας με «Περιοχές» μπορεί να έχει μία εγγραφή ανά δημοτική ενότητα, και ένας άλλος πίνακας με «Μετρήσεις» να περιλαμβάνει τις μετρήσεις PM2.5 για κάθε περιοχή. Η σχέση μεταξύ τους είναι one-to-many.

## Η δική μας λύση

Η εφαρμογή που αναπτύσσεται παρέχει ένα ενοποιημένο περιβάλλον προβολής περιβαλλοντικών και γεωγραφικών δεδομένων ανά Δήμο. Βασίζεται στα εξής καινοτόμα χαρακτηριστικά:

- Διαδραστικός χάρτης με δυνατότητα επιλογής Δήμου και εμφάνισης σχετικών στοιχείων.
- Δυναμική βάση δεδομένων με εγγραφές για ρύπους, βλάστηση, περιοχές.
- Σύστημα API που επιτρέπει την εξαγωγή των δεδομένων και σε τρίτες εφαρμογές.
- Δυνατότητα μελλοντικής επέκτασης με φωτογραφικό υλικό, αισθητήρες ή crowdsourcing.

## Γεωγραφική Περιγραφή & Δεδομένα

Η παρούσα εργασία επικεντρώνεται στον Νομό Θεσσαλονίκης, έναν από τους πιο πυκνοκατοικημένους και αστικοποιημένους νομούς της Ελλάδας, με πρωτεύουσα την πόλη της Θεσσαλονίκης. Ο νομός βρίσκεται στο βόρειο τμήμα της χώρας και αποτελεί αναπόσπαστο τμήμα της Κεντρικής Μακεδονίας. Η γεωγραφική του θέση, σε συνδυασμό με τον ισχυρό πολεοδομικό του χαρακτήρα, τον καθιστούν μία ιδιαίτερα ενδιαφέρουσα περιοχή για περιβαλλοντική μελέτη και ανάλυση.

### Γεωγραφική κατανομή και ιδιαιτερότητες

Η Θεσσαλονίκη συνδυάζει χαρακτηριστικά αστικού και περιαστικού περιβάλλοντος. Περιλαμβάνει πυκνοκατοικημένες περιοχές όπως το κέντρο της πόλης (Άνω Πόλη, Βαρδάρη, Καμάρα), βιομηχανικές ζώνες στα δυτικά (Σίνδος, Καλοχώρι), αλλά και περιαστικές περιοχές με περισσότερο πράσινο, όπως η Πυλαία, το Πανόραμα και ο Χορτιάτης (Dasarxeio) (TheOpinion) (Δημος Θεσσαλονίκης) (Σπανός).

Ιδιαίτερο ενδιαφέρον παρουσιάζει το γεγονός ότι ο νομός περιβάλλεται από σημαντικούς φυσικούς σχηματισμούς, όπως (Dasarxeio) (TheOpinion) :

- Ο Θερμαϊκός κόλπος στα δυτικά, που συμβάλλει στη ρύθμιση του μικροκλίματος.
- Ο Χορτιάτης στα ανατολικά, ο οποίος λειτουργεί ως φυσικό φίλτρο και ρυθμιστής της ποιότητας του αέρα.

- Ο Λαγκαδάς και η ευρύτερη πεδιάδα του Αγίου Βασιλείου, που προσφέρουν περισσότερες δυνατότητες για αγροτική και φυσική βλάστηση.

Ωστόσο, η ραγδαία αστικοποίηση έχει περιορίσει σημαντικά τους φυσικούς χώρους εντός του πολεοδομικού συγκροτήματος, οδηγώντας σε αύξηση των θερμοκρασιών, χαμηλότερη ποιότητα αέρα και περιορισμένο χώρο για αστικό πράσινο (Δημος Θεσσαλονίκης) (Σπανός).

## Επισκόπηση των δεδομένων βλάστησης

Η βλάστηση στον Δήμο Θεσσαλονίκης αποτελείται κυρίως από (Wilderness) (Voria):

- Καλλωπιστικά δέντρα σε πάρκα και δρόμους (πλατάνια, νεραντζιές, μουριές, ιπιές).
- Φυσική βλάστηση σε περιοχές όπως το Σείχ Σου (πευκοδάση), ο Χορτιάτης και τα γύρω βουνά.
- Αστική βλάστηση, η οποία έχει οργανωθεί κατά καιρούς μέσω δεντροφυτεύσεων από τοπικούς φορείς και περιβαλλοντικές οργανώσεις.

Τα δεδομένα βλάστησης που αξιοποιούνται στην εφαρμογή αφορούν τόσο γεωχωρικές θέσεις δέντρων όσο και πληροφορίες για το είδος, την ποσότητα και την τοποθεσία των δέντρων αυτών. Η δυνατότητα αποτύπωσης αυτών των στοιχείων στον χάρτη παρέχει ένα ισχυρό εργαλείο για την αξιολόγηση της πυκνότητας βλάστησης ανά περιοχή, την αναγνώριση αστικών "θερμών σημείων" χωρίς πράσινο, και τον σχεδιασμό νέων παρεμβάσεων.

## Επισκόπηση των δεδομένων ατμοσφαιρικής ρύπανσης

Τα δεδομένα ρύπανσης που αξιοποιούνται στην εφαρμογή προέρχονται από σταθμούς μέτρησης ατμοσφαιρικής ρύπανσης, που λειτουργούν σε διάφορα σημεία της πόλης και παρέχουν (Δελτίο Τύπου Θεσσαλονίκης) (Voria) (Wilderness) (GIS Θεσσαλονίκη):

- Τιμές για αιωρούμενα σωματίδια (PM10, PM2.5),
- Διοξείδιο του αζώτου (NO<sub>2</sub>),
- Όζον (O<sub>3</sub>) και
- Άλλα στοιχεία της ατμόσφαιρας όπως υγρασία.

Οι περιοχές με τη μεγαλύτερη συγκέντρωση ρύπων βρίσκονται κατά κύριο λόγο στο κέντρο της πόλης και στις δυτικές συνοικίες, λόγω της έντονης κυκλοφορίας και της γεινιάσης με βιομηχανικές μονάδες (Δελτίο Τύπου Θεσσαλονίκης) (Voria) (Wilderness) (GIS Θεσσαλονίκη) (Ανδρώνης and Καραθανάση).

Η συνδυαστική απεικόνιση αυτών των δεδομένων με τη βλάστηση μπορεί να δώσει σημαντικές ενδείξεις για (Ανδρώνης and Καραθανάση) :

- Τη συσχέτιση μεταξύ πρασίνου και ποιότητας αέρα,
- Την ανάγκη ενίσχυσης της βλάστησης σε επιβαρυσμένες περιοχές,
- Τον σχεδιασμό "πράσινων διαδρομών" μέσα στον αστικό ιστό για την ευκολότερη και πιο ασφαλή μετακίνηση πεζών και ποδηλατών.

## Τεχνολογική Υλοποίηση

Η εφαρμογή υλοποιείται με χρήση σύγχρονων τεχνολογιών Front-end και Back-end, όπως:

- HTML, CSS και JavaScript για τον σχεδιασμό της διεπαφής χρήστη (User Interface),
- PHP και SQL για την υλοποίηση της διασύνδεσης με τη βάση δεδομένων και την επιχειρησιακή λογική (Back-end),
- API σύνδεση με εξωτερικές πηγές δεδομένων για την αυτόματη άντληση περιβαλλοντικών μετρήσεων,
- Χρήση διαδραστικών χαρτών (π.χ. Leaflet ή Mapbox) για την οπτικοποίηση της πληροφορίας σε επίπεδο γειτονιάς.

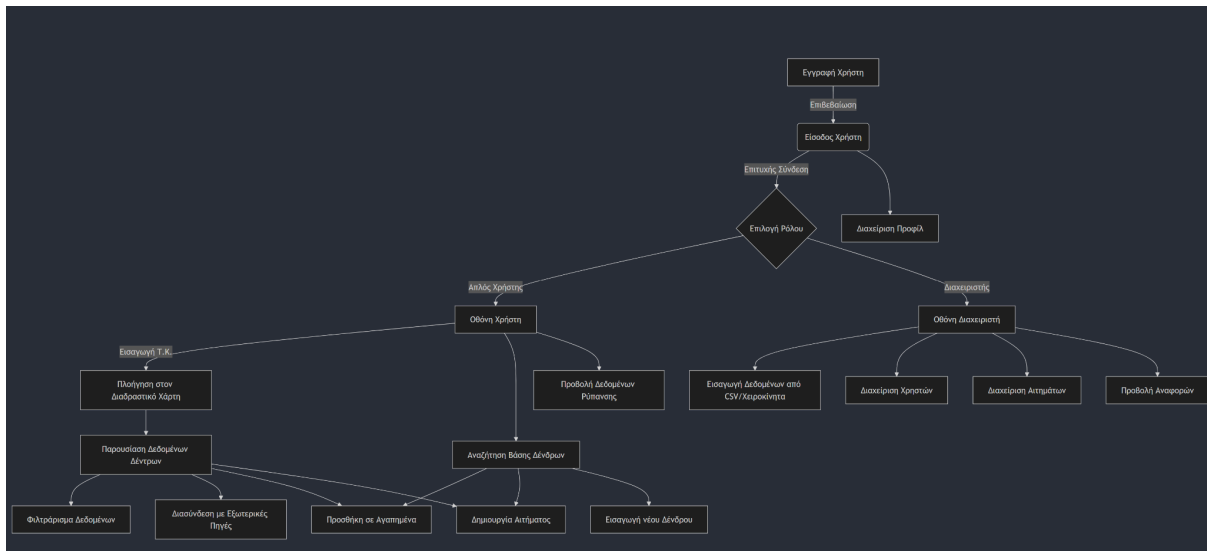
Η αρχιτεκτονική της εφαρμογής έχει σχεδιαστεί με τέτοιο τρόπο ώστε να υποστηρίζει επεκτασιμότητα για περισσότερους δήμους, να επιτρέπει εισαγωγή νέων δεδομένων από τους χρήστες, και να ενσωματώνει μηχανισμούς ανάλυσης και απεικόνισης που ενισχύουν τη λήψη αποφάσεων.

## Λειτουργικότητα

Οι υποστηριζόμενες λειτουργίες της εφαρμογής αναφέρονται στις:

1. **Εγγραφή Χρήστη:** Ο χρήστης έχει την δυνατότητα να εγγραφεί στην εφαρμογή με χρήση ενός username και password. Ο χρήστης μπορεί να κατέχει είτε τον ρόλο (1) απλού χρήστη είτε (2) διαχειριστή.
2. **Είσοδος Χρήστη:** Ο χρήστης συνδέεται στην εφαρμογή χρησιμοποιώντας όνομα χρήστη και κωδικό πρόσβασης. Η εφαρμογή υποστηρίζει βασική αυθεντικοποίηση για την προστασία των δεδομένων.
3. **Αναζήτηση Δήμου:** Ο χρήστης μπορεί να αναζητήσει την περιοχή που επιθυμεί στον χάρτη ή να χρησιμοποιήσει την πραγματική του θέση για να εντοπίσει τον Δήμο στον οποίο ανήκει. Η λειτουργία αυτή απευθύνεται στους κατοίκους του νομού Θεσσαλονίκης, καθώς προς το παρόν δεν υπάρχουν δεδομένα για άλλες περιοχές.
4. **Πλοήγηση και Παρουσίαση Δεδομένων:** Στην αρχική οθόνη εμφανίζεται ένας διαδραστικός χάρτης που επιτρέπει στον χρήστη να εντοπίσει τον Δήμο του και να μεταφερθεί στη δεύτερη ενότητα της εφαρμογής. Στη δεύτερη ενότητα, εμφανίζονται πληροφορίες για την πανίδα της περιοχής, προερχόμενες από τη βάση δεδομένων για τον επιλεγμένο Δήμο, όπως:
  - a. **Είδος δέντρου** (π.χ., πεύκο, κυπαρίσσι)
  - b. **Ονομασία δέντρου** (επιστημονική και κοινή ονομασία)
  - c. **Ποσότητα δέντρων** που έχουν φυτευτεί στον Δήμο.
5. **Ατμοσφαιρική Ρύπανση:** Η εφαρμογή στοχεύει επίσης να παρουσιάζει δεδομένα για την ατμοσφαιρική ρύπανση σε κάθε Δήμο, ένα κρίσιμο ζήτημα για την ποιότητα ζωής και τη δημόσια υγεία. Η παρουσίαση αυτών των δεδομένων θα γίνεται από την άντληση της πληροφορίας από τα σχετικά πεδία της βάσης δεδομένων.

Παρακάτω εμφανίζεται ένα διάγραμμα λειτουργιών για τους χρήστες και τους διαχειριστές.

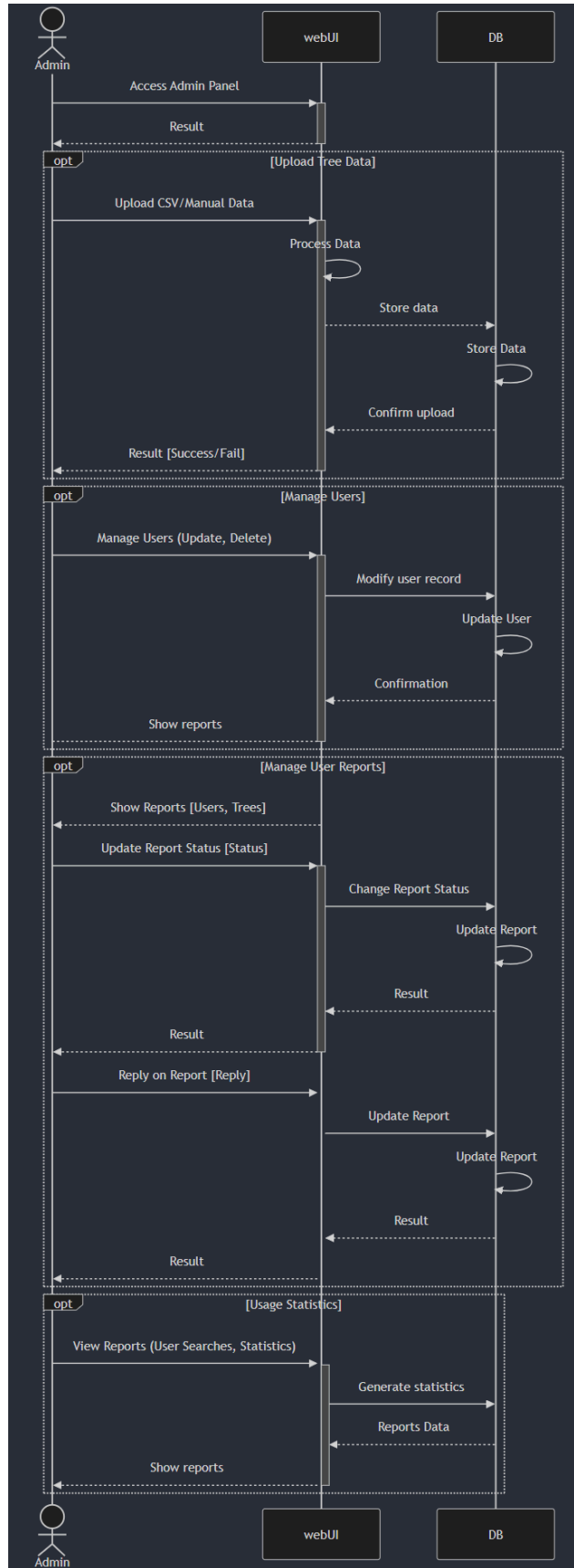


## Πρόσθετες Λειτουργικότητες

Η εφαρμογή στο σύνολό της θα υποστηρίξει τις εξής πρόσθετες λειτουργικότητες, διαχωρίζοντας τις οθόνες περιήγησης για τους δύο ρόλους χρηστών:

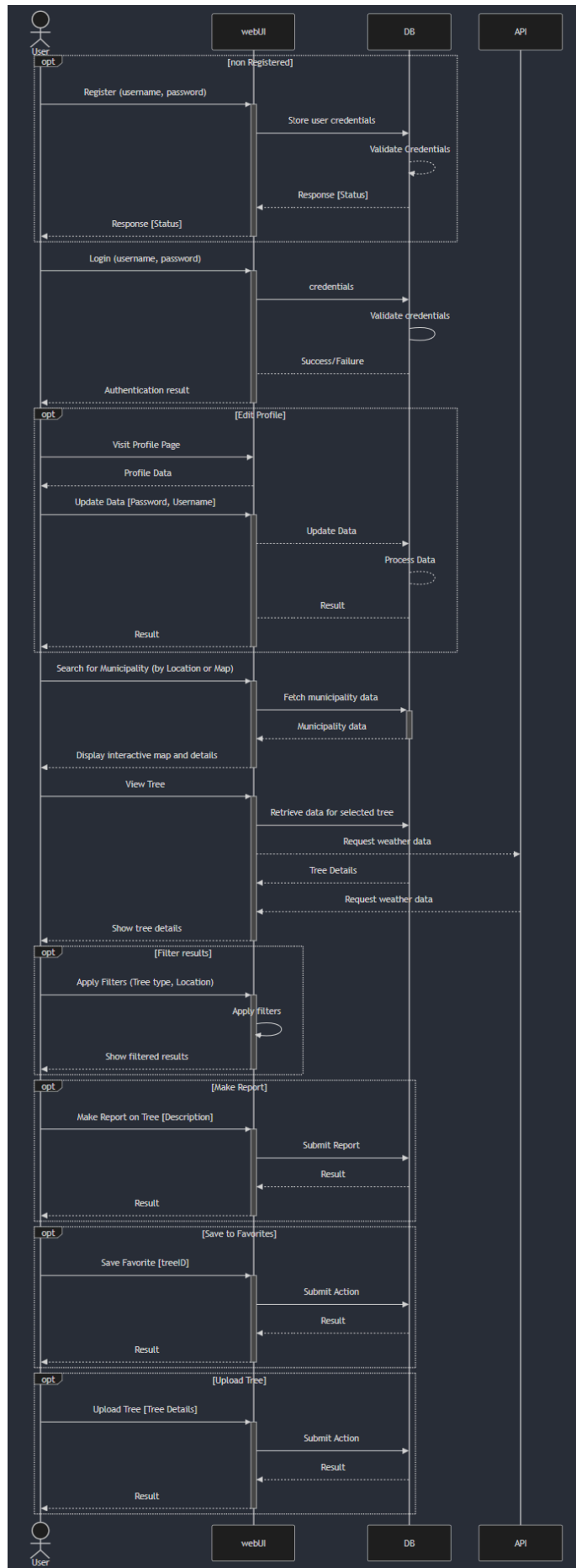
- **Οθόνη Διαχειριστή:** Για τους διαχειριστές της εφαρμογής, θα υπάρχει ξεχωριστή οθόνη που θα περιλαμβάνει λειτουργίες όπως:
  - Εισαγωγή δεδομένων μέσω αρχείων Excel/csv, που επιτρέπει την μαζική ενημέρωση της βάσης δεδομένων με πληροφορίες για δήμους, βλάστηση και ρύπανση.
  - Διαχείριση χρηστών: Δυνατότητα δημιουργίας, τροποποίησης ή διαγραφής λογαριασμών χρηστών και διαχειριστών.
  - Προβολή αναφορών: Παρουσίαση στατιστικών αναφορών σχετικά με τις αναζητήσεις χρηστών, που θα προσφέρει καλύτερη κατανόηση της χρήσης της εφαρμογής.

Το σύνολο των λειτουργιών του διαχειριστή παρουσιάζεται και στο ακόλουθο διάγραμμα ακολουθίας. Όπως φαίνεται σε αυτό, ο διαχειριστής μπορεί να εκτελέσει μία από τις προαναφερθέντες λειτουργίες, αφού πρώτα συνδεθεί στο λογαριασμό του. Ανάλογη οθόνη του διαχειριστή θα του επιτρέψει να δει τις ενεργές καταστάσεις χρηστών, αιτημάτων και να ανανεώσει υπάρχουσες πληροφορίες δεδομένων.



- **Οθόνη Απλού Χρήστη:** Οι απλοί χρήστες θα έχουν πρόσβαση στην κύρια οθόνη για την αναζήτηση Δήμων, όπου θα μπορούν να εισάγουν τον Τ.Κ. τους ή να χρησιμοποιούν τον χάρτη για την πλοήγηση και την πρόσβαση στις πληροφορίες.
  - **Φιλτράρισμα Δεδομένων:** Δυνατότητα αναζήτησης στοιχείων με συγκεκριμένα χαρακτηριστικά, π.χ., οι χρήστες να μπορούν να βλέπουν μόνο δέντρα συγκεκριμένου είδους ή να φιλτράρουν τα δεδομένα ρύπανσης με βάση την ποιότητα αέρα.
  - **Διασύνδεση με Εξωτερικές Πηγές Δεδομένων:** Ενσωμάτωση δεδομένων ατμοσφαιρικής ρύπανσης σε πραγματικό χρόνο από εξωτερικές πηγές, όπως το OpenWeather ή άλλες υπηρεσίες περιβαλλοντικών δεδομένων, εφόσον η επιλεγμένη περιοχή υποστηρίζεται.

Η ακόλουθη εικόνα ενός διαγράμματος ακολουθίας συνοψίζει τις παραπάνω λειτουργίες ενός χρήστη, καθώς και τις προαναφερθέντες. Γενικότερα οι χρήστες έχουν την δυνατότητα να δουν όλη την υπάρχουσα πληροφορία, να διαχειριστούν το προφίλ τους και να αποθηκεύσουν δεδομένα ως αγαπημένα ή να ανεβάσουν τα δικά τους καθώς και να δημιουργήσουν αιτήματα προς τους διαχειριστές.



# Τεχνολογίες

Οι τεχνολογίες που θα χρησιμοποιηθούν για την υλοποίηση του προαναφερθέν συστήματος αναφέρονται στις:

- **Front-end:** HTML, CSS, JavaScript για το περιβάλλον του χρήστη και την αλληλεπίδραση με τον χάρτη. Χρήση επίσης της PHP για την λήψη δεδομένων από την βάση.
- **Back-end:** Χρήση βάσης MySQL/Postgres για την αποθήκευση των δεδομένων της βάσης και PHP για την εξυπηρέτηση αιτημάτων και την επεξεργασία δεδομένων.
- **API Χαρτών:** Για την ενσωμάτωση του χάρτη πλοήγησης που επιτρέπει την αναζήτηση Δήμων.
- **Python Script:** Δημιουργία python script για την αρχική εισαγωγή των δεδομένων στην βάση. Τα συγκεκριμένα τμήματα κώδικα θα λαμβάνουν ως είσοδο κάποιο Excel/csv αρχείο και θα εισάγουν τα δεδομένα στην βάση.

Συγκεντρωτικά το σύνολο των τεχνολογιών εμφανίζεται στον ακόλουθο πίνακα.

Πίνακας 1: Τεχνολογίες που Χρησιμοποιήθηκαν

Τεχνολογία	Ρόλος στην Εφαρμογή
HTML5/CSS3	Δομή και Εμφάνιση διεπαφής
JavaScript	Διαδραστικότητα και δυναμικές λειτουργίες
PHP	Διαχείριση λογικής back-end
MySQL	Αποθήκευση και ανάκτηση δεδομένων
OpenStreet API	Απόκτηση δεδομένων χαρτών
Weather API	Απόκτηση δεδομένων περιβάλλοντος

# Απαιτήσεις & Σχεδίαση Συστήματος

Η ανάπτυξη του συστήματος βασίστηκε στην ανάλυση των αναγκών του τελικού χρήστη και των τεχνικών περιορισμών, με στόχο την παροχή μίας εύχρηστης και αξιόπιστης πλατφόρμας περιβαλλοντικής πληροφόρησης. Οι απαιτήσεις χωρίζονται σε Λειτουργικές και Μη Λειτουργικές, όπως περιγράφεται στη συνέχεια.

## Απαιτήσεις Συστήματος

### Λειτουργικές Απαιτήσεις

Οι λειτουργικές απαιτήσεις περιγράφουν τις βασικές υπηρεσίες και δυνατότητες που πρέπει να παρέχει το σύστημα ώστε να ικανοποιεί τις ανάγκες του χρήστη. Αυτές αναλύονται στην συνέχεια, με την παρουσίαση της ίδιας της πλατφόρμας και των δυνατοτήτων. Οι πιο σημαντικές περιλαμβάνουν:

- **Εγγραφή Χρήστη στο Σύστημα:** Ο χρήστης μπορεί να δημιουργήσει λογαριασμό εισάγοντας βασικά στοιχεία, ώστε να αποκτήσει πρόσβαση σε λειτουργίες που απαιτούν ταυτοποίηση, όπως η υποβολή αιτημάτων.
- **Αναζήτηση Δέντρου στον Δήμο μέσω Διαδραστικής Διαδικασίας:** Ο χρήστης δύναται να αναζητήσει ένα δέντρο με βάση την τοποθεσία του μέσω διαδραστικού χάρτη. Ο χάρτης μπορεί να παρέχει την απαραίτητη πληροφορία για την βλάστηση της κάθε περιοχής.
- **Υποβολή Αιτημάτων:** Ο χρήστης έχει τη δυνατότητα να συντάξει και να αποσταλεί αίτημα σχετικό με το δέντρο (π.χ. αναφορά προβλήματος, αίτηση φύτευσης) προσφέροντας έτσι τη συνεχή βελτίωση της πλατφόρμας μέσω της συλλογικής μέριμνας.
- **Υποβολή Δέντρου:** Ο χρήστης έχει την δυνατότητα να ανεβάσει ένα νέο δέντρο στην πλατφόρμα με την σχετική του πληροφορία, βοηθώντας έτσι στην ανάπτυξη και υποστήριξη της συλλογικότητας στην συλλογή πληροφοριών.

### Μη Λειτουργικές Απαιτήσεις

Οι μη λειτουργικές απαιτήσεις περιγράφουν χαρακτηριστικά που σχετίζονται με την ποιότητα λειτουργίας του συστήματος, την ασφάλεια και την αποδοτικότητα. Για το εν λόγω σύστημα τις συνοψίζουμε στις ακόλουθες:

- **Ασφάλεια Δεδομένων:** Προστασία των προσωπικών δεδομένων των χρηστών, κυρίως του κωδικού πρόσβασης, μέσω της κρυπτογραφημένης αποθήκευσής του στο σύστημα.
- **Απόδοση και Κλιμάκωση:** Το σύστημα πρέπει να διαχειρίζεται τα αιτήματα γρήγορα, ακόμα και σε περιόδους αυξημένης επισκεψιμότητας, και να μπορεί να επεκταθεί μελλοντικά για περισσότερους δήμους ή τύπους δέντρων.
- **Εύχρηστία και Προσβασιμότητα:** Η πλατφόρμα πρέπει να είναι απλή στη χρήση, φιλική προς το χρήστη, και προσβάσιμη από κινητές συσκευές και υπολογιστές.

- Συμβατότητα: Υποστήριξη για σύγχρονους browsers και προσαρμογή σε διαφορετικές αναλύσεις οθόνης.

## Σχεδίαση Συστήματος

Η σχεδίαση του συστήματος πραγματοποιήθηκε με στόχο τη δημιουργία μιας διαδραστικής, ασφαλούς και επεκτάσιμης πλατφόρμας για την απεικόνιση και επεξεργασία περιβαλλοντικών δεδομένων σε επίπεδο Δήμου. Κατά τον σχεδιασμό λήφθηκαν υπόψη η λειτουργικότητα, η ευκολία χρήσης και η δυνατότητα μελλοντικής επέκτασης.

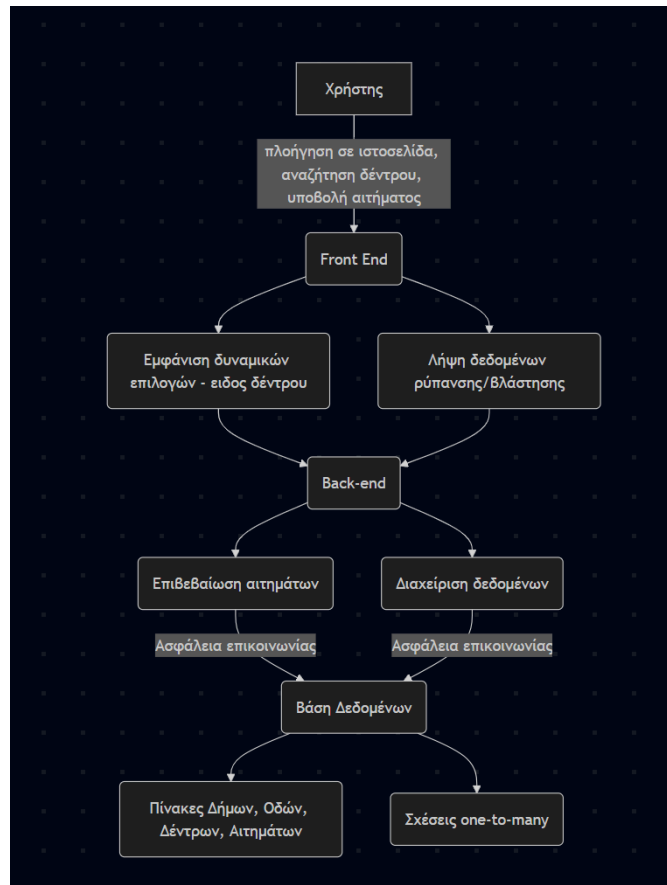
Το σύστημα υλοποιείται βάσει της κλασικής αρχιτεκτονικής διαχωρισμού σε Front-end και Back-end. Ο διαχωρισμός αυτός επιτρέπει την ανεξάρτητη ανάπτυξη και βελτιστοποίηση της διεπαφής χρήστη (user interface - UI) και της επιχειρησιακής λογικής (business logic), προσφέροντας μεγαλύτερη ευελιξία και καλύτερη διαχείριση της πολυπλοκότητας. Συγκεκριμένα:

- Μπροστά μέρος (Front-end): Ασχολείται με την αλληλεπίδραση με τον χρήστη, την παρουσίαση των δεδομένων και τη συλλογή εισροών (inputs) μέσω φιλικού περιβάλλοντος. Το Front-end αναπτύσσεται με τεχνολογίες HTML, CSS και JavaScript.
- Πίσω μέρος (Back-end): Υπεύθυνο για την επεξεργασία αιτημάτων, την αποθήκευση και ανάκτηση δεδομένων από τη βάση δεδομένων, καθώς και για την ασφάλεια των επικοινωνιών. Το Back-end υλοποιείται χρησιμοποιώντας PHP και SQL.

Η επικοινωνία μεταξύ Front-end και Back-end επιτυγχάνεται μέσω RESTful API κλήσεων, οι οποίες διασφαλίζουν την ανεξαρτησία των δύο πλευρών και την εύκολη μελλοντική συντήρηση ή αναβάθμιση του συστήματος. Συγκεκριμένα θα χρησιμοποιηθεί Ajax για την υλοποίηση των κλήσεων και την επικοινωνία της HTML-Javascript με την Php-SQL.

Επιπλέον, δίνεται έμφαση στην αξιοποίηση γεωχωρικών δεδομένων, μέσω των οποίων προσφέρονται λειτουργίες όπως ο εντοπισμός δέντρων βάσει τοποθεσίας χρήστη, η δυναμική εμφάνιση εικόνων και χαρακτηριστικών ανά είδος δέντρου και η αναζήτηση πληροφοριών για τη ρύπανση και τη βλάστηση της περιοχής.

Στην εικόνα εμφανίζεται το διάγραμμα ροής (flow-diagram) χρήστη της ιστοσελίδας από ένα χρήστη με τα βασικά χαρακτηριστικά που περιγράφηκαν. Ο χρήστης αλληλεπιδρά με το Front-end και το σύστημα διαχειρίζεται τα δεδομένα και την ασφαλή επικοινωνία με το back-end για την λήψη και παρουσίαση των πληροφοριών.



Η βάση δεδομένων έχει σχεδιαστεί σύμφωνα με τις αρχές της σχεσιακής μοντελοποίησης (Relational Modeling), περιλαμβάνοντας πίνακες που αναπαριστούν τους Δήμους, τις οδούς, τα δέντρα και τα αιτήματα των χρηστών, με κατάλληλες σχέσεις (one-to-many, many-to-one).

Η συνολική σχεδίαση επικεντρώνεται σε (1) απλότητα και ταχύτητα χρήσης για τον τελικό χρήστη, (2) ασφαλή διαχείριση και προστασία των προσωπικών δεδομένων και (3) επεκτασιμότητα για μελλοντική ενσωμάτωση περισσότερων περιοχών και παραμέτρων περιβάλλοντος.

## Αρχιτεκτονική Συστήματος

Το σύστημα θα αποτελείται από δύο βασικά στοιχεία, το (1) Front-end το οποίο είναι απαραίτητο για την παρουσίαση των δεδομένων στους χρήστες και την διάδρασή τους με αυτό και (2) το Back-end το οποίο είναι απαραίτητο για την αποθήκευση των δεδομένων και διαχείριση των αιτημάτων. Επιπλέον, θεωρείτε ως επιπλέον βασικό στοιχείο τα (3) Εξωτερικά Εργαλεία, τα οποία δεν αποτελούν άμεσο στοιχείο του συστήματος, αλλά είναι απαραίτητα για την διαχείριση των δεδομένων της βάσης.

### Front-end

Το front-end είναι υπεύθυνο για την διαχείριση του συστήματος και την πλοήγηση του μέσα σε αυτό. Παρέχει στον χρήστη μια φιλική και λειτουργική διεπαφή και είναι αυτό που βλέπτε

και με το οποίο αλληλεπιδρά ο χρήστης. Παρουσιάζεται η πληροφορία στον χρήστη και την διαχειρίζεται αρχικά με τα δικαιώματα που έχει και προϋπάρχουν από την πλευρά του συστήματος και στη συνέχεια ο ίδιος επιλέγει πως θα τα αξιοποιήσει. Για την παρουσίαση των πληροφοριών, σημάνσεων, προβολή χαρακτηριστικών και γενικά για το περιεχόμενο μιας ιστοσελίδας, υπάρχουν κάποια στοιχεία τα οποία πρέπει να τηρεί τις διαδικασίες για ένα web app. Η καθιερωμένη γλώσσα περιγραφής ιδιοτήτων για σήμανση των ιστοσελίδων είναι η HTML(HyperText Markup Language).

## Back-end

Το πιο σημαντικό στοιχείο και η βασική προϋπόθεση κάθε διαδικτυακής εφαρμογής είναι το Back-end. Είναι απαραίτητο στοιχείο το οποίο στηρίζει τη λογική μιας ιστοσελίδας και είναι υπεύθυνο για την διασύνδεση και την επικοινωνία ανάμεσα στον διακομιστή και στην βάση δεδομένων. Ο ρόλος που έχει σε μία ιστοσελίδα είναι επεξεργάζεται και να διαχειρίζεται δεδομένα δημιουργώντας άμεση επικοινωνία με την βάση δεδομένων που υπάρχει. Επιπρόσθετα, επιτυγχάνει την ασφάλεια και την αξιοπιστία της εφαρμογής.

Ένα back-end σύστημα μπορεί να αποτελείται στο σύνολό του από πολλαπλά εργαλεία, με τα πιο βασικά να είναι η Βάση Δεδομένων για την αποθήκευση της πληροφορίας χρηστών ή αυτής που παρουσιάζεται στο σύστημα, άλλα εργαλεία διασύνδεσης του Front-end με την βάση ή με άλλα εξωτερικά συστήματα ή άλλες λειτουργίες διαχείρισης των δεδομένων πριν αυτά αποθηκευτούν ή παρουσιαστούν στον χρήστη.

## Εξωτερικά Εργαλεία

Τα εξωτερικά εργαλεία χρησιμοποιούνται για την λήψη πληροφορίας από εξωτερικές πηγές ή αποστέλλουν πληροφορία σε αυτές. Αν και αποτελούν μέρος του συστήματος, συνήθως δεν έχουν άμεση σχέση με αυτό και εκτελούνται όταν απαιτείται να γίνει κάποια επιπλέον λειτουργία στο σύστημα, όπως η ανανέωση δεδομένων, η λήψη δεδομένων, η διαχείρισή τους, ή άλλες λειτουργίες ανεξάρτητες της συνολικής λειτουργίας της πλατφόρμας.

## Ρόλοι Χρηστών και Δυνατότητες

Η εφαρμογή υποστηρίζει δύο βασικούς ρόλους χρηστών:

1. **Χρήστης:** Ο απλός χρήστης που χρησιμοποιεί την πλατφόρμα για πρόσβαση στις βασικές λειτουργίες.
2. **Διαχειριστής:** Ένας εξουσιοδοτημένος χρήστης με πρόσβαση σε εργαλεία διαχείρισης και τροποποίησης δεδομένων.

## Χρήστης

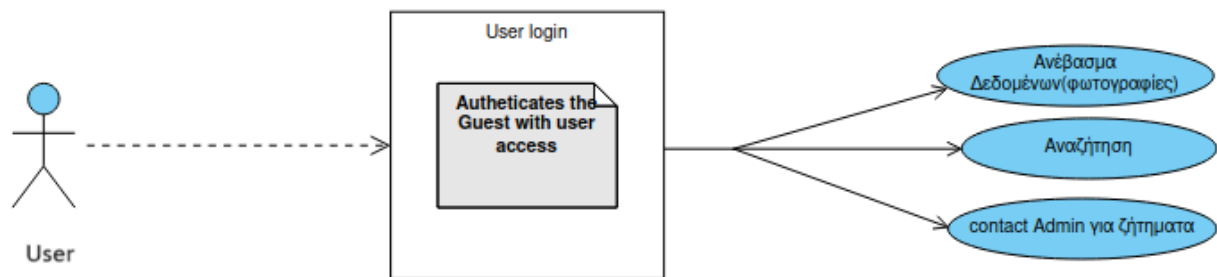
Ο ρόλος του χρήστη αναφέρεται στον απλό χρήστη που χρησιμοποιεί την πλατφόρμα.

Συγκεκριμένα οι δυνατότητες που μπορεί να κάνει είναι:

1. **Εγγραφή στην εφαρμογή:** Δημιουργία λογαριασμού με χρήση ονόματος χρήστη και κωδικού πρόσβασης.
2. **Σύνδεση και πρόσβαση σε δεδομένα:** Είσοδος στον λογαριασμό για πρόσβαση σε προσωποποιημένες λειτουργίες.

3. **Αναζήτηση Δήμου:** Δυνατότητα αναζήτησης του Δήμου με βάση τον Ταχυδρομικό Κώδικα (Τ.Κ.) ή μέσω διαδραστικού χάρτη.
4. **Προβολή δεδομένων:** Παρουσίαση πληροφοριών βλάστησης και ατμοσφαιρικής ρύπανσης για τον Δήμο.
5. **Επικοινωνία με Διαχειριστή:** Δημιουργία αιτημάτων ή ερωτήσεων προς τον διαχειριστή της εφαρμογής.

Στην εικόνα Χ.Χ εμφανίζεται το UML διάγραμμα χρήσης (Use Case) για τις δυνατότητες του απλού χρήστη.



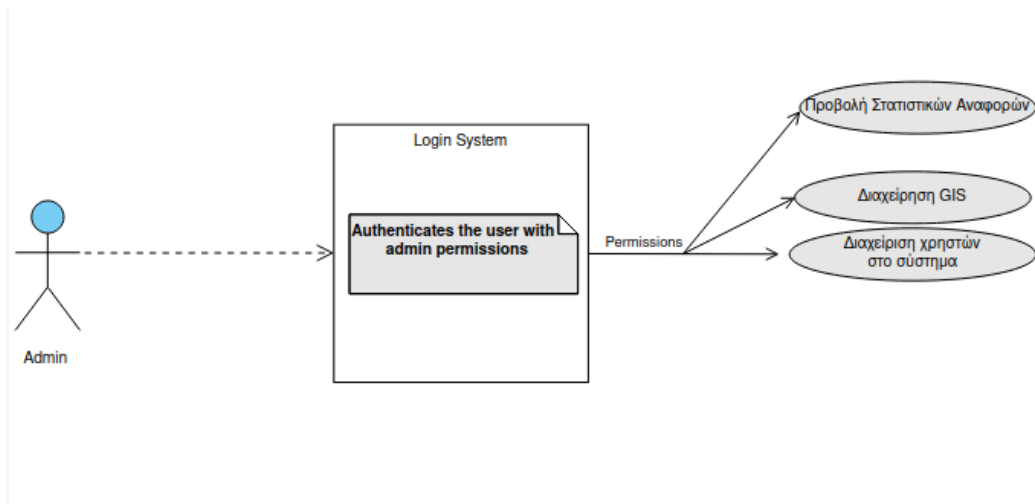
Διάγραμμα Χρήσης: Δυνατότητες του Χρήστη

## Διαχειριστής

Ο ρόλος του διαχειριστή περιλαμβάνει πρόσβαση σε λειτουργίες που επιτρέπουν τη διαχείριση της εφαρμογής και των δεδομένων. Οι βασικές δυνατότητες περιλαμβάνουν:

1. **Διαχείριση χρηστών:** Δημιουργία, τροποποίηση ή διαγραφή λογαριασμών χρηστών και διαχειριστών.
2. **Εισαγωγή δεδομένων:** Μαζική ενημέρωση της βάσης δεδομένων μέσω αρχείων Excel ή CSV, για πληροφορίες που σχετίζονται με Δήμους, βλάστηση και ατμοσφαιρική ρύπανση.
3. **Προβολή αναφορών:** Στατιστικές αναφορές σχετικά με τις αναζητήσεις χρηστών και τη χρήση της πλατφόρμας.
- 4.

Στην εικόνα Χ.Χ εμφανίζεται το UML διάγραμμα χρήσης (Use Case) για τις δυνατότητες του διαχειριστή.



Διάγραμμα Χρήσης: Δυνατότητες Διαχειριστή

## Τεχνολογικές Επιλογές

Για την δημιουργία μιας ιστοσελίδας περιεχομένου σήμερα υπάρχουν πολλαπλές επιλογές, με την δυνατότητα παραγωγής των συστημάτων για πολλαπλές συσκευές, όπως Android, iOS, Εφαρμογές Windows, Ιστοσελίδες και άλλες. Κάθε λύση έχει τα δικά της θετικά ή αρνητικά στοιχεία, προσφέροντας διαφορετικές δυνατότητες και ελευθερίες στον προγραμματιστή.

Η επιλογή των τεχνολογιών HTML, JavaScript, PHP και SQL για την ανάπτυξη της εφαρμογής βασίζεται στην ευρεία αποδοχή, την αξιοπιστία και την ευελιξία που προσφέρουν. Η HTML αποτελεί το θεμέλιο για τη δομή και την παρουσίαση του περιεχομένου της ιστοσελίδας, ενώ η JavaScript επιτρέπει τη δημιουργία διαδραστικών λειτουργιών και τη βελτίωση της εμπειρίας του χρήστη μέσω δυναμικών στοιχείων και ασύγχρονης επικοινωνίας με τον διακομιστή. Η χρήση της PHP ως γλώσσα backend διασφαλίζει την αποτελεσματική διαχείριση των αιτημάτων του χρήστη, την επεξεργασία δεδομένων και την αλληλεπίδραση με τη βάση δεδομένων. Τέλος, η SQL προσφέρει ένα σταθερό και αποδοτικό περιβάλλον για την αποθήκευση, αναζήτηση και διαχείριση των δεδομένων της εφαρμογής.

Ο συνδυασμός αυτών των τεχνολογιών επιτρέπει την υλοποίηση μιας διαδικτυακής εφαρμογής που είναι εύκολα επεκτάσιμη, συντηρήσιμη και προσβάσιμη από διαφορετικές συσκευές. Παράλληλα, η ευρεία τεκμηρίωση και η μεγάλη κοινότητα υποστήριξης που διαθέτουν τα συγκεκριμένα εργαλεία, διευκολύνουν την επίλυση προβλημάτων και την ενσωμάτωση νέων λειτουργιών στο μέλλον. Έτσι, διασφαλίζεται ότι η εφαρμογή θα μπορεί να ανταποκριθεί τόσο στις τρέχουσες όσο και στις μελλοντικές ανάγκες των χρηστών και του οργανισμού.

# Υλοποίηση Συστήματος

## Front-end

Η πλέον διαδεδομένη τεχνολογία για την υλοποίηση μιας ιστοσελίδας είναι ο συνδυασμός των τριών βασικών τεχνολογιών HTML, CSS και Javascript. Η HTML είναι ένα σύνολο για την διαμόρφωση της εμφάνισης και του περιεχομένου. Η HTML είναι υπεύθυνη για τον καθορισμό του περιεχομένου μιας ιστοσελίδας δηλαδή περιλαμβάνει κείμενα, εικόνες, βίντεο. Κατέχει σημαντικό ρόλο για τα αντικείμενα μέσα σε μία ιστοσελίδα, καθώς είναι υπεύθυνη για την εμφάνιση και δημιουργία τους.

Στη συνέχεια η CSS (Cascading Style Sheets) είναι μία γλώσσα υπολογιστή η οποία είναι υπεύθυνη για το πως φαίνονται τα περιεχόμενα σε ένα πρόγραμμα περιήγησης. Η CSS καθορίζει το στιλιστικό κομμάτι της HTML και το εμπλουτίζει με πλούσια εμφάνιση, καθορίζει τα χρώματα, τις γραμματοσειρές και τις διατάξεις κάθε σελίδας. Συνήθως, υπεύθυνοι για την λειτουργία αυτή είναι οι σχεδιαστές. Ωστόσο, στις περιπτώσεις μικρότερων ομάδων μπορεί ο ίδιος άνθρωπος να αναλαμβάνει την δημιουργία της ιστοσελίδας και στο στιλιστικό της κομμάτι.

Οι προαναφερθέντες τεχνολογίες δημιουργούν ένα στατικό υλικό. Για τον λόγο αυτό χρησιμοποιείται η Javascript, η οποία προσφέρει την δυνατότητα να δημιουργηθούν ιστοσελίδες που ανταποκρίνονται στις κινήσεις και τα αιτήματα του χρήστη. Μέσω τεχνολογιών, όπως Ajax ή PHP, δίνει την δυνατότητα της επικοινωνίας με την βάση δεδομένων και παρουσίαση των αποτελεσμάτων για τον χρήστη με δυναμικό τρόπο.

## Back-end

Η αποθήκευση των δεδομένων του συστήματος θα υποστηρίζεται μίας SQL Βάσης δεδομένων. Συγκεκριμένα, η βάση θα αποτελείται από δύο πίνακες, έναν για την αποθήκευση των χρηστών και ένα για την αποθήκευση των δεδομένων των δένδρων.

Παρακάτω εμφανίζεται ο πίνακας χρηστών.

Ο πίνακας users αποτελεί το βασικό στοιχείο για του πιθανούς χρήστες του συστήματος. Η λογική αυτή περιέχει όλες τις απαραίτητες πληροφορίες για την ταυτοποίηση, την ασφάλεια και την διαχείριση των χρηστών. Με την σωστή σχεδίαση και διαχείριση του πίνακα εξασφαλίζουμε την βελτιστοποίηση της λειτουργικότητας, της ασφάλειας και της επέκτασης.

Τα βασικά στοιχεία του πίνακα αναφέρονται στα:

1. Ένας μοναδικός τύπος ID για κάθε χρήστη το οποίο είναι και το αναγνωριστικό του. Το πεδίο ID χρησιμεύει για την διαχείριση του χρήστη και αναφορά του μέσα στην εφαρμογή. Η χρήση AUTO\_INCREMENT σιγουρεύει ότι η εγγραφή αποκτά αυτόματα έναν μοναδικό αριθμό χωρίς να χρειάζεται καταχώρηση από τον διαχειριστή. Η λειτουργικότητα αυτή εξασφαλίζει ότι δεν θα υπάρξουν διπλές εγγραφές στο σύστημα. Επίσης η χρήση ενός ακέραιου αριθμού ως πρωτεύον κλειδί αποτελεί πιο αποδοτική λύση για πιο εγγυημένες αναζητήσεις.
2. Το email είναι μοναδικό για κάθε χρήστη, συνεπώς ο χρησιμοποιείται για ταυτοποίηση στο σύστημα δηλαδή το (login) είτε ως επικοινωνία. Θα χρησιμοποιείται επομένως και για την ασφαλή διαδικασία ταυτοποίησης χρηστών. Στη συνέχεια μπορεί να χρησιμοποιηθεί για ειδοποιήσεις και αποστολή newsletter ή την επαναφορά κωδικού.

3. Στο πεδίο username αποθηκεύεται το μοναδικό όνομα χρήστη το οποίο ορίζει ο ίδιος. Είναι ένας τρόπος αναγνώρισης του χρήστη και καλύπτει περιπτώσεις όταν η χρήση του email δεν είναι αποδεκτή ή δεν χρησιμοποιείται. Βρίσκεται μοναδικά για τον κάθε χρήστη ξεχωριστά και δεν μπορεί να γίνει χρήση του ίδιου ονόματος με κάποιον άλλον. Το username συχνά χρησιμοποιείται για εφαρμογές κοινωνικού περιβάλλοντος είτε ακόμη για ιστοσελίδες. Είναι ένας κρυπτογραφημένος κωδικός πρόσβασης ο οποίος αποθηκεύεται και συντάσσεται από τον χρήστη. Διασφαλίζεται η ασφάλεια των δεδομένων του εκάστοτε χρήστη. Με την παρουσίαση του password κρυπτογραφημένα αποτρέπουμε την πιθανή πρόσβαση του μη εξουσιοδοτημένου ή γκακόβουλου χρήστη στο σύστημα, ακόμη και αν η βάση που χρησιμοποιούμε μπορεί να παραβιαστεί και να απειληθεί από κάποιον επιτήδιο. Επομένως πρόκειται για ένα απαραίτητο στοιχείο για την ασφάλεια και την προστασία του χρήστη μέσα στο σύστημα.
4. Χρήσιμο στοιχείο είναι και η αποθήκευση στη βάση την ημερομηνία και την ώρα δημιουργίας του λογαριασμού του χρήστη. Παρέχει πληροφορίες για την εγγραφή συγκεκριμένα για τον χρόνο που έγινε.
5. Καλή τακτική είναι η απόκλιση χρηστών από την πλατφόρμα. Επομένως το στοιχείο active θα δείχνει αν ο χρήστης είναι ή όχι ενεργός.
6. Τέλος, χρειαζόμαστε ένα ακόμα στοιχείο να αποθηκεύει το αν ο χρήστης είναι ή όχι διαχειριστής, ώστε να αποκτά πρόσβαση σε επιπλέον λειτουργικότητες της πλατφόρμας. Καθώς όλοι οι χρήστες έχουν κοινά στοιχεία προς αποθήκευση δεν απαιτείται η δημιουργία ενός ξεχωριστού πίνακα, αλλά η τροποποίηση του σχετικού πεδίου isAdmin σε True ή False, για την αναγωγή του χρήστη σε διαχειριστή και αντίθετα.

Παρακάτω εμφανίζεται αναλυτικά ο πίνακας με τα στοιχεία του χρήστη

Δεδομένα Χρηστών		
Όνομα	Τύπος	Περιγραφή
id	INT	Μοναδικό γνώρισμα AUTO_INCREMENT
email	VARCHAR(50)	Μοναδικό email για κάθε χρήστη
username	VARCHAR(25)	Μοναδικό username για κάθε χρήστη
password	VARCHAR(300)	Κρυπτογραφημένος κωδικός χρήστη
created_at	TIMESTAMP	Ημερομηνία και ώρα δημιουργίας χρήστη
status	VARCHAR(15)	Λαμβάνει τις τιμές PENDING   ACTIVE   SUSPENDED για να δηλώσει κατάσταση αναμονής στην εγγραφή, ενεργός ή ανενεργός χρήστης

role	VARCHAR(25)	Περιγράφει την κατάσταση χρήστη αν είναι διαχειριστής ή όχι. Δυνατές τιμές: admin   user
------	-------------	--

Για την αποθήκευση των δένδρων χρειαζόμαστε μία σειρά από δεδομένα τα οποία τα περιγράφουν για την σωστή τους αποθήκευση. Συγκεκριμένα χρειάζεται να αποθηκεύσουμε:

1. Το μοναδικό γνώρισμα για κάθε δένδρο, id, το οποίο δεχόμαστε από τα δεδομένα του σταθμού. Για τον λόγο αυτό δεν θα είναι αυτόματο γνώρισμα, αλλά θα εισάγεται χειροκίνητα.
2. Το αριθμό κωδικού του δένδρου που αναφέρεται στο είδος του, όπως αυτό εμφανίζεται στα δεδομένα. Εφόσον προσκομιστούν και τα δεδομένα των ειδών θα μπορέσει να γίνει διασύνδεση με χρήση αυτού ως εξωτερικού κλειδιού.
3. Το όνομα δέντρου είναι ένα επιπλέον απαραίτητο στοιχείο προς αποθήκευση.
4. Η απόλυτη θέση x και y, όπως εμφανίζεται στα δεδομένα, θα βοηθήσει στην απεικόνιση των στοιχείων στην περιοχή
5. Οι θέσεις latitude και longitude είναι επίσης απαραίτητα στοιχεία για την απεικόνιση των δένδρων στον χάρτη.
6. Κάθε δένδρο βρίσκεται σε μία τοποθεσία. Για τον λόγο αυτό, απαιτείται το εξωτερικό κλειδί location\_id, που αναφέρεται στον πίνακα τοποθεσιών, με αντίστοιχη πληροφορία.
7. Η φωτογραφία του δένδρου είναι επίσης ένα στοιχείο που απαιτεί αποθήκευση για την εμφάνιση στους χρήστες.
8. Σημαντική είναι και η αποθήκευση της ημέρας και ώρας που έγινε εισαγωγή του δένδρου στην πλατφόρμα για ανάλυση στατιστικών.

Παρακάτω εμφανίζεται ο πίνακας δεδομένων των δένδρων

Δεδομένα Δένδρων		
Όνομα	Τύπος	Περιγραφή
id	INT	Μοναδικό γνώρισμα δένδρου
type_code	INT	Ο κωδικός του είδους του δένδρου, ως εξωτερικό κλειδί του πίνακα είδους
name	VACHAR(50)	Το όνομα του δένδρου
absolute_position_x	FLOAT	Η απόλυτη θέση x βάση της τοπικότητας
absolute_position_y	FLOAT	Η απόλυτη θέση y βάση της τοπικότητας
lat	FLOAT	Το latitude της τοποθεσίας του δέντρου

lon	FLOAT	Το longitude της τοποθεσίας του δέντρου
inserted_at	TIMESTAMP	Η ημερομηνία και ώρα εισαγωγής της εγγραφής
url	VARCHAR(100)	Το path αποθήκευσης της φωτογραφίας του δέντρου.
location_id	INT	Εξωτερικό κλειδί στον πίνακα Δεδομένα Τοποθεσιών, αναφερόμενο στο ID της τοποθεσίας
inserted_by	VARCHAR(125)	Ο χρήστης που εισήγαγε το δένδρο. Από DEFAULT ισούται με admin.

Επιπλέον χρειάζεται να αποθηκεύσουμε τα στοιχεία για τον τύπο του δέντρου και σχετικά χαρακτηριστικά.

Δεδομένα Τύπου Δένδρων		
Όνομα	Τύπος	Περιγραφή
id	INT	Μοναδικό γνώρισμα είδους δένδρου για κάθε νέα εισαγωγή
type_id	INT	Μοναδικό γνώρισμα είδους δένδρου όπως αυτό εμφανίζεται στα δεδομένα
greek_name	VARCHAR(50)	Όνομα τύπου σε ελληνικούς χαρακτήρες
scientific_name	VACHAR(50)	Όνομα τύπου σε λατινικούς χαρακτήρες με τον επιστημονικό του ορισμό
amount	int	Ο αριθμός των δένδρων αυτού του τύπου στο γεωγραφικό διαμέρισμα αναφοράς

Όπως προαναφέρθηκε είναι απαραίτητη η αποθήκευση των τοποθεσιών των δένδρων. Για την αποθήκευση αυτών απαιτούνται τα εξής στοιχεία:

1. Ένα μοναδικό γνώρισμα για κάθε δρόμο, ID, το οποίο θα εισάγεται αυτόματα από το σύστημα μέσω AUTO\_INCREMENT
2. Ο κωδικός του δρόμου όπως αυτός εισέρχεται από τα υπάρχοντα δεδομένα

3. Το όνομα του δρόμου
4. Ο αριθμός του δρόμου
5. Ο ταχυδρομικός κώδικας του δήμου στον οποίο ανήκει ο δρόμος
6. Ο κωδικός του δημοτικού διαμερίσματος που ανήκει ο δρόμος

Παρακάτω εμφανίζεται ο πίνακας περιγραφής των τοποθεσιών που θα χρησιμοποιηθεί προς αποθήκευση αυτών.

Δεδομένα Τοποθεσιών		
Όνομα	Τύπος	Περιγραφή
id	INT	Μοναδικός κωδικός AUTO_INCREMENT για την τοποθεσία δρόμου
tax_code	INT	Ο ταχυδρομικός κωδικός της τοποθεσίας
street_id	INT	ο κωδικός του δρόμου της τοποθεσίας
street_name	VARCHAR(100)	Το όνομα του δρόμου της τοποθεσίας
street_number	INT	Ο αριθμός τοποθεσίας του δέντρου
area_id	INT	Κωδικός δημοτικού διαμερίσματος

Τέλος, έχουμε έναν πίνακα στον οποίο θα αποθηκεύεται πληροφορία που αφορά την αποθήκευση των αγαπημένων δένδρων για κάθε χρήστη. Η διασύνδεση αυτή απαιτεί την διασύνδεση των δύο πινάκων μέσω των μοναδικών ids από τους δύο πίνακες.

Δεδομένα Αγαπημένων		
Όνομα	Τύπος	Περιγραφή
id	INT	Μοναδικός κωδικός AUTO_INCREMENT για το αγαπημένο
user_id	INT	Εξωτερικό κλειδί στον χρήστη που έχει αποθηκεύσει το δέντρο στα αγαπημένα
tree_id	INT	Εξωτερικό κλειδί στο δένδρο που έχει αποθηκευτεί από τον χρήστη στα αγαπημένα

created_at	TIMESTAMP	Ημερομηνία και ώρα για την στιγμή δημιουργίας της εγγραφής
------------	-----------	--

Τέλος ένας χρήστης μπορεί να εκτελεί αιτήματα προς τους διαχειριστές. Τα αιτήματα αυτά θα πρέπει να αποθηκεύονται και να εμφανίζονται στο σύστημα. Συγκεκριμένα ένα αίτημα αποτελείται από

1. Τον μοναδικό κωδικό του αιτήματος το οποίο δημιουργείται αυτόματα από το σύστημα
2. Τον κωδικό χρήστη που έκανε το αίτημα.
3. Τον τύπο αιτήματος, που είναι είτε να αφορά σε κάποιο ΔΕΝΔΡΟ είτε αίτημα ΠΡΟΦΙΛ είτε ΆΛΛΟ.
4. Τον κωδικό δέντρου που αφορά το αίτημα, αν αφορά δέντρο.
5. Το κείμενο με το αίτημα που ζητάει ο χρήστης
6. Την κατάσταση αιτήματος ανα πάσα στιγμή, όπως ΕΝΕΡΓΟ είτε ΚΛΕΙΣΤΟ
7. Απαραίτητη είναι η ημερομηνία δημιουργίας του αιτήματος.
8. Απαραίτητη είναι η ημερομηνία επίλυσης του αιτήματος.

Παρακάτω εμφανίζεται ο πίνακας περιγραφής των αιτημάτων

Δεδομένα Αιτημάτων		
Όνομα	Τύπος	Περιγραφή
id	INT	Μοναδικός κωδικός AUTO_INCREMENT για τον κωδικό αιτήματος
user_id	INT	Ο μοναδικός κωδικός χρήστη που έκανε το αίτημα, που λειτουργεί ως εξωτερικό κλειδί
type	VARCHAR(20)	Ο τύπος του αιτήματος που μπορεί να λάβει τιμές TREE PROFILE OTHER
target_id	INT	Ο μοναδικός κωδικός δένδρου που αφορά το δένδρο ή του χρήστη που αφορά το αίτημα. Μπορεί να είναι NULL σε περίπτωση OTHER τύπου
description	TEXT	Το κείμενο του αιτήματος του χρήστη
reply	TEXT	Το κείμενο της απάντησης από κάποιον ADMIN χρήστη

status	VARCHAR(20)	Η κατάσταση του αιτήματος αν είναι ενεργό, κλειστό ή υπό επεξεργασία (PENDING, OPEN, CLOSED)
created_at	TIMESTAMP	Η ημερομηνία δημιουργίας του αιτήματος
resolved_at	TIMESTAMP	Η ημερομηνία επίλυσης του αιτήματος
opened_at	TIMESTAMP	Η ημερομηνία απάντησης στο αίτημα

Επιπλέον αποθηκεύουμε τους ξεχωριστούς σταθμούς από τους οποίους λαμβάνουμε τις μετρήσεις για την ρύπανση.

Δεδομένα Σταθμών		
id	INT	Το primary key το οποίο αυξάνεται αυτόματα από την βάση.
name	VARCHAR(100)	Το όνομα του σταθμού
latitude	FLOAT	Η τοποθεσία latitude το σταθμού
longitude	FLOAT	Η τοποθεσία longitude του σταθμού

Επιπλέον χρειάζεται να αποθηκεύσουμε τα στοιχεία για την ρύπανση του περιβάλλοντα χώρου. Τα στοιχεία αυτού αναφέρονται στα

- Τον μοναδικό κωδικό id που αναφέρεται στην μέτρηση
- Τον αύξων αριθμό όπως αυτή προέρχεται από τις εξωτερικές πηγές των δεδομένων
- Την τοποθεσία που παρατηρήθηκαν οι μετρήσεις
- Την ημερομηνία που καταγράφηκε η μέτρηση
- Το όνομα της ημέρας της καταγραφής
- Το έτος καταγραφής
- Οι ακόλουθες μετρήσεις για τα δεδομένα στοιχεία
  - So2
  - PM10
  - PM2,5
  - CO
  - NO
  - NO2
  - O3
  - Θερμοκρασία

- Υγρασία (%)

Ο παρακάτω πίνακας εμφανίζει αυτά τα στοιχεία.

Δεδομένα Ρύπανσης		
Όνομα	Τύπος	Περιγραφή
id	INT	Μοναδικό γνώρισμα εισαγωγής
number	INT	Εξωτερικός A/A προερχόμενος από τα δεδομένα
station_id	INT	Το εξωτερικό κλειδί για το όνομα του σταθμού της μέτρησης
datetime	DATETIME	Η ημερομηνία και ώρα που έγινε η καταμέτρηση
date	DATE	Η ημερομηνία της μέτρησης
day	String(25)	Το όνομα της ημέρας της μέτρησης
year	INT	Η χρονολογία της μέτρησης
so2	FLOAT	Η μέτρηση διοξειδίου του θείου την δεδομένη στιγμή
pm10	FLOAT	Η μέτρηση της ποιότητας του αέρα
pm25	FLOAT	Η μέτρηση της μόλυνσης του αέρα
co	FLOAT	Η μέτρηση μονοξειδίου άνθρακα
no	FLOAT	Η μέτρηση του αζώτου την δεδομένη στιγμή
no2	FLOAT	Η μέτρηση του διοξειδίου αζώτου την δεδομένη στιγμή
o3	FLOAT	Η μέτρηση οξυγόνου την δεδομένη στιγμή
temperature	FLOAT	Η μέτρηση της θερμοκρασίας την δεδομένη στιγμή σε βαθμούς κελσίου

humidity	FLOAT	Η μέτρηση του ποσοστού της υγρασίας την δεδομένη στιγμή
----------	-------	---

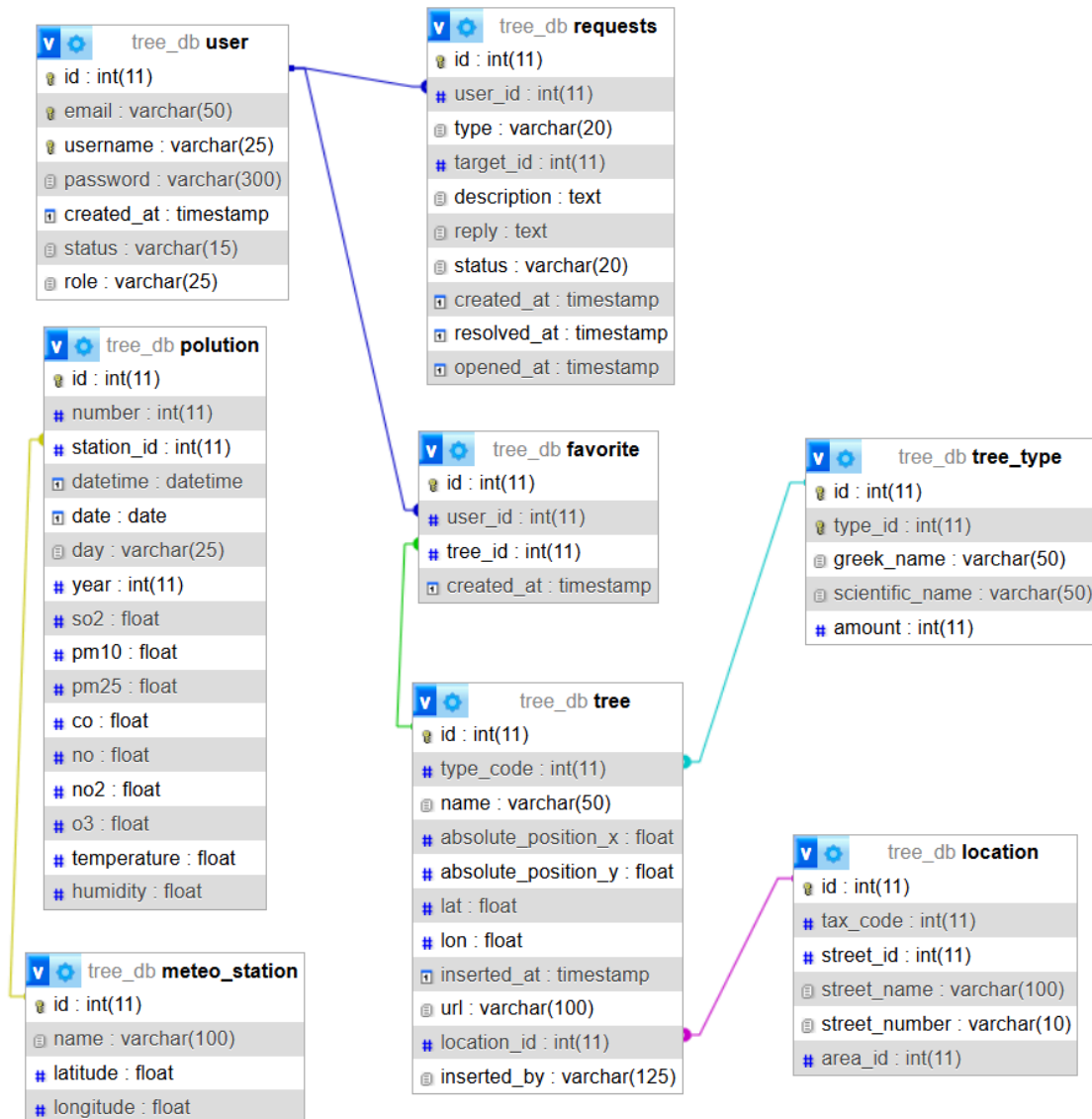
Οι πίνακες αυτοί συνδέονται με σχετικές συσχετίσεις, όπως εμφανίζεται στην επόμενη υποενότητα.

## Υλοποίηση Βάσης Δεδομένων

Η βάση έχει υλοποιηθεί σε γλώσσα MySQL. Η υλοποίηση της βάσης έχει γίνει μέσω του εργαλείου XAMPP για Windows συσκευές. Αρχικά πραγματοποιήθηκε η υλοποίηση της βάσης δεδομένων, με όνομα 'tree\_db'. Στην συνέχεια, κατασκευάστηκαν οι πίνακες που αναφέρθηκαν στην προηγούμενη υποενότητα με τα σχετικά τους στοιχεία. το αρχείο create\_db.sql περιέχει όλη την απαραίτητη πληροφορία για την δημιουργία της βάσης δυναμικά.

Επόμενο βήμα αφορά η εισαγωγή των δεδομένων. Τα δεδομένα αυτά μπορούν να εισαχθούν μέσα του αρχείου 'insert\_data.sql'. Τα δεδομένα αυτά μπορούν να πραγματοποιηθούν είτε χειροκίνητα, είτε κάνοντας χρήση ενός python κώδικα, όπως αναλύεται στην επόμενη υποενότητα.

Στην εικόνα εμφανίζεται η τελική εικόνα της βάσης δεδομένων από το εργαλείο XAMPP.



## Εξωτερικά Εργαλεία

Η κύρια χρήση των εξωτερικών εργαλείων αναφέρεται στην διαχείριση των δεδομένων της Βάσης αλλά και στην λήψη αναγκαίων πληροφοριών από εξωτερικές πηγές. Επομένως, η χρήση των εξωτερικών εργαλείων αφορά την αρχική εισαγωγή των δεδομένων με κατάλληλους τρόπους στην βάση αλλά και την συλλογή πληροφοριών από εξωτερικές πηγές, όπως στοιχεία για τον καιρό.

Τα δεδομένα που χρησιμοποιήθηκαν για την δημιουργία των δεδομένων και διαχειρίστηκαν οι κώδικες ρυθμοί, αντλήθηκαν από την πύλη ανοικτών δεδομένων του δήμου Θεσσαλονίκης<sup>1</sup> και από το ίδρυμα Κοπέρνικος<sup>2</sup>.

<sup>1</sup>

[https://opendata.thessaloniki.gr/el/search/field\\_topic/%CF%80%CE%B5%CF%81%CE%B9%CE%B2%CE%AC%CE%BB%CE%BB%CE%BF%CE%BD-26](https://opendata.thessaloniki.gr/el/search/field_topic/%CF%80%CE%B5%CF%81%CE%B9%CE%B2%CE%AC%CE%BB%CE%BB%CE%BF%CE%BD-26)

<sup>2</sup> <https://www.copernicus.eu/el>

## Python

Στην εφαρμογή, για την αρχική εισαγωγή, χρησιμοποιείται ένα Python script το οποίο αναλαμβάνει την λήψη των δεδομένων από τα xls αρχεία, όπως αυτά βρίσκονται στην βάση του Δήμου Θεσσαλονίκης και στην συνέχεια την παραγωγή του ανάλογου συνόλου 'Insert' εντολών για την SQL, με την προαναφερθέντα δομή. Για τον λόγο αυτό παράγει το αρχείο 'insert\_data.sql' το οποίο περιλαμβάνει όλες τις insert εντολές των δεδομένων στην βάση που έχει οριστεί.

Αξίζει να τονίσουμε ότι η Python έχει την δυνατότητα να εισαγάγει κατευθείαν τα δεδομένα μέσα στην βάση, χωρίς το ενδιάμεσο βήμα παραγωγής script. Για την παρούσα εργασία επιλέχθηκε η Python να εκτελεί και τις δύο προσεγγίσεις, και την παραγωγή δηλαδή των εντολών και την αποθήκευση αυτών σε σχετικό αρχείο, αλλά και την άμεση εισαγωγή τους στην βάση, ενημερώνοντας τον χρήστη για την επιτυχή προσθήκη των δεδομένων.

Παρακάτω εμφανίζεται ο κώδικας python που χρησιμοποιείται.

Για την υλοποίηση χρησιμοποιήθηκαν οι βιβλιοθήκες:

- Pandas: Ανάγνωση αρχείων XLS και χρήση/ αποθήκευσή τους μέσω σχετικών δομών
- Mysql-Connector: Σύνδεση με την βάση δεδομένων

### Βήματα Παραγωγής Κώδικα Μετεωρολογικών Σταθμών

Λόγω της ιδιομορφίας που παρουσιάζουν τα αρχεία με τα δεδομένα απαιτείται ιδιαίτερη διαχείριση αυτών. Συγκεκριμένα, ορισμένες περιοχές έχουν ελλιπή δεδομένα, επομένως απαιτείται πριν την εισαγωγή των δεδομένων από την Python, να εισάγουμε τις αντίστοιχες στήλες, με κενά δεδομένα. Εκτελώντας αυτές τις ενέργειες ορίζουμε κοινή διαχείριση για όλα τα δεδομένα στο ίδιο αρχείο και συνεπώς την ομοιομορφία του κώδικα.

Συγκεκριμένα, χρειάστηκε στο αρχείο 'Polution.xlsx', στα δύο τελευταία φύλλα του εγγράφου (Μτ.Στ. Πεδίο Άρεως και Μτ.Στ. Δώμα Παπαρρηγοπούλου), προσθέσαμε τις στήλες για τις περιβαλλοντολογικές μετρήσεις. Στο πρώτο φύλλο βρίσκεται η βιβλιογραφία, και επομένως την προσπερνάμε ως προς την διαχείριση.

Έχοντας μορφοποιήσει τα δεδομένα, θα κάνουμε χρήση του αρχείου ώστε να λάβουμε μόνο τα ονόματα των Μετεωρολογικών Σταθμών και να τα εισάγουμε στον αντίστοιχο πίνακα βάσης.

Στο τέλος προσθέτουμε έναν επιπλέον σταθμό με όνομα 'Copernicus' για να εισάγουμε τα επιπλέον δεδομένα χρονοσειράς που συλλέχθηκαν.

Ο κώδικας είναι ο παρακάτω.

```
def read_meteo_stations_data(excel_file):
    """Read meteorological stations data from Excel and generate SQL insert commands"""
    # Read all sheets
    excel = pd.ExcelFile(excel_file)
    sheet_names = excel.sheet_names[1:] # Skip first sheet
    sql_commands = []

    # Create a set for unique station names
    unique_stations = set()

    # First pass: collect unique station names
    for sheet_name in sheet_names:
        df = pd.read_excel(excel_file, sheet_name=sheet_name, header=None)
        station_name = re.sub(r'\s+', ' ', str(df.iloc[0, 0])).strip()
        unique_stations.add(station_name)

    # Generate INSERT statements for unique stations
    for station_name in unique_stations:
        sql = f"""
```

```

INSERT INTO tree_db.meteo_station (
    name
)
VALUES (
    '{station_name}'
);
"""
sql_commands.append(sql)

# Append Copernicus Data
sql = f"""
INSERT INTO tree_db.meteo_station (
    name
)
VALUES ('Copernicus');
"""
sql_commands.append(sql)

print(f"Found {len(unique_stations)} unique meteorological stations")
return sql_commands

```

## Βήματα Παραγωγής Κώδικα Ρύπανσης

Αφορά το ίδιο αρχείο της προηγούμενης υπο ενότητας, επομένως απαιτεί να γίνουν τα προαναφερθέντα βήματα. Έχοντας μορφοποιήσει όλα τα φύλλα, προχωράμε στην ξεχωριστή διαχείριση κάθε φύλλου του εγγράφου, διαβάζοντας τα δεδομένα ανα γραμμές και δημιουργώντας τις ανάλογες εντολές εισαγωγής sql.

Για την εξακρίβωση αν μία σειρά περιέχει έγκυρα δεδομένα, εξετάζουμε αν έχει εισαχθεί ημερομηνία στην γραμμή αυτή. Σε περίπτωση μη εισαχθείας ημερομηνίας θεωρούμε την γραμμή ως μη έγκυρη. Αυτό είναι απαραίτητο για να προσπεράσουμε γραμμές οι οποίες δεν περιέχουν δεδομένα, αλλά είναι από λάθος κενές.

Επιπλέον, η στήλη D αφορά το όνομα της ημέρας που παρατηρήθηκαν τα δεδομένα. Ωστόσο, αυτό εμφανίζεται με το όνομα της ημέρας, αλλά αποθηκεύεται με την αντίστοιχη ημερομηνία στο αρχείο. Επομένως απαιτείται από την Python η σωστή διαχείριση της ημερομηνίας - καθώς αυτή διαβάζεται - και η αποθήκευση της αντίστοιχης ημέρας.

Για την αποθήκευση των σταθμών, χρησιμοποιούμε το εξωτερικό κλειδί, από τον πίνακα που παράχθηκε στην προηγούμενη υποενότητα.

Καθώς για την εισαγωγή των δεδομένων εξετάζουμε τον αριθμό κάθε στήλης, μπορούμε να αλλάξουμε στο xlsx αρχείο τα ονόματα των Τίτλων (Headers) όχι όμως την σειρά εμφάνισής τους. Συγκεκριμένα η αντιστοίχιση αφορά την παρακάτω:

- Στήλη A: Γραμμή 0 το όνομα του Σταθμού
- Στήλη B: Αύξων αριθμός
- Στήλη C: Ημερομηνία
- Στήλη D: Ημέρα σε μορφή ημερομηνίας
- Στήλη E:SO
- Στήλη F:PM10
- Στήλη G:PM2.5
- Στήλη H:CO
- Στήλη I:NO
- Στήλη J:NO2
- Στήλη K:O3
- Στήλη L: Όνομα σταθμού και έτος (την προσπερνάμε)
- Στήλη M: Θερμοκρασία
- Στήλη N:Υγρασία (%)

Η σχετική συνάρτηση python εμφανίζεται παρακάτω:

```
def read_pollution_data(excel_file):
    """Read pollution data from Excel and generate SQL insert commands"""
    # Read all sheets
    excel = pd.ExcelFile(excel_file)
    sheet_names = excel.sheet_names

    # Skip only the first sheet
    relevant_sheets = sheet_names[1:]
    sql_commands = []

    # Add these day name mappings at the start of the function
    day_names = {
        0: 'Mon',
        1: 'Tue',
        2: 'Wed',
        3: 'Thu',
        4: 'Fri',
        5: 'Sat',
        6: 'Sun'
    }

    for sheet_name in relevant_sheets:
        # Read Excel without headers
        df = pd.read_excel(excel_file, sheet_name=sheet_name, header=None)

        # Get location name from the first row, first column and clean it
        location_name = re.sub(r'\s+', ' ', str(df.iloc[0, 0])).strip()

        # Skip the first row as it's usually a title
        df = df.iloc[1:]

        # Reset index to make sure we can access the rows properly
        df = df.reset_index(drop=True)

        for index, row in df.iterrows():
            try:
                # Skip row if date (column C) is empty
                if pd.isna(row[2]):
                    continue

                # Get values by column position
                # B column (index 1) = A.A.
                # C column (index 2) = Date
                # D column (index 3) = Day
                # E column (index 4) = SO2
                # F column (index 5) = PM10
                # G column (index 6) = PM2.5
                # H column (index 7) = CO
                # I column (index 8) = NO
                # J column (index 9) = NO2
                # K column (index 10) = O3
                # Last two columns = temperature and humidity

                # Convert date string to datetime with flexible format
                date_str = str(row[2]) # C column
                try:
                    # Try first format (dd/mm/yy)
                    date_obj = pd.to_datetime(date_str, format='%d/%m/%y')
                except:
                    try:
                        # Try second format (yyyy-mm-dd)
                        date_obj = pd.to_datetime(date_str)
                    except:
                        print(f"Could not parse date: {date_str} in row {index}")
                        continue

                # Get day name from date_obj
                day_of_week = day_names[date_obj.weekday()]
                year = date_obj.year

                # Get values by column position with better NULL handling
                values = {
```

```

'NULL',      'SO2': row[4] if pd.notna(row[4]) and str(row[4]).strip() != '' else
# E column
'NULL',      'PM10': row[5] if pd.notna(row[5]) and str(row[5]).strip() != '' else
# F column
'NULL',      'PM2.5': row[6] if pd.notna(row[6]) and str(row[6]).strip() != '' else
# G column
'NULL',      'CO': row[7] if pd.notna(row[7]) and str(row[7]).strip() != '' else
# H column
'NULL',      'NO': row[8] if pd.notna(row[8]) and str(row[8]).strip() != '' else
# I column
'NULL',      'NO2': row[9] if pd.notna(row[9]) and str(row[9]).strip() != '' else
# J column
'NULL',      'O3': row[10] if pd.notna(row[10]) and str(row[10]).strip() != '' else
# K column
' else 'NULL', # M column
'humidity': row[13] if pd.notna(row[13]) and str(row[13]).strip() != ''
else 'NULL'    # N column
}

sql = f"""
SET @station_id = (
    SELECT id FROM tree_db.meteo_station
    WHERE name = '{location_name}'
    LIMIT 1
);
INSERT INTO tree_db.polution (
    number, station_id, date, datetime, day, year,
    so2, pm10, pm25, co, no, no2, o3,
    temperature, humidity
)
VALUES (
    {row[1]},
    @station_id,
    '{date_obj.strftime('%Y-%m-%d')}',
    '{date_obj.strftime('%Y-%m-%d')} 00:00:00',
    '{day_of_week}',
    {year},
    {values['SO2']},
    {values['PM10']},
    {values['PM2.5']},
    {values['CO']},
    {values['NO']},
    {values['NO2']},
    {values['O3']},
    {values['temperature']},
    {values['humidity']}
);
"""
sql_commands.append(sql)

except Exception as e:
    print(f"Error processing row {index} in sheet {sheet_name}: {e}")
    print(f"Row data: {row}")
    continue

return sql_commands

```

Επιπλέον στην διάθεση μας διατίθενται και δεδομένα χρονοσειρών (timeseries) που αφορούν τα δεδομένα που συλλέχθηκαν από τον Κοπερνίκο. Ο κώδικας αυτός διαβάζει τα δεδομένα από το αντίστοιχο csv αρχείο και τα εισάγει με ανάλογο τρόπο.

```

def read_pollution_copernicus_data(csv_file):
    """Read pollution data from Copernicus CSV and generate SQL insert commands"""
    # Read CSV with headers since column names are needed
    df = pd.read_csv(csv_file)
    sql_commands = []

    # Add these day name mappings
    day_names = {
        0: 'Mon',

```

```

1: 'Tue',
2: 'Wed',
3: 'Thu',
4: 'Fri',
5: 'Sat',
6: 'Sun'
}

for index, row in df.iterrows():
    try:
        # Convert time string to datetime
        date_obj = pd.to_datetime(row['time'])

        # Get day name and year
        day_of_week = day_names[date_obj.weekday()]
        year = date_obj.year

        # Handle values with better NULL handling
        values = {
            'SO2': row['so2_conc'] if pd.notna(row['so2_conc']) and
str(row['so2_conc']).strip() != '' else 'NULL',
            'NO': row['no_conc'] if pd.notna(row['no_conc']) and
str(row['no_conc']).strip() != '' else 'NULL',
            'NO2': row['no2_conc'] if pd.notna(row['no2_conc']) and
str(row['no2_conc']).strip() != '' else 'NULL',
            'O3': row['o3_conc'] if pd.notna(row['o3_conc']) and
str(row['o3_conc']).strip() != '' else 'NULL',
            'CO': row['co_conc'] if pd.notna(row['co_conc']) and
str(row['co_conc']).strip() != '' else 'NULL',
        }

        sql = f"""
SET @station_id = (
    SELECT id FROM tree_db.meteo_station
    WHERE name = 'Copernicus'
    LIMIT 1
);
INSERT INTO tree_db.polution (
    number, station_id, date, datetime, day, year,
    so2, co, no, no2, o3,
    temperature, humidity, pm10, pm25
)
VALUES (
    {index + 1},
    @station_id,
    '{date_obj.strftime('%Y-%m-%d')}',
    '{date_obj.strftime('%Y-%m-%d %H:%M:%S')}',
    '{day_of_week}',
    {year},
    {values['SO2']},
    {values['CO']},
    {values['NO']},
    {values['NO2']},
    {values['O3']},
    NULL,
    NULL,
    NULL,
    NULL
);
"""
        sql_commands.append(sql)

    except Exception as e:
        print(f"Error processing row {index}: {e}")
        print(f"Row data: {row}")
        continue

return sql_commands

```

## Βήματα Παραγωγής Κώδικα Τύπου Δένδρων

Το αρχείο αυτό αφορά ένα csv αρχείο, όπως προέρχεται από τις πηγές. Πέραν ορισμένων κενών δεδομένων, το αρχείο αυτό δεν παρουσιάζει κάποια ιδιαίτερη μορφοποίηση,

επομένως διαβάζουμε κάθε γραμμή και αποθηκεύουμε τα δεδομένα των στηλών. Ως διαχωριστικό έχουμε το delimiter (','). Στην περίπτωση αυτών των δεδομένων, χρησιμοποιείται και το αυτόματο ID, παραγωγής από την βάση δεδομένων, αλλά και αυτό που εισέρχεται από τα δεδομένα, σε περίπτωση που αλλάξει κάποια ρύθμιση στην βάση των σταθμών, ώστε να είναι πάντα συγχρονισμένα.

Συγκεκριμένα η αντιστοίχιση στηλών είναι:

- Στήλη A: ID (εξωτερικό από τους σταθμούς)
- Στήλη B: Ελληνικό όνομα
- Στήλη C: Επιστημονικό όνομα σε Λατινικούς χαρακτήρες
- Στήλη D: md1
- Στήλη E: md2
- Στήλη F: md3
- Στήλη G: md4
- Στήλη H: md5
- Στήλη I: md6
- Στήλη J: md7
- Στήλη K: Συνολικό
- Στήλη L: Περιοχή σε τετραγωνικά μέτρα (m2)
- Στήλη M: Όγκος (crown volume)
- Στήλη M: Μέσος Όγκος (average crown volume)

Καθώς στην παρούσα εργασία ασχολούμαστε με ένα συγκεκριμένο γεωγραφικό διαμέρισμα, θα λάβουμε και θα αποθηκεύσουμε μόνο το md3, το οποίο αφορά την ποσότητα των ανάλογων δένδρων για το διαμέρισμα αναφοράς.

Ο κώδικας εισαγωγής των δεδομένων τύπων είναι ο παρακάτω.

```
def read_tree_types_data(csv_file):
    """Read tree types data from CSV and generate SQL insert commands"""
    # Read CSV without headers
    df = pd.read_csv(csv_file, header=None)
    sql_commands = []

    for index, row in df.iterrows():
        try:
            # Skip header row
            if index == 0:
                continue

            # Clean string values and handle NULL values
            # Column positions:
            # 0: type_id
            # 1: greek_name
            # 2: scientific_name
            # 3-8: md1-md6
            # 9: total
            # 10: area_m2
            # 11: crown_volume_m3
            # 12: avg_crown_volume_m3

            greek_name = re.sub(r'\s+', ' ', str(row[1])).strip() if pd.notna(row[1]) else ''
            scientific_name = re.sub(r'\s+', ' ', str(row[2])).strip() if pd.notna(row[2]) else ''

            # Handle numeric values
            values = {
                'md3': row[5] if pd.notna(row[5]) else 'NULL',
            }

            sql = f"""
            INSERT INTO tree_db.tree_type (
                type_id, greek_name, scientific_name, amount
```

```

    )
    VALUES (
        {row[0]},
        '{greek_name}',
        '{scientific_name}',
        {values['md3']}
    );
    """
    sql_commands.append(sql)

except Exception as e:
    print(f"Error processing row {index}: {e}")
    print(f"Row data: {row}")
    continue

return sql_commands

```

## Βήματα Παραγωγής Κώδικα Οδών Δένδρων

Το αρχείο με τα δέντρα αποτελείται μόνο από ένα φύλλο εργασίας με τους τίτλους (headings) να είναι σε αγγλικούς χαρακτήρες. Επομένως δύναται η άμεση ανάγνωσή τους. Στο αρχείο αυτό, παρατηρούμε ότι τα δεδομένα των περιοχών (οδών παρουσίας δένδρων) εμφανίζονται πολλαπλές φορές για πολλά δέντρα. Συνεπώς, αποθηκεύουμε τα δεδομένα αυτά σε ξεχωριστό πίνακα, και αναφερόμαστε σε αυτά μέσω εξωτερικού κλειδιού, για να μην έχουμε διπλότυπη πληροφορία.

Λόγω ιδιοτροπίας δεδομένων, ορισμένα μπορεί να έχουν λανθασμένα χαρακτηριστικά. Σε περίπτωση που ο αριθμός ενός δρόμου είναι κενός εισάγουμε τον αριθμό 0. Σε περίπτωση που ο αριθμός των δεδομένων είναι 'ΑΓΝΩΣΤΟΣ' ή ο ταχυδρομικός κώδικας κενός, επίσης εισάγουμε 0.

Λόγω εξάρτησης του εξωτερικού κλειδιού (foreign key), απαιτείται αρχικά να γίνει η εισαγωγή των τοποθεσιών και στην συνέχεια η εισαγωγή των δένδρων.

Η προσέγγιση του κώδικα εδώ αφορά την ανάγνωση των δεδομένων και την αποθήκευση των ξεχωριστών στοιχείων των οδών. Βασίζομαστε πάλι στις στήλες για την σωστή εισαγωγή των δεδομένων, επομένως τα ονόματα των τίτλων μπορούν να αλλάξουν, αλλά όχι η σειρά τους. Συγκεκριμένα έχουμε την αντιστοίχιση στηλών:

- Στήλη B: Ταχυδρομικός κώδικας
- Στήλη C: Κωδικός οδού
- Στήλη D: Όνομα οδού
- Στήλη E: Αριθμός οδού
- Στήλη F: Δημοτικό Διαμέρισμα (κωδικός)

Ο κώδικας εισαγωγής τοποθεσιών είναι ο παρακάτω.

```

def read_locations_data(excel_file):
    """Read locations data from Excel and generate SQL insert commands"""
    # Read Excel without headers
    df = pd.read_excel(excel_file, header=None)
    sql_commands = []

    # Skip header row
    df = df.iloc[1:]

    # Create a set of unique locations using relevant columns
    unique_locations = {} # Changed to dictionary to store all numbers for each street
    for index, row in df.iterrows():
        try:
            # Clean and prepare values
            tax_code = str(row[1]).strip() if pd.notna(row[1]) else '' # tk (column B)
            street_id = str(row[2]).strip() if pd.notna(row[2]) else '' # odos_id (column
C)
            street_name = re.sub(r'\s+', ' ', str(row[3])).strip() if pd.notna(row[3]) else
'' # onoma (column D)

```

```

        street_number = str(row[4]).strip() if pd.notna(row[4]) else '' # arithmos
(column E)
        area_id = int(float(row[5])) if pd.notna(row[5]) else 'NULL' #
dimotiko_diamerismo (column F)

        # Skip if essential fields are empty
        if not all([tax_code, street_id, street_name]):
            continue

        # Create key for uniqueness check (without street number)
        location_key = (tax_code, street_id, street_name)

        # Add to dictionary if it's a new street or update numbers if it exists
        if location_key not in unique_locations:
            unique_locations[location_key] = {
                'area_id': area_id,
                'numbers': set([street_number]) if street_number else set()
            }
        else:
            if street_number:
                unique_locations[location_key]['numbers'].add(street_number)
    except Exception as e:
        print(f"Error processing row {index}: {e}")
        print(f"Row data: {row}")
        continue

# Generate SQL commands for unique locations
for (tax_code, street_id, street_name), data in unique_locations.items():
    # Create one entry without number if no numbers exist
    if not data['numbers']:
        sql = f"""
INSERT INTO tree_db.location (
    tax_code, street_id, street_name, street_number, area_id
)
VALUES (
    '{tax_code}',
    '{street_id}',
    '{street_name}',
    NULL,
    {data['area_id']}
);
"""
        sql_commands.append(sql)
    else:
        # Create one entry for each unique number
        for number in data['numbers']:
            sql = f"""
INSERT INTO tree_db.location (
    tax_code, street_id, street_name, street_number, area_id
)
VALUES (
    '{tax_code}',
    '{street_id}',
    '{street_name}',
    {number},
    {data['area_id']}
);
"""
            sql_commands.append(sql)

print(f"Found {len(unique_locations)} unique streets with {sum(len(data['numbers']) for
data in unique_locations.values())} total addresses")
return sql_commands

```

## Βήματα Παραγωγής Κώδικα Δένδρων

Το αρχείο με τα δένδρα αποτελείται μόνο από ένα φύλλο εργασίας με τους τίτλους (headings) να είναι σε αγγλικούς χαρακτήρες. Επομένως δύναται η άμεση ανάγνωσή τους. Έχοντας ήδη εισάγει τα στοιχεία των οδών που βρίσκονται, προχωράμε στην παραγωγή του κώδικα για την εισαγωγή μόνο των στοιχείων των δένδρων, αλλά και την σύνδεση με των

εξωτερικών κλειδιών που έχουν παραχθεί από τις προηγούμενες εντολές. Για να παραλάβουμε το ανάλογο ID, εκτελούμε το sql query στον πίνακα τοποθεσιών, όπου θέλουμε όλα τα στοιχεία να είναι ίδια:

```
SELECT location.id FROM location I WHERE I.tax_code='{tc}' AND I.street_id='{sid}' AND I.street_name='{sn}' AND I.street_number='{stnum}' OR (I.street_number IS NULL AND 'stnum' = 'NULL').
```

Ο κώδικας είναι ο παρακάτω.

```
def read_trees_data(excel_file):
    """Read trees data from Excel and generate SQL insert commands"""
    # Read Excel without headers
    df = pd.read_excel(excel_file, header=None)
    sql_commands = []

    # Skip header row
    df = df.iloc[1:]

    for index, row in df.iterrows():
        try:
            # Skip row if essential data is missing
            if pd.isna(row[7]) or pd.isna(row[8]) or pd.isna(row[9]) or pd.isna(row[10]):
                # lat/lon coordinates
                continue

            # Get location components for foreign key reference
            tax_code = str(row[1]).strip() if pd.notna(row[1]) else '' # tk (column B)
            street_id = str(row[2]).strip() if pd.notna(row[2]) else '' # odos_id (column
C)
            street_name = re.sub(r'\s+', ' ', str(row[3])).strip() if pd.notna(row[3]) else
'' # onoma (column D)
            street_number = str(row[4]).strip() if pd.notna(row[4]) else 'NULL' # arithmos
(column E)

            sql = f"""
INSERT INTO tree_db.tree (
    type_code, location_id, absolute_position_x, absolute_position_y,
    lat, lon, common_name, inserted_at
)
SELECT
    '{row[6]}', # common_name (column G)
    l.id, # Get location_id from locations table
    {row[7]}, # x coordinate (column H)
    {row[8]}, # y coordinate (column I)
    {row[9]}, # latitude (column J)
    {row[10]}, # longitude (column K)
    '{row[6]}', # common_name (column G)
    '{datetime.now().strftime('%Y-%m-%d %H:%M:%S')}'
FROM tree_db.location l
WHERE l.tax_code = '{tax_code}'
AND l.street_id = '{street_id}'
AND l.street_name = '{street_name}'
AND (l.street_number = {street_number} OR (l.street_number IS NULL AND
{street_number} = 'NULL'))
LIMIT 1;
"""
            sql_commands.append(sql)

        except Exception as e:
            print(f"Error processing row {index}: {e}")
            print(f"Row data: {row}")
            continue

    return sql_commands
```

## Εισαγωγή στην Βάση

Η εισαγωγή των δεδομένων στην υπάρχουσα βάση γίνεται εκτελώντας όλες τις εντολές που έχουν παραχθεί από τα προηγούμενα βήματα. Λόγω των εξαρτήσεων, εισάγουμε πρώτα τα

δεδομένα μόλυνσης, μετά τους τύπους των δένδρων, τις τοποθεσίες και τέλος τα δεδομένα των δένδρων. Για την επικοινωνία με την βάση χρησιμοποιούμε το MYSQL CONNECTOR, και επικοινωνούμε με την τοπική Βάση Δεδομένων που κατασκευάσαμε, με τα εξής στοιχεία:

- Host: localhost
- User: root
- Password: <Κενό>
- Database: tree\_db

Ο κώδικας εισαγωγής είναι ο παρακάτω.

```
def import_data_to_db(sql_commands):
    """Import data to MySQL database"""
    try:
        connection = mysql.connector.connect(
            host='localhost',
            user='root', # replace with your MySQL username
            password='', # replace with your MySQL password
            database='tree_db'
        )

        cursor = connection.cursor()

        for command in sql_commands:
            cursor.execute(command)

        connection.commit()
        print("Data imported successfully!")

    except mysql.connector.Error as error:
        print(f"Failed to import data into MySQL table: {error}")

    finally:
        if connection.is_connected():
            cursor.close()
            connection.close()
            print("MySQL connection closed")
```

## APIs

- Ενσωμάτωση διαδραστικού χάρτη και χρήση API (π.χ. google maps)
- Εξωτερικά δεδομένα περιοχών (π.χ. weather acu )

Στην ενότητα αυτή αναφέρονται τα APIs, ως εξωτερικές κλήσεις, που χρησιμοποιήθηκαν για την υλοποίηση λειτουργιών της ιστοσελίδας. Συγκεκριμένα, χρησιμοποιήθηκαν OpenStreetMap για την αποτύπωση των δεδομένων σε χάρτες και η Weather Acu, για την αποτύπωση των τρεχόντων καιρικών συνθηκών στον τομέα εμφάνισης.

### Open Street Maps

Τα OpenStreetMaps, έχουν χρησιμοποιηθεί ως οι βασικοί χάρτες της εφαρμογής. Αυτά εμφανίζονται στην σελίδα περιήγησης 'TREE MAP' όπου έχουμε μία παρουσίαση όλων των δεδομένων των δέντρων επί του χάρτη. Η υλοποίηση αυτή εμφανίζεται στο αρχείο trees.php, όπου έχουμε την λήψη όλων των δέντρων από την βάση μας, τα οποία περιέχουν τις γεωγραφικές τους συντεταγμένες, και τελικά την ενσωμάτωση και κλήση των χαρτών στην σελίδα treeMap.html.

Η εφαρμογή είναι ανοικτού κώδικα (open Source) και συνεπώς δεν χρειάζεται κάποια ιδιαίτερη παραμετροποίηση [ <https://www.openstreetmap.org/> ] . Ο κώδικας εκτέλεσης εμφανίζεται παρακάτω.

```
function initMap() {
    // Initialize map with a default center
```

```

map = L.map('map').setView([0, 0], 13);

// Add OpenStreetMap tiles
L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
  maxZoom: 19,
  attribution: '© OpenStreetMap contributors'
}).addTo(map);

// Try to get user's location
if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition(
    (position) => {
      const pos = {
        lat: position.coords.latitude,
        lng: position.coords.longitude,
      };
      map.setView([pos.lat, pos.lng], 13);
      loadTrees();
    },
    () => {
      // If geolocation fails, just load trees
      loadTrees();
    }
  );
} else {
  loadTrees();
}
}

```

Η συνάρτηση `loadTrees()` καλεί ανάλογα SQL ερωτήματα προς την βάση για την λήψη των πληροφοριών κάθε δέντρου και την απεικόνισή του στον χάρτη, κάνοντας χρήση των συναρτήσεων στο `trees.php`.

Ο λόγος επιλογής αυτού του εργαλείου έναντι των Google Maps, αφορά το γεγονός ότι το δεύτερο απαιτεί την ύπαρξη ενός λογαριασμού με ενεργή πιστωτική κάρτα και την κατασκευή ανάλογου project, ώστε να επιφέρει πιθανές χρεώσεις με την χρήση του.

## Weather API

Η ιστοσελίδα <https://www.weatherapi.com/> αποτελεί μία σύγχρονη ιστοσελίδα που παρέχει στους χρήστες ένα σύνολο απο API για την αίτηση και λήψη πληροφοριών σχετικά με τις καιρικές συνθήκες που επικρατούν σε μία περιοχή. Ο χρήστης, κάνοντας εγγραφή σε αυτή, μπορεί να χρησιμοποιήσει τις συναρτήσεις που διατίθενται για λήψη της σχετικής πληροφορίας.

Βάση αυτού, έχει κατασκευαστεί μία συνάρτηση javascript η οποία ζητάει από το API να στείλει τις απαραίτητες πληροφορίες για τα καιρικά φαινόμενα και τις καιρικές συνθήκες τις περιοχής. Ο σχετικός κώδικας εμφανίζεται στην ιστοσελίδα `treeMap.html` και είναι ο ακόλουθος.

```

async function fetchWeatherData(lat, lon) {
  try {
    // Replace 'YOUR_API_KEY' with your WeatherAPI.com API key
    const apiKey = 'ff7a4bb9584f49dda48122650252404';
    const response = await
fetch(`https://api.weatherapi.com/v1/current.json?key=${apiKey}&q=${lat},${lon}&aqi=no`);
    const data = await response.json();

    if (data.current) {
      updateWeatherDisplay(data);
    } else {

```

```

                                console.error('Weather data fetch failed:',
data.error?.message);
    }
    } catch (error) {
        console.error('Error fetching weather data:', error);
    }
}

function updateWeatherDisplay(data) {
    const current = data.current;

    // Update weather icon
    document.getElementById('weather-icon').innerHTML =
        ``;

    // Update main weather info
    document.getElementById('weather-main').textContent =
current.condition.text;
    document.getElementById('weather-temp').textContent =
        `${Math.round(current.temp_c)}°C`;

    // Update weather details
    document.getElementById('weather-humidity').textContent =
        `${current.humidity}%`;
    document.getElementById('weather-wind').textContent =
        `${Math.round(current.wind_kph)} km/h`;
    document.getElementById('weather-pressure').textContent =
        `${current.pressure_mb} hPa`;
}

```

Όπως εμφανίζεται στον κώδικα, για την λήψη των πληροφοριών απαιτείται το API KEY το οποίο λαμβάνουμε κάνοντας την εγγραφή μας στο σύστημα.

# Λειτουργικότητα & Αξιολόγηση

Στο κεφάλαιο αυτό παρουσιάζονται οι βασικές λειτουργίες και οθόνες χρήσης από πλευράς χρηστών και διαχείρισης από πλευράς διαχειριστή. Στο τέλος της ενότητας παρουσιάζονται τα βήματα με τα οποία μπορεί να γίνει η εγκατάσταση του συστήματος.

## Εγγραφή & Είσοδος Χρήστη

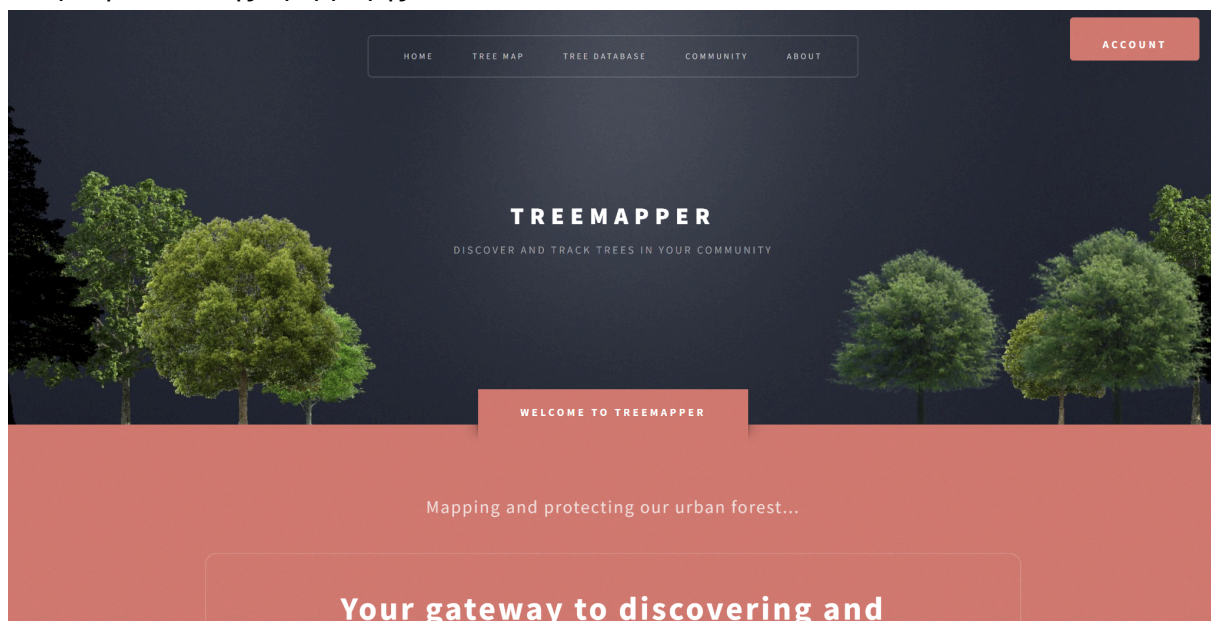
Ο χρήστης θα έχει περιορισμένη πρόσβαση στην ιστοσελίδα, εάν δεν έχει λογαριασμό. Για την δημιουργία του λογαριασμού υπάρχει σχετική οθόνη στην οποία ο χρήστης μπορεί να εισάγει τα στοιχεία με τα οποία επιθυμεί να εγγραφεί. Συγκεκριμένα εισάγει:

- username
- email
- password

Τα στοιχεία του username και email πρέπει να είναι μοναδικά. Ο κωδικός αποθηκεύεται στην βάση με hash, για να επιτευχθεί η ασφάλεια του συστήματος όσον αφορά την αποθήκευση των δεδομένων. Κατά την εγγραφή του ο χρήστης βρίσκεται σε κατάσταση 'PENDING' και αναμένεται η επιβεβαίωση από κάποιον χρήστη με ρόλο 'Admin' να επιτρέψει την είσοδό του στο σύστημα. Πάντα ένας χρήστης εγγράφεται ως απλός χρήστης ('user') και στην συνέχεια δύναται η αναβάθμισή του από κάποιον διαχειριστή. Τα στοιχεία username και password χρησιμοποιούνται στην συνέχεια για την είσοδο του χρήστη στην εφαρμογή.

Η προαναφερθέντα λογική υλοποιείται από ένα σύνολο html αρχείων για την γραφική απεικόνιση των στοιχείων σε συνδυασμό με τον css κώδικα για την μορφοποίηση, ένα σύνολο php αρχείων που αφορούν την επικοινωνία με την βάση δεδομένων καθώς και ένα σύνολο από javascript αρχεία για την επικοινωνία και την διαχείριση των αιτημάτων μεταξύ html και php.

Για την είσοδο/εγγραφή του χρήστη πρέπει να επιλέξει την επιλογή 'Account' από το πάνω δεξί μέρος της οθόνης ώστε να κατευθυνθεί στην σελίδα σύνδεσης. Παρακάτω εμφανίζεται η κεντρική σελίδα της εφαρμογής.



Η παρακάτω εικόνα εμφανίζει την αντίστοιχη σελίδα εισόδου/ εγγραφής του χρήστη.

# TREEMAPPER

DISCOVER AND TRACK TREES IN YOUR COMMUNITY

ACCOUNT ACCESS

**SIGN IN**

**SIGN UP**

admin

.....

**SIGN IN**

**RESET**

Η σελίδα εγγραφής (Sign UP) λειτουργεί με αντίστοιχο τρόπο, ζητώντας τα επιπλέον στοιχεία του χρήστη.

## ACCOUNT ACCESS

**SIGN IN**

**SIGN UP**

*Email*

*Username*

*Password*

*Confirm Password*

**SIGN UP**

**RESET**

Ο χρήστης ενημερώνεται για τις πράξεις που εκτελεί. Αν προσπαθήσει να κάνει 'Sign In' με ένα username το οποίο δεν υπάρχει, θα ενημερωθεί καταλλήλως, αφού ο κώδικας ελέγχει το μοναδικό username που εισήγαγε ο χρήστης εάν υπάρχει στην βάση δεδομένων. Αν υπάρχει ελέγχει-συγκρίνει τον εισαχθέν κωδικό με αυτό της βάσης. Αν τα στοιχεία συμπίπτουν ο χρήστης κάνει επιτυχώς σύνδεση στην πλατφόρμα, διαφορετικά υπάρχει ενημέρωση πως τα στοιχεία που δόθηκαν είναι λανθασμένα και άρα θα πρέπει να επανατοποθετήσει τα σωστά στοιχεία.

# TREEMAPPER

DISCOVER AND TRACK TREES IN YOUR COMMUNITY

ACCOUNT ACCESS

Invalid credentials

SIGN IN

SIGN UP

admin

.....

SIGN IN

RESET

Σε επιτυχή είσοδο του λογαριασμού, ο χρήστης ενημερώνεται και οδηγείται πίσω στην αρχική σελίδα. Η κατάσταση εισόδου είναι ορατή από το πάνω δεξιό σκέλος της ιστοσελίδας όπου φαίνεται το όνομα χρήστη του εισαχθέν ατόμου, και η δυνατότητα για έξοδο από το σύστημα (Log out).

Αντίστοιχα στην εγγραφή του χρήστη έχουμε τα ανάλογα βήματα και ενημέρωση του χρήστη..

ACCOUNT ACCESS

Registration successful

SIGN IN

SIGN UP

admin@mail.com

admin

....

....

Όταν ένας χρήστης εγγράφεται αναμένει επιβεβαίωση της εγγραφής. Σε περίπτωση χρήσης username ή email που υπάρχει ήδη στο σύστημα ενημερωνόμαστε καταλλήλως.

## ACCOUNT ACCESS

Email or username already exists

SIGN IN

SIGN UP

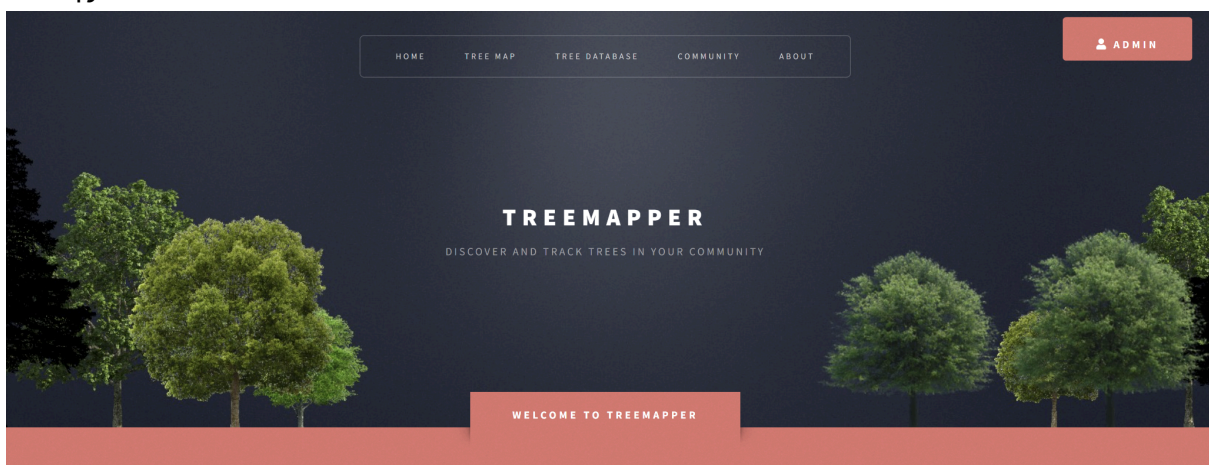
admin@mail.com

admin

Για την είσοδο στην ιστοσελίδα μπορούμε να χρησιμοποιήσουμε τον παρακάτω λογαριασμό admin:

- username: admin
- password: admin

Με την είσοδο στην σελίδα, έχουμε την εμφάνιση του ονόματος χρήστη στο πάνω μέρος της οθόνης.



Επιλέγοντας το αντίστοιχο κουμπί, οδηγούμαστε στην σελίδα 'My Profile' όπου μπορούμε να δούμε την σχετική πληροφορία.

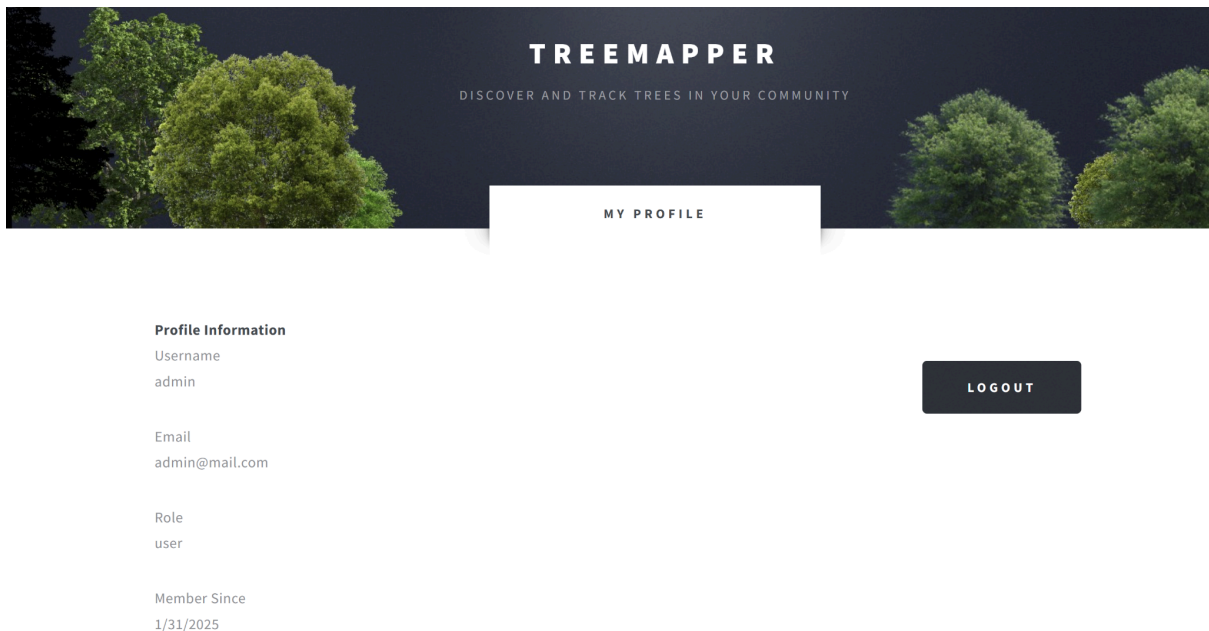
# Πλοήγηση Χρήστη

Στην ενότητα αυτή παρουσιάζονται οι βασικές οθόνες πλοήγησης χρήστη στο περιβάλλον της ιστοσελίδας.

## Οθόνη Προφίλ

Στην οθόνη αυτή ο χρήστης μπορεί να δει τις απαραίτητες πληροφορίες για τον λογαριασμό του. Συγκεκριμένα βλέπει:

- Το email με το οποίο έχει εισαχθεί
- Το username που χρησιμοποιεί
- Τον ρόλο του
- Την ημέρα εγγραφής του
- Αλλαγή κωδικού πρόσβασης
- Έξοδος από το σύστημα



Ο χρήστης μπορεί να αλλάξει τα παραπάνω στοιχεία, εφόσον ικανοποιούνται οι περιορισμοί που έχουμε προαναφέρει. Το email για λόγους ασφαλείας δεν μπορεί να αλλαχθεί και το username απαιτείται να είναι μοναδικό. Ο ρόλος δεν μπορεί να αλλαχθεί στο σημείο αυτό ούτε η ημερομηνία εγγραφής. Ο χρήστης μπορεί να αλλάξει τον τρέχων κωδικό του, εφόσον εισάγει πρώτα επιτυχώς τον τρέχων κωδικό.

Member Since  
1/31/2025

#### Change Username

UPDATE USERNAME

#### Change Password

CHANGE PASSWORD

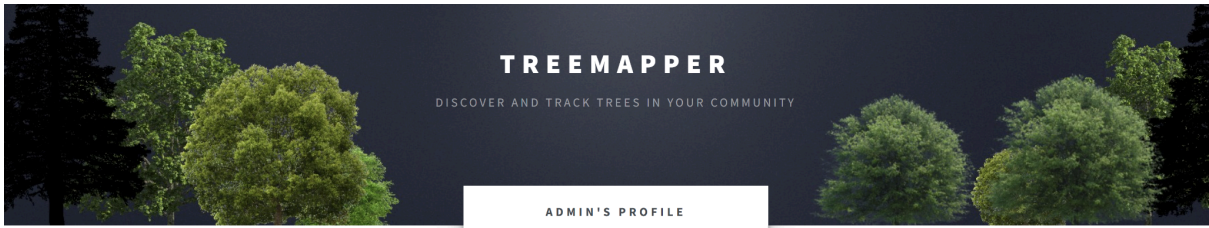
Ο χρήστης ενημερώνεται στην περίπτωση που έχει εισάγει λάθος στοιχεία ή δεν έχει εισάγει σωστά τις παραμέτρους.

#### Change Username

⚠ Please fill out this field.

UPDATE USERNAME

Επιπλέον ένας χρήστης επιτρέπεται να επισκεφτεί την σελίδα ενός άλλου χρήστη και να δει την σχετική πληροφορία που αφορά τα δέντρα που έχει ανεβάσει αυτός ο χρήστης αλλά και τα δέντρα που έχουν αποθηκευτεί ως αγαπημένα.



**ADMIN'S PROFILE**

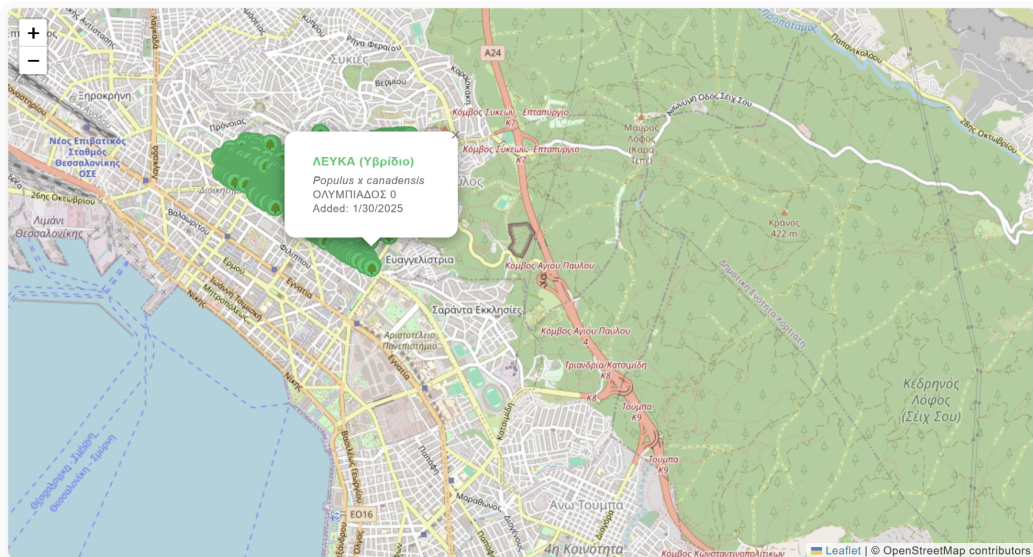
**admin**  
Member since: 1/31/2025 Uploads: 2 Favorites: 0 [REPORT USER](#)

**UPLOADS**   **FAVORITES**

 <b>ΛΙΓΟΥΣΤΡΟ</b> Type: ΛΙΓΟΥΣΤΡΟ (Ligustrum japonicum) Location: ΠΑΠΑΔΟΠΟΥΛΟΥ ΑΛΕΞΑΝΔΡΑΣ 30	 <b>ΣΟΦΟΡΑ</b> Type: ΣΟΦΟΡΑ (Sophora japonica) Location: ΑΓΙΟΥ ΔΗΜΗΤΡΙΟΥ 105
---	---

## Οθόνη Χάρτη Δένδρων

Σε αυτή την οθόνη οι χρήστες μπορούν να δουν τα δεδομένα των δέντρων τα οποία έχουν εισαχθεί στην ιστοσελίδα, τα οποία εμφανίζονται πάνω σε χάρτη, βάση της τοποθεσίας κάθε δέντρου. Σε αυτή την οθόνη έχουν πρόσβαση μόνο εγγεγραμμένοι χρήστες της σελίδας. Στην σελίδα εμφανίζονται όλα τα καταγεγραμμένα δέντρα με την σχετική τους πληροφορία, όπως είναι το όνομα, σχετικές μετρήσεις, τοποθεσία και άλλα.



Ο χρήστης επιλέγοντας ένα δέντρο από τον χάρτη μπορεί να δει τις σχετικές του πληροφορίες αναλυτικά.

#### Current Weather



Moderate or heavy rain with thunder

22°C

Humidity

57%

Wind Speed

17 km/h

Pressure

1008 hPa

#### Tree Information

No image available

##### Name

ΦΛΑΜΟΥΡΙΑ

##### Type Code

Greek Name: ΦΛΑΜΟΥΡΙΑ

Scientific Name: Tilia sp

##### Location

Street: ΜΩΡΕΑΣ 3

Tax Code: 546

Area ID: 3

##### Added On

2/25/2025

##### Inserted By

default

##### Coordinates

40.6381, 22.9543

##### Position

X: 411429, Y: 4498820

Amount

17

SUBMIT REQUEST

Στο πάνω μέρος εμφανίζονται οι πληροφορίες σχετικά με τον καιρό και το κλίμα της περιοχής όπως αυτές λαμβάνονται ζωντανά από την ιστοσελίδα <https://www.weatherapi.com> κάνοντας χρήση του ανάλογου API της εφαρμογής. Το δένδρο που έχουμε επιλέξει για προβολή, βρίσκεται σε μία τοποθεσία (latitude, longitude) και συνεπώς βάση αυτών των στοιχείων λαμβάνουμε τα στοιχεία για την αντίστοιχη περιοχή.

### Φίλτρα Αναζήτησης

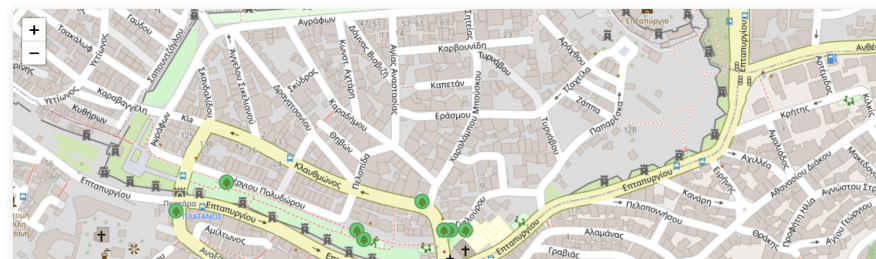
Επιπλέον, ο χρήστης μπορεί να κάνει αναζήτηση (filter) βάση συγκεκριμένων μετρικών, όπως είναι ο τύπος του δέντρου ή το όνομα αυτού, εμφανίζοντας έτσι στον χάρτη μόνο τα δεδομένα με τα σχετικά γνωρίσματα. Συγκεκριμένα, τα γνωρίσματα αναζήτησης αφορούν:

- Αναζήτηση βάση Ελληνικού ονόματος: Ο χρήστης μπορεί να εισάγει το ελληνικό όνομα του δέντρου και να εμφανιστούν τα σχετικά αποτελέσματα



Tree Type: All Types | Search by Name: ΠΛΑΤΑΝΙ | Search by Radius (meters): Enter radius... | LOCATION

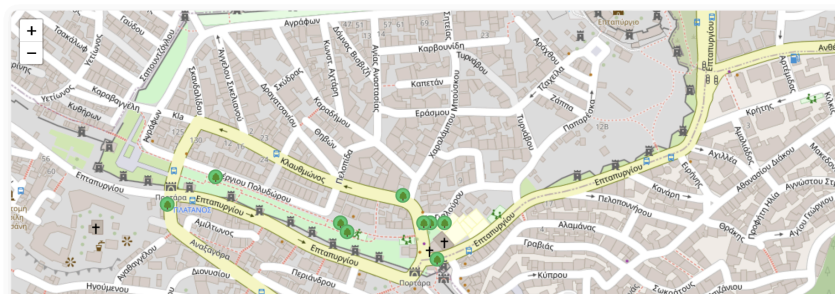
APPLY FILTERS | RESET



- Αναζήτηση βάση Επιστημονικού ονόματος: Ο χρήστης μπορεί να εισάγει το επιστημονικό όνομα ενός δέντρου και να εμφανιστούν τα σχετικά αποτελέσματα

Tree Type: All Types | Search by Name: Platanus | Search by Radius (meters): Enter radius... | LOCATION

APPLY FILTERS | RESET



- Αναζήτηση βάση Τύπου δέντρου: Ο χρήστης μπορεί να επιλέξει τον επιθυμητό τύπου δέντρου και να εμφανιστούν μόνο τα δεδομένα που ικανοποιούν αυτόν τον τύπο



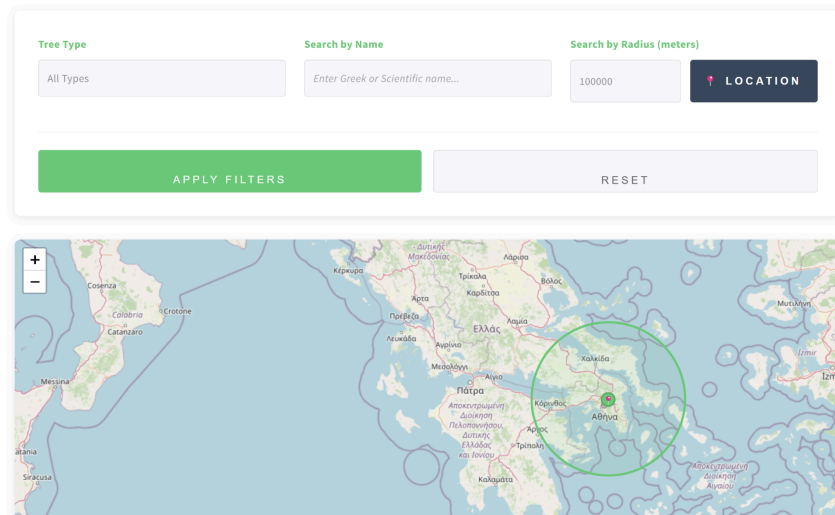
Tree Type: ΑΧΑΔΙΑ (Pyrus communis) | Search by Name: Enter Greek or Scientific name... | Search by Radius (meters): Enter radius... | LOCATION

APPLY FILTERS | RESET



Ο τύπος των δέντρων είναι ένα DropDown στοιχείο, το οποίο εμφανίζει δυναμικά όλους τους τύπους δέντρων που έχουν εισαχθεί.

- Αναζήτηση βάση Απόστασης από τοποθεσία με Χ μέτρα ακτίνας εύρους: Ο χρήστης μπορεί να επιλέξει την τωρινή του τοποθεσία, η οποία λαμβάνεται αυτόματα από την συσκευή του χρήστη, και τα μέτρα ακτίνας, και να εμφανιστούν μόνο τα δεδομένα που βρίσκονται σε ακτίνα Χ μέτρα από την δοθείσα τοποθεσία.

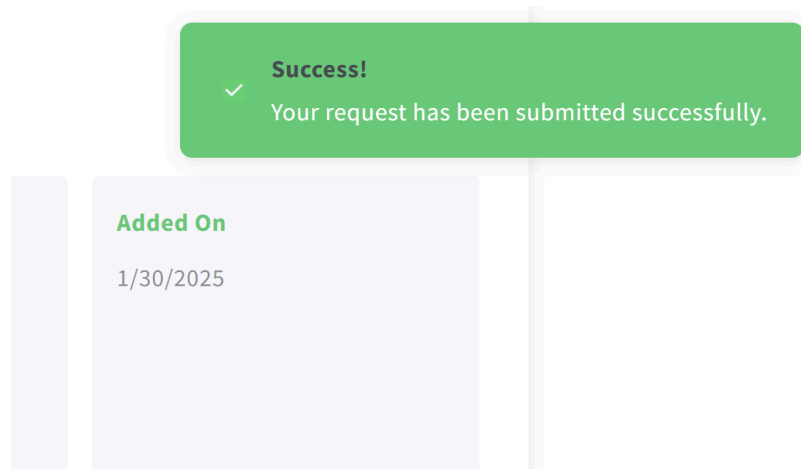


Ο χρήστης μπορεί να επιλέξει να εφαρμόσει τα φίλτρα (Apply Filters) στην περίπτωση της αναζήτησης με τοποθεσία, ή να καθαρίσει τα φίλτρα που έχουν εισαχθεί (Reset), δείχνοντας πάλι τα αρχικά δεδομένα. Τα δεδομένα για τα υπόλοιπα στοιχεία εφαρμόζονται δυναμικά.

### Αιτήματα Δέντρων

Επιλέγοντας ένα δέντρο ο χρήστης μπορεί να δημιουργήσει ένα νέο αίτημα προς αυτό το δέντρο. Ο χρήστης επιλέγει τον τύπο του αιτήματος (TREE ως προεπιλογή) και καταγράφει τον λόγο που αφορά το αίτημα που κάνει.

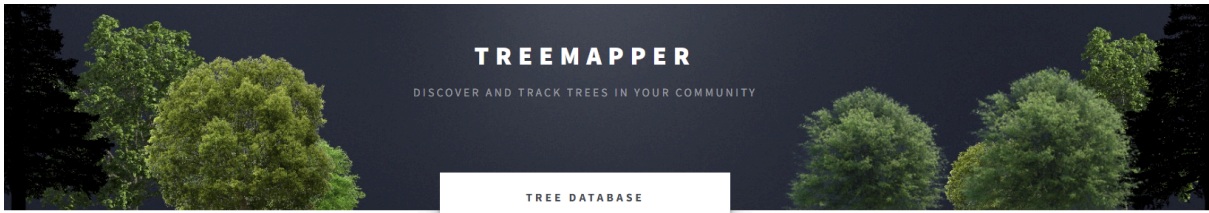
Το αίτημα, με χρήση του requests.php, καταγράφεται στην βάση και θα είναι ορατό πλέον από κάποιος διαχειριστή να ασχοληθεί με το σχετικό ζήτημα. Ο χρήστης ενημερώνεται με σχετικό μήνυμα, ότι το αίτημα έχει καταχωρηθεί.



## Οθόνη Βάσης Δέντρων

Η σχετική σελίδα από το μενού 'Tree Database' παρουσιάζει τις πληροφορίες για τα δέντρα και τους τύπους δέντρων που έχουν καταγεγραφεί. Η είσοδος επιτρέπεται μόνο σε χρήστες που έχουν συνδεθεί. Η σελίδα αυτή χωρίζεται σε τρεις βασικές σελίδες (1) τύποι των δέντρων, (2) τα δέντρα που έχουν εισαχθεί και (3) την δυνατότητα ανεβάσματος (upload) νέου δέντρου.

Συγκεκριμένα ο χρήστης στην πρώτη σελίδα μπορεί να δει τους τύπους δέντρων που έχουν εισαχθεί, με ανάλογα στοιχεία γενικών πληροφοριών.



TREE DATABASE

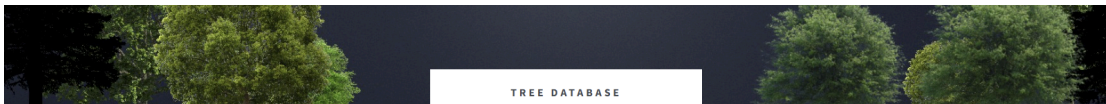
TREE TYPES    TREE DATABASE    UPLOAD TREE

Tree Types Database

<b>Total Types</b> <b>92</b>	<b>Total Amount</b> <b>1,406</b>	<b>Average Amount</b> <b>15.28</b>
---------------------------------	-------------------------------------	---------------------------------------

<b>_ΑΓΝΩΣΤΟ ΕΙΔΟΣ_</b> <i>Άγνωστο είδος</i> <b>Amount: 132</b>	<b>ΑΓΓΕΛΙΚΗ</b> <i>Pittosporum tobira</i> <b>Amount: 6</b>	<b>ΑΕΙΛΑΝΘΟΣ</b> <i>Ailanthus altissima</i> <b>Amount: 2</b>	<b>ΑΚΑΚΙΑ ΚΩΝΣΤΑΝΤΙΝΟΥΠΟΛΕΩΣ</b> <i>Albizia julibrissin</i> <b>Amount: 80</b>
<b>ΑΚΑΚΙΑ ΣΦ</b> <i>Acacia sp</i> <b>Amount: 5</b>	<b>ΑΜΥΓΔΑΛΙΑ</b> <i>Prunus amygdalus</i> <b>Amount: 1</b>	<b>ΑΡΙΑ</b> <i>Quercus ilex</i> <b>Amount: 1</b>	<b>ΑΡΙΖΟΝΑ</b> <i>Cupressus arizonae</i> <b>Amount: 8</b>

Στην συνέχεια η δεύτερη σελίδα παρουσιάζει όλα τα δεδομένα των δέντρων που έχουν εισαχθεί. Στο άνω μέρος της σελίδας παρουσιάζονται γενικά στοιχεία και μεταβλητές για τα δεδομένα, όπως συνολικός αριθμός δέντρων, διαφορετικοί τύποι που εμφανίζονται και το όνομα της τελευταίας εισαγωγής που έχει πραγματοποιηθεί.



TREE TYPES    TREE DATABASE    UPLOAD TREE

Tree Database

<b>Total Trees</b> <b>1407</b>	<b>Total Types</b> <b>53</b>	<b>Latest Addition</b> <b>Peukidi</b>
-----------------------------------	---------------------------------	--

Type Distribution

<b>ΠΕΥΚΟ</b> 0% 3 trees	<b>ΠΛΑΤΑΝΙ</b> 1% 21 trees	<b>ΑΚΑΚΙΑ ΚΩΝΣΤΑΝΤΙΝΟΥΠΟΛΕΩΣ</b> 5% 71 trees	<b>ΨΕΥΔΑΚΑΚΙΑ</b> 7% 93 trees
<b>ΝΕΡΑΤΣΙΑ</b> 3% 40 trees	<b>ΛΙΓΟΥΣΤΡΟ</b> 11% 150 trees	<b>_ΑΓΝΩΣΤΟ ΕΙΔΟΣ_</b> 16% 226 trees	<b>ΙΒΙΣΚΟΣ</b> 1% 12 trees
<b>ΔΑΦΝΗ</b> 3% 36 trees	<b>ΣΟΦΟΡΑ</b> 19% 265 trees	<b>ΑΡΙΖΟΝΑ</b> 1% 8 trees	<b>ΤΟΥΓΙΑ</b> 0% 7 trees
<b>ΛΩΤΟΣ</b> 0% 2 trees	<b>ΠΙΚΡΟΔΑΦΝΗ</b> 0% 6 trees	<b>ΜΟΥΡΙΑ</b> 1% 10 trees	<b>ΚΥΠΑΡΙΣΣΙ</b> 1% 10 trees
<b>ΔΑΦΝΟΚΕΡΑΣΟ</b> 0% 0 trees	<b>ΕΛΙΑ</b> 1% 14 trees	<b>ΧΑΜΑΙΡΩΠΑΣ</b> 0% 0 trees	<b>ΥΓΡΑΜΒΑΡΗ</b> 1% 14 trees

Στο κάτω μέρος της σελίδας, έχουμε την παρουσίαση όλων των δέντρων και συνεπώς της βάσης δεδομένων. Κάθε δέντρο εμφανίζεται με την σχετική του πληροφορία.

1 trees   1 trees   1 trees   1 trees

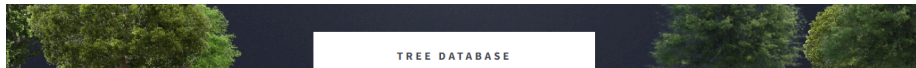
**ΚΑΜΕΛΙΑ** 0%  
1 trees

🔍 Search trees by name...   🌳 All Types   ↻ RESET FILTERS

<p><b>Peukidi</b> 1</p> <p>Type: ΠΕΥΚΟ Scientific Name: Pinus sp Location: ΑΙΣΧΥΛΟΥ 9 Coordinates: 35.129, 23 Added By: admin Added On: 2/11/2025</p>	<p><b>ΝΕΡΑΤΖΙΑ</b> 2</p> <p>Type: ΝΕΡΑΤΖΙΑ Scientific Name: Citrus aurantium Location: ΚΛΑΥΘΜΩΝΟΣ 39 Coordinates: 40.6423, 22.9582 Added By: admin Added On: 2/10/2025</p>	<p><b>ΛΙΓΟΥΣΤΡΟ</b> 3</p> <p>Type: ΛΙΓΟΥΣΤΡΟ Scientific Name: Ligustrum japonicum Location: ΠΟΛΥΔΡΟΥ 0 Coordinates: 40.6426, 22.9555 Added By: admin Added On: 2/10/2025</p>	<p><b>ΑΚΑΚΙΑ ΚΩΝΣΤΑΝΤΙΝΟΥΠΟΛΕΩΣ</b> 4</p> <p>Type: ΑΚΑΚΙΑ ΚΩΝΣΤΑΝΤΙΝΟΥΠΟΛΕΩΣ Scientific Name: Albizia julibrissin Location: ΠΑΠΑΡΕΣΚΑ 0 Coordinates: 40.643, 22.9614 Added By: admin Added On: 2/10/2025</p>
<p><b>ΨΕΥΔΑΚΑΚΙΑ</b> 5</p> <p>Type: ΨΕΥΔΑΚΑΚΙΑ Scientific Name: Robinia pseudoacacia Location: ΒΕΝΙΖΕΛΟΥ ΕΛΕΥΘΕΡΙΟΥ (ΟΡΙΑ ΑΓΙΟΣ ΠΑΥΛΟΣ) 0 Coordinates: 40.6417, 22.9599 Added By: admin Added On: 2/10/2025</p>	<p><b>ΝΕΡΑΤΖΙΑ</b> 6</p> <p>Type: ΝΕΡΑΤΖΙΑ Scientific Name: Citrus aurantium Location: ΜΟΥΣΚΟΥ ΧΑΡΑΛ 21 Coordinates: 40.6429, 22.9594 Added By: admin Added On: 2/10/2025</p>	<p><b>ΨΕΥΔΑΚΑΚΙΑ</b> 7</p> <p>Type: ΨΕΥΔΑΚΑΚΙΑ Scientific Name: Robinia pseudoacacia Location: ΙΠΑΤΡΟΥ ΔΗΜΗΤΡΙΟΥ 0 Coordinates: 40.6421, 22.9605 Added By: admin Added On: 2/10/2025</p>	<p><b>ΣΟΦΟΡΑ</b> 8</p> <p>Type: ΣΟΦΟΡΑ Scientific Name: Sophora japonica Location: ΜΟΥΣΚΟΥ ΧΑΡΑΛ 0 Coordinates: 40.6417, 22.9591 Added By: admin Added On: 2/10/2025</p>

Επιτρέπεται η αναζήτηση με όνομα και τύπο.

Τέλος, δύναται το ανέβασμα ενός νέου δέντρου από την σχετική οθόνη.



TREE TYPES   TREE DATABASE   **UPLOAD TREE**

Upload New Tree

**Tree Name**

**Tree Type**

**Latitude**   **Longitude**  
  

**Location**

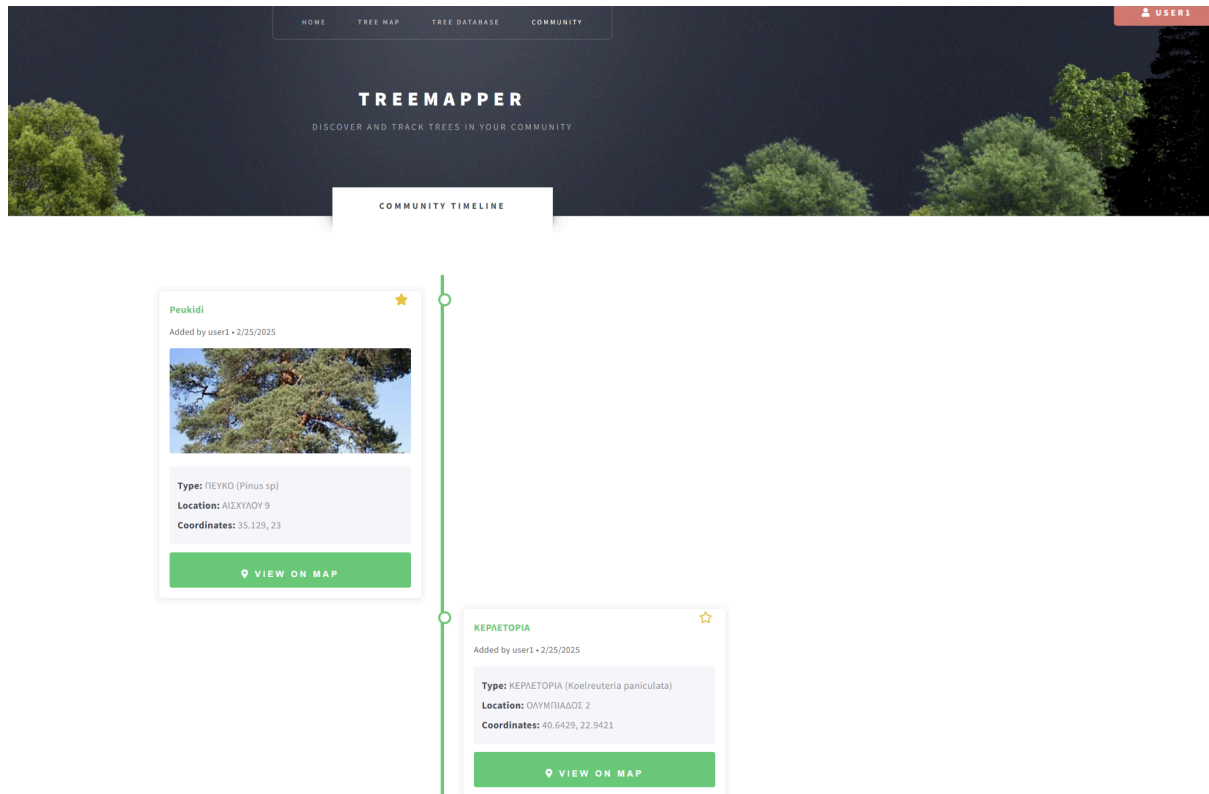
**Add New Location**

**Image**  
 No file chosen  
Accepted formats: JPG, PNG, GIF (max 5MB)

Για νέα εισαγωγή δέντρου απαιτείται από τον χρήστη να εισάγει το όνομα του δέντρου, τον τύπο και την τοποθεσία του, συμπληρώνοντας τόσο την γεωγραφική τοποθεσία του (latitude, longitude) αλλά και την οδό στην οποία βρίσκεται. Αν δεν έχει ήδη εισαχθεί η σχετική οδός, επιτρέπεται η εισαγωγή νέας τοποθεσίας. Τέλος ο χρήστης επιλέγει την φωτογραφία του δέντρου που επιθυμεί να εισάγει.

## Οθόνη Κοινότητας

Η οθόνη αυτή είναι κοινή και για τους χρήστες και για τους διαχειριστές. Εδώ εμφανίζεται σε χρονική σειρά οι αναρτήσεις που πραγματοποιήθηκαν από τους χρήστες με την σχετική τους πληροφορία.



Στην σελίδα αυτή δύναται ο χρήστης να επιλέξει μία ανάρτηση ως 'Αγαπημένη'. Το αγαπημένο αποθηκεύονται στην βάση δεδομένων και μπορεί αργότερα να τα ανατρέξει από την σελίδα προφίλ του.

## Οθόνη Ρύπανσης

Η οθόνη αυτή είναι κοινή και για τους χρήστες και για τους διαχειριστές. Εδώ εμφανίζεται μία σειρά από γραφήματα που αφορούν τις μετρήσεις ρύπανσης από τους διαφορετικούς σταθμούς. Οι χρήστες μπορούν να επιλέξουν το χρονικό διάστημα μέτρησης (π.χ. 7 ημέρες, 1 χρόνο, κλπ) καθώς και συγκεκριμένη μέτρηση (π.χ. ο3) ή όλες τις μετρήσεις. Στην συνέχεια η σελίδα παρουσιάζει στον χρήστη τα αποτελέσματα της αναζήτησης βάση όσων έχει ορίσει. Τα γραφήματα είναι ποικίλα και περιλαμβάνουν τα παρακάτω:

- Εικονογράφηση μόλυνσης σε χάρτη
- Χρονοδιαγράμματα
- Κυκλικά διαγράμματα
- Ραβδογράμματα

Κάθε διάγραμμα παρουσιάζει σχετικές μετρήσεις, εφόσον αυτές είναι διαθέσιμες και επιλεγμένες για το εν λόγω χρονικό διάγραμμα.

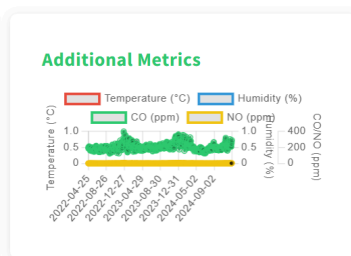
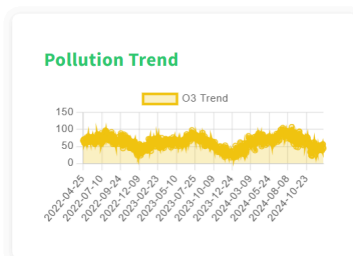
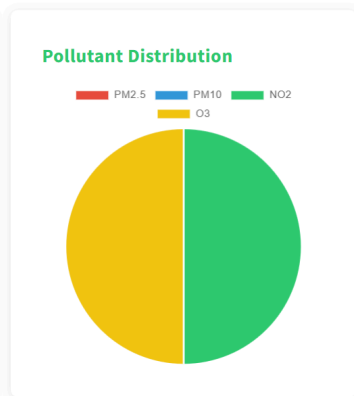
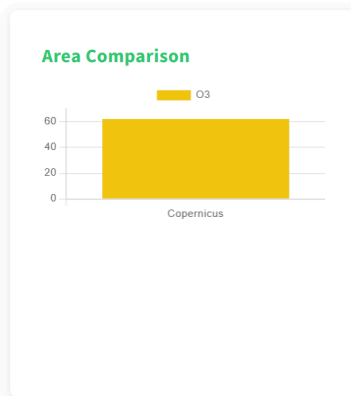
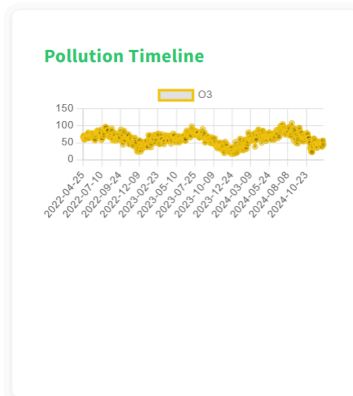
Ο χρήστης μπορεί να επιλέξει τις μετρήσεις κατά βήμα. Δηλαδή, μπορεί αρχικά να επιλέξει να λάβει μόνο τις μετρήσεις σχετικές με το O3 και στην συνέχεια για το NO2, δίνοντας έτσι την δυνατότητα στον χρήστη να συγκρίνει τις διαφορετικές τιμές που καταγράφηκαν στα δεδομένα χρονικά διαστήματα.



**Time Range**  
Last 3 Years

**Pollutant Type**  
O3

**UPDATE DATA**



## Οθόνη Διαχειριστή

Ο Διαχειριστής (Admin) έχει στην διάθεσή του μία επιπλέον σελίδα πλοήγησης, 'Admin Dashboard', η οποία αφορά όλα τα διαχειριστικά κομμάτια και δυνατότητες. Συγκεκριμένα, σε αυτή την σελίδα ο χρήστης δύναται να δει όλους τους χρήστες που βρίσκονται εγγεγραμμένοι στην ιστοσελίδα, με την κατάσταση τους (ACTIVE | PENDING | SUSPENDED), αλλά και τα ενεργά αιτήματα που έχουν δημιουργηθεί από χρήστες.

### Ενεργοί Χρήστες

Ο διαχειριστής, μπορεί να επιλέξει έναν χρήστη και να αλλάξει την κατάσταση του, επιτρέποντάς του έτσι την είσοδο του και χρήση των δυνατοτήτων της ιστοσελίδας. Αντίστοιχα, μετατρέποντας έναν χρήστη σε κατάσταση SUSPENDED, αποτρέπει την είσοδό του.



#### User Management

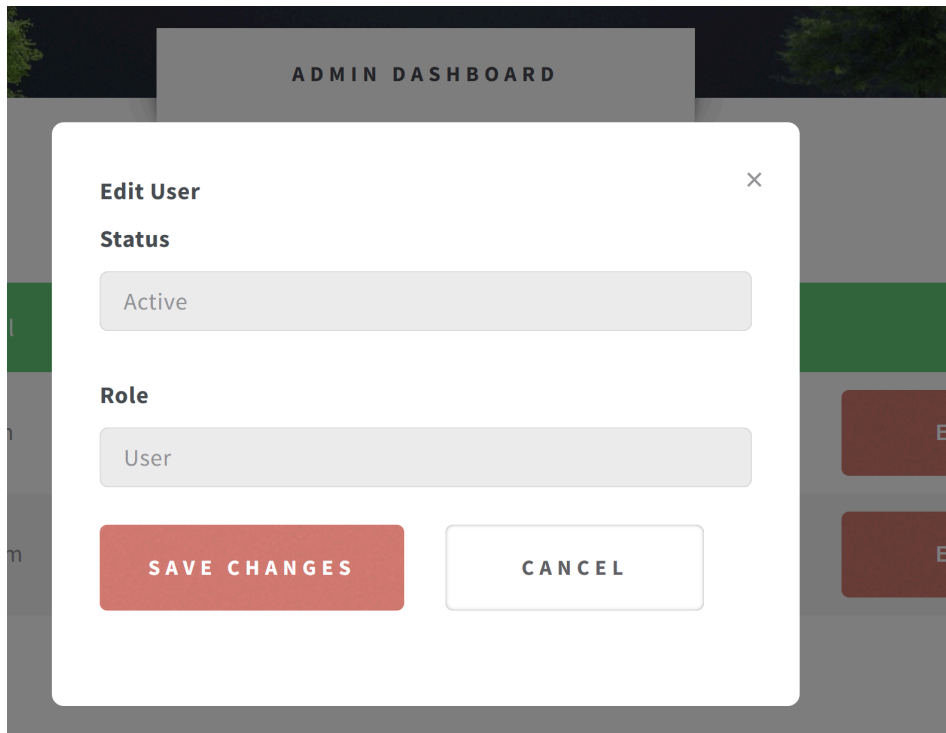
Username	Email	Status	Role	Member Since	Actions
user1	user1@mail.com	ACTIVE	user	2/3/2025	EDIT
admin	admin@mail.com	ACTIVE	admin	1/31/2025	EDIT

#### Request Management

User	Type	Status	Description	Created	Opened	Resolved	Actions
------	------	--------	-------------	---------	--------	----------	---------



No requests found. All caught up!



## Αιτήματα

Επιπλέον, στην σελίδα αυτή ο διαχειριστής μπορεί να δει τα ενεργά αιτήματα που έχουν γίνει από τους χρήστες και να απαντήσει σε αυτά και να αλλάξει κατά αντιστοιχεί την κατάσταση κάθε αιτήματος.

Request Management							
User	Type	Status	Description	Created	Opened	Resolved	Actions
admin	TREE	PENDING	The scientific name is badly formatted	2/3/2025	-	-	EDIT

Τα αιτήματα αφορούν (1) δέντρα, (2) άλλους χρήστες ή (3) άλλον λόγο. Ο διαχειριστής μπορεί να ανοίξει ένα αίτημα και να δει το κείμενο που έχει σημειώσει ο χρήστης, να απαντήσει σε αυτόν ή να αλλάξει την κατάσταση του αιτήματος.

Ανάλογα με τις επιλογές, η βάση δεδομένων ανανεώνεται αυτόματα.

x

### Edit Request

**Request Details**

**User:**

**Type:** TREE

**Tree ID:** ΛΙΓΟΥΣΤΡΟ

**Description:**

**Status**

Pending

**Admin Reply**

*Enter your response to this request*

SAVE CHANGES

CANCEL

Ο διαχειριστής, απαντάει στον χρήστη και ενημερώνει την κατάσταση σε 'Open'.

#### Request Management

User	Type	Status	Description	Created	Opened	Resolved	Actions
admin	TREE	OPEN	The scientific name is badly formatted	2/3/2025	2/3/2025	-	EDIT

Στην περίπτωση που ένα αίτημα είναι σε 'Open' κατάσταση, μπορεί ο διαχειριστής στην συνέχεια να το καταχωρήσει ως 'Closed', ενημερώνοντας όλα τα σχετικά πεδία στην βάση δεδομένων.

^

**Edit Request**

**Request Details**

**User:**

**Type:** TREE

**Tree ID:** ΛΙΓΟΥΣΤΡΟ

**Description:**

**Status**

Closed

**Admin Reply**

We are looking over it. Data are correctly indicated

**SAVE CHANGES** **CANCEL**

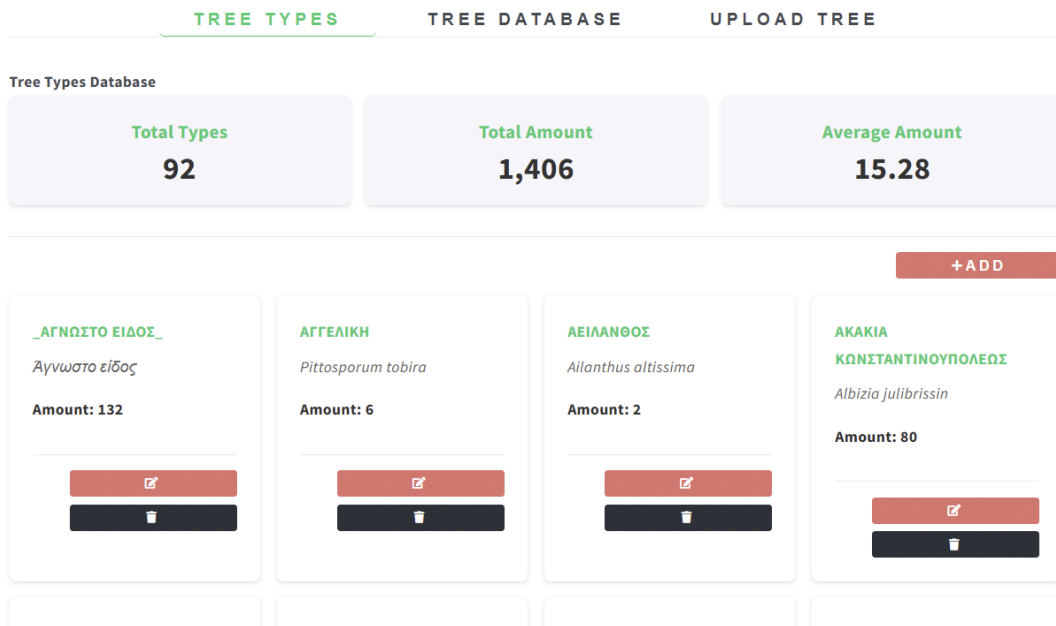
**Request Management**

User	Type	Status	Description	Created	Opened	Resolved	Actions
admin	TREE	CLOSED	The scientific name is badly formatted	2/3/2025	2/3/2025	2/3/2025	<b>EDIT</b>

## Οθόνη Βάσης Δέντρων

Η οθόνη αυτή είναι ίδια με αυτή που βλέπει ο χρήστης για την παρουσίαση των (1) τύπων των δέντρων, (2) των δέντρων και (3) το ανέβασμα δέντρου. Η διαφοροποίηση εδώ αφορά ότι ο διαχειριστής έχει την δυνατότητα να διαγράψει και να επεξεργαστεί τα δεδομένα που υπάρχουν ήδη στην βάση. Επομένως, στο σημείο αυτό οι επιπλέον δυνατότητες το διαχειριστή αφορούν τα δύο επιπλέον κουμπιά σε κάθε εγγραφή για επεξεργασία και διαγραφή, καθώς και το κουμπί εισαγωγής νέου είδους δέντρου.

Kata



Για την εισαγωγή νέου τύπου ζητούνται τα απαραίτητα στοιχεία όπως το ελληνικό και το επιστημονικό όνομα.

×

**Add New Tree Type**

**Greek Name**

**Scientific Name**

ADD TREE TYPE
CANCEL

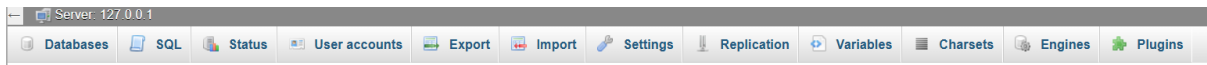
## Δομή Παραδοτέων

Στα παραδοτέα της πτυχιακής δίνονται τα εξής:

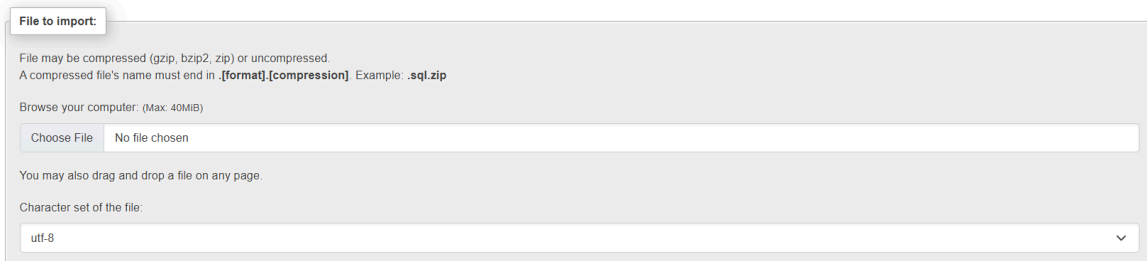
- **Database:** Περιέχει τον κώδικα για την εγκατάσταση της βάσης δεδομένων μαζί με τα αρχικά δεδομένα για λόγους παρουσίασης. Η εξαγωγή έχει πραγματοποιηθεί μέσω του εργαλείου XAMPP.
- **Python:** Περιέχει τον κώδικα Python για την εξαγωγή των δεδομένων από τα αρχικά ακατέργαστα αρχεία, καθώς και τα ίδια τα αρχεία. Ο κώδικας python θα εξαγάγει τόσο τα δεδομένα σε νέα αρχεία, αλλά επίσης θα προχωρήσει και την εισαγωγή τους στην βάση δεδομένων.
- **tree\_web:** Αφορά τα αρχεία για το html,css,javascript και php τα οποία δημιουργούν το μπροστά μέρος της ιστοσελίδας και βοηθούν στην διάδραση του χρήστη με αυτή και το back-end του συστήματος.

Τα βήματα εγκατάστασης αφορούν:

1. Εγκατάσταση του XAMPP (<https://www.apachefriends.org/download.html>) ή ανάλογου εργαλείου
2. Δημιουργία της βάσης δεδομένων μέσα από το προφίλ χρήσης.
  - a. Επιλέγουμε από το πάνω Menu την επιλογή ' Import' \



- b. Επιλέγουμε το αρχείο tree\_db



- c. Περιμένουμε να γίνει η εισαγωγή της βάσης και των δεδομένων
3. Εφόσον έχουμε κατασκευάσει την βάση και εισάγει τα δεδομένα, μεταφέρουμε τον φάκελο tree\_web στο C:\xampp\htdocs\
  4. Μπαίνουμε στο περιβάλλον του ιστότοπου από το url:  
http://localhost/tree\_web/index.html

Για την ευκολία χρήσης έχουν κατασκευαστεί οι ακόλουθοι λογαριασμοί:

Χρήστης:

username: plainuser

pass: pass

Διαχειριστής

username: admin

password: admin

## Συμπεράσματα & Μελλοντικές Επεκτάσεις

### Συμπεράσματα

Η παρούσα ιστοσελίδα αποτελεί ένα σύγχρονο και χρήσιμο εργαλείο για την απεικόνιση περιβαλλοντικών δεδομένων και τη χαρτογράφηση της αστικής βλάστησης στον Δήμο Θεσσαλονίκης. Μέσα από την πολυεπίπεδη αρχιτεκτονική της, η εφαρμογή καταφέρνει να συνδυάζει αποτελεσματικά την επεξεργασία ακατέργαστων δεδομένων, τη φιλική προς τον χρήστη διεπαφή και την ασφαλή αποθήκευση πληροφορίας.

Συγκεκριμένα:

- Οι **Python κώδικες** επιτρέπουν την αξιόπιστη εισαγωγή και προτυποποίηση των δεδομένων, κάνοντάς τα κατάλληλα προς παρουσίαση και ανάλυση.

- Η **διεπαφή HTML/PHP** προσφέρει έναν διαδραστικό τρόπο αλληλεπίδρασης με τα δεδομένα, ενισχύοντας την εμπειρία του χρήστη.
- Η **βάση δεδομένων SQL** διασφαλίζει την ακεραιότητα και τη διαρκή προσβασιμότητα των πληροφοριών.

Η υποστήριξη δύο επιπέδων χρηστών – βασικού χρήστη και διαχειριστή – επιτρέπει την ασφαλή και στοχευμένη λειτουργία του συστήματος. Ο διαχειριστής διαθέτει εργαλεία για την παρακολούθηση και διαχείριση της πλατφόρμας, εξασφαλίζοντας την αξιοπιστία και την σωστή παρουσίαση του περιεχομένου.

Ένα ιδιαίτερα καινοτόμο χαρακτηριστικό της εφαρμογής είναι η δυνατότητα συμμετοχής των πολιτών μέσω της εισαγωγής προσωπικών παρατηρήσεων ή δεδομένων. Η προσέγγιση αυτή ενισχύει την έννοια της *συλλογικής αντίληψης (collective perception)*, ενθαρρύνοντας την ενεργή συμβολή της κοινωνίας στην περιβαλλοντική καταγραφή και ενίσχυση της αστικής πληροφορίας.

Συνολικά, η εφαρμογή προσφέρει:

- Έναν οπτικοποιημένο τρόπο παρουσίασης της περιβαλλοντικής πληροφορίας, διευκολύνοντας την κατανόηση και την αξιοποίησή της από αρμόδιες αρχές, πολίτες και ερευνητές.
- Ένα ανοιχτό περιβάλλον εξέλιξης, όπου μπορούν να προστεθούν νέες λειτουργικότητες, όπως προβλέψεις ρύπανσης, ανάλυση χρονικών τάσεων ή χαρτογράφηση με επιπλέον φίλτρα.
- Ένα πολύτιμο εργαλείο για την περιβαλλοντική πολιτική, καθώς διευκολύνει τη λήψη αποφάσεων με γνώμονα την πραγματική εικόνα της πόλης.

Η επιτυχία αυτής της πρωτοβουλίας υπογραμμίζει τη σημασία της ψηφιακής καινοτομίας στον τομέα του περιβάλλοντος και θέτει τις βάσεις για μελλοντικές βελτιώσεις και επεκτάσεις.

## Προκλήσεις & Περιορισμοί

Κατά τη διαδικασία υλοποίησης της εφαρμογής παρουσιάστηκαν ορισμένες σημαντικές προκλήσεις και περιορισμοί που επηρέασαν τόσο την κατασκευή όσο και τη λειτουργικότητά της.

Μία από τις σημαντικότερες προκλήσεις εντοπίστηκε στα ίδια τα αρχικά ακατέργαστα δεδομένα. Τα δεδομένα προέρχονταν από διαφορετικούς σταθμούς μέτρησης και αρμόδιους φορείς, γεγονός που οδήγησε σε μεγάλη ετερογένεια. Συγκεκριμένα, παρατηρήθηκε ότι:

- Οι σταθμοί κατέγραφαν διαφορετικές μετρικές (π.χ. CO<sub>2</sub>, NO<sub>2</sub>, κ.ά.),
- Οι μετρήσεις αφορούσαν διαφορετικές χρονικές περιόδους και συχνότητες,

- Πολλά από τα αρχεία περιείχαν ελλιπή ή ελλείποντα δεδομένα.

Η ανομοιογένεια αυτή δημιούργησε σημαντικές δυσκολίες τόσο στην εισαγωγή των δεδομένων στη βάση όσο και στη σχεδίαση του σχήματος της βάσης, ώστε να είναι σε θέση να υποστηρίξει ποικίλα και ασύμμετρα σύνολα πληροφορίας.

Επιπλέον, προκλήσεις παρουσιάστηκαν και στον συνδυασμό των δεδομένων, ιδιαίτερα όταν πρόκειται για αντιστοιχίες δέντρων και τύπων δέντρων με συγκεκριμένες περιοχές και χρονικές στιγμές. Η συχνή απουσία τιμών δυσχεραίνει την αξιόπιστη απεικόνιση και την εξαγωγή ασφαλών συμπερασμάτων.

Ο Python κώδικας, ύστερα από αναλυτική μελέτη και πολλαπλές δοκιμές, κατάφερε σε μεγάλο βαθμό να αντιμετωπίσει αυτές τις διαφοροποιήσεις. Η χρήση της βιβλιοθήκης Pandas αποδείχθηκε καθοριστική στην ομογενοποίηση των δεδομένων, την αφαίρεση κενών τιμών και τη δομική κατηγοριοποίηση των εγγραφών.

Ωστόσο, παρά τις τεχνικές βελτιστοποίησης, η έλλειψη πληροφορίας εξακολούθησε να αποτελεί περιοριστικό παράγοντα τόσο για την πληρότητα της παρουσίασης όσο και για την εγκυρότητα των στατιστικών ή χρονικών αναλύσεων. Οι μεγάλοι χρονικοί «θόρυβοι» σε συνδυασμό με τα κενά δεδομένων είχαν ως αποτέλεσμα, σε ορισμένες περιπτώσεις, παραμορφωμένη ή αποσπασματική απεικόνιση των αποτελεσμάτων.

Παρόλα αυτά, η εφαρμογή καταφέρνει να παρουσιάσει με σαφήνεια τα διαθέσιμα δεδομένα και να προσφέρει μία χρήσιμη βάση για μελλοντική επέκταση και εμπλουτισμό της βάσης δεδομένων με πιο πλήρη και συνεκτικά σύνολα πληροφορίας.

## Μελλοντικές Επεκτάσεις

Η υλοποιηθείσα εφαρμογή θέτει τις βάσεις για μία ευρύτερη και πιο ολοκληρωμένη πλατφόρμα διαχείρισης και απεικόνισης περιβαλλοντικών και αστικών δεδομένων. Στο πλαίσιο αυτό, διαμορφώνονται αρκετές κατευθύνσεις για μελλοντικές επεκτάσεις, με σκοπό την ενίσχυση της χρηστικότητας, της πληρότητας και της διαλειτουργικότητας του συστήματος.

### Υποστήριξη ανεβάσματος αρχείων (CSV/Excel)

Η ενσωμάτωση δυνατότητας ανεβάσματος αρχείων από τους χρήστες (π.χ. σε μορφή .csv ή .xlsx) θα επιτρέψει την περαιτέρω ενίσχυση της βάσης δεδομένων, αξιοποιώντας δεδομένα που συλλέγονται και από ανεξάρτητους φορείς ή ιδιώτες. Η λειτουργία αυτή μπορεί να συνοδεύεται από αυτόματους ελέγχους εγκυρότητας των δεδομένων (data validation), για την αποφυγή σφαλμάτων και την διασφάλιση ποιότητας. Μία διαφορετική προσέγγιση αφορά την κατασκευή ενός συγκεκριμένου προτύπου (template) για το ανέβασμα νέων δεδομένων.

### Επέκταση σε περισσότερους Δήμους

Η εφαρμογή μπορεί να επεκταθεί ώστε να καλύπτει περισσότερους δήμους σε εθνικό επίπεδο, δημιουργώντας έναν πανελλαδικό χάρτη με στοιχεία ρύπανσης και βλάστησης. Με αυτόν τον τρόπο, επιτυγχάνεται μία συγκεντρωτική εικόνα του περιβάλλοντος ανά περιοχή, και διευκολύνονται διαδημοτικές συγκρίσεις ή μελέτες. Για την υλοποίηση αυτού απαιτούνται επιπλέον στοιχεία από τους αρμόδιους φορείς.

## Εμπλουτισμός πληροφορίας και κάλυψη ελλείψεων

Μία ακόμη επέκταση αφορά την προσθήκη περισσότερων πεδίων πληροφορίας για τα δέντρα (π.χ. ηλικία, ύψος, κατάσταση, εποχικότητα) και για τη ρύπανση (π.χ. ταχύτητα ανέμου). Η συνεργασία με φορείς ή χρήση αισθητήρων IoT μπορεί να βοηθήσει στον συστηματικό και πλήρη εμπλουτισμό των μετρήσεων, περιορίζοντας τα σημερινά κενά. Η δημιουργία αυτής της λειτουργικότητας θα μπορέσει να προσφέρει πολύ σημαντικές πληροφορίες σε ζωντανό χρόνο από πολλά διαφορετικά διαμερίσματα και τοποθεσίες, ακόμα και από το εξωτερικό.

## Συστήματα ειδοποίησης και ανάλυσης

Η προσθήκη μηχανισμών ειδοποίησης για υψηλά επίπεδα ρύπανσης, καθώς και γραφικές αναπαραστάσεις δεδομένων με δυνατότητα φιλτραρίσματος ανά περιοχή, χρονικό διάστημα ή δείκτη, μπορούν να βελτιώσουν τη χρησιμότητα της εφαρμογής. Παράλληλα, μπορεί να υλοποιηθεί ένα σύστημα βασικής ανάλυσης δεδομένων, το οποίο να εξάγει συμπεράσματα ή τάσεις βάσει των τιμών που καταγράφονται. Η υλοποίηση αυτής της λειτουργικότητας είναι άμεσα συνδυσασμένη με την προηγούμενη επέκταση, καθώς η χρήση IoT συσκευών θα οδηγήσει σε άμεσες ενέργειες και εντοπισμό μετρήσεων που διαφεύγουν από τις καθιερωμένες μετρήσεις.

## Χάρτης με live δεδομένα και χρονοδιάγραμμα μεταβολών

Η χρήση διαδραστικών χαρτών με δυνατότητα αναπαραγωγής ιστορικών δεδομένων ("time slider") θα προσφέρει στους χρήστες τη δυνατότητα να παρακολουθούν τις μεταβολές ρύπανσης ή δέντρων μέσα στον χρόνο. Επιπλέον, η ενσωμάτωση live δεδομένων από πραγματικούς σταθμούς μέτρησης ή APIs τρίτων φορέων θα ενισχύσει τη δυναμικότητα της εφαρμογής. Η τωρινή κατάσταση αφορά τα δεδομένα τα οποία υπάρχουν εισαγμένα στην βάση από προϋπάρχοντες μετρήσεις. Η επέκταση αυτή θα βοηθήσει στην ακόμα πιο δυναμική και ζωντανή παρουσίαση της πληροφορίας.

## Εφαρμογή για κινητά

Η ανάπτυξη mobile εφαρμογής (Android/iOS) θα επιτρέψει την ευρύτερη χρήση της πλατφόρμας και την ενεργή συμμετοχή των πολιτών, οι οποίοι θα μπορούν να αναφέρουν σημεία με προβλήματα, να ανεβάζουν φωτογραφίες ή να εντοπίζουν κοντινά σημεία πρασίνου, όπως πάρκα, ή εντοπισμό συγκεκριμένων δένδρων, σε πραγματικό χρόνο.





Πεταλάς, Κωνσταντίνος Φ. “ΑΤΜΟΣΦΑΙΡΙΚΗ ΡΥΠΑΝΣΗ ΚΑΙ ΑΛΛΕΡΓΙΕΣ.” *DR Peletas*

*News*, 03 09 2021,

<https://www.drpetalas.gr/el/%CE%B1%CF%84%CE%BC%CE%BF%CF%83%CF%86%CE%B1%CE%B9%CF%81%CE%B9%CE%BA%CE%B7-%CF%81%CF%85%CF%80%CE%B1%CE%BD%CF%83%CE%B7-%CE%BA%CE%B1%CE%B9-%CE%B1%CE%BB%CE%BB%CE%B5%CF%81%CE%B3%CE%B9%CE%B5%CF%83>.

Accessed 2025.

Κωλέτση, Βαρβάρα. *Βιώσιμες πόλεις: Η καταπολέμηση του φαινομένου της αστικής θερμικής νησίδας μέσω πράσινων - μπλε υποδομών*. 2024. *Ιδρυματικό Αποθετήριο Πολυνόη*,

<https://polynoe.lib.uniwa.gr/xmlui/handle/11400/7282?locale-attribute=en>.

Τσίγκου, Κωνσταντίνα. *Περιβαλλοντική Διαχείριση*. 06 2022. *Πανεπιστήμιο Δυτικής*

*Μακεδονίας*,

<https://eclass.uowm.gr/modules/document/file.php/CHEMENG285/%CE%A0%CE%95%CE%A1%CE%99%CE%92.%20%CE%94%CE%99%CE%91%CE%A7%CE%95%CE%99%CE%A1%CE%99%CE%A3%CE%97-%20%CE%94%CE%B9%CE%AC%CE%BB%CE%B5%CE%BE%CE%B7%202.pdf>. Accessed 04 2025.

Τζωρτζάκης, Ιωάννης. *Εφαρμογές Γεωπληροφορικής στα Τεχνικά Έργα*. 2 ed., Creative

Commons, 2016,

[https://www.iep.edu.gr/images/IEP/EPISTIMONIKI\\_YPIRESIA/Epist\\_Monades/B\\_Kyklos/Tee/2016/GEpal/2016\\_GEpal\\_Geopliroriki\\_symplyliko.pdf](https://www.iep.edu.gr/images/IEP/EPISTIMONIKI_YPIRESIA/Epist_Monades/B_Kyklos/Tee/2016/GEpal/2016_GEpal_Geopliroriki_symplyliko.pdf).

Χαλκιάς, Χρίστος. *Τι είναι τα Συστήματα Γεωγραφικών Πληροφοριών*. 2020. *Χαροκόπειο*

*Πανεπιστήμιο*,

<https://eclass.hua.gr/modules/document/file.php/GEO174/GIS/%CE%A0%CE%91%CE%A1%CE%9F%CE%A5%CE%A3%CE%99%CE%91%CE%A3%CE%95%CE%99%CE%A3/%CE%A4%CE%B9%20%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9%20%CF%84%CE%B1%20GIS.pdf>. Accessed 20 04 2025.

Κόκκινος, Κωνσταντίνος. “Εφαρμογές Γεωπληροφορικής στο Περιβάλλον.” ΕΘΑΑΕ, 2024, <https://ds.uth.gr/6th-semester/environmental-geoinformatics-applications/>. Accessed 26 04 2025.

Σπανός, Κωνσταντίνος. *Ο σχεδιασμός ενός ολοκληρωμένου δικτύου αστικού και περιαστικού πράσινου*. 2024. *Ερευνα & Καινοτομία*, [https://www.elgo.gr/images/pdf/publications/demeter\\_magazine/dmtr7p23-25.pdf](https://www.elgo.gr/images/pdf/publications/demeter_magazine/dmtr7p23-25.pdf).

Παγωνάκης, Αντώνιος. *Διερεύνηση μεθόδων διαμεταγωγής γεωχωρικών δεδομένων νέφους σημείων σε λογισμικά GIS*. Διπλωματική Εργασία. 2025. ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ, <https://polynoe.lib.uniwa.gr/xmlui/bitstream/handle/11400/8475/%CE%94%CE%B9%CE%B5%CF%81%CE%B5%CF%8D%CE%BD%CE%B7%CF%83%CE%B7%20%CE%BC%CE%B5%CE%B8%CF%8C%CE%B4%CF%89%CE%BD%20%CE%B4%CE%B9%CE%B1%CE%BC%CE%B5%CF%84%CE%B1%CE%B3%CF%89%CE%B3%CE%AE%CF%82%20%CE%B3%CE%>.

Ποζουκίδου, Γ, and Α Τασσπούλου. *ΓΕΩΧΩΡΙΚΑ ΔΕΔΟΜΕΝΑ. Η ΣΧΕΣΗ ΜΕ ΤΟ ΣΧΕΔΙΑΣΜΟ ΚΑΙ ΤΗΝ ΑΝΑΠΤΥΞΗ*. 1 12 2017. *Τεχνικό Επιμηλητήριο Ελλάδας*, [https://tkm.tee.gr/wp-content/uploads/2018/01/Gewxwrika\\_Dedomena.pdf](https://tkm.tee.gr/wp-content/uploads/2018/01/Gewxwrika_Dedomena.pdf). Accessed 25 04 2025.

Ανδρώνης, Β, and Β Καραθανάση. *Χαρτογράφηση Πυκνότητας Βλάστησης με Χρήση Υπερφασματικών Απεικονίσεων*. 2018, [https://mirc.ntua.gr/foundation/8th\\_conference/ergasies/47\\_ANDRONIS\\_BASILHS\\_KARAUANASH\\_BASILEIA.pdf](https://mirc.ntua.gr/foundation/8th_conference/ergasies/47_ANDRONIS_BASILHS_KARAUANASH_BASILEIA.pdf).

AmsterdamCity. “Amsterdam Smart Citizens Lab.” *Waag Society Project*, 2016, <https://amsterdamsmartcity.com/updates/project/amsterdam-smart-citizens-lab-39010h7g>. Accessed 21 04 2025.

Barcelona. “Smart City | Info Barcelona | Barcelona City Council.” *Ajuntament de Barcelona*, <https://www.barcelona.cat/infobarcelona/en/tema/smart-city>. Accessed 25 April 2025.

- Caterino, Eugenio. "What is Environmental Data? Examples, Types, and Databases." *Datarade*, 2025, <https://datarade.ai/data-categories/environmental-data>. Accessed 26 April 2025.
- CityGreen. "The Ultimate Guide On How To Plant Urban Trees." *Citygreen*, 2 August 2023, <https://citygreen.com/ultimate-guide-on-how-to-plant-trees-in-urban-areas/>. Accessed 25 April 2025.
- CurieJet. "Understanding PM2.5 Sensors." *Curie Jet*, 2025, <https://www.curiejet.com/en/blogs/detail/understanding-pm25-sensors>. Accessed 28 April 2025.
- Dasarxeio. *Το Περιαστικό Δάσος της Θεσσαλονίκης: Η Ιστορία του, το φυσικό περιβάλλον και η διαχείρισή του*. 2023. *Dasarxeio*, <https://dasarxeio.com/2023/01/25/121797/>. Accessed 2025.
- EPA. "Integrated Urban Air Toxics Strategy | US EPA." *Environmental Protection Agency*, 6 November 2024, <https://www.epa.gov/haps/integrated-urban-air-toxics-strategy>. Accessed 25 April 2025.
- Gajare, Tanaji S. "Ways to Reduce Air Pollution in Cities." *Perfect Pollucon Services*, <https://www.ppsthane.com/blog/ways-to-reduce-air-pollution-in-cities>. Accessed 25 April 2025.
- GeoForest. "Συστήματα Γεωγραφικών Πληροφοριών." *GSIS*, 2025, <https://geoforest.gr/efarmoges-gis/>. Accessed 03 05 2025.
- Geomeletiki. "Γεωγραφικά Συστήματα Πληροφοριών στην διαχείριση δεδομένων και λήψη αποφάσεων." *Γεωμελετητική ΕΤΕ*, 2022, <https://www.geomeletitiki.gr/3073-2>. Accessed 27 04 2025.
- GISGeography. "Latitude, Longitude and Coordinate System Grids - GIS Geography." *GISGeography*, 2024, <https://gisgeography.com/latitude-longitude-coordinates/>. Accessed 28 April 2025.
- Google. "Search by latitude & longitude in Google Maps - Computer - Google Maps Help." *Google Help*, 2025,



- Jain, Sandeep. "Difference between Relational database and NoSQL." *GeeksforGeeks*, 25 October 2022,  
<https://www.geeksforgeeks.org/difference-between-relational-database-and-nosql/>.  
Accessed 25 April 2025.
- Jain, Sandeep. "Difference Between REST API and RESTful API." *GeeksforGeeks*, 4 April 2025,  
<https://www.geeksforgeeks.org/know-the-difference-between-rest-api-and-restful-api/>.  
Accessed 28 April 2025.
- Jain, Sandeep. "Relationships in SQL – One-to-One, One-to-Many, Many-to-Many." *GeeksforGeeks*, 2 April 2024,  
<https://www.geeksforgeeks.org/relationships-in-sql-one-to-one-one-to-many-many-to-many/>. Accessed 28 April 2025.
- Kontonikas, M. "Η Αστική Θερμική Νησίδα στην Αθήνα." *Kontonikas + Associates*, 2019,  
<https://kontonikas.gr/2019/08/astiki-thermiki-nisida/>. Accessed 25 04 2025.
- LAMDA. "Βιώσιμη ανάπτυξη και αστικό τοπίο: Το μέλλον είναι πράσινο." *LAMDA Blog*, 2025,  
<https://www.lamdadev.com/blog/biosimi-anaptuksi-kai-astiko-topio-to-mellon-einai-prasino>. Accessed 25 04 2025.
- LatLong. "Latitude and Longitude Finder." *Latitude and Longitude Finder on Map Get Coordinates*, 2025, <https://www.latlong.net/>. Accessed 28 April 2025.
- Microsoft. "Video: Create many-to-many relationships." *Microsoft Support*, 2025,  
<https://support.microsoft.com/en-us/office/video-create-many-to-many-relationships-e65bcc53-8e1c-444a-b4fb-1c0b8c1f5653>. Accessed 28 April 2025.
- Mittermaier, Pascal. "Want to make your city healthier? Plant a tree." *World Economic Forum*, 2016,  
<https://www.weforum.org/stories/2016/11/want-to-make-your-city-healthier-plant-a-tree/>. Accessed 25 04 2025.

MongoDB. "Relational Vs. Non-Relational Databases | MongoDB." *MongoDB*,  
<https://www.mongodb.com/resources/compare/relational-vs-non-relational-databases>  
. Accessed 25 April 2025.

myEnv. "Get real-time environmental data via myENV." *myEnv*, 2025,  
<https://www.tech.gov.sg/products-and-services/for-citizens/environment/myenv/>.  
Accessed 24 04 2025.

NYC Trees. *NYC Tree Map*, <https://tree-map.nycgovparks.org/>. Accessed 25 April 2025.

PlumeLabs. *Plume Labs: Be empowered against air pollution*, <https://plumelabs.com/en/>.  
Accessed 25 April 2025.

Redhat. "What is a REST API?" *Red Hat*, 8 May 2020,  
<https://www.redhat.com/en/topics/api/what-is-a-rest-api>. Accessed 28 April 2025.

SES, Σύλλογος Ελλήνων Συγκοινωνιολόγων. *ΟΙ ΘΕΣΕΙΣ ΤΟΥ ΣΕΣ ΓΙΑ ΤΗΝ ΠΡΟΣΤΑΣΙΑ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΑΠΟ ΤΗΝ ΚΑΤΑΣΚΕΥΗ & ΛΕΙΤΟΥΡΓΙΑ ΣΥΓΚΟΙΝΩΝΙΑΚΩΝ ΕΡΓΩΝ*. 2011. ΣΕΣ, SES,  
<https://www.ses.gr/wp-content/uploads/2019/11/ses-2011-06-thesis-gia-perivalon-1.pdf>.

Singh, Ashbindu. *Environmental Databases Environmental Databases*. 2018. *UNEP Division of Early Warning and Assessment UNEP Division of Early Warning and Assessment-North America*,  
<https://sedac.ciesin.columbia.edu/GSDworkshop/Singh.pdf>.

Sowjanya, Pedada. "Τι είναι η γεωχωρική ανάλυση; Το σχέδιο πριν από το πραγματικό σχέδιο." *Mind the Graph*, 5 May 2023,  
<https://mindthegraph.com/blog/el/what-is-geospatial-analysis/>. Accessed 25 April 2025.

TheOpinion. "Το Περιαστικό Δάσος της Θεσσαλονίκης μέσα από τα μάτια των ειδικών: Ένα βιβλίο για την ιστορία και το φυσικό περιβάλλον της πόλης." *TheOpinion*, 2023,  
<https://menoumethess.wordpress.com/2023/01/30/%CF%84%CE%BF-%CF%80%CE%B5%CF%81%CE%B9%CE%B1%CF%83%CF%84%CE%B9%CE%BA%CF%8C>

-%CE%B4%CE%AC%CF%83%CE%BF%CF%82-%CF%84%CE%B7%CF%82-%CE%B8%CE%B5%CF%83%CF%83%CE%B1%CE%BB%CE%BF%CE%BD%CE%AF%CE%BA%CE%B7%CF%82/. Accessed 25 04 2025.

Treepedia. "Exploring the Green Canopy in cities around the world." *Treepedia*, 2025, <https://senseable.mit.edu/treepedia>. Accessed 25 04 2025.

Virgo, James. "Relationships in SQL - complete guide with examples." *Dittofi*, <https://www.dittofi.com/relationships-in-sql-complete-guide-with-examples/>. Accessed 28 April 2025.

Voigt, K. *NON-SPATIAL ENVIRONMENTAL DATA*. 2 ed., ENVIRONMENTAL SYSTEMS, 2018, <https://www.eolss.net/sample-chapters/c09/e4-20-02-02.pdf>.

Voria. "Πρασινίζουν γειτονιές στον δήμο Θεσσαλονίκης - Δενδροφυτεύσεις σε Νέα Παραλία και δυτική είσοδο." 2024, <https://www.voria.gr/article/prasinizoy-n-geitonies-ston-dimo-thessalonikis-dendrofytey-seis-se-nea-paralia-kai-dytiki>. Accessed 25 04 2025.

WaagFutureLab. "Smart Citizens Lab." *Waag Future Lab*, 2025, <https://waag.org/en/project/amsterdam-smart-citizens-lab/>. Accessed 21 04 2025.

White Oak. "Enhancing Cities With Tree Planting Initiatives." *White Oak Tree Care*, 2025, <https://whiteoaktreecare.net/enhancing-cities-tree-planting-initiatives/>. Accessed 25 04 2025.

WikiPedia. "Geographic coordinate system." *Wikipedia*, 2024, [https://en.wikipedia.org/wiki/Geographic\\_coordinate\\_system](https://en.wikipedia.org/wiki/Geographic_coordinate_system). Accessed 28 April 2025.

Wilderness. "ΓΙΑ ΝΑ ΓΝΩΡΙΣΟΥΜΕ ΚΑΛΥΤΕΡΑ ΤΟ ΣΕΙΧ ΣΟΥ." 2009, <https://forestsos.wordpress.com/2009/02/08/%CE%B3%CE%B9%CE%B1-%CE%BD%CE%B1-%CE%B3%CE%BD%CF%89%CF%81%CE%B9%CF%83%CE%BF%CF%85%CE%BC%CE%B5-%CE%BA%CE%B1%CE%BB%CF%85%CF%84%CE%B5%CF%81%CE%B1-%CF%84%CE%BF-%CF%83%CE%B5%CE%B9%CF%87-%CF%83%CE%BF%CF%85/>. Accessed 26 04 2025.

Xiaojiang, Li, et al. "Assessing street-level urban greenery using Google Street View and a modified green view index." *ELSEVIER*, vol. 14, no. Urban Forestry & Urban Greening, 2015, pp. 675-685. *ELSEVIER*,  
[https://senseable.mit.edu/treepedia/treepedia\\_publication.pdf](https://senseable.mit.edu/treepedia/treepedia_publication.pdf). Accessed 2025.