

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη Web-GIS Συστήματος  
για τη Χωροθέτηση και την Οπτικοποίηση  
Εγκαταστάσεων Ανανεώσιμων Πηγών Ενέργειας»



Του φοιτητή  
Κωνσταντίνου  
Ταντανάση-Χαραλαμπίδη  
Αρ. Μητρώου: iee2019162

Επιβλέπων:  
Ιορδάνης Κιοσκερίδης

Ημερομηνία: 6/12/2025

Τίτλος Δ.Ε. «Ανάπτυξη Web-GIS Συστήματος  
για τη Χωροθέτηση και την Οπτικοποίηση  
Εγκαταστάσεων Ανανεώσιμων Πηγών Ενέργειας»  
Κωδικός Δ.Ε. 25292

Ονοματεπώνυμο φοιτητή: Κωνσταντίνος Ταντανάσης-Χαραλαμπίδης

Ονοματεπώνυμο εισηγητή: Ιορδάνης Κιοσκερίδης

Ημερομηνία ανάληψης Δ.Ε.: 17/8/2025

Ημερομηνία περάτωσης Δ.Ε.: 6/12/2025

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Κωνσταντίνου Ταντανάση-Χαραλαμπίδη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

## Πρόλογος

Η επιλογή της παρούσας διπλωματικής εργασίας προέκυψε από το έντονο ενδιαφέρον μου για τις τεχνολογίες Web-GIS και τον ρόλο τους στη βιώσιμη ενεργειακή μετάβαση. Η ραγδαία ανάπτυξη των Ανανεώσιμων Πηγών Ενέργειας και η ανάγκη για ορθολογικό σχεδιασμό των εγκαταστάσεών τους καθιστούν απαραίτητη την ύπαρξη εργαλείων που συνδυάζουν γεωχωρική ανάλυση και σύγχρονες διαδικτυακές τεχνολογίες. Η εργασία αυτή μου έδωσε τη δυνατότητα να εμβαθύνω τόσο στις τεχνικές πτυχές της ανάπτυξης ενός ολοκληρωμένου Web-GIS συστήματος όσο και στη θεωρητική θεμελίωση που διέπει τη διαχείριση και αξιοποίηση γεωχωρικών δεδομένων.

Μέσα από τη διαδικασία σχεδιασμού, υλοποίησης και αξιολόγησης της εφαρμογής απέκτησα σημαντική εμπειρία σε τεχνολογίες όπως React, TypeScript, Leaflet, PostgreSQL/PostGIS και Supabase, αλλά και σε ζητήματα όπως η αρχιτεκτονική client-server, η διαλειτουργικότητα γεωχωρικών υπηρεσιών και η χαρτογραφική οπτικοποίηση. Η εργασία συνέβαλε ουσιαστικά στην κατανόηση των απαιτήσεων και των προκλήσεων που συνοδεύουν την ανάπτυξη ενός λειτουργικού συστήματος χαρτογράφησης για τις ΑΠΕ, εμπλουτίζοντας τις γνώσεις και τις δεξιότητές μου στον τομέα των γεωπληροφορικών εφαρμογών.

## Περίληψη

Η παρούσα πτυχιακή εργασία επικεντρώνεται στην ανάπτυξη και αξιολόγηση ενός Web-GIS συστήματος για τη χωροθέτηση και οπτικοποίηση εγκαταστάσεων Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ). Η εφαρμογή βασίστηκε σε σύγχρονες τεχνολογίες ανοιχτού κώδικα, όπως React, TypeScript, Vite και Leaflet για τη χαρτογραφική απεικόνιση, καθώς και σε PostgreSQL/PostGIS σε συνδυασμό με το Supabase για τη διαχείριση των γεωχωρικών δεδομένων. Η υλοποίηση σε επίπεδο PoC (Proof of Concept) περιλαμβάνει διαδραστική χαρτογραφική διεπαφή, δυνατότητα εισαγωγής νέων εγκαταστάσεων, φιλτράρισμα ανά περιφέρεια και άμεση απεικόνιση των δεδομένων μέσω GeoJSON. Η εργασία καταδεικνύει την αποτελεσματικότητα της ενσωμάτωσης τεχνολογιών Web και GIS για την υποστήριξη αποφάσεων που σχετίζονται με την ενεργειακή αξιοποίηση και τη βιώσιμη ανάπτυξη και τα βήματα που προϋποθέτουν την ανάπτυξη μιας λειτουργικής τέτοιας WEB/GIS εφαρμογής.

Λέξεις-κλειδιά: Web-GIS, ΑΠΕ, React, Leaflet, PostgreSQL, PostGIS, Supabase, γεωχωρικά δεδομένα

# «Development of a Web-GIS System for the Siting and Visualization of Renewable Energy Installations»

«Konstantinos Tantanasis-Charalampidis»

## **Abstract**

This undergraduate thesis focuses on the development and evaluation of a Web-GIS system designed for the spatial placement and visualization of Renewable Energy Installations (REI). The application leverages modern open-source technologies, including React, TypeScript, Vite, and Leaflet for web-based cartographic visualization, as well as PostgreSQL/PostGIS integrated with Supabase for spatial data management. The implementation at the PoC (Proof of Concept) level offers an interactive mapping interface with features such as adding new installations, filtering by administrative region, and displaying real-time data using GeoJSON. The project demonstrates the effectiveness of combining Web and GIS technologies to support decision-making in energy utilization and sustainable development and the steps that are required for the development of such a functioning WEB/GIS application.

Keywords: Web-GIS, Renewable Energy, React, Leaflet, PostgreSQL, PostGIS, Supabase, spatial data

## Ευχαριστίες

Ολοκληρώνοντας αυτή τη διπλωματική εργασία, θα ήθελα να εκφράσω την ειλικρινή μου ευγνωμοσύνη σε όλους όσους συνέβαλαν ουσιαστικά στην προσπάθειά μου.

Πρώτα απ' όλα, οφείλω ιδιαίτερες ευχαριστίες στον επιβλέποντα καθηγητή μου, Δρ. Ιορδάνη Κιοσκερίδη, για την πολύτιμη καθοδήγηση, τη συνεχή υποστήριξη και τις εύστοχες παρατηρήσεις του καθ' όλη τη διάρκεια της ερευνητικής διαδικασίας. Η συμβολή του υπήρξε σημαντική για την πρόοδο και την επιτυχή ολοκλήρωση της εργασίας μου.

Θα ήθελα επίσης να ευχαριστήσω τα μέλη της εξεταστικής επιτροπής, καθώς και τους/τις συναδέλφους μου, για τις γόνιμες ανταλλαγές απόψεων, τις προτάσεις και τις ιδέες που ενίσχυσαν ουσιαστικά την ανάπτυξη της παρούσας μελέτης.

Τέλος, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένειά μου για την αδιάκοπη στήριξη, την κατανόηση και την ενθάρρυνση που μου προσέφεραν σε κάθε στάδιο αυτής της πορείας.

Με εκτίμηση και ευγνωμοσύνη,

**Κωνσταντίνος Ταντανάσης-Χαραλαμπίδης**

# Περιεχόμενα

|                                                                             |      |
|-----------------------------------------------------------------------------|------|
| Πρόλογος .....                                                              | iii  |
| Περίληψη .....                                                              | iv   |
| Abstract .....                                                              | v    |
| Ευχαριστίες .....                                                           | vi   |
| Κατάλογος Σχημάτων και Εικόνων .....                                        | ix   |
| Κατάλογος Πινάκων .....                                                     | ix   |
| Συνοτομογραφίες .....                                                       | x    |
| Εισαγωγή .....                                                              | xi   |
| Δομή της εργασίας .....                                                     | xiii |
| 1. Βιβλιογραφική Ανασκόπηση – Θεωρητική Τεκμηρίωση .....                    | 1    |
| 1.1. GIS και web εφαρμογές στην ενέργεια.....                               | 1    |
| 1.2. GIS frameworks (Leaflet.js, OpenLayers, Google Maps API). .....        | 4    |
| 1.3. Βάσεις δεδομένων χωρικών δεδομένων (PostGIS, MongoDB). .....           | 7    |
| 1.4. Client-server αρχιτεκτονικές για geospatial apps .....                 | 10   |
| 1.5. Παραδείγματα από υπάρχουσες εφαρμογές Web-GIS σε ΑΠΕ. ....             | 12   |
| 1.6. Μελέτη Περίπτωσης: Geoportal του γεωπληροφοριακού χάρτη της ΠΑΑΕΥ..... | 16   |
| 1.7. Πρότυπα και ρυθμιστικές αρχές για γεωχωρικά δεδομένα .....             | 18   |
| 1.8. Τα πρότυπα OGC και η σημασία τους στα Web-GIS Συστήματα .....          | 20   |
| 1.9. Συμπεράσματα Βιβλιογραφικής Ανασκόπησης.....                           | 22   |
| 2. Ανάπτυξη της GIS γεωπληροφοριακής εφαρμογής.....                         | 24   |
| 2.1. Ανάπτυξη της εφαρμογής ως PoC (Proof of Concept).....                  | 24   |
| 2.2. Τεχνολογίες, εργαλεία και υπηρεσίες που χρησιμοποιούνται.....          | 26   |
| 2.3. Διαδικασία ανάπτυξης της εφαρμογής.....                                | 31   |
| 2.4. Οι πίνακες και τα views στην βάση δεδομένων της εφαρμογής.....         | 38   |
| 2.5. Περιγραφή και δημιουργία του schema της Βάσης Δεδομένων .....          | 49   |
| 2.6. Περιγραφή της λειτουργικότητας των βασικών αρχείων της εφαρμογής ..... | 52   |
| 2.7. Η εμφάνιση και η λειτουργικότητα της αναρτημένης εφαρμογής .....       | 58   |
| 3. Συμπεράσματα – Περιορισμοί – Μελλοντικές προεκτάσεις.....                | 62   |
| 3.1. Συμπεράσματα.....                                                      | 62   |
| 3.2. Περιορισμοί .....                                                      | 63   |
| 3.3. Μελλοντικές προεκτάσεις.....                                           | 64   |
| Βιβλιογραφία .....                                                          | 66   |
| ΠΑΡΑΡΤΗΜΑ Α:Παράθεση SQL με SCHEMA της Β.Δ. ....                            | 71   |

|                                                         |           |
|---------------------------------------------------------|-----------|
| <b>ΠΑΡΑΡΤΗΜΑ Β: Παράθεση κώδικα της εφαρμογής .....</b> | <b>73</b> |
|---------------------------------------------------------|-----------|

## Κατάλογος Σχημάτων και Εικόνων

|                                                                                                                    |    |
|--------------------------------------------------------------------------------------------------------------------|----|
| Εικόνα 1.1. Η ιστοσελίδα του Geoportal της ΠΑΑΕΥ.....                                                              | 16 |
| Σχήμα 2.1. Οπτικοποιημένη περιγραφή των διασυνδέσεων των συστατικών της εφαρμογής.....                             | 34 |
| Σχήμα 2.2. Η ροή της ανάπτυξης της εφαρμογής .....                                                                 | 35 |
| Εικόνα 2.1. Η δενδρική απεικόνιση των βασικών αρχείων της εφαρμογής.....                                           | 52 |
| Εικόνα 2.2. Τα μέρη στην αρχική σελίδα της εφαρμογής.....                                                          | 58 |
| Εικόνα 2.3. Drop-down φίλτρου Περιφέρειας και εμφάνιση πληροφοριών σε πολύγωνο<br>καταχωρημένης εγκατάστασης ..... | 58 |
| Εικόνα 2.4. Το παράθυρο εισαγωγής των δεδομένων της Νέας Εγκατάστασης .....                                        | 59 |
| Εικόνα 2.5. Το πολύγωνο και τα δεδομένα της Νέας Εγκατάστασης εμφανιζόμενα πάνω στον χάρτη<br>της εφαρμογής.....   | 59 |

## Κατάλογος Πινάκων

|                                                           |    |
|-----------------------------------------------------------|----|
| Πίνακας 2.1. Πίνακας areas (base table) .....             | 38 |
| Πίνακας 2.2. Πίνακας installations (base table) .....     | 40 |
| Πίνακας 2.3. Πίνακας spatial_ref_sys (base table).....    | 42 |
| Πίνακας 2.4. Πίνακας geography_columns (view).....        | 43 |
| Πίνακας 2.5. Πίνακας geometry_columns (view) .....        | 45 |
| Πίνακας 2.6. Πίνακας v_areas_geojson (view) .....         | 47 |
| Πίνακας 2.7. Πίνακας v_installations_geojson (view) ..... | 49 |

## Συντομογραφίες

|            |                                                 |
|------------|-------------------------------------------------|
| Δ.Ε.       | Διπλωματική Εργασία                             |
| ΔΙ.ΠΑ.Ε.   | Διεθνές Πανεπιστήμιο Ελλάδος                    |
| Π.Ε.       | Πτυχιακή Εργασία                                |
| GIS        | Geographical Information Systems                |
| Α.Π.Ε.     | Ανανεώσιμες Πηγές Ενέργειας                     |
| P.A.A.E.Y. | Ρυθμιστική Αρχή Αποβλήτων, Ενέργειας και Υδάτων |
| P.A.E.     | Ρυθμιστική Αρχή Ενέργειας                       |
| JSON       | JavaScript Object Notation                      |
| GeoJSON    | Geographical JavaScript Object Notation         |
| UUID       | Universally Unique Identifier                   |
| SRID       | Spatial Reference ID                            |
| Β.Δ.       | Βάση Δεδομένων                                  |
| SQL        | Structured Query Language                       |
| OGC        | Open Geospatial Consortium                      |
| WMS        | Web Map Service                                 |
| WFS        | Web Feature Service                             |
| WCS        | Web Coverage Service                            |
| EPSG       | European Petroleum Survey Group                 |
| WKT        | Well-Known Text                                 |
| PoC        | Proof of Concept                                |
| PostGIS    | PostgreSQL Geographical Information System      |

## Εισαγωγή

Η μετάβαση προς μια πιο βιώσιμη ενεργειακή πραγματικότητα αποτελεί στρατηγική προτεραιότητα για τις σύγχρονες κοινωνίες. Οι Ανανεώσιμες Πηγές Ενέργειας (ΑΠΕ) βρίσκονται στο επίκεντρο αυτής της αλλαγής, καθώς συμβάλλουν στη μείωση της εξάρτησης από τα ορυκτά καύσιμα και στον περιορισμό των εκπομπών αερίων του θερμοκηπίου. Παράλληλα, η αυξανόμενη ζήτηση για ενέργεια καθιστά επιτακτική την ανάγκη για βέλτιστη αξιοποίηση των φυσικών πόρων, μέσα από την ορθή χωροθέτηση των ενεργειακών εγκαταστάσεων.

Η διαδικασία της χωροθέτησης των ΑΠΕ δεν είναι απλή. Απαιτεί συνδυασμό επιστημονικών, τεχνικών, περιβαλλοντικών και κοινωνικών παραμέτρων. Η επιλογή των κατάλληλων τοποθεσιών πρέπει να λαμβάνει υπόψη παράγοντες όπως την διαθεσιμότητα πόρων (ηλιακή ακτινοβολία, άνεμος, υδατικοί πόροι), την γεωμορφολογία του εδάφους, την απόσταση από δίκτυα μεταφοράς, αλλά και ενδεχόμενους περιβαλλοντικούς περιορισμούς ή οι τοπικές κοινωνικές ευαισθησίες.

Σε αυτό το πλαίσιο, τα Γεωγραφικά Συστήματα Πληροφοριών – Geographical Information Systems (GIS) προσφέρουν ένα ισχυρό εργαλείο για την ανάλυση και την υποστήριξη λήψης αποφάσεων. Μέσω της δυνατότητας συλλογής, επεξεργασίας και οπτικοποίησης γεωχωρικών δεδομένων, οι τεχνολογίες GIS διευκολύνουν την πολύπλευρη αξιολόγηση των πιθανών περιοχών εγκατάστασης ΑΠΕ. Η ενσωμάτωση αυτών των δεδομένων σε διαδικτυακά περιβάλλοντα (Web-GIS) ενισχύει την προσβασιμότητα και την αλληλεπίδραση χρηστών με τις διαθέσιμες πληροφορίες.

Οι σύγχρονες διαδικτυακές τεχνολογίες, όπως τα web frameworks και οι βιβλιοθήκες Leaflet.js, OpenLayers και Google Maps API, επιτρέπουν τη δυναμική απεικόνιση γεωχωρικών δεδομένων στο διαδίκτυο. Σε συνδυασμό με βάσεις δεδομένων που υποστηρίζουν χωρικά ερωτήματα (π.χ. PostGIS, MongoDB), καθίσταται εφικτή η ανάπτυξη εφαρμογών που δεν περιορίζονται μόνο στην παρουσίαση στατικών χαρτών, αλλά επεκτείνονται σε διαδραστικές λειτουργίες, όπως φιλτράρισμα, ανάλυση και χωρική αναζήτηση.

Η σημασία τέτοιων εργαλείων είναι διπλή. Από τη μια πλευρά, υποστηρίζουν τους φορείς λήψης αποφάσεων, προσφέροντας μια αντικειμενική βάση δεδομένων και μεθόδων για την αξιολόγηση εναλλακτικών σεναρίων. Από την άλλη, συμβάλλουν στη διαφάνεια και την

ενημέρωση του κοινού, καθώς επιτρέπουν στους πολίτες να κατανοήσουν καλύτερα τις παραμέτρους και τα κριτήρια που σχετίζονται με τη χωροθέτηση ενεργειακών έργων.

Η ανάπτυξη Web-GIS εφαρμογών για ΑΠΕ εντάσσεται επίσης στο ευρύτερο πλαίσιο της «έξυπνης ενέργειας» και των «έξυπνων πόλεων», όπου η τεχνολογία διαδραματίζει ρόλο-κλειδί στην αποδοτική διαχείριση πόρων και στην προώθηση της βιώσιμης ανάπτυξης. Η χρήση ανοικτών δεδομένων (open datasets) και η αξιοποίηση cloud τεχνολογιών μπορούν να ενισχύσουν περαιτέρω την επεκτασιμότητα και τη χρηστικότητα τέτοιων εφαρμογών.

Στο πεδίο της ακαδημαϊκής έρευνας, παρατηρείται ολοένα και μεγαλύτερο ενδιαφέρον για την ανάπτυξη πρωτοτύπων και πιλοτικών Web-GIS εφαρμογών που συνδυάζουν την επιστημονική γνώση με τη σύγχρονη τεχνολογία λογισμικού. Τέτοιες εφαρμογές αποτελούν πολύτιμα παραδείγματα για το πώς η πληροφορική μπορεί να υπηρετήσει την ενεργειακή μετάβαση, ενισχύοντας τη διατομεακή συνεργασία ανάμεσα σε μηχανικούς, επιστήμονες περιβάλλοντος και φορείς χάραξης πολιτικής.

Η παρούσα εργασία εντάσσεται σε αυτό το πλαίσιο, επιδιώκοντας να μελετήσει και να υλοποιήσει μια διαδικτυακή γεωχωρική εφαρμογή για τη χωροθέτηση και την απεικόνιση εγκαταστάσεων ΑΠΕ. Μέσα από τη μελέτη υπάρχοντων εργαλείων και τεχνολογιών, τη σχεδίαση κατάλληλης βάσης δεδομένων και την ανάπτυξη frontend και backend υποδομών, στοχεύει να αναδείξει τις δυνατότητες αλλά και τις προκλήσεις τέτοιων λύσεων.

Το ερευνητικό ερώτημα που τίθεται είναι: Πώς μπορεί να αναπτυχθεί μια ολοκληρωμένη Web-GIS εφαρμογή που θα υποστηρίζει τη χωροθέτηση και την απεικόνιση εγκαταστάσεων Ανανεώσιμων Πηγών Ενέργειας, αξιοποιώντας σύγχρονες τεχνολογίες διαδικτύου και γεωχωρικών δεδομένων;

Στόχος της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη και αξιολόγηση ενός Web-GIS συστήματος που θα επιτρέπει τη χωροθέτηση και την οπτικοποίηση εγκαταστάσεων ΑΠΕ, αξιοποιώντας τεχνολογίες React, TypeScript, Vite, Leaflet για τη χαρτογραφική απεικόνιση, και PostgreSQL/PostGIS σε συνδυασμό με Supabase για την αποθήκευση και διαχείριση των γεωχωρικών δεδομένων. Μέσω της εφαρμογής θα είναι δυνατή η συλλογή, αποθήκευση και διαδραστική προβολή γεωχωρικών δεδομένων, με στόχο τη βελτιωμένη ανάλυση και παρουσίαση πληροφοριών που σχετίζονται με την ενεργειακή αξιοποίηση. Η ανάπτυξη της εφαρμογής στο βασίζεται στο github και netlify. Η εφαρμογή θα βασιστεί σε εικονικά συνθετικά και/ή ανοικτά δεδομένα (open datasets) για τις ανάγκες της ανάπτυξης και δοκιμής,

προκειμένου να διασφαλιστεί η ορθή λειτουργικότητα του συστήματος και η δυνατότητα αναπαραγωγής των αποτελεσμάτων.

## **Δομή της εργασίας**

Η παρούσα πτυχιακή εργασία με τίτλο «Ανάπτυξη και αξιολόγηση ενός Web-GIS συστήματος για χωροθέτηση και οπτικοποίηση εγκαταστάσεων ΑΠΕ» διαρθρώνεται σε τρία βασικά κεφάλαια, ακολουθώντας μια λογική και προοδευτική πορεία από τη θεωρητική θεμελίωση έως την τεχνική υλοποίηση και αξιολόγηση του συστήματος.

Στο Πρώτο Κεφάλαιο παρουσιάζεται το θεωρητικό και τεχνολογικό υπόβαθρο της έρευνας. Εξετάζονται οι βασικές αρχές των Γεωγραφικών Συστημάτων Πληροφοριών (GIS) και των Web-GIS εφαρμογών, η έννοια των Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ) και ο ρόλος των γεωχωρικών δεδομένων στη λήψη αποφάσεων για τη χωροθέτηση ενεργειακών εγκαταστάσεων. Επιπλέον, αναλύονται τα διεθνή και εθνικά πρότυπα διαχείρισης γεωχωρικών δεδομένων, οι ρυθμιστικές αρχές, καθώς και οι αρχές διαλειτουργικότητας και ανοικτών δεδομένων που διέπουν τα σύγχρονα γεωχωρικά συστήματα.

Το Δεύτερο Κεφάλαιο εστιάζει στις τεχνολογίες, τις υπηρεσίες και τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Παρουσιάζονται αναλυτικά τα τεχνολογικά στοιχεία του οικοσυστήματος ανάπτυξης, όπως οι React, TypeScript, Leaflet, PostgreSQL/PostGIS, Supabase, Netlify και Node.js, καθώς και τα πρότυπα δεδομένων JSON και GeoJSON. Επιπλέον, περιγράφονται οι αρχιτεκτονικές client-server, τα πρότυπα OGC (Open Geospatial Consortium) (WMS - Web Map Service, WFS - Web Feature Service, WCS - Web Coverage Service) και η φιλοσοφία ανάπτυξης Proof of Concept (PoC), που ακολουθήθηκε ώστε να επιβεβαιωθεί η τεχνική εφικτότητα της προτεινόμενης λύσης.

Στη συνέχεια περιγράφεται η ερευνητική μεθοδολογία και η διαδικασία σχεδιασμού του Web-GIS συστήματος. Παρουσιάζονται τα στάδια ανάλυσης απαιτήσεων, ο καθορισμός των λειτουργικών και μη λειτουργικών προδιαγραφών, καθώς και το διάγραμμα αρχιτεκτονικής της εφαρμογής. Εξηγείται η ροή δεδομένων μεταξύ front-end, back-end και βάσης, ενώ γίνεται αναφορά στις διαδικασίες ενσωμάτωσης (integration) και δοκιμών (testing) που πραγματοποιήθηκαν για τη διασφάλιση της αξιοπιστίας και της συνέπειας του συστήματος.

Το κεφάλαιο αυτό αναφέρεται τελικά στην υλοποίηση και αξιολόγηση της εφαρμογής. Παρουσιάζονται αναλυτικά η δομή και οι επιμέρους λειτουργίες του συστήματος, η διεπαφή χρήστη (UI) και οι χάρτες που ενσωματώθηκαν για την απεικόνιση των εγκαταστάσεων ΑΠΕ. Περιγράφεται η λειτουργικότητα της εφαρμογής ως Proof of Concept και τα αποτελέσματα της τεχνικής αξιολόγησης, η οποία βασίστηκε σε κριτήρια όπως η χρηστικότητα, η απόδοση, η ταχύτητα φόρτωσης και η διαλειτουργικότητα. Επίσης, αναλύονται οι δυνατότητες μελλοντικής επέκτασης του συστήματος με τεχνικές πολυκριτηριακής ανάλυσης, τηλεπισκόπησης και big data.

Τέλος, στο Τρίτο Κεφάλαιο συνοψίζονται τα συμπεράσματα που προέκυψαν από την ανάπτυξη και αξιολόγηση του Web-GIS συστήματος. Αναδεικνύεται η σημασία της χρήσης ανοιχτών τεχνολογιών και γεωχωρικών προτύπων για τον βιώσιμο ενεργειακό σχεδιασμό και διατυπώνονται προτάσεις για μελλοντική έρευνα. Μεταξύ αυτών περιλαμβάνονται η ενσωμάτωση εργαλείων τεχνητής νοημοσύνης για την πρόβλεψη ενεργειακής απόδοσης, η ανάπτυξη συμμετοχικών (participatory) εφαρμογών για πολίτες, καθώς και η σύνδεση με εθνικά και ευρωπαϊκά geoportals για ολοκληρωμένη χαρτογράφηση των ΑΠΕ.

Επίσης είναι απαραίτητο να αναφερθεί ότι, στο τέλος της παρούσας εργασίας, μετά την παράθεση των βιβλιογραφικών αναφορών κατά IEEE, στα επισυναπτόμενα παραρτήματα παρατίθενται και τα κρίσιμότερα και κυριότερα τμήματα του κώδικα που υλοποιεί την εφαρμογή.

# **1. Βιβλιογραφική Ανασκόπηση – Θεωρητική Τεκμηρίωση**

## **1.1. GIS και web εφαρμογές στην ενέργεια.**

Τα GIS (Geographic Information Systems) είναι συστήματα που στηρίζονται σε ηλεκτρονικούς υπολογιστές για τη διαχείριση, ανάλυση και απεικόνιση γεωγραφικών δεδομένων σε ψηφιακή μορφή, όπως ορίζεται από τον Goodchild (2009) <sup>[1]</sup>. Οι δυνατότητές τους περιλαμβάνουν χωρική ανάλυση, διαχείριση βάσεων δεδομένων και υποστήριξη λήψης αποφάσεων σε πολλούς επιστημονικούς και τεχνολογικούς τομείς (Goodchild 2009) <sup>[1]</sup>.

Η γενικότερη χρησιμότητα των GIS συνίσταται στην οπτικοποίηση πολύπλοκων χωρικών δεδομένων και στην υποβοήθηση λήψης αποφάσεων μέσω της δυνατότητας σύνθετης χωρικής ανάλυσης, όπως ανάλυση καταλληλότητας περιοχών, πληροφορία για οικοσυστήματα, χρήσεις γης και υποδομές (Goodchild 2009) <sup>[1]</sup>.

Στον τομέα της ενέργειας και ειδικότερα των Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ), τα GIS αξιοποιούνται για τον εντοπισμό των καταλληλότερων θέσεων εγκατάστασης φωτοβολταϊκών και ανεμογεννητριών, λαμβάνοντας υπόψη γεωμορφολογία, ακτινοβολία, άνεμο, υδρογραφικά στοιχεία, δίκτυα διανομής και περιβαλλοντικούς περιορισμούς (Avtar et al. 2019) <sup>[2]</sup>.

Κριτικές ανασκοπήσεις έχουν επισημάνει ότι τα εργαλεία GIS, όταν συνδυάζονται με μοντέλα τεχνοοικονομικής ανάλυσης, αποτελούν ισχυρά εργαλεία για την αξιολόγηση και σχεδίαση εγκαταστάσεων ΑΠΕ, με δυνατότητες προσομοίωσης και υποστήριξης πολιτικών (Tethys 2023) <sup>[3]</sup>.

Η ανάπτυξη "cadastre" Ανανεώσιμων Πηγών Ενέργειας έχει προταθεί ως μέθοδος για την καταγραφή της υπάρχουσας υποδομής και της δυναμικότητας εγκαταστάσεων, με γεωχωρικά δεδομένα που επιτρέπουν την αξιολόγηση νέων περιοχών και τη χάραξη πολιτικής (Bieda 2021) <sup>[4]</sup>.

Ο όρος cadastre (στα ελληνικά: κτηματολόγιο) προέρχεται από τη γαλλική λέξη cadastre και αναφέρεται σε ένα επίσημο, συστηματικό μητρώο που καταγράφει πληροφορίες για ακίνητα γης. Παραδοσιακά, το cadastre περιλαμβάνει: τα όρια και την έκταση των ακινήτων, τα στοιχεία ιδιοκτησίας (ποιος είναι ο κάτοχος), τη χρήση της γης (κατοικία, γεωργία, δάσος κ.λπ.), και τυχόν νομικά ή πολεοδομικά βάρη (π.χ. δικαιώματα διέλευσης).

Στη σύγχρονη εποχή, ένα cadastre είναι ψηφιακό, συνδεδεμένο με GIS (Geographic Information Systems), ώστε να προσφέρει ακριβή γεωχωρική απεικόνιση και εύκολη πρόσβαση σε θεσμικούς φορείς και πολίτες.

Υπάρχουν και ειδικοί τύποι cadastre, π.χ.:

- Solar cadastre: χάρτης/μητρώο που δείχνει την ηλιακή δυναμικότητα κτιρίων.
- Energy cadastre: βάση δεδομένων για το δυναμικό και τις εγκαταστάσεις ΑΠΕ.
- Forest cadastre: καταγραφή δασικών εκτάσεων.

Στην Ελλάδα, το αντίστοιχο Ηλεκτρονικό Κτηματολόγιο είναι το Εθνικό Κτηματολόγιο, που συγκεντρώνει γεωχωρικά και νομικά δεδομένα για όλα τα ακίνητα, με στόχο τη διαφάνεια και την ασφάλεια στις συναλλαγές. Επομένως γενικότερα, το cadastre είναι ένας επίσημος "χάρτης-μητρώο" γης και ιδιοκτησίας, και όταν συνδέεται με την ενέργεια (π.χ. solar cadastre), γίνεται ένα εργαλείο GIS για την καταγραφή και αξιολόγηση του ενεργειακού δυναμικού.

Πρακτικά παραδείγματα "συντεταγμένων cadastre" περιλαμβάνουν την ανάπτυξη solar cadastre στη Θεσσαλονίκη, όπου αξιολογείται η δυναμικότητα ηλιακής ενέργειας σε κτίρια λαμβάνοντας υπόψη σκίαση από γύρω κτίρια και δέντρα (Chatzigeorgiou et al.) [5].

Σε διεθνές επίπεδο, υπάρχουν εργαλεία όπως το Global Solar Atlas και το Global Wind Atlas, τα οποία παρέχουν διαδραστικούς χάρτες ακτινοβολίας και ανέμου αντίστοιχα, με δυνατότητα κατεβάσματος δεδομένων σε μορφή GIS για περαιτέρω ανάλυση (World Bank, ESMAP) [6], [7].

Αυτά τα web-based εργαλεία δίνουν τη δυνατότητα σε πολιτικούς, ερευνητές και επενδυτές να εκτιμήσουν ενεργειακό δυναμικό σε παγκόσμια κλίμακα, ενισχύοντας τον σχεδιασμό και την ευαισθητοποίηση (World Bank, ESMAP) [6], [7].

Σε πιο πρακτικό επίπεδο, εταιρείες όπως η Enel Green Power έχουν αναπτύξει εφαρμογές GIS που προβλέπουν την παραγωγή φωτοβολταϊκών εγκαταστάσεων με χρήση δεδομένων ακτινοβολίας και μετεωρολογικών στοιχείων, ενισχύοντας την ακρίβεια προβλέψεων και τη λήψη αποφάσεων (Enel Green Power 2023) [8].

Η συνδυασμένη χρήση GIS με τεχνικές πολυκριτηριακής απόφασης (MCDA) έχει αποδειχθεί αποτελεσματική για επιλογή τοποθεσιών, όπως σε μελέτες για ανεμογεννήτριες ή φωτοβολταϊκά που λαμβάνουν υπόψη τεχνικά, οικονομικά, οικολογικά και κοινωνικά κριτήρια (Bertsiou et al. 2025) <sup>[9]</sup>.

Παράλληλα, η χρησιμοποίηση του GIS σε συνδυασμό με remote sensing και τεχνητή νοημοσύνη δημιουργεί νέες δυνατότητες αυτοματοποιημένης ανάλυσης, όπως ανίχνευση μονάδων παραγωγής ενέργειας με χρήση CNN (Convolutional Neural Networks - Συνελκτικά Νευρωνικά Δίκτυα) και Vision Transformers (Z. Guo et al. 2024, K. Ioannou et al.) <sup>[109],[110]</sup>.

Τα Vision Transformers (ViTs) αποτελούν πιο σύγχρονη προσέγγιση σε σύγκριση με τα CNN και βασίζονται στην αρχιτεκτονική Transformer (Vaswani et al. 2017; Dosovitskiy et al. 2020) <sup>[10]</sup>, <sup>[11]</sup>. Εφαρμόζουν την αρχή της προσοχής (attention), χωρίζοντας την εικόνα σε μικρά «patches» και εστιάζοντας στα πιο σημαντικά μέρη της, γεγονός που τα καθιστά ιδιαίτερα αποδοτικά (Khan et al. 2022) <sup>[12]</sup>. Νεότερες παραλλαγές, όπως το Swin Transformer (Liu et al. 2021) <sup>[13]</sup> και τα data-efficient ViTs (Touvron et al. 2021) <sup>[14]</sup>, έχουν δείξει ακόμη καλύτερες επιδόσεις σε σύνθετες εφαρμογές αναγνώρισης εικόνας.

Ένα άλλο παράδειγμα χρήσης GIS στην ενεργειακή διαχείριση αφορά την παρακολούθηση και ανάλυση δικτύων χαμηλής τάσης, όπου GIS εργαλεία βοηθούν στην ανίχνευση σφαλμάτων και την προσομοίωση συμπεριφορών κατανάλωσης, προσφέροντας λύσεις βελτίωσης της ποιότητας ισχύος (Antic & Capuder 2023) <sup>[15]</sup>.

Στην Ελλάδα, η χρήση του Hellenic Cadastre (Ηλεκτρονικό Κτηματολόγιο) προωθείται για πολλούς σκοπούς, συμπεριλαμβανομένης της χρήσης του για GIS εφαρμογές, αφού παρέχει ακριβή γεωχωρικά δεδομένα που μπορούν να υποστηρίξουν χωροθέτηση ενεργειακών έργων (UN-GGIM 2015) <sup>[16]</sup>.

Ο συνδυασμός του Ελληνικού Κτηματολογίου με Web-GIS υποδομές θα μπορούσε να ενισχύσει τη διάχυση και αξιοποίηση δεδομένων ιδιοκτησιών, χρήσεων γης και ενεργειακού δυναμικού, προσφέροντας σαφή εικόνα για τη βέλτιστη χωροθέτηση εγκαταστάσεων ΑΠΕ.

Σε επίπεδο ανοικτών ενεργειακών δεδομένων, υπάρχουν πλατφόρμες όπως το energydata.info, το OpenEI, το Open Power System Data, καθώς και το Renewables.ninja, οι οποίες προσφέρουν GIS δεδομένα σχετικών με ενεργειακά συστήματα, παραγωγή και πόρους (Open energy system databases 2025) <sup>[17]</sup>.

Τα πλεονεκτήματα της χρήσης web-GIS στην ενέργεια περιλαμβάνουν τη διαδραστική απεικόνιση, τον διαμοιρασμό πληροφορίας σε ευρύ κοινό, τη βελτίωση αποφάσεων χάραξης πολιτικής και επενδύσεων, καθώς και τη δυνατότητα άμεσης ενημέρωσης δεδομένων.

Ωστόσο, υπάρχουν και μειονεκτήματα: η εξάρτηση από την ποιότητα των δεδομένων, οι τεχνικές απαιτήσεις για υποδομές, θέματα διαλειτουργικότητας, ασφάλειας και προστασίας προσωπικών δεδομένων, αλλά και οι δυσκολίες ολοκλήρωσης με άλλα συστήματα (Tethys 2023; Bieda 2021) <sup>[3]</sup>, <sup>[4]</sup>.

Το μέλλον των GIS στην ενέργεια δείχνει ιδιαίτερα υποσχόμενο: η ενοποίηση με ενεργειακά μοντέλα, real time δεδομένα, τεχνολογίες cloud, AI, και participatory GIS (PPGIS), μπορεί να μετατρέψει τα Web-GIS εργαλεία σε καθολικά συστήματα λήψης αποφάσεων με συμμετοχικό χαρακτήρα (Tethys 2023; Participatory GIS) <sup>[3]</sup>, <sup>[18]</sup>. Συμπερασματικά, το GIS και τα web-GIS εργαλεία αναδεικνύονται σε βασικά εργαλεία για:

- Εκτίμηση δυναμικού ΑΠΕ (ηλιακή και αιολική),
- Χωροθέτηση εγκαταστάσεων με πολυκριτηριακή αξιολόγηση,
- Οπτικοποίηση δεδομένων για λήψη αποφάσεων,
- Ανοιχτή πρόσβαση σε ενεργειακά και γεωχωρικά δεδομένα,
- Υποστήριξη πολιτικής και δημόσιας ενημέρωσης, με σημαντικά οφέλη αλλά και τεχνολογικές και οργανωτικές προκλήσεις.

## **1.2. GIS frameworks (Leaflet.js, OpenLayers, Google Maps API).**

Τα GIS frameworks αποτελούν βασικά εργαλεία ανάπτυξης διαδραστικών εφαρμογών χαρτογράφησης στο διαδίκτυο. Πρόκειται για βιβλιοθήκες λογισμικού που επιτρέπουν την απεικόνιση, πλοήγηση, ανάλυση και αλληλεπίδραση με γεωχωρικά δεδομένα σε διαδικτυακό περιβάλλον. Η σημασία τους είναι μεγάλη, καθώς επιτρέπουν τη σύνδεση μεταξύ των δεδομένων GIS και των τελικών χρηστών, χωρίς την ανάγκη εξειδικευμένου λογισμικού ή υπολογιστικών πόρων στην πλευρά του χρήστη (Haklay & Weber 2008) <sup>[19]</sup>.

Η χρήση GIS frameworks έχει διευκολύνει την ανάπτυξη Web-GIS εφαρμογών σε πολλούς τομείς: από την αστική ανάπτυξη και την περιβαλλοντική παρακολούθηση, έως τις μεταφορές

και τις Ανανεώσιμες Πηγές Ενέργειας (ΑΠΕ). Η δυνατότητα αξιοποίησης ανοικτών δεδομένων και υπηρεσιών διαδικτύου, όπως τα WMS (Web Map Services), τα WFS (Web Feature Services) και τα APIs τρίτων, καθιστά τα frameworks ευέλικτα εργαλεία για έρευνα και εφαρμογές πολιτικής (Neis & Zielstra 2014) <sup>[20]</sup>.

Στον τομέα της ενέργειας, οι πλατφόρμες αυτές επιτρέπουν την απεικόνιση εγκαταστάσεων ΑΠΕ, την εκτίμηση δυναμικού, τη χωροθέτηση και την ενημέρωση ενδιαφερομένων. Για παράδειγμα, εφαρμογές που συνδυάζουν δορυφορικά δεδομένα με Leaflet ή OpenLayers προσφέρουν στους πολίτες διαδραστικούς χάρτες με περιοχές κατάλληλες για φωτοβολταϊκά ή αιολικά έργα (Avtar et al. 2019) <sup>[21]</sup>.

Το Leaflet.js είναι ένα από τα πιο διαδεδομένα ελαφριά frameworks για web mapping. Αναπτύχθηκε με γνώμονα την απλότητα και την απόδοση, ώστε να μπορεί να χρησιμοποιηθεί εύκολα σε εφαρμογές που δεν απαιτούν βαριά επεξεργασία δεδομένων. Το βασικό του πλεονέκτημα είναι το μικρό του μέγεθος και η ευκολία χρήσης, γεγονός που το καθιστά ιδανικό για εφαρμογές σε κινητές συσκευές και για έργα με περιορισμένους πόρους (Agafonkin 2011) <sup>[22]</sup>.

Ένα σημαντικό παράδειγμα χρήσης Leaflet είναι οι χάρτες του OpenStreetMap, καθώς και πλήθος δημοτικών ή περιβαλλοντικών εφαρμογών που επιτρέπουν την παρακολούθηση δεδομένων σε πραγματικό χρόνο, όπως ποιότητα αέρα ή επίπεδα θορύβου (Neis & Zielstra 2014) <sup>[20]</sup>. Στις ΑΠΕ, Leaflet χρησιμοποιείται σε εφαρμογές που παρουσιάζουν δεδομένα ηλιακής ακτινοβολίας σε επίπεδο κτιρίου, προσφέροντας στους χρήστες εύκολη και γρήγορη απεικόνιση (Chatzigeorgiou et al. 2023) <sup>[23]</sup>.

Στα πλεονεκτήματα του Leaflet συγκαταλέγονται η απλότητα, η μεγάλη κοινότητα ανάπτυξης και ο πλούτος προσθηκών (plugins). Ωστόσο, μειονεκτεί σε πιο απαιτητικές εφαρμογές με μεγάλα γεωχωρικά datasets, καθώς δεν έχει την ίδια υπολογιστική ισχύ και κλιμάκωση που προσφέρουν άλλες πλατφόρμες, όπως το OpenLayers (Peterson 2012) <sup>[24]</sup>.

Το OpenLayers, αντίθετα, αποτελεί ένα πιο ισχυρό και ευέλικτο framework, που προσφέρει προηγμένες δυνατότητες GIS, όπως υποστήριξη για WMS, WFS και εξειδικευμένα formats (GeoJSON, KML, GML). Χρησιμοποιείται κυρίως σε έργα όπου απαιτείται μεγάλη ακρίβεια και κλιμάκωση, όπως εθνικές χωρικές υποδομές δεδομένων και γεωπληροφοριακά συστήματα δημόσιων οργανισμών (Hazzard 2011) <sup>[25]</sup>.

Παράδειγμα εφαρμογής OpenLayers αποτελεί το Hellenic Cadastre (Ελληνικό Κτηματολόγιο), το οποίο χρησιμοποιεί web-based τεχνολογίες για την παροχή χαρτογραφικών δεδομένων σε πολίτες και φορείς (UN-GGIM 2015) <sup>[26]</sup>. Στο πεδίο της ενέργειας, το OpenLayers συχνά ενσωματώνεται σε πλατφόρμες χωροθέτησης αιολικών και φωτοβολταϊκών πάρκων, επιτρέποντας πολυκριτηριακή αξιολόγηση σε συνδυασμό με δεδομένα χρήσεων γης και περιβαλλοντικούς περιορισμούς (Bertsiou et al. 2025) <sup>[27]</sup>.

Το κύριο πλεονέκτημα του OpenLayers είναι η ισχυρή του αρχιτεκτονική, που επιτρέπει την επεξεργασία μεγάλων datasets και τη δημιουργία προηγμένων εργαλείων ανάλυσης. Στα μειονεκτήματά του συγκαταλέγονται η πολυπλοκότητα στη χρήση και η καμπύλη εκμάθησης, γεγονός που το καθιστά πιο δύσκολο για αρχάριους προγραμματιστές (Hazzard 2011) <sup>[25]</sup>.

Το Google Maps API αποτελεί ένα από τα πιο γνωστά frameworks παγκοσμίως, προσφέροντας χάρτες υψηλής ποιότητας, δορυφορικές εικόνες, δυνατότητες πλοήγησης και πρόσβαση σε δεδομένα της Google. Η χρήση του έχει διαδοθεί σε πλήθος εφαρμογών, από εμπορικές ιστοσελίδες μέχρι εφαρμογές logistics, τουρισμού και μεταφορών (Peterson 2012) <sup>[24]</sup>.

Στον τομέα της ενέργειας, το Google Maps API χρησιμοποιείται για την απεικόνιση υποδομών ΑΠΕ, όπως πάρκα ανεμογεννητριών και φωτοβολταϊκών, καθώς και για τη χαρτογράφηση δικτύων μεταφοράς και διανομής ηλεκτρικής ενέργειας. Ένα χαρακτηριστικό παράδειγμα είναι το εργαλείο Google Project Sunroof, που εκτιμά την ηλιακή δυναμικότητα στέγης χρησιμοποιώντας δορυφορικά δεδομένα και GIS τεχνικές (Google 2017) <sup>[28]</sup>.

Το βασικό πλεονέκτημα του Google Maps API είναι η αξιοπιστία και η ευκολία ενσωμάτωσης σε εφαρμογές. Επιπλέον, η πλατφόρμα διαθέτει ισχυρά δεδομένα χαρτογράφησης και συχνές ενημερώσεις. Ωστόσο, το κύριο μειονέκτημα είναι η εξάρτηση από την πολιτική χρήσης της Google, καθώς και το γεγονός ότι δεν είναι πλήρως ανοιχτή πλατφόρμα. Η εμπορική διάσταση περιορίζει τη δυνατότητα ενσωμάτωσης σε εφαρμογές ερευνητικού ή δημόσιου χαρακτήρα με περιορισμένους πόρους (Haklay & Weber 2008) <sup>[19]</sup>.

Συγκριτικά, το Leaflet ξεχωρίζει για την απλότητα και την ταχύτητα ανάπτυξης, το OpenLayers για την ισχύ και τις προηγμένες λειτουργίες GIS, ενώ το Google Maps API για την προσβασιμότητα και τα υψηλής ποιότητας δεδομένα. Η επιλογή μεταξύ τους εξαρτάται από τις απαιτήσεις της εκάστοτε εφαρμογής, όπως ο όγκος δεδομένων, η ανάγκη για ακρίβεια ή οι περιορισμοί κόστους.

Οι εξελίξεις στην ανάπτυξη Web-GIS frameworks δείχνουν μια τάση προς μεγαλύτερη διαλειτουργικότητα, αξιοποίηση cloud τεχνολογιών και ενσωμάτωση τεχνητής νοημοσύνης. Στο μέλλον, οι πλατφόρμες αυτές αναμένεται να ενισχυθούν με δυνατότητες real-time ανάλυσης και προηγμένες τεχνικές οπτικοποίησης, συμβάλλοντας στη λήψη αποφάσεων σε πολύπλοκα ενεργειακά συστήματα (Avtar et al. 2019) <sup>[21]</sup>.

Η ενσωμάτωση GIS frameworks στην ενέργεια δεν περιορίζεται μόνο σε τεχνικό επίπεδο. Αποτελεί επίσης εργαλείο πολιτικής και συμμετοχής, αφού οι πολίτες μπορούν να έχουν πρόσβαση σε ανοικτά δεδομένα και να συμμετέχουν σε διαδικασίες χωροθέτησης έργων ΑΠΕ, ενισχύοντας τη διαφάνεια και την κοινωνική αποδοχή (Haklay & Weber 2008) <sup>[19]</sup>.

Συνολικά, τα GIS frameworks λειτουργούν ως γέφυρα μεταξύ γεωχωρικής πληροφορίας και χρηστών, μετατρέποντας πολύπλοκα δεδομένα σε εύχρηστες και διαδραστικές πλατφόρμες. Στον ενεργειακό τομέα, αποτελούν απαραίτητο συστατικό για την ανάπτυξη εφαρμογών που θα υποστηρίξουν την ενεργειακή μετάβαση και τη βιώσιμη ανάπτυξη.

### **1.3. Βάσεις δεδομένων χωρικών δεδομένων (PostGIS, MongoDB).**

Οι βάσεις δεδομένων χωρικών δεδομένων αποτελούν αναπόσπαστο κομμάτι των σύγχρονων Web-GIS εφαρμογών, καθώς προσφέρουν τη δυνατότητα αποθήκευσης, ανάκτησης και ανάλυσης γεωχωρικών πληροφοριών. Σε αντίθεση με τις παραδοσιακές βάσεις δεδομένων που διαχειρίζονται μόνο αλφαριθμητικά δεδομένα, οι βάσεις αυτές είναι σχεδιασμένες να χειρίζονται πολύπλοκες γεωμετρικές δομές όπως σημεία, γραμμές και πολύγωνα, καθώς και να εκτελούν χωρικά ερωτήματα (Shekhar & Chawla 2003) <sup>[29]</sup>.

Η ανάπτυξη εξειδικευμένων συστημάτων βάσεων για GIS συνδέεται στενά με την ανάγκη για ολοκληρωμένες χωρικές αναλύσεις. Η δυνατότητα χρήσης SQL με χωρικές επεκτάσεις, αλλά και η ανάπτυξη NoSQL βάσεων που υποστηρίζουν χωρικές δομές, οδήγησαν σε ευρεία χρήση εργαλείων όπως το PostGIS και το MongoDB (Rigaux, Scholl & Voisard 2001) <sup>[30]</sup>.

Το PostGIS είναι μια χωρική επέκταση του PostgreSQL, ενός από τα πιο ισχυρά και δημοφιλή συστήματα σχεσιακών βάσεων δεδομένων ανοιχτού κώδικα. Αναπτύχθηκε το 2001 από την Refractions Research στον Καναδά, με στόχο να παρέχει υποστήριξη για γεωχωρικά δεδομένα και χωρικά ερωτήματα, σύμφωνα με τα πρότυπα της Open Geospatial Consortium (OGC) (Obe & Hsu 2011) <sup>[31]</sup>.

Η κύρια χρησιμότητα του PostGIS είναι ότι μετατρέπει το PostgreSQL σε πλήρως λειτουργικό Spatial Database Management System (SDBMS). Υποστηρίζει γεωμετρικούς τύπους δεδομένων, όπως σημεία, γραμμές, πολύγωνα και πολυγωνικά πλέγματα, καθώς και λειτουργίες όπως υπολογισμός αποστάσεων, εμβαδών και σχέσεων τοπολογίας (Steiniger & Hunter 2013) <sup>[32]</sup>.

Ένα από τα μεγαλύτερα πλεονεκτήματα του PostGIS είναι η συμβατότητά του με άλλα GIS εργαλεία και η υποστήριξη προτύπων OGC όπως WMS και WFS. Αυτό καθιστά δυνατή τη διαλειτουργικότητα με λογισμικά όπως QGIS, ArcGIS και GeoServer, επιτρέποντας την απρόσκοπτη ροή δεδομένων (Obe & Hsu 2011) <sup>[31]</sup>.

Στον τομέα των ΑΠΕ, το PostGIS χρησιμοποιείται για την αποθήκευση και ανάλυση δεδομένων που σχετίζονται με τη χωροθέτηση εγκαταστάσεων, όπως υψομετρικά μοντέλα, δεδομένα ηλιακής ακτινοβολίας, θέσεις ανεμογεννητριών και δίκτυα μεταφοράς. Έρευνες έχουν δείξει ότι μπορεί να υποστηρίξει πολύπλοκα χωρικά μοντέλα για την εκτίμηση καταλληλότητας περιοχών (Avtar et al. 2019) <sup>[33]</sup>.

Ωστόσο, το PostGIS έχει και μειονεκτήματα. Η χρήση του απαιτεί εξειδικευμένη γνώση SQL και χωρικών συναρτήσεων, ενώ η διαχείριση πολύ μεγάλων datasets μπορεί να είναι απαιτητική σε υπολογιστικούς πόρους. Επιπλέον, η διαδικασία κλιμάκωσης σε περιβάλλοντα big data είναι πιο δύσκολη σε σχέση με βάσεις NoSQL (Steiniger & Hunter 2013) <sup>[32]</sup>.

Η MongoDB, από την άλλη, είναι μια NoSQL βάση δεδομένων, που αναπτύχθηκε το 2007 από την εταιρεία 10gen (σήμερα MongoDB Inc.). Αντί να χρησιμοποιεί σχέσεις και πίνακες, αποθηκεύει δεδομένα σε μορφή εγγράφων (documents) τύπου JSON, γεγονός που προσφέρει ευελιξία και ταχύτητα στη διαχείριση μη δομημένων δεδομένων (Chodorow 2013) <sup>[34]</sup>.

Η υποστήριξη χωρικών δεδομένων στο MongoDB έχει εισαχθεί μέσω της δυνατότητας αποθήκευσης και αναζήτησης δεδομένων γεωχωρικού τύπου. Αυτό επιτρέπει ερωτήματα όπως εύρεση κοντινών σημείων (nearest neighbor search), υπολογισμός αποστάσεων και χωρικό φιλτράρισμα (Banker 2016) <sup>[35]</sup>.

Ένα από τα βασικά πλεονεκτήματα του MongoDB είναι η επεκτασιμότητά του (scalability). Χάρη στην αρχιτεκτονική του, μπορεί να διαχειριστεί τεράστιες ποσότητες δεδομένων σε κατανεμημένα περιβάλλοντα, γεγονός που το καθιστά ιδανικό για εφαρμογές big data και real-time ανάλυσης (Pokorný 2016) <sup>[36]</sup>.

Στον τομέα των ΑΠΕ, το MongoDB χρησιμοποιείται για την ανάλυση δεδομένων αισθητήρων σε πραγματικό χρόνο, όπως μετεωρολογικά δεδομένα, δεδομένα παραγωγής ενέργειας και συντήρησης εγκαταστάσεων. Για παράδειγμα, μπορεί να αποθηκεύει συνεχείς ροές δεδομένων από ανεμογεννήτριες και να υποστηρίζει εφαρμογές προγνωστικής συντήρησης (Huang et al. 2017) <sup>[37]</sup>.

Το βασικό μειονέκτημα του MongoDB σε σχέση με το PostGIS είναι ότι οι χωρικές του δυνατότητες είναι πιο περιορισμένες. Ενώ είναι εξαιρετικό για γρήγορα queries σε γεωχωρικά σημεία και πολυγωνικές περιοχές, δεν προσφέρει το ίδιο εύρος χωρικών αναλύσεων και συναρτήσεων που παρέχει το PostGIS (Chodorow 2013) <sup>[34]</sup>.

Η ιστορική εξέλιξη των δύο συστημάτων δείχνει μια συμπληρωματική πορεία: το PostGIS αναπτύχθηκε από την ανάγκη ακριβούς και προτυποποιημένης χωρικής ανάλυσης, ενώ το MongoDB δημιουργήθηκε για την αντιμετώπιση της ταχύτητας και του όγκου των δεδομένων που εμφανίστηκαν με την εξάπλωση του web και των IoT συστημάτων (Pokorný 2016) <sup>[36]</sup>.

Σε πρακτικό επίπεδο, πολλές σύγχρονες Web-GIS εφαρμογές συνδυάζουν τις δύο βάσεις: PostGIS για την αυστηρή χωρική ανάλυση και MongoDB για τη διαχείριση real-time ή μη δομημένων δεδομένων. Αυτό το υβριδικό μοντέλο αξιοποιεί τα πλεονεκτήματα και των δύο τεχνολογιών (Steiniger & Hunter 2013) <sup>[32]</sup>.

Η επιλογή της κατάλληλης βάσης εξαρτάται σε μεγάλο βαθμό από τις ανάγκες της εφαρμογής: έργα που απαιτούν πολύπλοκα χωρικά ερωτήματα και τήρηση προτύπων συνήθως επιλέγουν PostGIS, ενώ εφαρμογές που επικεντρώνονται σε big data και real-time ανάλυση προτιμούν MongoDB.

Συμπερασματικά, τόσο το PostGIS όσο και το MongoDB έχουν καθιερωθεί ως βασικά εργαλεία για την ανάπτυξη συστημάτων που διαχειρίζονται χωρικά δεδομένα. Στον τομέα της ενέργειας, η σωστή επιλογή ή ο συνδυασμός τους μπορεί να αποτελέσει καταλύτη για την ανάπτυξη έξυπνων Web-GIS εφαρμογών που θα υποστηρίξουν τη βιώσιμη χωροθέτηση και παρακολούθηση εγκαταστάσεων ΑΠΕ.

#### 1.4. Client-server αρχιτεκτονικές για geospatial apps

Οι client-server αρχιτεκτονικές αποτελούν θεμέλιο λίθο για τη λειτουργία σύγχρονων γεωχωρικών εφαρμογών. Βασίζονται στην αρχή της διάκρισης μεταξύ ενός πελάτη (client), που είναι υπεύθυνος για την απεικόνιση και την αλληλεπίδραση με τον χρήστη, και ενός εξυπηρετητή (server), που παρέχει τα δεδομένα, εκτελεί αναλύσεις και διαχειρίζεται τις βάσεις δεδομένων (Tanenbaum & Steen 2007) <sup>[38]</sup>.

Η ιστορική εξέλιξη της client-server αρχιτεκτονικής συνδέεται με την ανάπτυξη καταμεμημένων συστημάτων τις δεκαετίες 1980 και 1990, όπου η ανάγκη για πολυχρηστικά περιβάλλοντα και πρόσβαση σε κεντρικά δεδομένα οδήγησε στη δημιουργία του μοντέλου αυτού (Coulouris, Dollimore & Kindberg 2005) <sup>[39]</sup>. Στα GIS, η μετάβαση από το desktop στο web περιβάλλον κατέστησε την client-server προσέγγιση απαραίτητη, καθώς επέτρεψε την παροχή γεωχωρικών δεδομένων σε χρήστες μέσω διαδικτύου (Peng & Tsou 2003) <sup>[40]</sup>.

Στο πλαίσιο των γεωχωρικών εφαρμογών, ο client μπορεί να είναι ένα πρόγραμμα περιήγησης (browser), μια κινητή εφαρμογή ή ακόμη και ένα desktop GIS που ζητά δεδομένα από τον server. Ο server μπορεί να είναι ένας web server, ένας map server (π.χ. GeoServer, MapServer) ή μια βάση χωρικών δεδομένων που απαντά σε χωρικά αιτήματα (Zhang, Goodchild & Batty 2012) <sup>[41]</sup>.

Υπάρχουν δύο βασικοί τύποι client-server αρχιτεκτονικών: οι thin clients και οι thick clients. Στην πρώτη περίπτωση, η επεξεργασία γίνεται κυρίως στον server, ενώ ο client περιορίζεται στην απεικόνιση. Στη δεύτερη περίπτωση, ο client εκτελεί μέρος της ανάλυσης, μειώνοντας το φόρτο στον server αλλά απαιτώντας περισσότερους πόρους στον τελικό χρήστη (Peng & Tsou 2003) <sup>[40]</sup>.

Η χρήση thin client αρχιτεκτονικών είναι ιδιαίτερα συχνή σε εφαρμογές web mapping, όπως το Google Maps, όπου ο server διαχειρίζεται μεγάλα datasets και ο browser προβάλλει τα αποτελέσματα. Αυτό το μοντέλο διευκολύνει την πρόσβαση από οποιαδήποτε συσκευή, ανεξαρτήτως υπολογιστικής ισχύος (Peterson 2012) <sup>[42]</sup>.

Αντίθετα, οι thick clients συναντώνται συχνότερα σε επαγγελματικά περιβάλλοντα GIS, όπου το λογισμικό στο client (π.χ. QGIS) μπορεί να κατεβάσει δεδομένα από τον server και να εκτελέσει αναλύσεις τοπικά. Αυτό επιτρέπει μεγαλύτερη ευελιξία αλλά περιορίζει την καθολική πρόσβαση (Steiniger & Hunter 2013) <sup>[43]</sup>.

Ένα σημαντικό στοιχείο στη client-server αρχιτεκτονική είναι τα web services. Πρότυπα όπως το WMS (Web Map Service), WFS (Web Feature Service) και WCS (Web Coverage Service) αναπτύχθηκαν από το OGC (Open Geospatial Consortium) ώστε να επιτρέπουν τη διαλειτουργικότητα μεταξύ διαφορετικών συστημάτων και εφαρμογών (Percivall 2008) <sup>[44]</sup>.

Η ανάπτυξη map servers, όπως το GeoServer και το MapServer, βασίζεται στην αρχιτεκτονική client-server και παρέχει δεδομένα σε μορφές συμβατές με τα παραπάνω πρότυπα. Αυτά τα συστήματα αποτελούν τη ραχοκοκαλιά πολλών δημόσιων γεωχωρικών υποδομών (OGC 2015) <sup>[45]</sup>.

Στον τομέα των ΑΠΕ, οι client-server αρχιτεκτονικές χρησιμοποιούνται για εφαρμογές που επιτρέπουν σε χρήστες να εξετάσουν χάρτες ηλιακής ακτινοβολίας, δεδομένα ανέμου ή πιθανές τοποθεσίες εγκατάστασης μέσω ενός browser. Ο server εκτελεί τις χωρικές αναλύσεις και ο client παρουσιάζει τα αποτελέσματα με διαδραστικό τρόπο (Avtar et al. 2019) <sup>[46]</sup>.

Τα πλεονεκτήματα των client-server αρχιτεκτονικών περιλαμβάνουν την κεντρική διαχείριση δεδομένων, την ασφάλεια και την ευκολία ενημέρωσης. Επιπλέον, διευκολύνουν τη συνεργασία μεταξύ διαφορετικών φορέων, καθώς τα δεδομένα παραμένουν σε έναν server και δεν απαιτείται αποστολή αντιγράφων (Zhang, Goodchild & Batty 2012) <sup>[41]</sup>.

Ωστόσο, υπάρχουν και μειονεκτήματα. Η εξάρτηση από τον server μπορεί να δημιουργήσει bottlenecks σε περιβάλλοντα με πολλούς χρήστες. Η απόδοση εξαρτάται επίσης από την ταχύτητα σύνδεσης στο διαδίκτυο και την υπολογιστική ισχύ του server. Επιπλέον, οι thin clients έχουν περιορισμένες δυνατότητες ανάλυσης σε σχέση με τα desktop GIS (Coulouris, Dollimore & Kindberg 2005) <sup>[39]</sup>.

Σημαντική εξέλιξη στον χώρο είναι η μετάβαση σε cloud-based αρχιτεκτονικές, που αποτελούν μια προέκταση του client-server μοντέλου. Σε αυτή την περίπτωση, οι server υπηρεσίες φιλοξενούνται σε cloud υποδομές, επιτρέποντας κλιμάκωση ανάλογα με τη ζήτηση και μειώνοντας το κόστος συντήρησης (Yang et al. 2011) <sup>[47]</sup>.

Η ανάπτυξη RESTful APIs αποτέλεσε επίσης κομβικό σημείο στην εξέλιξη των geospatial apps. Μέσω REST (REpresentational State Transfer που αποδίδεται ως Αρχιτεκτονική Μεταφοράς Αναπαριστώμενης Κατάστασης), οι clients μπορούν να επικοινωνούν με servers με ελαφριά πρωτόκολλα, ενισχύοντας την απόδοση και την απλότητα ενσωμάτωσης σε εφαρμογές τρίτων (Fielding 2000) <sup>[48]</sup>.

Οι αρχιτεκτονικές αυτές δεν περιορίζονται μόνο σε web apps. Στις κινητές εφαρμογές, για παράδειγμα, η χρήση client-server μοντέλου επιτρέπει την απεικόνιση χαρτών offline (μέσω caching) και την ενημέρωση δεδομένων όταν υπάρχει σύνδεση, βελτιώνοντας την εμπειρία του χρήστη (Haklay & Weber 2008) <sup>[49]</sup>.

Η ασφάλεια αποτελεί επίσης κρίσιμο ζήτημα. Σε περιβάλλοντα όπου διακινούνται ευαίσθητα δεδομένα (π.χ. ιδιοκτησιακά δικαιώματα, υποδομές ενέργειας), η client-server αρχιτεκτονική πρέπει να εξασφαλίζει κρυπτογράφηση, ελεγχόμενη πρόσβαση και μηχανισμούς auditing (Zhang, Goodchild & Batty 2012) <sup>[41]</sup>.

Στο πλαίσιο της βιώσιμης ανάπτυξης, οι client-server εφαρμογές GIS επιτρέπουν όχι μόνο την υποστήριξη αποφάσεων αλλά και τη συμμετοχική χαρτογράφηση (participatory mapping). Οι πολίτες μπορούν να εισάγουν δεδομένα από τον client, τα οποία αποθηκεύονται στον server, ενισχύοντας την κοινωνική διάσταση της τεχνολογίας (Goodchild 2007) <sup>[50]</sup>.

Συνολικά, η αρχιτεκτονική client-server έχει αποδείξει την ανθεκτικότητά της και παραμένει θεμέλιο για τα γεωχωρικά συστήματα. Η μελλοντική της εξέλιξη φαίνεται να συνδέεται με το cloud computing, τα μικρο-υπηρεσιακά συστήματα (microservices) και την τεχνητή νοημοσύνη, καθιστώντας τις geospatial apps ακόμη πιο ισχυρές και ευέλικτες (Yang et al. 2011) <sup>[47]</sup>.

## **1.5. Παραδείγματα από υπάρχουσες εφαρμογές Web-GIS σε ΑΠΕ.**

Η ανάπτυξη εφαρμογών Web-GIS στον τομέα των Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ) έχει γνωρίσει σημαντική άνθηση την τελευταία δεκαετία. Οι εφαρμογές αυτές αξιοποιούν χωρικά δεδομένα, δορυφορικές μετρήσεις, βάσεις δεδομένων και τεχνικές ανάλυσης για να υποστηρίξουν την αξιολόγηση, τον σχεδιασμό και την παρακολούθηση έργων ΑΠΕ. Η δυνατότητα διαδικτυακής πρόσβασης σε αυτά τα εργαλεία επιτρέπει σε επενδυτές, ερευνητές, φορείς χάραξης πολιτικής και πολίτες να έχουν διαδραστική και άμεση πληροφόρηση (Avtar et al. 2019) <sup>[51]</sup>.

Ένα από τα πλέον γνωστά διεθνή παραδείγματα είναι το Global Solar Atlas, που αναπτύχθηκε από την Παγκόσμια Τράπεζα και το ESMAP. Παρέχει υψηλής ανάλυσης δεδομένα ηλιακής ακτινοβολίας για όλες τις χώρες του κόσμου, με δυνατότητα λήψης δεδομένων σε μορφή GIS

και εκτίμησης της ηλιακής δυναμικότητας σε συγκεκριμένες τοποθεσίες (World Bank/ESMAP 2019) <sup>[52]</sup>.

Αντίστοιχο εργαλείο αποτελεί το Global Wind Atlas, που παρέχει δεδομένα αιολικού δυναμικού παγκοσμίως. Βασίζεται σε προσομοιώσεις μετεωρολογικών μοντέλων και στατιστική επεξεργασία μετρήσεων, δίνοντας στους χρήστες τη δυνατότητα να αξιολογήσουν περιοχές με βάση το αιολικό δυναμικό και τη γεωγραφική τους θέση (World Bank/ESMAP 2019) <sup>[52]</sup>.

Το πλεονέκτημα αυτών των εργαλείων είναι η δωρεάν πρόσβαση σε υψηλής ποιότητας δεδομένα, η απλή διεπαφή και η δυνατότητα χρήσης για προκαταρκτικές μελέτες έργων ΑΠΕ. Ωστόσο, το μειονέκτημά τους είναι ότι δεν παρέχουν πάντοτε την τοπική λεπτομέρεια που απαιτείται για τελικές μελέτες χωροθέτησης, καθώς στηρίζονται σε μοντέλα μεγάλης κλίμακας (World Bank/ESMAP 2019) <sup>[52]</sup>.

Στην Ευρώπη, σημαντική πρωτοβουλία είναι το European Solar Radiation Atlas (ESRA) και οι μετεξελίξεις του, που υποστηρίζουν την εκτίμηση ηλιακής ακτινοβολίας και την αξιολόγηση φωτοβολταϊκών εφαρμογών. Τα δεδομένα συνδυάζονται με GIS εργαλεία για την απεικόνιση σε τοπικό και περιφερειακό επίπεδο (Cros et al. 2004) <sup>[53]</sup>.

Στον χώρο των συμμετοχικών εφαρμογών, το OpenStreetMap έχει χρησιμοποιηθεί για την καταγραφή μικρών εγκαταστάσεων ΑΠΕ, όπως φωτοβολταϊκά σε στέγες. Μέσω κατάλληλων Web-GIS πλατφορμών, οι πολίτες μπορούν να προσθέσουν πληροφορίες, ενισχύοντας τη διαθεσιμότητα ανοικτών δεδομένων (Neis & Zielstra 2014) <sup>[54]</sup>.

Ένα χαρακτηριστικό παράδειγμα εμπορικής εφαρμογής είναι το Google Project Sunroof, το οποίο χρησιμοποιεί δεδομένα Google Maps και αλγορίθμους GIS για να υπολογίσει την ηλιακή δυναμικότητα στέγης σε αστικές περιοχές. Ο χρήστης μπορεί να εισαγάγει τη διεύθυνσή του και να λάβει εκτίμηση της παραγόμενης ενέργειας και των οικονομικών ωφελειών από εγκατάσταση φωτοβολταϊκών (Google 2017) <sup>[55]</sup>.

Στην Ελλάδα, σημαντική πρωτοβουλία αποτελεί ο Ηλεκτρονικός Χάρτης ΑΠΕ του Υπουργείου Περιβάλλοντος και Ενέργειας (ΥΠΕΝ). Ο χάρτης αυτός παρέχει πληροφορίες για αδειοδοτημένα έργα ΑΠΕ, περιοχές Natura 2000, προστατευόμενες ζώνες και δίκτυα μεταφοράς, υποστηρίζοντας τη διαφάνεια και την ενημέρωση πολιτών και επενδυτών (ΥΠΕΝ 2020) <sup>[56]</sup>.

Άλλη ελληνική εφαρμογή είναι το Solar Cadastre της Θεσσαλονίκης, που αξιολογεί τη δυναμικότητα ηλιακής ενέργειας σε κτίρια, λαμβάνοντας υπόψη σκίαση από γειτονικά κτίρια και μορφολογία εδάφους. Η εφαρμογή αυτή επιτρέπει σε πολίτες και φορείς να εκτιμήσουν τις δυνατότητες εγκατάστασης φωτοβολταϊκών σε επίπεδο γειτονιάς (Chatzigeorgiou et al. 2023) [57].

Στην Ισπανία, η πλατφόρμα Eolica 2.0 παρέχει πληροφορίες για την κατανομή του αιολικού δυναμικού και τις αδειοδοτημένες εγκαταστάσεις ανεμογεννητριών. Η εφαρμογή λειτουργεί με OpenLayers και προσφέρει δεδομένα σε πραγματικό χρόνο για τις τοπικές αρχές και τους επενδυτές (Rodríguez et al. 2015) [58].

Στη Γερμανία, το SolarGIS χρησιμοποιείται ευρέως για την εκτίμηση ηλιακής ακτινοβολίας και την προσομοίωση φωτοβολταϊκών συστημάτων. Η πλατφόρμα παρέχει δεδομένα υψηλής ακρίβειας που βασίζονται σε δορυφορικές παρατηρήσεις και είναι διαθέσιμα μέσω διαδικτυακής πρόσβασης (Šúri et al. 2007) [59].

Στη Δανία, το WindPRO αποτελεί λογισμικό και διαδικτυακό εργαλείο για τον σχεδιασμό αιολικών πάρκων. Περιλαμβάνει δεδομένα GIS για το αιολικό δυναμικό, καθώς και εργαλεία προσομοίωσης παραγωγής και ανάλυσης περιβαλλοντικών επιπτώσεων (EMD International 2019) [60].

Στην Κίνα, εφαρμογές Web-GIS χρησιμοποιούνται για την παρακολούθηση μεγάλων αιολικών πάρκων στην εσωτερική Μογγολία, με δυνατότητα real-time ανάλυσης δεδομένων παραγωγής και συντήρησης μέσω ενσωμάτωσης με IoT αισθητήρες (Huang et al. 2017) [61].

Στη Βόρεια Αμερική, ο National Renewable Energy Laboratory (NREL) έχει αναπτύξει σειρά εργαλείων Web-GIS όπως το PVWatts Calculator, που επιτρέπει σε χρήστες να εκτιμήσουν την παραγωγή φωτοβολταϊκών εγκαταστάσεων και να συγκρίνουν διαφορετικά σενάρια εγκατάστασης (Dobos 2014) [62].

Στη Γαλλία, η εφαρμογή BDPV (Base de Données Photovoltaïque) καταγράφει και απεικονίζει εγκαταστάσεις φωτοβολταϊκών που δηλώνονται από πολίτες, δημιουργώντας μια crowdsourced βάση δεδομένων για την ηλιακή ενέργεια (Lapostolle et al. 2015) [63].

Στην Ινδία, η πλατφόρμα India Solar Resource Atlas αναπτύχθηκε για να υποστηρίξει τον εθνικό σχεδιασμό ΑΠΕ, παρέχοντας γεωχωρικά δεδομένα ηλιακής ακτινοβολίας σε συνδυασμό με κλιματικά και γεωμορφολογικά στοιχεία (GIZ 2015) <sup>[64]</sup>.

Στην Αυστραλία, το AREMI (Australian Renewable Energy Mapping Infrastructure) προσφέρει διαδραστικούς χάρτες για την ηλιακή, αιολική και βιοενέργεια, ενσωματώνοντας δεδομένα από δημόσιες και ιδιωτικές πηγές (Geoscience Australia 2016) <sup>[65]</sup>.

Ένα ακόμη παράδειγμα είναι το RE Atlas της Νότιας Αφρικής, που παρέχει εργαλεία GIS για την αξιολόγηση δυναμικού ΑΠΕ και την υποστήριξη πολιτικών ενεργειακής μετάβασης στη χώρα (CSIR 2018) <sup>[66]</sup>.

Στον Καναδά, η πλατφόρμα RETScreen αποτελεί ένα ολοκληρωμένο εργαλείο ανάλυσης ενεργειακών έργων που συνδυάζει χωρικά δεδομένα με τεχνοοικονομική αξιολόγηση (Natural Resources Canada 2019) <sup>[67]</sup>.

Η Ελλάδα συμμετέχει επίσης σε ευρωπαϊκές πρωτοβουλίες όπως το EMHIREs (European Meteorological High-resolution RES time series) <sup>[68]</sup>, που παρέχει δεδομένα παραγωγής ΑΠΕ σε GIS μορφή, υποστηρίζοντας τον σχεδιασμό των δικτύων ηλεκτρικής ενέργειας (Gonzalez Aparicio et al. 2016) <sup>[69]</sup>.

Οι παραπάνω εφαρμογές δείχνουν ότι τα Web-GIS εργαλεία για ΑΠΕ καλύπτουν διαφορετικά επίπεδα: από παγκόσμια εκτίμηση πόρων, έως εθνικά και τοπικά εργαλεία σχεδιασμού και συμμετοχικής χαρτογράφησης.

Κοινό χαρακτηριστικό όλων είναι ότι αξιοποιούν χωρικά δεδομένα για να απαντήσουν σε ερωτήματα βιωσιμότητας, κόστους και περιβαλλοντικών επιπτώσεων. Παράλληλα, ενισχύουν τη διαφάνεια και την κοινωνική αποδοχή, αφού οι πληροφορίες γίνονται διαθέσιμες σε πολίτες και φορείς (Goodchild 2007) <sup>[70]</sup>.

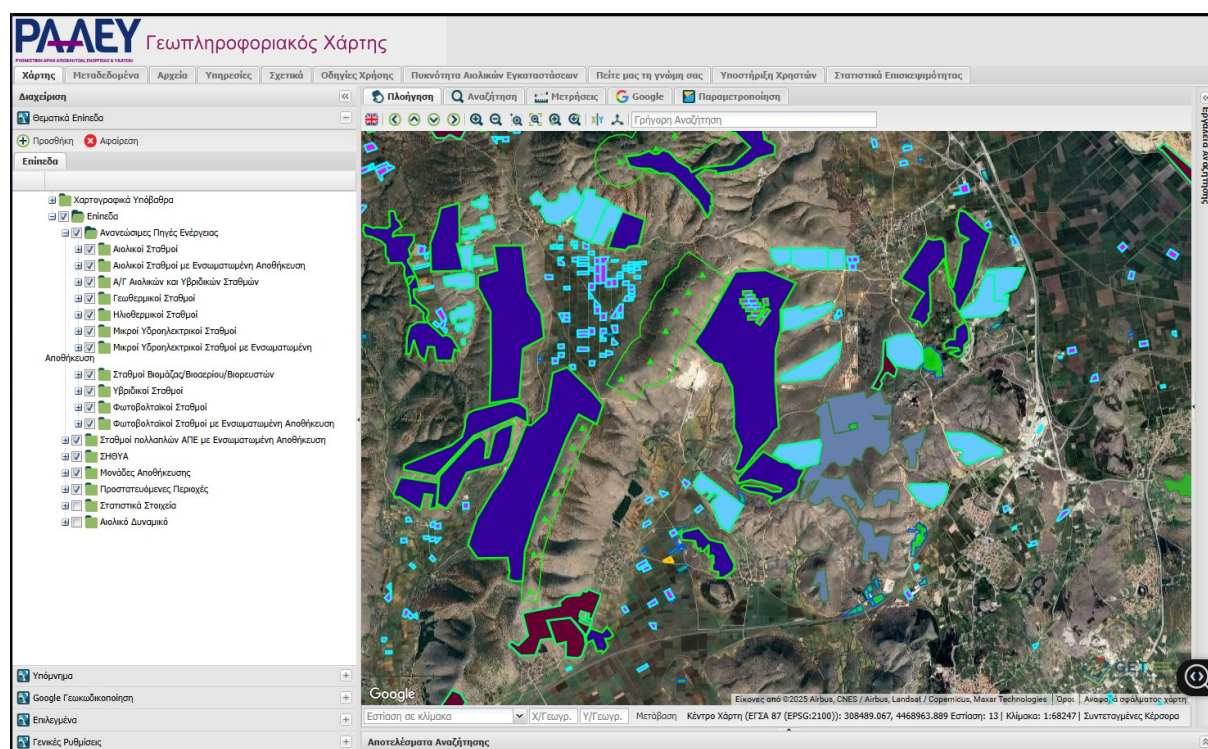
Το βασικό τους πλεονέκτημα είναι η προσβασιμότητα: οι χρήστες δεν χρειάζεται να έχουν εξειδικευμένο λογισμικό GIS, αλλά μπορούν να χρησιμοποιούν ένα απλό πρόγραμμα περιήγησης. Τα μειονεκτήματα σχετίζονται με την εξάρτηση από την ποιότητα των δεδομένων, τις απαιτήσεις υποδομής και την ανάγκη συνεχούς ενημέρωσης.

Συνολικά, οι υπάρχουσες εφαρμογές Web-GIS σε ΑΠΕ αποτελούν πολύτιμα παραδείγματα για το πώς μπορεί να αξιοποιηθεί η τεχνολογία GIS στην πράξη. Η μελέτη τους βοηθά να

εντοπιστούν βέλτιστες πρακτικές αλλά και τα όρια, τα οποία μπορούν να ξεπεραστούν μελλοντικά μέσω νέων τεχνολογιών όπως big data, cloud και AI.

## 1.6. Μελέτη Περίπτωσης: Geoportal του γεωπληροφοριακού χάρτη της PAAEY

Η Ρυθμιστική Αρχή Αποβλήτων, Ενέργειας και Υδάτων (PAAEY), πρώην ΡΑΕ (Ρυθμιστική Αρχή Ενέργειας), έχει αναπτύξει και διαθέτει στο κοινό μια online γεωπληροφοριακή εφαρμογή (Geoportal) μέσω του ιστοτόπου <https://geo.rae.gr>. Η εφαρμογή αυτή αποτελεί χαρακτηριστικό παράδειγμα Web-GIS εργαλείου το οποίο ενσωματώνει γεωχωρικά δεδομένα και πληροφορίες για έργα Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ) στην Ελλάδα.



Εικόνα 1.1. Η ιστοσελίδα του Geoportal της PAAEY

Η εφαρμογή παρέχει στον χρήστη τη δυνατότητα να προσεγγίσει χωρικά δεδομένα μέσω ενός διαδραστικού χάρτη. Ο χάρτης περιλαμβάνει πληροφορίες για αδειοδοτημένα έργα ΑΠΕ, όπως φωτοβολταϊκά, αιολικά, μικρά υδροηλεκτρικά, μονάδες βιομάζας και άλλες μορφές ενεργειακής παραγωγής (Avtar et al. 2019) <sup>[71]</sup>. Κάθε έργο συνοδεύεται από στοιχεία όπως το μέγεθος της εγκατάστασης, την ισχύ, το στάδιο αδειοδότησης και τον φορέα υλοποίησης.

Η οπτικοποίηση αυτών των δεδομένων σε χαρτογραφική μορφή επιτρέπει στους χρήστες να αντιληφθούν τις γεωγραφικές κατανομές των ενεργειακών έργων. Μέσω φίλτρων, μπορεί να επιλεγεί συγκεκριμένος τύπος έργου, στάδιο αδειοδότησης ή χρονικό διάστημα υποβολής αιτήσεων (Bieda 2022) <sup>[72]</sup>. Η δυνατότητα αυτή διευκολύνει ερευνητές, επενδυτές και πολίτες να εντοπίζουν έργα που τους ενδιαφέρουν.

Ένα από τα σημαντικότερα πλεονεκτήματα της εφαρμογής είναι η διαδραστικότητα. Ο χρήστης μπορεί να κάνει zoom, pan και να επιλέγει πολυγωνικά αντικείμενα για να εμφανίσει αναλυτικά χαρακτηριστικά. Η εφαρμογή υποστηρίζει επίσης τη δυνατότητα εξαγωγής δεδομένων, γεγονός που την καθιστά πολύτιμο εργαλείο για μελέτες σκοπιμότητας ή ακαδημαϊκή έρευνα (Goodchild 2009) <sup>[74]</sup>.

Η ΠΑΑΕΥ έχει ενσωματώσει τεχνολογίες ανοιχτών προτύπων, ώστε η εφαρμογή να είναι διαλειτουργική με άλλα GIS εργαλεία. Για παράδειγμα, η χρήση Web Map Service (WMS) και Web Feature Service (WFS) επιτρέπει τη σύνδεση του geoportal με άλλα λογισμικά GIS, όπως QGIS ή ArcGIS (UN-GGIM 2025) <sup>[78]</sup>. Έτσι, χρήστες με εξειδικευμένες ανάγκες μπορούν να ενσωματώσουν τα δεδομένα σε δικά τους αναλυτικά περιβάλλοντα.

Η τεχνολογική υλοποίηση της εφαρμογής βασίζεται σε web mapping βιβλιοθήκες όπως το Leaflet ή το OpenLayers, καθώς και σε server-side πλατφόρμες για τη διαχείριση χωρικών δεδομένων, όπως το GeoServer. Επιπλέον, η χρήση βάσεων δεδομένων όπως το PostgreSQL με επέκταση PostGIS επιτρέπει την αποθήκευση και την αποδοτική ανάκτηση μεγάλου όγκου χωρικών πληροφοριών (Tethys 2023) <sup>[77]</sup>.

Σημαντική είναι επίσης η σύνδεση της εφαρμογής με το εθνικό πλαίσιο χωρικών δεδομένων. Μέσα από το geoportal, ο χρήστης μπορεί να δει θεματικά επίπεδα, όπως προστατευόμενες περιοχές Natura 2000, όρια διοικητικών περιφερειών ή υφιστάμενα δίκτυα μεταφοράς ενέργειας (World Bank/ESMAP 2025) <sup>[79]</sup>. Η ολοκληρωμένη αυτή προσέγγιση επιτρέπει την αξιολόγηση της χωροθέτησης νέων έργων με βάση πολλαπλά κριτήρια.

Η αλληλεπίδραση με τον χρήστη ενισχύεται και μέσω της απλής, φιλικής διεπαφής. Το σύστημα παρέχει εργαλεία αναζήτησης με βάση κωδικό, περιοχή ή είδος έργου. Η διαφάνεια αυτή συμβάλλει στη συμμετοχικότητα και ενισχύει την εμπιστοσύνη του κοινού στις διαδικασίες αδειοδότησης (Participatory GIS, Wikipedia, 2025) <sup>[76]</sup>.

Επιπλέον, η δυνατότητα συνεχούς ενημέρωσης των δεδομένων καθιστά το εργαλείο αξιόπιστο για λήψη αποφάσεων. Τα δεδομένα προέρχονται από τις αιτήσεις που κατατίθενται στην ΡΑΑΕΥ και επικαιροποιούνται τακτικά, διασφαλίζοντας την ακρίβεια και τη συνάφεια των πληροφοριών (Enel Green Power 2023) <sup>[73]</sup>.

Η ύπαρξη τέτοιων εφαρμογών εντάσσεται σε μια διεθνή τάση για διάχυση γεωχωρικών δεδομένων ενέργειας. Παράλληλα με το *Global Solar Atlas* και το *Global Wind Atlas* της World Bank, το geoportal της ΡΑΑΕΥ αποτελεί εθνικό παράδειγμα αξιοποίησης Web-GIS για ενεργειακό σχεδιασμό (World Bank/ESMAP 2025, 2019) <sup>[79]</sup>, <sup>[80]</sup>.

Σε τεχνολογικό επίπεδο, η εφαρμογή μπορεί να επεκταθεί με την ενσωμάτωση εργαλείων ανάλυσης, όπως πολυκριτηριακή ανάλυση καταλληλότητας, προσομοίωση ενεργειακής παραγωγής και σύνδεση με δεδομένα τηλεπισκόπησης (Avtar 2019) <sup>[71]</sup>. Αυτό θα ενίσχυε την αξία της εφαρμογής για στρατηγικό σχεδιασμό.

Η διασύνδεση με ανοικτά δεδομένα αποτελεί ακόμα έναν άξονα βελτίωσης. Πλατφόρμες όπως το *Open Energy System Databases* δείχνουν ότι η ανοιχτή πρόσβαση σε ενεργειακά δεδομένα ενισχύει την καινοτομία και τη συμμετοχή της κοινωνίας (Wikipedia, 2025) <sup>[75]</sup>.

Η εφαρμογή του geo.rae.gr προσφέρει πολλαπλά οφέλη σε διαφορετικές ομάδες χρηστών. Για τους ερευνητές, αποτελεί πηγή δεδομένων για ανάλυση και αξιολόγηση. Για τους επενδυτές, είναι εργαλείο εντοπισμού ευκαιριών. Για τους πολίτες, προάγει τη διαφάνεια και την κατανόηση του ενεργειακού σχεδιασμού.

Συνολικά, η online εφαρμογή του γεωπληροφοριακού χάρτη της ΡΑΑΕΥ συνιστά ένα ώριμο και εξελισσόμενο παράδειγμα χρήσης Web-GIS τεχνολογιών στον τομέα των Ανανεώσιμων Πηγών Ενέργειας. Η συμβολή της στην ορατότητα, στη διαφάνεια και στη λήψη αποφάσεων υπογραμμίζει τη σημασία της γεωπληροφορικής για την ενεργειακή μετάβαση στην Ελλάδα.

## **1.7. Πρότυπα και ρυθμιστικές αρχές για γεωχωρικά δεδομένα**

Η ανάπτυξη και η χρήση γεωχωρικών δεδομένων σε εθνικό και διεθνές επίπεδο απαιτεί την υιοθέτηση προτύπων και κατευθυντήριων αρχών που διασφαλίζουν τη διαλειτουργικότητα και την ποιότητα των δεδομένων. Η ύπαρξη κοινών προτύπων καθιστά εφικτή την ανταλλαγή πληροφοριών μεταξύ διαφορετικών συστημάτων GIS και οργανισμών, μειώνοντας τον

κατακερματισμό των δεδομένων. Η τυποποίηση επιτρέπει επίσης την ομοιογενή αναπαράσταση χωρικών πληροφοριών, γεγονός που συμβάλλει στη διαφάνεια και στην αποτελεσματικότητα της λήψης αποφάσεων (Goodchild, 2009) <sup>[81]</sup>.

Ο Οργανισμός Open Geospatial Consortium (OGC) έχει διαδραματίσει κεντρικό ρόλο στην ανάπτυξη προτύπων για γεωχωρικές υπηρεσίες, όπως τα Web Map Service (WMS), Web Feature Service (WFS) και Web Coverage Service (WCS). Τα πρότυπα αυτά προάγουν τη διαλειτουργικότητα και επιτρέπουν σε διαφορετικά λογισμικά GIS να μοιράζονται και να αξιοποιούν δεδομένα χωρίς την ανάγκη ad hoc προσαρμογών (Percivall, 2008) <sup>[82]</sup>. Η ευρεία αποδοχή τους από κυβερνήσεις και διεθνείς οργανισμούς έχει καταστήσει το OGC σημείο αναφοράς για τις γεωχωρικές υποδομές δεδομένων.

Σε ευρωπαϊκό επίπεδο, η οδηγία INSPIRE (Infrastructure for Spatial Information in Europe) αποτελεί βασικό θεσμικό πλαίσιο για την οργάνωση και διάθεση γεωχωρικών δεδομένων. Η οδηγία καθορίζει τις προδιαγραφές για την παροχή εναρμονισμένων χωρικών δεδομένων μεταξύ των κρατών-μελών, ενισχύοντας την περιβαλλοντική πολιτική, τη χωροταξία και τον σχεδιασμό υποδομών. Η υλοποίηση της οδηγίας έχει οδηγήσει στη δημιουργία εθνικών υποδομών χωρικών δεδομένων, που ακολουθούν κοινούς κανόνες μεταδεδομένων, ποιότητας και πρόσβασης (UN-GGIM, 2015) <sup>[84]</sup>.

Σε διεθνές επίπεδο, τα Ηνωμένα Έθνη μέσω της επιτροπής *United Nations Committee of Experts on Global Geospatial Information Management* (UN-GGIM) αναπτύσσουν κατευθυντήριες αρχές για την ενίσχυση της συνεργασίας στον τομέα της γεωχωρικής πληροφορίας. Οι αρχές αυτές στοχεύουν στην ενδυνάμωση των χωρών σε θέματα χαρτογράφησης, ανταλλαγής δεδομένων και υιοθέτησης διεθνών προτύπων. Η συμβολή του UN-GGIM είναι κρίσιμη για τη διασφάλιση της παγκόσμιας συνέργειας σε γεωχωρικά ζητήματα (UN-GGIM, 2015) <sup>[84]</sup>.

Τέλος, η τήρηση προτύπων και ρυθμιστικών αρχών δεν περιορίζεται σε τεχνικό επίπεδο, αλλά συνδέεται άμεσα με ζητήματα διακυβέρνησης, διαφάνειας και συμμετοχής των πολιτών. Μέσα από εργαλεία όπως η συμμετοχική χαρτογράφηση, ενισχύεται η διάχυση των δεδομένων και η κοινωνική αποδοχή έργων που σχετίζονται με τη χρήση γης και τις Ανανεώσιμες Πηγές Ενέργειας (Goodchild, 2007) <sup>[83]</sup>. Η εναρμόνιση των τεχνολογικών και θεσμικών παραμέτρων οδηγεί έτσι σε πιο βιώσιμα και αποδεκτά συστήματα γεωχωρικής πληροφορίας.

## 1.8. Τα πρότυπα OGC και η σημασία τους στα Web-GIS Συστήματα

Η ανάπτυξη και η διαλειτουργικότητα των συστημάτων Γεωγραφικών Πληροφοριών (GIS) στηρίζονται σε διεθνή πρότυπα που εξασφαλίζουν την ομοιογένεια και τη συμβατότητα μεταξύ διαφορετικών εφαρμογών και πλατφορμών. Ο οργανισμός Open Geospatial Consortium (OGC) αποτελεί τον κύριο φορέα που καθορίζει και συντηρεί αυτά τα πρότυπα. Ιδρύθηκε τη δεκαετία του 1990 και έχει ως βασικό του στόχο να προωθήσει ανοιχτές προδιαγραφές για τη διαχείριση, επεξεργασία και ανταλλαγή γεωχωρικών δεδομένων σε διαφορετικά περιβάλλοντα και λογισμικά (OGC, 2023) <sup>[106]</sup>.

Τα OGC πρότυπα εξασφαλίζουν ότι τα δεδομένα GIS μπορούν να ανταλλάσσονται ανεξάρτητα από την πλατφόρμα ή το λογισμικό που χρησιμοποιεί κάθε φορέας. Αυτό σημαίνει πως δεδομένα που παράγονται, για παράδειγμα, από έναν κρατικό φορέα σε QGIS μπορούν να αξιοποιηθούν χωρίς πρόβλημα σε μια Web-GIS εφαρμογή που βασίζεται στο Leaflet ή στο ArcGIS Online. Η διαλειτουργικότητα αυτή είναι θεμελιώδης για έργα μεγάλης κλίμακας, όπως εθνικά κτηματολόγια ή ενεργειακοί χάρτες, που απαιτούν συνέργεια μεταξύ διαφορετικών πηγών δεδομένων (Goodchild, 2009) <sup>[104]</sup>.

Ένα από τα πιο διαδεδομένα πρότυπα του OGC είναι το WMS (Web Map Service), το οποίο καθορίζει έναν μηχανισμό παροχής χαρτών μέσω διαδικτύου με τη μορφή εικόνων (π.χ. PNG, JPEG, SVG). Ο χρήστης μπορεί να ζητήσει έναν χάρτη για μια συγκεκριμένη περιοχή και το σύστημα επιστρέφει μια στατική απεικόνιση που απεικονίζει το υπόβαθρο ή τα θεματικά επίπεδα ενδιαφέροντος. Το WMS χρησιμοποιείται ευρέως για οπτική αναπαράσταση γεωχωρικών δεδομένων, χωρίς όμως να επιτρέπει την άμεση πρόσβαση ή επεξεργασία των ίδιων των δεδομένων (De la Beaujardière, 2006) <sup>[103]</sup>.

Στην παρούσα εφαρμογή Web-GIS, η χρήση υπηρεσιών τύπου WMS θα μπορούσε να ενισχύσει τη χαρτογραφική εμπειρία, επιτρέποντας την ενσωμάτωση εξωτερικών υποβάθρων (όπως δορυφορικές απεικονίσεις, ορθοφωτοχάρτες ή διοικητικά όρια από δημόσιους φορείς). Με αυτόν τον τρόπο, οι εγκαταστάσεις ΑΠΕ προβάλλονται σε ένα οπτικά πλούσιο πλαίσιο, βελτιώνοντας την ερμηνεία και την αξιοπιστία της χωρικής πληροφορίας.

Ένα άλλο βασικό πρότυπο του OGC είναι το WFS (Web Feature Service), το οποίο επιτρέπει την πρόσβαση σε πραγματικά γεωχωρικά δεδομένα και όχι απλώς σε εικόνες. Μέσω WFS, οι χρήστες μπορούν να κάνουν ερωτήματα, να φιλτράρουν, να επεξεργαστούν και να κατεβάσουν δεδομένα σε μορφή GML (Geography Markup Language) ή GeoJSON. Αυτό καθιστά το WFS

ιδανικό για αναλυτικές και διαδραστικές εφαρμογές, όπου τα δεδομένα δεν είναι απλώς οπτικά, αλλά λειτουργικά και επεξεργάσιμα (OGC, 2021) <sup>[105]</sup>.

Στην περίπτωση της παρούσας εφαρμογής Web-GIS, ένα API βασισμένο σε WFS θα μπορούσε να επιτρέπει στους χρήστες ή στους ερευνητές να εξάγουν δεδομένα των φωτοβολταϊκών εγκαταστάσεων (όπως τοποθεσία, ισχύς μέγιστη ή μέση, εμβαδόν σε m<sup>2</sup>) για περαιτέρω ανάλυση σε GIS ή στατιστικά εργαλεία. Με αυτόν τον τρόπο, η εφαρμογή δεν λειτουργεί μόνο ως απλό εργαλείο απεικόνισης, αλλά και ως κόμβος γεωχωρικής πληροφορίας που προάγει τη γνώση και τη διαφάνεια.

Το τρίτο σημαντικό πρότυπο είναι το WCS (Web Coverage Service), το οποίο επιτρέπει την παροχή ψηφιδογραφικών (bitmap) ή πολυδιάστατων γεωχωρικών δεδομένων, όπως δορυφορικές εικόνες, χάρτες θερμοκρασίας, ηλιακής ακτινοβολίας ή ψηφιακά μοντέλα εδάφους. Σε αντίθεση με το WMS, που παρέχει μόνο την απεικόνιση, το WCS δίνει πλήρη πρόσβαση στα αριθμητικά δεδομένα κάθε pixel, διατηρώντας την πληροφορία τιμών και μονάδων μέτρησης (Baumann, 2018) <sup>[102]</sup>.

Στο πλαίσιο του παρόντος Web-GIS, ένα WCS θα μπορούσε να αξιοποιηθεί για τη χαρτογράφηση του δυναμικού ηλιακής ακτινοβολίας ή της κλίσης του εδάφους, παρέχοντας στους χρήστες τη δυνατότητα να εκτιμήσουν χωρικά τη βιωσιμότητα μιας περιοχής για εγκατάσταση φωτοβολταϊκών. Αυτή η πληροφορία, σε συνδυασμό με τα υπάρχοντα δεδομένα ισχύος και εμβαδού, θα ενίσχυε τη χωρική ανάλυση απόδοσης και τη λήψη αποφάσεων.

Επιπλέον, η υιοθέτηση των προτύπων του OGC επιτρέπει στη συγκεκριμένη εφαρμογή να είναι επεκτάσιμη και διαλειτουργική. Μελλοντικά, μπορεί να διασυνδεθεί με διεθνείς πλατφόρμες όπως το Copernicus Open Access Hub ή το Global Solar Atlas, ενσωματώνοντας δεδομένα χωρίς ανάγκη προσαρμογής ή μετατροπής. Έτσι, η εφαρμογή μετατρέπεται σε ένα ευέλικτο Web-GIS οικοσύστημα που μπορεί να υποστηρίξει ανάλυση, πολιτικό σχεδιασμό και περιβαλλοντική διαχείριση.

Τελικά, τα πρότυπα του OGC αποτελούν τον κορμό της σύγχρονης γεωχωρικής πληροφορικής. Εξασφαλίζουν τη συμβατότητα, την επεκτασιμότητα και τη βιωσιμότητα των Web-GIS εφαρμογών, επιτρέποντας την απρόσκοπτη ανταλλαγή δεδομένων μεταξύ φορέων, οργανισμών και πολιτών. Η ενσωμάτωση αυτών των προτύπων στο παρόν Web-GIS προσφέρει τη δυνατότητα περαιτέρω εξέλιξης της εφαρμογής σε ένα ανοιχτό, συνεργατικό και

τεχνικά ώριμο εργαλείο για τη χαρτογράφηση και ανάλυση των Ανανεώσιμων Πηγών Ενέργειας.

## **1.9. Συμπεράσματα Βιβλιογραφικής Ανασκόπησης**

Η βιβλιογραφική ανασκόπηση ανέδειξε ότι τα Γεωγραφικά Συστήματα Πληροφοριών (GIS) αποτελούν θεμελιώδες εργαλείο για τη διαχείριση και ανάλυση γεωχωρικών δεδομένων, με ιδιαίτερη σημασία στον τομέα των Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ). Η δυνατότητα συλλογής, αποθήκευσης, επεξεργασίας και οπτικοποίησης δεδομένων σε συνδυασμό με διαδικτυακές τεχνολογίες καθιστά τα GIS απαραίτητη υποδομή για τον σχεδιασμό και τη χωροθέτηση ενεργειακών έργων.

Η ανάλυση των διαθέσιμων GIS frameworks (Leaflet.js, OpenLayers, Google Maps API) έδειξε ότι υπάρχει πληθώρα τεχνολογικών επιλογών, καθεμία με τα δικά της πλεονεκτήματα και περιορισμούς. Τα ελαφριά εργαλεία, όπως το Leaflet.js, προσφέρονται για γρήγορες και απλές εφαρμογές, ενώ πιο ισχυρές βιβλιοθήκες όπως το OpenLayers υποστηρίζουν σύνθετες γεωχωρικές λειτουργίες. Από την άλλη πλευρά, εμπορικές λύσεις όπως το Google Maps API προσφέρουν ευκολία και ποιότητα δεδομένων, με περιορισμούς όμως στη διαλειτουργικότητα και στο κόστος.

Οι βάσεις δεδομένων χωρικών δεδομένων (PostGIS, MongoDB) αποτελούν κρίσιμο στοιχείο για την αποθήκευση και ανάλυση γεωχωρικών πληροφοριών. Το PostGIS ξεχωρίζει για την πληρότητα και την υποστήριξη προτύπων OGC, καθιστώντας το ιδανικό για χωρικές αναλύσεις μεγάλης ακρίβειας. Αντίθετα, το MongoDB παρέχει ευελιξία και επεκτασιμότητα για real-time δεδομένα και big data περιβάλλοντα. Ο συνδυασμός των δύο μπορεί να δώσει λύσεις σε εφαρμογές που απαιτούν τόσο αναλυτική ακρίβεια όσο και γρήγορη απόκριση σε μεγάλους όγκους δεδομένων.

Η μελέτη των client-server αρχιτεκτονικών κατέδειξε ότι αποτελούν τον βασικό κορμό λειτουργίας των Web-GIS εφαρμογών. Μέσα από την αλληλεπίδραση client και server, καθίσταται εφικτή η διάχυση γεωχωρικής πληροφορίας σε χρήστες ανεξάρτητα από την τοποθεσία τους. Η εξέλιξη προς cloud-based υποδομές και RESTful APIs ενισχύει την ευελιξία, την κλιμάκωση και τη δυνατότητα ενσωμάτωσης με άλλα συστήματα.

Η ανασκόπηση υπαρχουσών εφαρμογών Web-GIS σε ΑΠΕ, τόσο σε διεθνές όσο και σε ελληνικό επίπεδο, ανέδειξε την πρακτική αξία αυτών των εργαλείων. Παραδείγματα όπως το Global Solar Atlas, το Google Project Sunroof, ο Ηλεκτρονικός Χάρτης ΑΠΕ του ΥΠΕΝ και το Solar Cadastre Θεσσαλονίκης δείχνουν ότι τέτοιες πλατφόρμες ενισχύουν τη λήψη αποφάσεων, την ενημέρωση του κοινού και τη διαφάνεια στη διαδικασία χωροθέτησης. Παράλληλα, καθιστούν προσιτή την πληροφορία ακόμη και σε μη ειδικούς χρήστες.

Συνολικά, η βιβλιογραφική ανασκόπηση δείχνει ότι οι Web-GIS τεχνολογίες έχουν τη δυνατότητα να λειτουργήσουν ως κρίσιμος μοχλός για την ενεργειακή μετάβαση. Η συνδυαστική αξιοποίηση των κατάλληλων frameworks, βάσεων δεδομένων και αρχιτεκτονικών, σε συνδυασμό με την αξιοποίηση ανοικτών δεδομένων, μπορεί να οδηγήσει στην ανάπτυξη καινοτόμων εφαρμογών. Αυτές οι εφαρμογές δεν υποστηρίζουν μόνο τον τεχνικό σχεδιασμό έργων ΑΠΕ αλλά και τη συμμετοχική διακυβέρνηση, ενισχύοντας τη βιωσιμότητα και την κοινωνική αποδοχή των έργων.

## 2. Ανάπτυξη της GIS γεωπληροφοριακής εφαρμογής

### 2.1. Ανάπτυξη της εφαρμογής ως PoC (Proof of Concept).

Η ανάπτυξη μιας εφαρμογής ως Proof of Concept (PoC) αποτελεί μια κοινή πρακτική στον χώρο της πληροφορικής και της μηχανικής λογισμικού. Το PoC είναι μια αρχική υλοποίηση που στοχεύει να αποδείξει την τεχνική εφικτότητα μιας ιδέας, να επιβεβαιώσει την καταλληλότητα συγκεκριμένων τεχνολογιών και να δώσει μια πρώτη απτή μορφή σε μια θεωρητική σύλληψη. Δεν πρόκειται για ένα τελικό προϊόν, αλλά για ένα πειραματικό στάδιο που λειτουργεί ως «απόδειξη» ότι οι βασικές αρχές και τεχνικές μπορούν να εφαρμοστούν με επιτυχία (Ries, 2011; Dutta et al., 2017) <sup>[85]</sup> <sup>[86]</sup>.

Χαρακτηριστικό ενός PoC είναι η εστίαση στον πυρήνα της λειτουργικότητας. Αυτό σημαίνει ότι περιλαμβάνει μόνο τις απολύτως απαραίτητες δυνατότητες που απαιτούνται ώστε να επιβεβαιωθεί ότι η τεχνολογική προσέγγιση είναι εφαρμόσιμη. Στην περίπτωση μιας Web-GIS εφαρμογής, αυτό μπορεί να σημαίνει ότι ο χάρτης μπορεί να αποδώσει σωστά τα δεδομένα, ότι η βάση μπορεί να αποθηκεύσει και να ανακτήσει χωρικές πληροφορίες και ότι η διεπαφή χρήστη επιτρέπει βασική αλληλεπίδραση. Ωστόσο, τα συστήματα ελέγχου εγκυρότητας δεδομένων, η ασφάλεια, η πλήρης εργονομία και η αντοχή σε μεγάλο φόρτο χρηστών συνήθως δεν είναι προτεραιότητα σε αυτό το στάδιο (Croll & Yoskovitz, 2013) <sup>[87]</sup>.

Οι περιορισμοί ενός PoC είναι έμφυτοι στη φύση του. Επειδή δημιουργείται σε σύντομο χρονικό διάστημα με στόχο την ταχεία δοκιμή μιας ιδέας, συχνά υστερεί σε θέματα όπως η ολοκληρωμένη τεκμηρίωση, η σταθερότητα, η συμβατότητα με ευρύτερα συστήματα ή η ευχρηστία για τελικούς χρήστες. Επιπλέον, μπορεί να χρησιμοποιεί απλουστευμένα σχήματα βάσης δεδομένων, περιορισμένους ελέγχους εγκυρότητας ή ακόμα και «σκληρά» δεδομένα (hard-coded values) για να λειτουργήσει. Αυτοί οι περιορισμοί δεν υποβαθμίζουν τον ρόλο του, αλλά τονίζουν ότι πρόκειται για βήμα ενδιάμεσο πριν την ανάπτυξη ενός ολοκληρωμένου Minimum Viable Product (MVP) ή τελικού λογισμικού (Blank & Dorf, 2012) <sup>[88]</sup>.

Παρά τους περιορισμούς του, το PoC προσφέρει σημαντικά οφέλη. Δίνει τη δυνατότητα στους δημιουργούς να ελέγξουν τις τεχνολογικές επιλογές σε πραγματικό περιβάλλον, να εντοπίσουν έγκαιρα πιθανά προβλήματα και να λάβουν ανατροφοδότηση από χρήστες ή συνεργάτες σε πολύ πρώιμο στάδιο. Επίσης, συμβάλλει στη μείωση του ρίσκου, καθώς μια ιδέα μπορεί να απορριφθεί ή να βελτιωθεί πριν επενδυθούν σημαντικοί πόροι στην πλήρη ανάπτυξή της. Στην περίπτωση της παρούσας Web-GIS εφαρμογής, η PoC προσέγγιση επιτρέπει την επαλήθευση

ότι οι τεχνολογίες όπως React, Leaflet, PostgreSQL/PostGIS και Supabase μπορούν να συνεργαστούν αποτελεσματικά, θέτοντας έτσι τα θεμέλια για περαιτέρω ανάπτυξη και επέκταση της λειτουργικότητας (Olsson & Bosch, 2015) <sup>[89]</sup>.

## **2.2. Τεχνολογίες, εργαλεία και υπηρεσίες που χρησιμοποιούνται.**

Στην ενότητα αυτή παρουσιάζονται αναλυτικά οι βασικές τεχνολογίες, οι βιβλιοθήκες καθώς και οι διαδικτυακές υπηρεσίες που αξιοποιήθηκαν για την ανάπτυξη της εφαρμογής. Η επιλογή των συγκεκριμένων τεχνολογιών δεν είναι τυχαία, αλλά έγινε με γνώμονα την ευχρηστία, την αξιοπιστία, την αποδοτικότητα και τη δυνατότητα μελλοντικής επεκτασιμότητας του συστήματος.

Η ανάπτυξη ενός Web-GIS συστήματος προϋποθέτει τον συνδυασμό εργαλείων που καλύπτουν διαφορετικά επίπεδα λειτουργικότητας: από την απεικόνιση γεωχωρικών δεδομένων στον χάρτη και την αλληλεπίδραση με τον χρήστη, μέχρι την αποθήκευση, διαχείριση και ανάλυση των δεδομένων στον εξυπηρετητή (server). Για τον σκοπό αυτό χρησιμοποιήθηκαν τόσο τεχνολογίες ανοικτού κώδικα όσο και ευρέως διαδεδομένες υπηρεσίες διαδικτύου, που εξασφαλίζουν τη διαλειτουργικότητα και την ευκολία χρήσης.

Στα παρακάτω παρατίθενται τα εργαλεία και οι υπηρεσίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής Web-GIS:

### **2.2.1 React**

Η React (<https://react.dev>) είναι μια βιβλιοθήκη JavaScript που αναπτύχθηκε από το Facebook για τη δημιουργία διαδραστικών διεπαφών χρήστη (UIs). Χρησιμοποιεί την έννοια των components, δηλαδή επαναχρησιμοποιήσιμων δομικών στοιχείων, και το virtual DOM για αποδοτικό rendering. Η δημοτικότητά του οφείλεται στην ευκολία δημιουργίας δυναμικών και responsive web εφαρμογών, καθώς και στη μεγάλη κοινότητα και οικοσύστημά του (Griffith, 2021) <sup>[90]</sup>.

Στην παρούσα εφαρμογή, το React χρησιμοποιείται για τη δόμηση του front-end περιβάλλοντος και της διεπαφής του χρήστη. Μέσα από components υλοποιούνται λειτουργίες όπως η εισαγωγή νέων εγκαταστάσεων, η εμφάνιση χαρτών, και η δυναμική ενημέρωση δεδομένων σε πραγματικό χρόνο. Το React επιτρέπει την αποδοτική διαχείριση καταστάσεων (state management) που απαιτεί η αλληλεπίδραση με το GIS υπόβαθρο.

### 2.2.2 TypeScript

Το TypeScript (<https://www.typescriptlang.org>) είναι ένα υπερσύνολο της JavaScript που εισάγει στατική τυποποίηση (types), με αποτέλεσμα τη βελτίωση της αναγνωσιμότητας και συντηρησιμότητας του κώδικα. Επιτρέπει την έγκαιρη ανίχνευση λαθών κατά το στάδιο της ανάπτυξης και διευκολύνει την κλιμάκωση μεγάλων έργων. Η χρήση του έχει αυξηθεί σημαντικά τα τελευταία χρόνια, ειδικά σε σύνθετες εφαρμογές (Bierman et al., 2014) <sup>[91]</sup>.

Στην εφαρμογή Web-GIS, το TypeScript χρησιμοποιείται σε όλα τα μέρη του front-end για να διασφαλίσει την ορθότητα των δεδομένων που ανταλλάσσονται μεταξύ χαρτών, API και βάσης. Για παράδειγμα, οι τύποι που δηλώνονται για τα γεωχωρικά δεδομένα εξασφαλίζουν ότι οι γεωμετρίες και οι τιμές εμφανίζονται σωστά στον χάρτη και αποθηκεύονται αξιόπιστα στη βάση.

### 2.2.3 Vite

Το Vite (<https://vite.dev>) είναι ένα σύγχρονο build tool και development server που σχεδιάστηκε για ταχύτητα και αποδοτικότητα. Χρησιμοποιεί native ES modules για την ανάπτυξη, και προσφέρει ταχύτατο HMR (Hot Module Replacement), βελτιστοποιώντας την εμπειρία ανάπτυξης (You, 2021) <sup>[92]</sup>. Σε σχέση με παραδοσιακά εργαλεία όπως το Webpack, είναι σημαντικά πιο ελαφρύ και γρήγορο.

Στην εργασία αυτή, το Vite χρησιμοποιείται για το build και το bundling του κώδικα React/TypeScript, επιτρέποντας γρήγορη ανάπτυξη και εύκολη διάθεση της εφαρμογής στο Netlify. Η χρήση του καθιστά την εφαρμογή πιο ευέλικτη, καθώς οι αλλαγές στον κώδικα ενσωματώνονται άμεσα και οι τελικοί χρήστες λαμβάνουν ένα βελτιστοποιημένο πακέτο.

### 2.2.4 Leaflet

Το Leaflet (<https://leafletjs.com>) είναι μια ανοιχτού κώδικα βιβλιοθήκη JavaScript για χαρτογραφικές εφαρμογές στο διαδίκτυο. Είναι ελαφριά, επεκτάσιμη και προσφέρει λειτουργικότητες όπως προβολή πλακιδίων (tiles), markers, polygons και αλληλεπιδράσεις με τον χάρτη. Χρησιμοποιείται ευρέως σε εφαρμογές GIS λόγω της απλότητας και της επεκτασιμότητάς του (Agafonkin, 2011) <sup>[93]</sup>.

Στην εφαρμογή Web-GIS, το Leaflet αποτελεί τον πυρήνα της χαρτογραφικής απεικόνισης. Μέσω αυτού προβάλλονται οι εγκαταστάσεις ΑΠΕ πάνω στο υπόβαθρο, ενώ ο χρήστης μπορεί να δει, να εισάγει και να επεξεργαστεί πολύγωνα που απεικονίζουν τις εγκαταστάσεις. Οι λειτουργίες zoom, pan και επιλογής περιοχών διευκολύνουν την πλοήγηση και την ανάλυση.

### 2.2.5 PostgreSQL / PostGIS

Το PostgreSQL ( <https://www.postgresql.org> και <https://postgis.net> ) είναι ένα ισχυρό σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων ανοικτού κώδικα, γνωστό για την αξιοπιστία και την υποστήριξη πολύπλοκων ερωτημάτων.

Το PostGIS είναι επέκταση του PostgreSQL που προσθέτει χωρική υποστήριξη και επιτρέπει την αποθήκευση και ανάλυση γεωχωρικών δεδομένων (Obe & Hsu, 2011) <sup>[94]</sup>.

Στην παρούσα εφαρμογή, το PostgreSQL/PostGIS χρησιμοποιείται για την αποθήκευση των γεωχωρικών δεδομένων των εγκαταστάσεων ΑΠΕ. Τα πολύγωνα που εισάγονται από τον χρήστη αποθηκεύονται ως γεωμετρίες, και το PostGIS επιτρέπει τον υπολογισμό χαρακτηριστικών όπως το εμβαδό. Έτσι, η βάση λειτουργεί όχι μόνο ως αποθήκη αλλά και ως μηχανή ανάλυσης.

### 2.2.6 Supabase

Το Supabase (<https://supabase.com>) είναι μια backend-as-a-service πλατφόρμα που βασίζεται στο PostgreSQL και παρέχει εργαλεία για αυθεντικοποίηση, API δημιουργία, real-time δεδομένα και διαχείριση αποθηκευμένων αρχείων. Θεωρείται ως μια «ανοιχτού κώδικα εναλλακτική» του Firebase (Supabase, 2023) <sup>[95]</sup>.

Στην εφαρμογή Web-GIS, το Supabase χρησιμοποιείται για την επικοινωνία του front-end με τη βάση PostgreSQL/PostGIS. Μέσω των APIs του, επιτυγχάνεται η εισαγωγή νέων εγκαταστάσεων, η άντληση δεδομένων και η δυναμική εμφάνιση των αποτελεσμάτων στον χάρτη. Επιπλέον, το Supabase εξασφαλίζει την ασφαλή διαχείριση χρηστών και τη συνδεσιμότητα με την απομακρυσμένη βάση.

### 2.2.7 GitHub

Το GitHub (<https://github.com>) είναι η μεγαλύτερη πλατφόρμα φιλοξενίας κώδικα και συνεργασίας για έργα λογισμικού, βασισμένη στο σύστημα ελέγχου εκδόσεων Git. Χρησιμοποιείται για την αποθήκευση, τον έλεγχο εκδόσεων, και τη συνεργασία μεταξύ προγραμματιστών (Loeliger & McCullough, 2012) <sup>[96]</sup>.

Στην παρούσα εργασία, το GitHub χρησιμοποιείται για τη διαχείριση του κώδικα της εφαρμογής, την παρακολούθηση αλλαγών και τη συνεργασία. Κάθε αλλαγή στον κώδικα καταγράφεται σε commits, ενώ η σύνδεση με το Netlify επιτρέπει την αυτόματη ανάπτυξη (CI/CD) με κάθε ενημέρωση του repository.

### 2.2.8 Netlify

Το Netlify (<https://www.netlify.com>) είναι μια πλατφόρμα συνεχούς ανάπτυξης (CI/CD) και φιλοξενίας web εφαρμογών. Υποστηρίζει αυτόματες αναπτύξεις από Git repositories, serverless functions, και παρέχει παγκόσμιο CDN για γρήγορη διάθεση του περιεχομένου (Netlify, 2023) <sup>[97]</sup>.

Στην παρούσα εφαρμογή Web-GIS, το Netlify αποτελεί την πλατφόρμα ανάπτυξης και διάθεσης. Κάθε ενημέρωση που ανεβαίνει στο GitHub ενεργοποιεί αυτόματα build και deploy διαδικασίες στο Netlify, ώστε η εφαρμογή να είναι διαθέσιμη online. Έτσι διασφαλίζεται γρήγορη ανάπτυξη και συνεχής διαθεσιμότητα.

### 2.2.9 Node.js

Το Node.js (<https://nodejs.org/en>) είναι ένα περιβάλλον εκτέλεσης JavaScript από την πλευρά του διακομιστή, που βασίζεται στη μηχανή V8 της Google Chrome. Παρέχει μηχανισμούς ασύγχρονης εκτέλεσης και event-driven αρχιτεκτονική, καθιστώντας το ιδιαίτερα αποδοτικό για εφαρμογές με μεγάλο αριθμό ταυτόχρονων αιτημάτων.

Χρησιμοποιείται ευρέως για ανάπτυξη web servers, APIs, real-time εφαρμογών, αλλά και για εργαλεία ανάπτυξης όπως build systems και task runners (Tilkov & Vinoski, 2010) <sup>[98]</sup>.

Στην παρούσα εφαρμογή Web-GIS, το Node.js χρησιμοποιείται κυρίως στο backend οικοσύστημα που υποστηρίζει την ανάπτυξη και διάθεση του κώδικα. Μέσω του Node.js εκτελούνται εργαλεία όπως το Vite, ενώ οι Netlify Functions βασίζονται σε περιβάλλον Node.js για την υλοποίηση serverless λειτουργιών, όπως η επικοινωνία με τη βάση δεδομένων μέσω Supabase. Έτσι, το Node.js εντάσσεται στην υποδομή της εφαρμογής ως θεμέλιο για την εκτέλεση εργαλείων και serverless API endpoints.

#### 2.2.10 Json και GeoJSON

Το JSON (JavaScript Object Notation) ([www.json.org](http://www.json.org) και <https://geojson.org>) είναι μια ελαφριά και ευρέως χρησιμοποιούμενη μορφή ανταλλαγής δεδομένων, βασισμένη στη σύνταξη της JavaScript αλλά ανεξάρτητη από συγκεκριμένη γλώσσα προγραμματισμού. Χρησιμοποιεί απλή δομή «κλειδιού–τιμής», επιτρέποντας την αποθήκευση και μεταφορά δεδομένων με τρόπο που είναι ευανάγνωστος από τον άνθρωπο και εύκολα επεξεργάσιμος από μηχανές (Bray, 2014) <sup>[99]</sup>.

Η δημοτικότητά του προκύπτει από την ικανότητά του να αναπαριστά πολύπλοκες δομές όπως λίστες και πίνακες με απλό τρόπο, καθιστώντας το πρότυπο εργαλείο για web εφαρμογές και APIs, καθώς και για την επικοινωνία μεταξύ frontend και backend συστημάτων (Nurseitov et al., 2009) <sup>[100]</sup>.

Στην παρούσα εφαρμογή, το JSON αξιοποιείται ως ο κύριος μηχανισμός ανταλλαγής δεδομένων μεταξύ του frontend (React/Leaflet) και του backend (PostgreSQL/PostGIS μέσω Supabase και Netlify Functions). Κατά την εισαγωγή νέας εγκατάστασης ΑΠΕ, τα δεδομένα (π.χ. code, power\_max, region, coords, wkt) πακετάρονται σε JSON αντικείμενο και αποστέλλονται μέσω HTTP request προς τον server. Αντίστοιχα, τα δεδομένα που επιστρέφουν από τη βάση, ιδίως οι γεωμετρικές πληροφορίες, αποδίδονται σε μορφή GeoJSON, το οποίο αποτελεί εξειδίκευση του JSON για γεωχωρικά δεδομένα (Butler et al., 2016) <sup>[101]</sup>.

Αυτό επιτρέπει την άμεση απεικόνιση πολυγώνων και σημείων στον χάρτη, καθιστώντας το JSON το «γλωσσικό» μέσο που συνδέει δυναμικά τη βάση δεδομένων με την οπτική χαρτογραφική απεικόνιση.

### 2.3. Διαδικασία ανάπτυξης της εφαρμογής

Η ανάπτυξη της εφαρμογής πραγματοποιήθηκε ακολουθώντας μια συστηματική διαδικασία, η οποία ξεκινά από τη φάση της ανάλυσης απαιτήσεων και οδηγεί στον καθορισμό των λειτουργικών και μη λειτουργικών προδιαγραφών, θέτοντας τα θεμέλια για την τελική υλοποίηση. Η μεθοδολογία που υιοθετήθηκε βασίστηκε σε μια ευέλικτη προσέγγιση ανάπτυξης (Agile), προσαρμοσμένη στις ανάγκες ενός Proof of Concept (PoC), με στόχο τη σταδιακή επαλήθευση της τεχνικής εφικτότητας και τη διασφάλιση της συνέπειας μεταξύ τεχνολογιών, βάσεων δεδομένων και χαρτογραφικής διεπαφής — μια προσέγγιση σύμφωνη με τη χρήση spikes / PoC στην Agile μεθοδολογία για την έγκαιρη αξιολόγηση τεχνικών λύσεων. (H. Al Hashimi et al., 2019) <sup>[111]</sup>.

Η ανάλυση απαιτήσεων αποτέλεσε το πρώτο στάδιο της διαδικασίας ανάπτυξης και περιλάμβανε τη συλλογή, καταγραφή και κατηγοριοποίηση των αναγκών που έπρεπε να καλύπτει η εφαρμογή. Η φάση αυτή στόχευε στον εντοπισμό των χρηστών-στόχων (π.χ. ερευνητές, φορείς ενέργειας, πολίτες) και των βασικών λειτουργιών που θα υποστήριζε το Web-GIS σύστημα, όπως η απεικόνιση, εισαγωγή, επεξεργασία και φιλτράρισμα δεδομένων έργων ΑΠΕ. Εξετάστηκαν επίσης οι τύποι δεδομένων που θα διαχειριζόταν η εφαρμογή όπως πολύγωνα, σημεία και μεταδεδομένα για κάθε ενεργειακή εγκατάσταση — καθώς και οι πηγές προέλευσης (βάσεις PostgreSQL/PostGIS, APIs, αρχεία GeoJSON).

Με βάση τα αποτελέσματα της ανάλυσης, ακολούθησε η διαμόρφωση των λειτουργικών απαιτήσεων. Οι λειτουργικές απαιτήσεις περιγράφουν το σύνολο των ενεργειών που πρέπει να μπορεί να εκτελεί το σύστημα από την πλευρά του χρήστη και του διαχειριστή. Ενδεικτικά, καθορίστηκαν λειτουργίες όπως:

- η προβολή χαρτογραφικού υποβάθρου μέσω Leaflet,
- η δυναμική φόρτωση και απεικόνιση των εγκαταστάσεων ΑΠΕ,
- η δυνατότητα εισαγωγής νέας εγκατάστασης μέσω φόρμας,
- η αποθήκευση των γεωμετρικών δεδομένων στη βάση PostgreSQL/PostGIS,
- η ανανέωση του χάρτη σε πραγματικό χρόνο μέσω Supabase APIs,
- και η φιλτραρισμένη αναζήτηση έργων ανά κατηγορία, ισχύ ή περιοχή.

Οι λειτουργικές αυτές προδιαγραφές τεκμηριώθηκαν λεπτομερώς, ώστε να εξασφαλιστεί η σωστή αλληλεπίδραση μεταξύ των επιμέρους επιπέδων της εφαρμογής — διεπαφής χρήστη, διακομιστή και βάσης δεδομένων.

Παράλληλα, καθορίστηκαν οι μη λειτουργικές προδιαγραφές, που αφορούν την ποιότητα, την απόδοση και τη συνολική εμπειρία χρήσης. Αυτές περιλαμβάνουν:

- Απόδοση: ο χάρτης και τα δεδομένα να φορτώνουν εντός 2–3 δευτερολέπτων σε τυπική σύνδεση δικτύου,
- Δυνατότητα κλιμάκωσης: η εφαρμογή να μπορεί να υποστηρίξει μελλοντική επέκταση (π.χ. προσθήκη νέων τύπων δεδομένων ή χρηστών),
- Αξιοπιστία: οι λειτουργίες CRUD (Create, Read, Update, Delete) να εκτελούνται χωρίς απώλεια δεδομένων,
- Διαλειτουργικότητα: συμβατότητα με διεθνή πρότυπα OGC (WMS, WFS, GeoJSON),
- Ασφάλεια: ασφαλής διαχείριση δεδομένων και ελεγχόμενη πρόσβαση χρηστών μέσω του Supabase authentication,
- Ευχρηστία: καθαρή και φιλική διεπαφή με έμφαση στην ευκολία πλοήγησης και στη σαφή παρουσίαση των δεδομένων.

Στη συνέχεια, πραγματοποιήθηκε η ανάλυση αρχιτεκτονικής του συστήματος, προκειμένου να προσδιοριστεί η ακριβής διάρθρωση και ροή δεδομένων. Η εφαρμογή οργανώθηκε σε τρία βασικά επίπεδα:

- α. το front-end (React/TypeScript/Leaflet), όπου υλοποιείται η αλληλεπίδραση με τον χρήστη και η χαρτογραφική απεικόνιση,
- β. το back-end (Supabase APIs, Node.js, Netlify Functions), το οποίο διαχειρίζεται τη λογική των αιτημάτων, την επικοινωνία με τη βάση και τη διαχείριση εξουσιοδοτήσεων,
- γ. τη βάση δεδομένων (PostgreSQL/PostGIS), όπου αποθηκεύονται οι γεωμετρικές και περιγραφικές πληροφορίες.

Η δομή αυτή διασφαλίζει την ανεξαρτησία των επιπέδων και επιτρέπει εύκολη συντήρηση και επεκτασιμότητα.

Στο επόμενο στάδιο, υλοποιήθηκε ένα διάγραμμα ροής δεδομένων (Data Flow Diagram - DFD) για να αποτυπωθεί η επικοινωνία μεταξύ χρήστη, server και βάσης. Το διάγραμμα αυτό βοήθησε στη χαρτογράφηση των βασικών διαδικασιών: από την εισαγωγή ενός νέου έργου ΑΠΕ, την αποστολή των δεδομένων στο API, την αποθήκευσή τους στη βάση PostGIS και την επαναφορά τους στο Leaflet για οπτικοποίηση στον χάρτη.

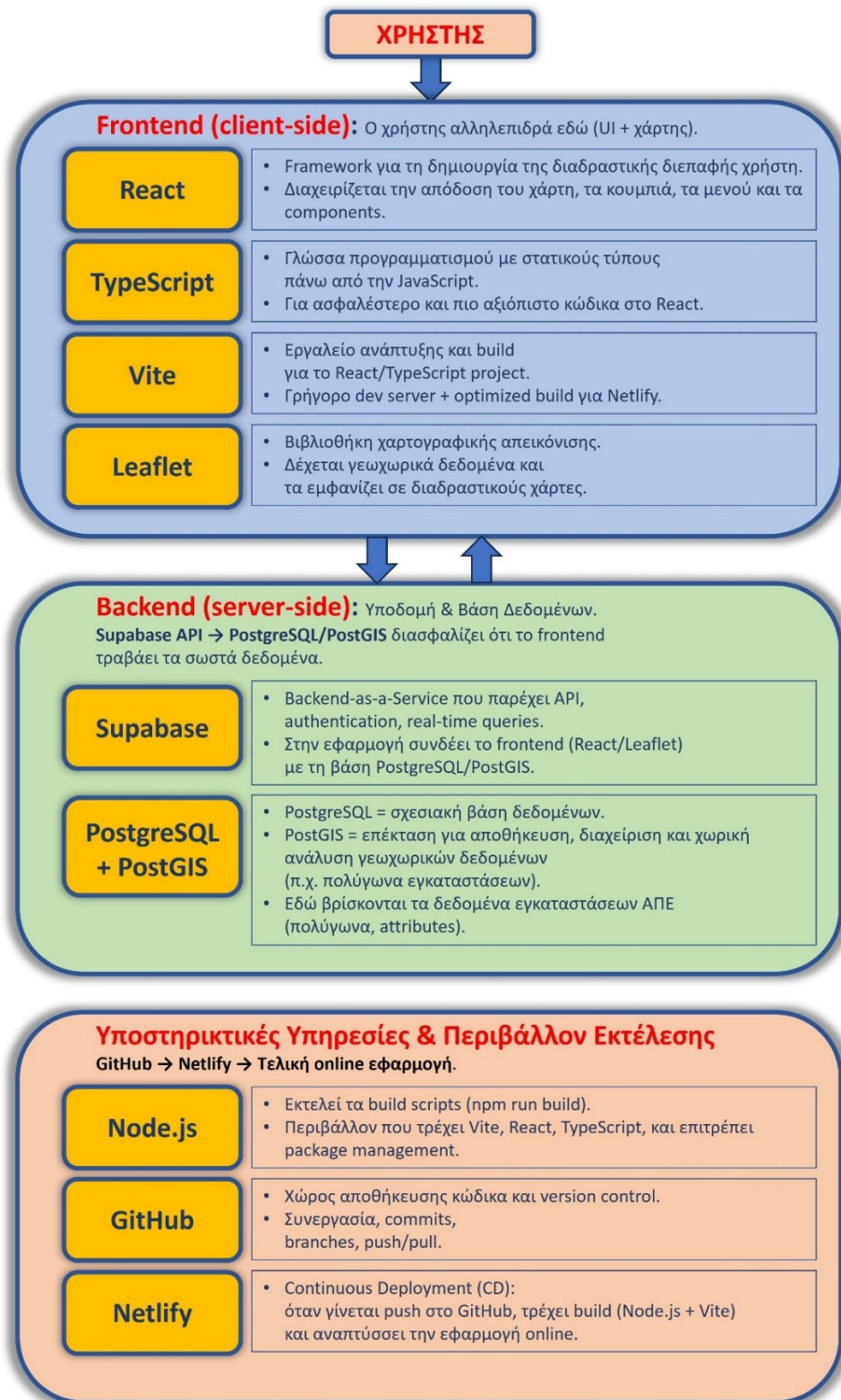
Παράλληλα, αναπτύχθηκαν mockups για τη διεπαφή χρήστη, ώστε να διασφαλιστεί η εργονομία και η συνοχή του σχεδιασμού.

Η φάση της υλοποίησης βασίστηκε στην αρχή της επαναληπτικής ανάπτυξης (iterative development). Σε κάθε κύκλο ανάπτυξης, επιλεγόταν μια μικρή ομάδα λειτουργιών (π.χ. προβολή χάρτη, εισαγωγή έργου, δυναμική ενημέρωση) που αναπτύσσονταν, ελέγχονταν και ενσωματώνονταν σταδιακά. Με τον τρόπο αυτό, εξασφαλίστηκε η σταδιακή σταθεροποίηση της εφαρμογής και η δυνατότητα εντοπισμού και επίλυσης σφαλμάτων σε πρώιμο στάδιο.

Τέλος, η φάση ελέγχου και επαλήθευσης (testing and validation) περιλάμβανε τη δοκιμή των λειτουργιών τόσο από την πλευρά του τελικού χρήστη (UI testing) όσο και του back-end (API testing). Ελέγχθηκαν η ορθότητα των χωρικών συναρτήσεων PostGIS, η εγκυρότητα των δεδομένων GeoJSON και η συνέπεια της οπτικοποίησης στο Leaflet.

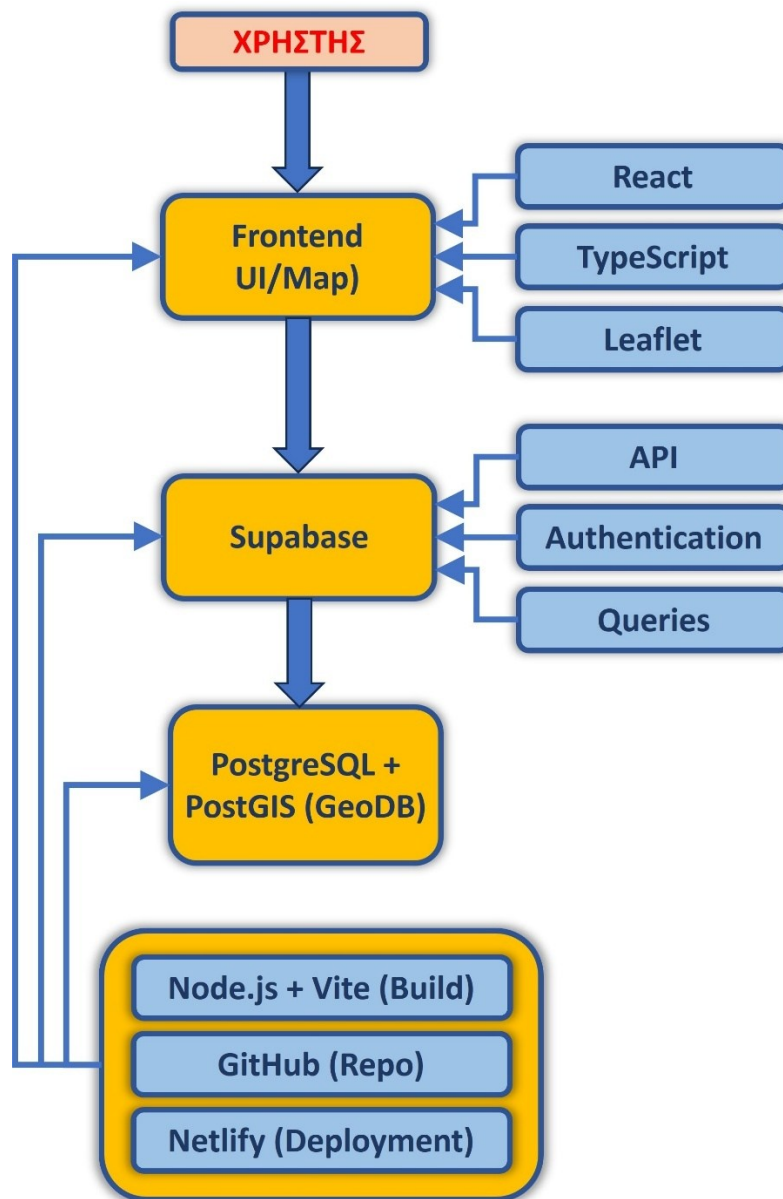
Η επιτυχής ολοκλήρωση των δοκιμών επιβεβαίωσε ότι η εφαρμογή ικανοποιεί τις απαιτήσεις και πληροί τις λειτουργικές και μη λειτουργικές προδιαγραφές που τέθηκαν στην αρχή.

Στην παρακάτω εικόνα παρατίθενται με οπτικοποιημένο τρόπο οι διασυνδέσεις των βασικών συστατικών της εφαρμογής.



Σχήμα 2.1. Οπτικοποιημένη περιγραφή των διασυνδέσεων των συστατικών της εφαρμογής

Στην εικόνα που ακολουθεί παρατίθεται η ροή ανάπτυξης της εφαρμογής με χρήση των εργαλείων και υπηρεσιών που έχουν προαναφερθεί.



Σχήμα 2.2. Η ροή της ανάπτυξης της εφαρμογής

- Ο Χρήστης αλληλεπιδρά με το Frontend (React, TypeScript, Leaflet).
- Το Frontend επικοινωνεί με το Supabase (API/Auth/Queries).
- Το Supabase συνδέεται με τη βάση PostgreSQL/PostGIS για τα γεωχωρικά δεδομένα.
- Το Node.js + Vite βοηθούν στο build.
- Το GitHub φιλοξενεί τον κώδικα, και το Netlify αναλαμβάνει το deployment, παρέχοντας το UI (User Interface) στον χρήστη.

Η διαδικασία ανάπτυξης της εφαρμογής Web-GIS για τη χωροθέτηση και οπτικοποίηση εγκαταστάσεων Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ) ξεκίνησε με τη δημιουργία της βάσης δεδομένων στο περιβάλλον του Supabase, το οποίο λειτουργεί ως cloud backend service πάνω σε PostgreSQL. Στο στάδιο αυτό ορίστηκαν οι βασικοί πίνακες (areas, installations) και οι απαραίτητες χωρικές δομές δεδομένων. Η χρήση του PostGIS επέτρεψε την αποθήκευση και επεξεργασία γεωμετρικών δεδομένων τύπου geometry, υποστηρίζοντας πολύγωνα και σημεία με ακρίβεια συντεταγμένων. Παράλληλα, δημιουργήθηκαν και views όπως `v_areas_geojson` και `v_installations_geojson`, τα οποία μετατρέπουν τα δεδομένα της βάσης σε μορφή GeoJSON, κατάλληλη για άμεση απεικόνιση στο Leaflet.

Στη συνέχεια πραγματοποιήθηκε η σχεδίαση της δομής της εφαρμογής στο περιβάλλον React. Η αρχιτεκτονική που επιλέχθηκε είναι αρθρωτή (modular), επιτρέποντας τη διαχείριση του κάθε υποσυστήματος ως ανεξάρτητο component. Δημιουργήθηκαν βασικά αρχεία, όπως το `MapView.tsx` για την αρχική απεικόνιση του χάρτη, το `InstallationsLayer.tsx` για τη φόρτωση και προβολή των εγκαταστάσεων, και το `AreasLayer.tsx` για την απεικόνιση των γεωγραφικών περιοχών. Η οργάνωση των αρχείων και η τήρηση διακριτών ευθυνών διευκόλυνε τη συνεργασία μεταξύ front-end και back-end τμημάτων της εφαρμογής.

Η επόμενη φάση αφορούσε την ενσωμάτωση του Leaflet, μιας ελαφριάς βιβλιοθήκης JavaScript για τη χαρτογραφική απεικόνιση. Μέσω του Leaflet έγινε δυνατή η απεικόνιση των πολυγώνων και των σημείων στον διαδραστικό χάρτη, η προσθήκη επιπέδων (layers), καθώς και η παροχή εργαλείων πλοήγησης (zoom in/out, pan). Επιπλέον, προστέθηκαν λειτουργίες interactive markers και pop-ups, ώστε ο χρήστης να μπορεί να ανακτά πληροφορίες για κάθε εγκατάσταση, όπως ονομασία, ισχύ (σε kW) και εμβαδό του πολυγώνου.

Για την διαχείριση των δεδομένων χρησιμοποιήθηκε το Supabase SDK, μέσω του οποίου η εφαρμογή πραγματοποιεί ασύγχρονα αιτήματα (queries) προς τη βάση. Το αρχείο `supabaseClient.ts` ορίζει τη σύνδεση με το backend και εξασφαλίζει ασφαλή πρόσβαση μέσω των μεταβλητών περιβάλλοντος (`VITE_SUPABASE_URL`, `VITE_SUPABASE_ANON_KEY`). Οι λειτουργίες select, insert και update υλοποιήθηκαν με τρόπο που επιτρέπει την άμεση ενημέρωση του χάρτη χωρίς ανανέωση της σελίδας. Έτσι, η επικοινωνία μεταξύ βάσης και frontend είναι δυναμική και real-time.

Σημαντικό μέρος της ανάπτυξης αποτέλεσε η προσθήκη νέων εγγραφών μέσω της διεπαφής `NewInstallation.tsx`. Το αρχείο αυτό επιτρέπει στον χρήστη να δημιουργεί νέες εγκαταστάσεις

ΑΠΕ επιλέγοντας το σημείο ή το πολύγωνο απευθείας πάνω στον χάρτη. Οι γεωμετρικές συντεταγμένες αποθηκεύονται αυτόματα στη βάση, μαζί με πρόσθετα χαρακτηριστικά (όπως ισχύς και επιφάνεια). Ο χρήστης έχει τη δυνατότητα να δει σε πραγματικό χρόνο το αποτέλεσμα της εισαγωγής του, γεγονός που καθιστά τη διεπαφή ιδιαίτερα φιλική και κατανοητή.

Επιπλέον, αναπτύχθηκε το σύστημα φιλτραρίσματος περιοχών, το οποίο υλοποιείται στο αρχείο `RegionFilter.tsx`. Μέσω ενός drop-down επιλογέα, ο χρήστης μπορεί να περιορίσει τα αποτελέσματα του χάρτη σε μία από τις δεκατρείς ελληνικές περιφέρειες. Το φίλτρο αυτό ενεργοποιεί δυναμικά τη φόρτωση των αντίστοιχων εγγραφών από τη βάση, προσφέροντας μια στοχευμένη και καθαρή απεικόνιση των δεδομένων. Η δυνατότητα αυτή ενισχύει σημαντικά την ευχρηστία της εφαρμογής, επιτρέποντας την εστίαση σε γεωγραφικές ενότητες.

Η αισθητική και λειτουργική ολοκλήρωση του χάρτη έγινε με τη χρήση του `MapView.tsx` ως κεντρικού ενοποιητικού component. Επάνω στο βασικό χάρτη προστέθηκαν όλα τα επιμέρους layers (εγκαταστάσεις, περιοχές, νέα γεωμετρία) και κουμπιά ελέγχου, όπως το “New Installation”. Ιδιαίτερη προσοχή δόθηκε στη θέση των κουμπιών, ώστε να μην επικαλύπτονται με τα εργαλεία zoom του Leaflet, διασφαλίζοντας εργονομία στη διεπαφή. Η απόκριση της εφαρμογής παραμένει ομαλή χάρη στη χρήση hooks και state management του React.

Τέλος, πραγματοποιήθηκε η φιλοξενία και συνεχής ανάπτυξη (CI/CD) μέσω του GitHub και της πλατφόρμας Netlify. Η σύνδεση του αποθετηρίου GitHub με το Netlify επιτρέπει αυτόματη ανανέωση της εφαρμογής κάθε φορά που πραγματοποιείται commit στον κύριο κλάδο. Η διαδικασία build εκτελείται με το Vite, ενώ το Netlify φροντίζει για τη διάθεση της εφαρμογής στο διαδίκτυο. Με αυτόν τον τρόπο, η ανάπτυξη του Web-GIS συστήματος έγινε πλήρως αυτοματοποιημένη, σύγχρονη και επεκτάσιμη, παρέχοντας ένα ολοκληρωμένο οικοσύστημα ανάπτυξης και διάθεσης.

## 2.4. Οι πίνακες και τα views στην βάση δεδομένων της εφαρμογής

### 2.4.1 Πίνακας: areas (base table)

Ο πίνακας areas λειτουργεί ως βασική οντότητα αποθήκευσης γεωχωρικών δεδομένων για περιοχές ενδιαφέροντος (π.χ. ζώνες εγκαταστάσεων ΑΠΕ). Τα πεδία id και geom είναι κρίσιμα για την ταυτοποίηση και χαρτογραφική απεικόνιση, το properties δίνει δυναμική και επεκτασιμότητα στα μεταδεδομένα, ενώ τα name και created\_at βελτιώνουν την ευχρηστία και την ανιχνευσιμότητα της πληροφορίας.

Πίνακας 2.1. Πίνακας areas (base table)

| column_name | data_type                | is_nullable | column_default    |
|-------------|--------------------------|-------------|-------------------|
| id          | uuid                     | NO          | gen_random_uuid() |
| name        | text                     | YES         | null              |
| properties  | jsonb                    | NO          | '{}'::jsonb       |
| geom        | USER-DEFINED             | NO          | null              |
| created_at  | timestamp with time zone | NO          | now()             |

#### id (uuid):

- Γενικά: Το uuid (Universally Unique Identifier) είναι ένας μοναδικός αναγνωριστικός αριθμός που εξασφαλίζει πως κάθε εγγραφή στον πίνακα είναι μοναδική.
- Στην εφαρμογή: Χρησιμοποιείται ως πρωτεύον κλειδί για τις περιοχές (areas). Επειδή το uuid δημιουργείται αυτόματα με τη συνάρτηση gen\_random\_uuid(), εξασφαλίζεται πως κάθε περιοχή έχει ένα σταθερό και μοναδικό ID ανεξάρτητο από την ακολουθία των εγγραφών. Αυτό είναι κρίσιμο για τη διασύνδεση με τα γεωχωρικά δεδομένα και την ανάκτηση συγκεκριμένων περιοχών από το frontend.

#### name (text):

- Γενικά: Το πεδίο text επιτρέπει την αποθήκευση ελεύθερου κειμένου, π.χ. περιγραφών ή ονομάτων.
- Στην εφαρμογή: Εδώ μπορεί να αποθηκευτεί η ονομασία της περιοχής (π.χ. "Περιφέρεια Δυτικής Μακεδονίας" ή "Ζώνη Αιολικού Πάρκου"). Το πεδίο είναι προαιρετικό (nullable), καθώς κάποιες περιοχές μπορεί να μην έχουν όνομα κατά την εισαγωγή και να συμπληρωθεί αργότερα.

**properties (jsonb):**

- Γενικά: Το jsonb είναι τύπος δεδομένων της PostgreSQL που επιτρέπει την αποθήκευση JSON αντικειμένων με δυνατότητα ευέλικτων ερωτημάτων.
- Στην εφαρμογή: Στο πεδίο αυτό αποθηκεύονται μεταδεδομένα της περιοχής, όπως π.χ. κωδικός, τύπος εγκατάστασης, ιδιοκτήτης ή κανονιστικές πληροφορίες. Η χρήση JSONB προσφέρει ευελιξία, γιατί μπορεί να εμπλουτιστεί με νέα properties χωρίς αλλαγές στο schema του πίνακα. Αυτό καθιστά την εφαρμογή πιο επεκτάσιμη.

**geom (USER-DEFINED):**

- Γενικά: Πρόκειται για το γεωχωρικό πεδίο, το οποίο σε PostgreSQL/PostGIS είναι τύπου geometry. Περιέχει σχήματα όπως Polygon, MultiPolygon, Point κ.λπ.
- Στην εφαρμογή: Αποθηκεύει τη γεωμετρία της εκάστοτε περιοχής σε χωρική μορφή. Αυτό είναι το κλειδί για την απεικόνιση των περιοχών στον χάρτη Leaflet, καθώς από εδώ αντλούνται οι συντεταγμένες που μετατρέπονται σε GeoJSON και εμφανίζονται ως πολύγωνα στο frontend.

**created\_at (timestamp with time zone):**

- Γενικά: Το timestamp with time zone καταγράφει τη χρονική στιγμή που δημιουργήθηκε η εγγραφή.
- Στην εφαρμογή: Χρησιμοποιείται για ιστορικό και auditing, π.χ. πότε δημιουργήθηκε μια νέα περιοχή. Αυτό είναι σημαντικό για να γίνεται παρακολούθηση του πότε προστέθηκαν νέες περιοχές και για σκοπούς αξιοπιστίας δεδομένων.

**2.4.2 Πίνακας: installations (base table)**

Ο πίνακας installations είναι η βασική οντότητα της εφαρμογής, όπου αποθηκεύονται όλα τα δεδομένα για τις εγκαταστάσεις ΑΠΕ (π.χ. φωτοβολταϊκά, αιολικά πάρκα). Τα πεδία του εξυπηρετούν την ταυτοποίηση (id, code), την ενεργειακή πληροφόρηση (power\_max, power\_avg), την χωρική κατανομή (region, geom, area\_m2) και την ιστορικότητα (created\_at). Σε συνδυασμό με τον πίνακα areas, δίνει τη δυνατότητα ολοκληρωμένης χαρτογράφησης και φιλτραρίσματος των εγκαταστάσεων στην εφαρμογή Web-GIS.

Πίνακας 2.2.. Πίνακας *installations* (base table)

| column_name | data_type                | is_nullable | column_default                            |
|-------------|--------------------------|-------------|-------------------------------------------|
| id          | bigint                   | NO          | nextval('installations_id_seq'::regclass) |
| code        | text                     | NO          | null                                      |
| power_max   | numeric                  | YES         | null                                      |
| power_avg   | numeric                  | YES         | null                                      |
| region      | USER-DEFINED             | NO          | null                                      |
| geom        | USER-DEFINED             | NO          | null                                      |
| area_m2     | double precision         | NO          | 0                                         |
| created_at  | timestamp with time zone | NO          | now()                                     |

#### id (bigint):

- Γενικά: Το `bigint` είναι τύπος ακεραίου μεγάλου μεγέθους που επιτρέπει αποθήκευση πολύ μεγάλων αριθμητικών τιμών. Συχνά χρησιμοποιείται για `primary keys` με αυτόματη αρίθμηση (`serial` ή `sequence`).
- Στην εφαρμογή: Είναι το πρωτεύον κλειδί κάθε εγγραφής εγκατάστασης. Χρησιμοποιείται το `nextval('installations_id_seq'::regclass)` ώστε να παράγεται αυτόματα ένας μοναδικός αύξων αριθμός. Αυτό διευκολύνει την παρακολούθηση και αναφορά σε κάθε εγκατάσταση ΑΠΕ.

#### code (text):

- Γενικά: Το πεδίο `text` χρησιμοποιείται για αποθήκευση ελεύθερου κειμένου, συχνά μοναδικών κωδικών ή περιγραφικών στοιχείων.
- Στην εφαρμογή: Εδώ αποθηκεύεται ένας μοναδικός κωδικός της εγκατάστασης (π.χ. “PV-001” για φωτοβολταϊκό ή “WIND-023” για ανεμογεννήτρια). Ο κωδικός αυτός λειτουργεί ως σημείο αναφοράς για την αναζήτηση και ταυτοποίηση της εγκατάστασης στο GIS.

#### power\_max (numeric):

- Γενικά: Ο τύπος `numeric` υποστηρίζει αποθήκευση αριθμών μεγάλης ακρίβειας, συχνά για οικονομικά ή τεχνικά μεγέθη.
- Στην εφαρμογή: Αποθηκεύει τη μέγιστη ισχύ (σε kW) που μπορεί να παράγει η εγκατάσταση. Είναι προαιρετικό πεδίο, καθώς ίσως κάποια δεδομένα να μην είναι διαθέσιμα κατά την εισαγωγή.

**power\_avg (numeric):**

- Γενικά: Όπως και το power\_max, αποθηκεύει αριθμητικές τιμές μεγάλης ακρίβειας.
- Στην εφαρμογή: Εδώ καταγράφεται η μέση ισχύς (σε kW) της εγκατάστασης, π.χ. βάσει ιστορικών δεδομένων ή εκτιμήσεων. Αυτό το πεδίο είναι επίσης προαιρετικό, με στόχο να δίνει πιο ρεαλιστική εικόνα της ενεργειακής απόδοσης.

**region (USER-DEFINED):**

- Γενικά: Στις PostgreSQL μπορεί να οριστεί USER-DEFINED τύπος, όπως enum ή domain, που αποθηκεύει περιορισμένες τιμές.
- Στην εφαρμογή: Αποθηκεύει την περιφέρεια στην οποία ανήκει η εγκατάσταση (π.χ. “Δυτική Μακεδονία”, “Αττική”). Αυτό είναι κρίσιμο για το dropdown φίλτρο στο frontend, που επιτρέπει στον χρήστη να φιλτράρει εγκαταστάσεις ανά περιφέρεια.

**geom (USER-DEFINED):**

- Γενικά: Όπως και στον πίνακα areas, αυτό είναι γεωχωρικό πεδίο τύπου geometry από το PostGIS, που μπορεί να περιέχει Point, Polygon κ.λπ.
- Στην εφαρμογή: Αποθηκεύει τη γεωμετρία της εγκατάστασης (συνήθως Polygon ή Point) ώστε να αποτυπώνεται στον χάρτη Leaflet. Αυτό είναι το θεμέλιο για την οπτικοποίηση στο Web-GIS.

**area\_m2 (double precision):**

- Γενικά: Ο τύπος double precision είναι αριθμητικός με υψηλή ακρίβεια κινητής υποδιαστολής, κατάλληλος για μετρήσεις.
- Στην εφαρμογή: Χρησιμοποιείται για να καταγράφει την έκταση της εγκατάστασης σε τετραγωνικά μέτρα. Από προεπιλογή παίρνει τιμή 0, αλλά σε πλήρη δεδομένα ενημερώνεται βάσει της γεωμετρίας.

**created\_at (timestamp with time zone):**

- Γενικά: Καταγράφει χρονική στιγμή με πληροφορία ζώνης ώρας.
- Στην εφαρμογή: Καταγράφει την ημερομηνία και ώρα εισαγωγής κάθε εγκατάστασης. Έτσι υπάρχει ιστορικό για το πότε δημιουργήθηκαν ή καταχωρήθηκαν δεδομένα στο σύστημα.

### 2.4.3 Πίνακας: spatial\_ref\_sys (base table)

Ο πίνακας spatial\_ref\_sys δεν περιέχει άμεσα δεδομένα για τις εγκαταστάσεις ή τις περιοχές, αλλά είναι απολύτως κρίσιμος για την ορθή απεικόνιση των γεωμετριών. Χωρίς αυτόν, οι πίνακες areas και installations δεν θα μπορούσαν να "ξέρουν" σε ποιο σύστημα συντεταγμένων ανήκουν τα χωρικά τους δεδομένα, κάτι που θα οδηγούσε σε λάθη στην οπτικοποίηση στον χάρτη.

Πίνακας 2.3. Πίνακας spatial\_ref\_sys (base table)

| column_name | data_type         | is_nullable | column_default |
|-------------|-------------------|-------------|----------------|
| srid        | integer           | NO          | null           |
| auth_name   | character varying | YES         | null           |
| auth_srid   | integer           | YES         | null           |
| srttext     | character varying | YES         | null           |
| proj4text   | character varying | YES         | null           |

#### srid (integer):

- Γενικά: Το srid (Spatial Reference ID) είναι ο μοναδικός αριθμητικός κωδικός που αναγνωρίζει ένα σύστημα συντεταγμένων (π.χ. WGS84 = 4326, ΕΓΣΑ87 = 2100).
- Στην εφαρμογή: Το πεδίο αυτό επιτρέπει στο PostGIS να καταλαβαίνει σε ποιο σύστημα αναφοράς βρίσκονται οι γεωμετρίες που αποθηκεύονται στους πίνακες areas και installations. Έτσι διασφαλίζεται ότι όλα τα χωρικά δεδομένα προβάλλονται σωστά στον χάρτη.

#### auth\_name (character varying):

- Γενικά: Το auth\_name δηλώνει την αρχή/οργανισμό που ορίζει το σύστημα συντεταγμένων (π.χ. EPSG, ESRI).
- Στην εφαρμογή: Επιτρέπει να γίνεται συσχέτιση του SRID με διεθνή πρότυπα, π.χ. το EPSG:4326. Αυτό εξασφαλίζει διαλειτουργικότητα όταν τα δεδομένα εξάγονται ή εισάγονται από άλλα GIS συστήματα.

#### auth\_srid (integer):

- Γενικά: Είναι ο αριθμητικός κωδικός SRID του συστήματος συντεταγμένων, όπως έχει καθοριστεί από την αρχή (auth\_name).
- Στην εφαρμογή: Χρησιμοποιείται για αντιστοίχιση με εξωτερικά συστήματα, έτσι ώστε τα δεδομένα που αποθηκεύονται (π.χ. γεωμετρίες εγκαταστάσεων) να μπορούν να διαβαστούν και από άλλα εργαλεία GIS όπως το QGIS ή το ArcGIS.

### **srtext (character varying):**

- Γενικά: Περιέχει την περιγραφή του συστήματος αναφοράς σε μορφή Well-Known Text (WKT), που είναι κατανοητή από πολλές εφαρμογές GIS.
- Στην εφαρμογή: Εξασφαλίζει ότι οι γεωμετρίες μπορούν να αναλυθούν και να παρουσιαστούν με ακρίβεια. Για παράδειγμα, το WKT του EPSG:4326 καθορίζει ότι πρόκειται για γεωγραφικές συντεταγμένες σε μοίρες.

### **proj4text (character varying)**

- Γενικά: Περιέχει την περιγραφή του συστήματος συντεταγμένων σε μορφή Proj4 string, που είναι άλλη μια ευρέως χρησιμοποιούμενη μορφή ορισμού χωρικών συστημάτων.
- Στην εφαρμογή: Το πεδίο αυτό επιτρέπει στον PostGIS και σε βιβλιοθήκες όπως το Leaflet ή το OpenLayers να ερμηνεύουν σωστά τις γεωμετρίες και να προβάλλουν τις εγκαταστάσεις στον σωστό χάρτη.

## **2.4.4 Πίνακας: geography\_columns (view)**

Το geography\_columns είναι system view του PostGIS που λειτουργεί ως “μετα-πληροφορία” (metadata) για να ξέρουμε σε ποιους πίνακες και στήλες υπάρχουν γεωγραφικά δεδομένα, τι διαστάσεις έχουν, σε ποιο SRID ανήκουν και ποιον τύπο γεωμετρίας αποθηκεύουν.

Στη παρούσα εφαρμογή δεν χρησιμοποιείται το geography αλλά το geometry, άρα το αντίστοιχο view που είναι πιο χρήσιμο για αυτή την εφαρμογή είναι το geometry\_columns. Παρ’ όλα αυτά, το geography\_columns παραμένει διαθέσιμο για μελλοντική επέκταση (π.χ. αν αποφασίσουμε να χρησιμοποιήσουμε 3D γεωδαιτικά δεδομένα σε σφαιρική επιφάνεια αντί για απλό επίπεδο χάρτη).

*Πίνακας 2.4. Πίνακας geography\_columns (view)*

| column_name        | data_type | is_nullable | column_default |
|--------------------|-----------|-------------|----------------|
| f_table_catalog    | name      | YES         | null           |
| f_table_schema     | name      | YES         | null           |
| f_table_name       | name      | YES         | null           |
| f_geography_column | name      | YES         | null           |
| coord_dimension    | integer   | YES         | null           |
| srid               | integer   | YES         | null           |
| type               | text      | YES         | null           |

**f\_table\_catalog (name):**

- Γενικά: Δείχνει σε ποιο database catalog (βάση δεδομένων) ανήκει ο πίνακας που περιέχει γεωγραφικό πεδίο τύπου geography.
- Στην εφαρμογή: Εξασφαλίζει ότι μπορούμε να αναγνωρίσουμε σε ποια βάση Supabase βρίσκονται οι πίνακες με γεωγραφικά δεδομένα (π.χ. installations, areas).

**f\_table\_schema (name):**

- Γενικά: Δείχνει το schema (π.χ. public) στο οποίο βρίσκεται ο πίνακας με πεδία τύπου geography.
- Στην εφαρμογή: Επιτρέπει οργάνωση των γεωχωρικών δεδομένων στο κατάλληλο schema, ώστε να αποφεύγονται συγκρούσεις με άλλα schemas.

**f\_table\_name (name):**

- Γενικά: Το όνομα του πίνακα που περιέχει στήλη τύπου geography.
- Στην εφαρμογή: Αν στον πίνακα installations ή areas χρησιμοποιούσαμε πεδίο τύπου geography (αντί για geometry), εδώ θα εμφανιζόταν το όνομά τους.

**f\_geography\_column (name):**

- Γενικά: Το όνομα της στήλης του πίνακα που είναι τύπου geography.
- Στην εφαρμογή: Θα έδειχνε το πεδίο π.χ. geom εφόσον ήταν geography.

**coord\_dimension (integer):**

- Γενικά: Δείχνει τη διάσταση των συντεταγμένων: 2 (XY), 3 (XYZ), ή 4 (XYZM).
- Στην εφαρμογή: Αν είχαμε υψομετρικά δεδομένα, θα μπορούσε να είναι 3. Στην παρούσα εφαρμογή θα είναι 2 (γεωγραφικό πλάτος/μήκος).

**srid (integer):**

- Γενικά: Το Spatial Reference ID (όπως στον spatial\_ref\_sys). Καθορίζει το σύστημα συντεταγμένων.
- Στην εφαρμογή: Χρήσιμο για να επιβεβαιώνουμε ότι όλα τα δεδομένα στο πεδίο geography είναι π.χ. στο EPSG:4326.

**type (text):**

- Γενικά: Ο τύπος της γεωμετρίας (π.χ. Point, Polygon, LineString).
- Στην εφαρμογή: Θα έδειχνε π.χ. Polygon για τις περιοχές εγκαταστάσεων ΑΠΕ, εφόσον το πεδίο ήταν geography.

### 2.4.5 Πίνακας: geometry\_columns (view)

Το geometry\_columns είναι ένα είδος «ευρετηρίου γεωμετριών» που το PostGIS ενημερώνει αυτόματα και το οποίο επιτρέπει στην εφαρμογή να δουλεύει με ασφάλεια με γεωμετρικά δεδομένα, γνωρίζοντας schema, SRID, τύπο και πίνακα.

Ο πίνακας geometry\_columns είναι view που δημιουργείται αυτόματα από το PostGIS και δεν είναι «κανονικός πίνακας». Ο ρόλος του είναι να κρατάει μεταδεδομένα για τα γεωμετρικά πεδία που υπάρχουν στη βάση, ώστε το σύστημα (και οι εφαρμογές που συνδέονται) να ξέρουν πού βρίσκονται τα γεωμετρικά δεδομένα, τι τύπο έχουν, σε ποιο SRID βρίσκονται και πώς μπορούν να χρησιμοποιηθούν.

Στη παρούσα Web-GIS εφαρμογή για ΑΠΕ, το view αυτό βοηθάει έμμεσα: δεν εμφανίζεται απευθείας στον χρήστη, αλλά είναι θεμέλιο για το PostGIS ώστε να γνωρίζει πώς να ερμηνεύσει τα πεδία geom σε πίνακες όπως το installations και το areas, τα οποία τροφοδοτούν το frontend (React + Leaflet) με τα πολύγωνα και τα σημεία.

Πίνακας 2.5. Πίνακας geometry\_columns (view)

| column_name       | data_type         | is_nullable | column_default |
|-------------------|-------------------|-------------|----------------|
| f_table_catalog   | character varying | YES         | null           |
| f_table_schema    | name              | YES         | null           |
| f_table_name      | name              | YES         | null           |
| f_geometry_column | name              | YES         | null           |
| coord_dimension   | integer           | YES         | null           |
| srid              | integer           | YES         | null           |
| type              | character varying | YES         | null           |

#### **f\_table\_catalog (character varying):**

- Γενικά: Το όνομα του database catalog (συνήθως το όνομα της ίδιας της βάσης).
- Στην εφαρμογή: Δεν χρησιμοποιείται άμεσα, αλλά δηλώνει σε ποια βάση βρίσκονται τα γεωμετρικά δεδομένα. Στο Supabase, που τρέχει σε PostgreSQL, είναι πάντα η τρέχουσα βάση του project. Χρήσιμο για debugging ή σε multi-database περιβάλλοντα.

#### **f\_table\_schema (name):**

- Γενικά: Το όνομα του schema (π.χ. public) που περιέχει τον πίνακα με το γεωμετρικό πεδίο.

- Στην εφαρμογή: Σημαντικό για να ξέρει η εφαρμογή (μέσω Supabase SQL queries) πού να αναζητήσει τα δεδομένα. Σχεδόν όλα τα tables σου (installations, areas) ανήκουν στο public.

**f\_table\_name (name):**

- Γενικά: Το όνομα του πίνακα που περιέχει το γεωμετρικό πεδίο (π.χ. installations, areas).
- Στην εφαρμογή: Δείχνει πού ακριβώς είναι αποθηκευμένα τα γεωμετρικά δεδομένα. Π.χ. στον πίνακα installations είναι τα πολύγωνα των εγκαταστάσεων ΑΠΕ, ενώ στον πίνακα areas οι περιοχές αναφοράς.

**f\_geometry\_column (name):**

- Γενικά: Το όνομα της στήλης που περιέχει το γεωμετρικό αντικείμενο (π.χ. geom).
- Στην εφαρμογή: Εδώ φαίνεται το βασικό πεδίο geom που διαβάζει η παρούσα εφαρμογή και το μετατρέπει σε GeoJSON για να το εμφανίσει στον Leaflet χάρτη.

**coord\_dimension (integer):**

- Γενικά: Ο αριθμός των συντεταγμένων που έχει η γεωμετρία (π.χ. 2 για X-Y, 3 για X-Y-Z).
- Στην εφαρμογή: Στη συγκεκριμένη περίπτωση είναι σχεδόν πάντα 2 (γεωγραφικό πλάτος και μήκος). Αν είχε επιλεγθεί να ενσωματωθούν και υψομετρικά δεδομένα, θα γινόταν 3D.

**srid (integer):**

- Γενικά: Το Spatial Reference ID (π.χ. 4326 για WGS84). Δείχνει σε ποιο σύστημα αναφοράς ανήκουν οι συντεταγμένες.
- Στην εφαρμογή: Πολύ κρίσιμο. Γιατί όλα τα δεδομένα που εμφανίζονται στον Leaflet πρέπει να είναι στο ίδιο SRID. Συνήθως είναι 4326 (WGS84), το διεθνές πρότυπο για GIS και WebMaps.

**type (character varying):**

- Γενικά: Ο τύπος γεωμετρίας (π.χ. POINT, POLYGON, MULTIPOLYGON).
- Στην εφαρμογή: Εδώ θα φαίνεται αν η γεωμετρία είναι σημείο (π.χ. θέση εγκατάστασης) ή πολύγωνο (π.χ. περιοχή που καλύπτει). Η παρούσα εφαρμογή το χρειάζεται για να ξέρει πώς να σχεδιάσει το σχήμα στον χάρτη.

#### 2.4.6 Πίνακας: v\_areas\_geojson (view)

Το v\_areas\_geojson λειτουργεί ως γέφυρα μεταξύ του PostgreSQL/PostGIS backend και της React εφαρμογής. Δίνει τα δεδομένα στη σωστή μορφή για web mapping (GeoJSON) και έτσι απλοποιεί σημαντικά τη διασύνδεση της βάσης με τον χάρτη.

Το v\_areas\_geojson είναι view που έχει φτιαχτεί πάνω στον πίνακα areas ώστε να παρέχει τα δεδομένα σε μορφή GeoJSON, την οποία μπορεί να χρησιμοποιήσει εύκολα το frontend (React + Leaflet). Ουσιαστικά, είναι ένας «ενδιάμεσος μετασχηματιστής» που παίρνει τις στήλες από τον base table (areas) και τις αποδίδει σε JSON μορφή φιλική για web mapping.

Στην παρούσα εφαρμογή Web-GIS για ΑΠΕ, το view αυτό είναι κομβικό: επιτρέπει στη React εφαρμογή να ζητάει έτοιμα GeoJSON features χωρίς να χρειάζεται να κάνει parsing SQL queries ή PostGIS συναρτήσεις. Έτσι, μειώνεται η πολυπλοκότητα στο frontend και επιταχύνεται η απόδοση, αφού η Leaflet δουλεύει απευθείας με GeoJSON.

Πίνακας 2.6. Πίνακας v\_areas\_geojson (view)

| column_name | data_type                | is_nullable | column_default |
|-------------|--------------------------|-------------|----------------|
| id          | uuid                     | YES         | null           |
| name        | text                     | YES         | null           |
| properties  | jsonb                    | YES         | null           |
| created_at  | timestamp with time zone | YES         | null           |
| geometry    | json                     | YES         | null           |

##### id (uuid):

- Γενικά: Μοναδικό αναγνωριστικό για κάθε περιοχή, με μορφή UUID.
- Στην εφαρμογή: Χρησιμοποιείται ως κλειδί ώστε η κάθε περιοχή (π.χ. μια ζώνη ενδιαφέροντος για εγκαταστάσεις ΑΠΕ) να αναγνωρίζεται μοναδικά στον χάρτη και στη βάση.

##### name (text):

- Γενικά: Κείμενο που αποθηκεύει την ονομασία της περιοχής.
- Στην εφαρμογή: Δίνει περιγραφικό τίτλο στην περιοχή (π.χ. “Ζώνη Αιολικών Κοζάνης”), που μπορεί να εμφανίζεται σε popups ή λίστες.

**properties (jsonb):**

- Γενικά: Στήλη τύπου JSONB που αποθηκεύει επιπλέον χαρακτηριστικά σε δυναμική μορφή (π.χ. metadata, περιγραφικά tags).
- Στην εφαρμογή: Επιτρέπει ευελιξία για να αποθηκεύονται έξτρα δεδομένα χωρίς να αλλάζει το schema (π.χ. πληροφορίες για περιβαλλοντικούς περιορισμούς, βαθμό καταλληλότητας).

**created\_at (timestamp with time zone):**

- Γενικά: Χρονική σήμανση δημιουργίας της εγγραφής.
- Στην εφαρμογή: Χρήσιμο για audit trail — π.χ. πότε καταχωρήθηκε μια νέα περιοχή ή για να γίνεται φιλτράρισμα με βάση τον χρόνο εισαγωγής.

**geometry (json):**

- Γενικά: Η γεωμετρία της περιοχής σε μορφή JSON (GeoJSON).
- Στην εφαρμογή: Είναι το πιο σημαντικό πεδίο για το frontend. Η Leaflet διαβάζει κατευθείαν αυτό το JSON ώστε να εμφανίσει το πολύγωνο στον χάρτη. Έτσι, δεν χρειάζεται να γίνονται σύνθετοι μετασχηματισμοί από WKB/WKT.

**2.4.7 Πίνακας: v\_installations\_geojson (view)**

Το v\_installations\_geojson λειτουργεί σαν έτοιμο API επίπεδο δεδομένων για το frontend: μειώνει τα βήματα που απαιτούνται για την αναπαράσταση των εγκαταστάσεων ΑΠΕ στον χάρτη και παρέχει μια καθαρή, τυποποιημένη μορφή (GeoJSON) που η Leaflet χρησιμοποιεί άμεσα.

Το v\_installations\_geojson είναι ένα view που δημιουργήθηκε πάνω στον πίνακα installations με σκοπό να παρέχει τις εγκαταστάσεις ΑΠΕ σε μορφή GeoJSON Feature. Ουσιαστικά, «μεταφράζει» τις εγγραφές των εγκαταστάσεων σε JSON αντικείμενα συμβατά με το πρότυπο GeoJSON, ώστε να μπορούν να χρησιμοποιηθούν απευθείας από το frontend (React + Leaflet).

Στην παρούσα εφαρμογή Web-GIS, το view αυτό εξυπηρετεί την άμεση χαρτογραφική απεικόνιση των εγκαταστάσεων στον διαδραστικό χάρτη. Αντί το frontend να χρειάζεται να συνθέτει GeoJSON από ακατέργαστα δεδομένα (όπως WKT/WKB γεωμετρίες και attributes), το backend σε PostgreSQL/PostGIS μέσω Supabase παρέχει έτοιμο το GeoJSON Feature, μειώνοντας το φόρτο επεξεργασίας στην React εφαρμογή και βελτιώνοντας την ταχύτητα φόρτωσης.

Πίνακας 2.7. Πίνακας v\_installations\_geojson (view)

| column_name | data_type | is_nullable | column_default |
|-------------|-----------|-------------|----------------|
| id          | bigint    | YES         | null           |
| feature     | jsonb     | YES         | null           |

#### id (bigint):

- Γενικά: Αριθμητικό μοναδικό αναγνωριστικό κάθε εγκατάστασης.
- Στην εφαρμογή: Χρησιμοποιείται ως πρωτεύον κλειδί για να συνδέει το κάθε GeoJSON feature με την αντίστοιχη εγγραφή στον πίνακα installations. Χάρη σε αυτό, η εφαρμογή μπορεί να αναγνωρίζει και να εμφανίζει σωστά πληροφορίες για κάθε εγκατάσταση (π.χ. κωδικό, ισχύ, περιοχή).

#### feature (jsonb):

- Γενικά: Στήλη τύπου JSONB που αποθηκεύει ολόκληρο το GeoJSON Feature, δηλαδή ένα αντικείμενο που περιλαμβάνει τόσο την γεωμετρία (geometry: point/polygon/multipolygon) όσο και τις ιδιότητες (properties: κωδικός, ισχύς, μέση τιμή, εμβαδόν, κ.ά.).
- Στην εφαρμογή: Είναι το κεντρικό στοιχείο που περνάει απευθείας στη Leaflet ώστε να γίνει η απεικόνιση των εγκαταστάσεων στον χάρτη. Εξασφαλίζει ότι όλα τα απαραίτητα attributes (π.χ. ισχύς, περιοχή, εμβαδόν) είναι διαθέσιμα στο popup/tooltip που βλέπει ο χρήστης.

## 2.5. Περιγραφή και δημιουργία του schema της Βάσης Δεδομένων

Το αρχείο create db.sql είναι το βασικό σενάριο αρχικοποίησης της βάσης δεδομένων της εφαρμογής. Σκοπός του είναι να δημιουργήσει από την αρχή τη δομή που θα υποστηρίζει την αποθήκευση και διαχείριση γεωχωρικών δεδομένων σχετικών με τις εγκαταστάσεις ΑΠΕ.

Παρακάτω ακολουθεί η περιγραφή λειτουργίας του αρχείου create db.sql που περιέχει τον κώδικα SQL για την δημιουργία της Βάσης Δεδομένων. Ο κώδικας του αρχείου φαίνεται στο Παράρτημα Α.

### 1. Καθαρισμός (DROP)

Στην αρχή, γίνεται διαγραφή τυχόν υπαρχόντων views, triggers, functions και πινάκων που σχετίζονται με τις εγκαταστάσεις, ώστε να αποφευχθούν συγκρούσεις σε περιπτώσεις επανεκτέλεσης του script.

### 2. Δημιουργία ENUM τύπου (gr\_region)

Ορίζεται ένας τύπος δεδομένων gr\_region που περιλαμβάνει τις 13 διοικητικές περιφέρειες της Ελλάδας. Αυτό επιτρέπει την αποθήκευση των εγκαταστάσεων με αναφορά σε μια συγκεκριμένη περιφέρεια, διασφαλίζοντας έλεγχο εγκυρότητας.

### 3. Πίνακας installations

Δημιουργείται ο βασικός πίνακας με πεδία:

- id (πρωτεύον κλειδί),
- code (μοναδικός κωδικός εγκατάστασης),
- power\_max, power\_avg (ισχύς – μέγιστη, μέση),
- region (περιφέρεια, συνδεδεμένη με το ENUM),
- geom (γεωμετρία πολύγωνου με SRID 4326),
- area\_m2 (υπολογιζόμενη επιφάνεια σε m<sup>2</sup>),
- created\_at (ημερομηνία δημιουργίας).

Επίσης, προστίθενται indexes για βελτιωμένη απόδοση σε χωρικά queries και φιλτράρισμα ανά περιφέρεια.

### 4. Trigger για area\_m2

Δημιουργείται trigger που αυτόματα υπολογίζει το εμβαδόν (ST\_Area) κάθε γεωμετρίας όταν γίνεται εισαγωγή ή ενημέρωση. Έτσι, διασφαλίζεται η ακρίβεια χωρίς χειροκίνητους υπολογισμούς από το frontend.

### 5. RPC function api\_insert\_installation

Παρέχεται μια function που επιτρέπει την εισαγωγή εγκαταστάσεων μέσω Supabase / PostgreSQL API. Η function:

- Ελέγχει αν η τιμή του region είναι έγκυρη ως προς το ENUM.
- Μετατρέπει WKT string σε γεωμετρία.
- Επιστρέφει το id της νέας εγγραφής.

Η function είναι διαθέσιμη στους ρόλους anon και authenticated, ώστε να μπορεί να καλείται από το frontend.

#### 6. View `v_installations_geojson`

Δημιουργείται ένα view που επιστρέφει τις εγκαταστάσεις σε μορφή GeoJSON Feature. Αυτό επιτρέπει στο frontend (Leaflet/React) να διαβάζει απευθείας χωρικά δεδομένα έτοιμα για απεικόνιση χωρίς πρόσθετους μετασχηματισμούς.

#### 7. Δικαιώματα & Refresh

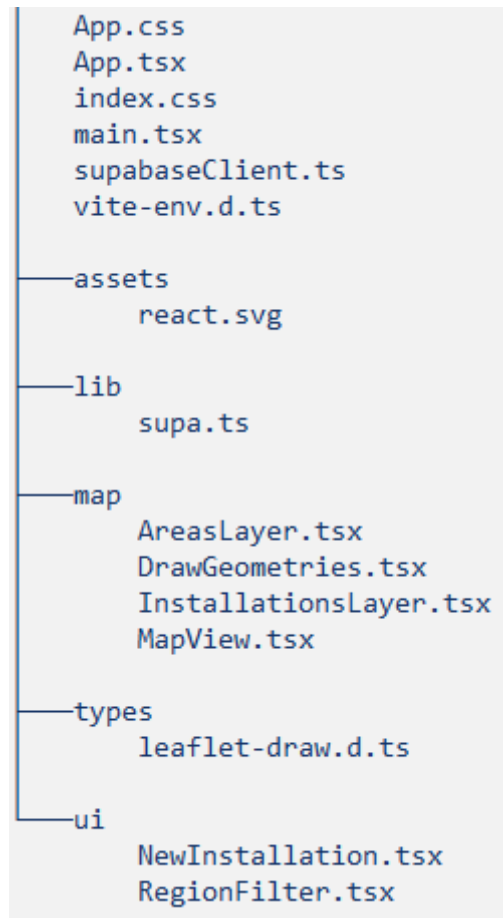
Δίνονται δικαιώματα SELECT/EXECUTE στους χρήστες του API και γίνεται notify στο PostgREST για ανανέωση του schema cache.

Σχέση με τα υπόλοιπα μέρη της εφαρμογής:

- Το `installations` table είναι η αποθήκη των δεδομένων που βλέπουν οι χρήστες στο χάρτη.
- Το `v_installations_geojson` συνδέεται άμεσα με το `InstallationsLayer.tsx`, το οποίο διαβάζει τα δεδομένα και τα εμφανίζει στο Leaflet map.
- Η `api_insert_installation` χρησιμοποιείται από το `NewInstallation.tsx`, ώστε όταν εισάγει ο χρήστης ένα νέο πολύγωνο, να καταχωρείται σωστά στη ΒΔ.
- Ο trigger εξασφαλίζει την αυτόματη ενημέρωση του `area_m2`, που εμφανίζεται στο UI.
- Ο ορισμός ENUM `gr_region` επιτρέπει στο `RegionFilter.tsx` να φιλτράρει εγγραφές με βάση τις περιφέρειες.

## 2.6. Περιγραφή της λειτουργικότητας των βασικών αρχείων της εφαρμογής

Τα αρχεία της εφαρμογής παρατίθενται στην δενδρική απεικόνιση στην παρακάτω εικόνα.



Εικόνα 2.1. Η δενδρική απεικόνιση των βασικών αρχείων της εφαρμογής

Στην συνέχεια ακολουθεί αναλυτική παρουσίαση της λειτουργικότητας των βασικών αρχείων.

### 2.6.1 Το αρχείο src/map/InstallationsLayer.tsx

Το αρχείο src/map/InstallationsLayer.tsx αποτελεί ένα από τα βασικά κομμάτια της εφαρμογής, καθώς είναι υπεύθυνο για τη φόρτωση και την απεικόνιση στον χάρτη των εγκαταστάσεων ΑΠΕ που έχουν καταχωρηθεί στη βάση δεδομένων. Πιο συγκεκριμένα, χρησιμοποιεί το Supabase για να αντλήσει από τον πίνακα installations τις γεωμετρίες (πολύγωνα) και τα μεταδεδομένα (όπως ισχύς, κωδικός, περιοχή) και στη συνέχεια τα μετατρέπει σε αντικείμενα GeoJSON, τα οποία απεικονίζονται με το Leaflet πάνω στον χάρτη. Με αυτό τον τρόπο, οι χρήστες βλέπουν τις εγκαταστάσεις χωροθετημένες στο γεωγραφικό τους πλαίσιο και έχουν τη δυνατότητα αλληλεπίδρασης με αυτές (π.χ. μέσω popups ή tooltips).

Η λειτουργία του αρχείου διασυνδέεται άμεσα με άλλα μέρη της εφαρμογής: συνεργάζεται με το `MapView.tsx`, που παρέχει το γενικό container του χάρτη και ενσωματώνει τα layers· επηρεάζεται από το `RegionFilter.tsx`, που αλλάζει το φίλτρο ανά περιφέρεια ώστε να φορτώνονται μόνο οι σχετικές εγκαταστάσεις· και συνδέεται με το `AreasLayer.tsx` για να απεικονίζονται τα όρια περιοχών μέσα στις οποίες εντάσσονται οι εγκαταστάσεις. Επιπλέον, η σχέση με το `NewInstallation.tsx` είναι λειτουργική, αφού ό,τι εισάγεται ως νέα εγκατάσταση στη βάση, μέσω του συγκεκριμένου component, μπορεί να προβληθεί στη συνέχεια στο layer των εγκαταστάσεων. Συνολικά, το `InstallationsLayer.tsx` είναι το κρίσιμο κομμάτι που μετατρέπει τα δεδομένα της βάσης σε οπτική πληροφορία πάνω στον χάρτη, αποτελώντας έτσι τον σύνδεσμο μεταξύ της αποθήκευσης/διαχείρισης δεδομένων και της χαρτογραφικής απεικόνισης. Ο κώδικας του αρχείου παρατίθεται στο Παράρτημα B1.

### 2.6.2 Το αρχείο `src/map/MapView.tsx`

Το αρχείο `src/map/MapView.tsx` είναι το κεντρικό αρχείο διαχείρισης του χάρτη στην εφαρμογή και λειτουργεί ως το "container" μέσα στο οποίο ενσωματώνονται όλα τα υπόλοιπα layers και λειτουργικότητες. Χρησιμοποιεί το Leaflet για να δημιουργήσει και να προβάλλει τον διαδραστικό χάρτη στον οποίο απεικονίζονται τόσο οι εγκαταστάσεις ΑΠΕ όσο και τα όρια περιοχών ή άλλες γεωχωρικές πληροφορίες. Παράλληλα, παρέχει στον χρήστη βασικά εργαλεία αλληλεπίδρασης, όπως τα κουμπιά μεγέθυνσης/σμίκρυνσης (zoom in/zoom out), τον επιλογέα για την εισαγωγή νέων εγκαταστάσεων, καθώς και το drop-down μενού για φιλτράρισμα των δεδομένων ανά περιφέρεια. Με τον τρόπο αυτό, το `MapView.tsx` λειτουργεί σαν η "κύρια σκηνή" όπου προβάλλεται το αποτέλεσμα της εφαρμογής.

Η διασύνδεσή του με τα υπόλοιπα αρχεία είναι καθοριστική: (1) φορτώνει το `InstallationsLayer.tsx`, το οποίο εμφανίζει στον χάρτη τις εγκαταστάσεις από τη βάση δεδομένων, (2) ενσωματώνει το `AreasLayer.tsx` για να φαίνονται τα όρια των περιοχών, (3) αξιοποιεί το `RegionFilter.tsx` ώστε ο χρήστης να μπορεί να φιλτράρει τα δεδομένα ανά γεωγραφική ενότητα και (4) επικοινωνεί με το `NewInstallation.tsx`, μέσω του οποίου εισάγονται νέες εγκαταστάσεις και εμφανίζονται άμεσα στον χάρτη. Ουσιαστικά, το `MapView.tsx` είναι ο κεντρικός κόμβος που συντονίζει και παρουσιάζει τα δεδομένα και τις λειτουργίες της εφαρμογής, εξασφαλίζοντας τη συνοχή και την αλληλεπίδραση μεταξύ χρήστη, δεδομένων και απεικόνισης. Ο κώδικας του αρχείου παρατίθεται στο Παράρτημα B2.

### 2.6.3 Το αρχείο src/map/AreasLayer.tsx

Το αρχείο src/map/AreasLayer.tsx είναι υπεύθυνο για την απεικόνιση των γεωγραφικών περιοχών (areas) στον χάρτη της εφαρμογής. Χρησιμοποιεί δεδομένα που αποθηκεύονται στον πίνακα areas της βάσης PostgreSQL/PostGIS μέσω του Supabase, και τα μετατρέπει σε πολύγωνα GeoJSON τα οποία προβάλλονται στο Leaflet. Κάθε περιοχή μπορεί να έχει ιδιότητες αποθηκευμένες στο πεδίο properties (τύπου JSONB), δίνοντας τη δυνατότητα εμπλουτισμού των πολυγώνων με μεταδεδομένα, όπως ονόματα, κατηγορίες ή άλλα χαρακτηριστικά. Το αρχείο διαχειρίζεται την ανάκτηση των δεδομένων, τη μετατροπή τους στη σωστή μορφή, και την εμφάνιση τους με χρωματισμούς και αλληλεπίδραση πάνω στον χάρτη.

Η διασύνδεση του AreasLayer.tsx με τα υπόλοιπα μέρη της εφαρμογής είναι σημαντική: (1) συνεργάζεται με το MapView.tsx, το οποίο είναι ο κεντρικός "καμβάς" όπου εμφανίζονται όλα τα layers, (2) λειτουργεί συμπληρωματικά με το InstallationsLayer.tsx, το οποίο εμφανίζει τις εγκαταστάσεις ΑΠΕ, έτσι ώστε οι περιοχές να δίνουν χωρικό πλαίσιο στις εγκαταστάσεις· και συνδέεται έμμεσα με το RegionFilter.tsx, αφού η επιλογή περιφέρειας μπορεί να περιορίσει τόσο τις εγκαταστάσεις όσο και τα όρια περιοχών που εμφανίζονται στον χάρτη. Συνολικά, το αρχείο αυτό συμβάλλει στη γεωγραφική οριοθέτηση και στην πιο σαφή χωρική κατανόηση των δεδομένων που διαχειρίζεται η εφαρμογή. Ο κώδικας του αρχείου παρατίθεται στο Παράρτημα Β3.

### 2.6.4 Το αρχείο src/map/DrawGeometries.tsx

Το αρχείο src/map/DrawGeometries.tsx έχει ως κύριο ρόλο να παρέχει στον χρήστη τη δυνατότητα σχεδίασης νέων γεωμετριών (κυρίως πολυγώνων) απευθείας πάνω στον χάρτη της εφαρμογής. Η λειτουργικότητα αυτή υλοποιείται με τη βοήθεια εργαλείων της βιβλιοθήκης Leaflet (και πιθανών πρόσθετων της, όπως το Leaflet Draw), τα οποία επιτρέπουν στον χρήστη να δημιουργεί, να μετακινεί και να επεξεργάζεται χωρικά σχήματα. Με τον τρόπο αυτό, ο χρήστης μπορεί να ορίσει την ακριβή τοποθεσία και το σχήμα μιας νέας εγκατάστασης ΑΠΕ ή μιας περιοχής ενδιαφέροντος, πριν αυτά αποθηκευτούν στη βάση. Ο κώδικας του αρχείου χειρίζεται επίσης τα απαραίτητα events (π.χ. ολοκλήρωση σχεδίασης, διαγραφή γεωμετρίας), ώστε να ενημερώνεται κατάλληλα η εφαρμογή.

Η διασύνδεση του DrawGeometries.tsx με τα υπόλοιπα μέρη της εφαρμογής είναι άμεση: (1) συνεργάζεται με το NewInstallation.tsx, όπου η γεωμετρία που σχεδιάζεται χρησιμοποιείται για να δημιουργηθεί μια νέα εγγραφή εγκατάστασης στη βάση, (2) επικοινωνεί με το MapView.tsx, που είναι ο κεντρικός χώρος εμφάνισης και χειρισμού του χάρτη και (3) επηρεάζει έμμεσα το InstallationsLayer.tsx, αφού οι νέες γεωμετρίες που εισάγονται γίνονται μελλοντικά τμήμα του layer των εγκαταστάσεων. Με αυτό τον τρόπο, το αρχείο αποτελεί κρίσιμο εργαλείο για την ενεργή αλληλεπίδραση του χρήστη με τον χάρτη, διευκολύνοντας τη δημιουργία και καταγραφή χωρικών δεδομένων μέσα στην εφαρμογή. Ο κώδικας του αρχείου παρατίθεται στο Παράρτημα B4.

### **2.6.5 Το αρχείο src/ui/NewInstallation.tsx**

Το αρχείο src/ui/NewInstallation.tsx είναι υπεύθυνο για τη διαχείριση της φόρμας εισαγωγής νέας εγκατάστασης ΑΠΕ από τον χρήστη. Μέσα από το UI component που παρέχει, ο χρήστης μπορεί να εισάγει στοιχεία όπως ο κωδικός (code), η μέγιστη και μέση ισχύς (power\_max, power\_avg), η περιφέρεια (region) και να συνδέσει τη γεωμετρία που σχεδιάστηκε προηγουμένως στον χάρτη. Το αρχείο περιλαμβάνει λογική για την ανάγνωση και την επικύρωση των τιμών που δίνονται, και τελικά καλεί τις αντίστοιχες συναρτήσεις για την αποθήκευση των δεδομένων στη βάση μέσω Supabase. Επιπλέον, συνδυάζει τα στοιχεία της φόρμας με το υπολογιζόμενο εμβαδό (area\_m2), ώστε η νέα εγκατάσταση να καταγράφεται πλήρως.

Η διασύνδεση του NewInstallation.tsx με τα υπόλοιπα μέρη της εφαρμογής είναι θεμελιώδης: (1) συνεργάζεται με το DrawGeometries.tsx, αφού η γεωμετρία που έχει σχεδιάσει ο χρήστης περνάει στη φόρμα για να συνδυαστεί με τα υπόλοιπα δεδομένα, (2) επικοινωνεί με το InstallationsLayer.tsx, καθώς οι νέες εγγραφές που εισάγονται εμφανίζονται μετά στο layer των εγκαταστάσεων στον χάρτη και (3) ενσωματώνεται στο MapView.tsx, το οποίο προσφέρει το περιβάλλον χαρτογράφησης όπου εκτελείται όλη η διαδικασία. Με λίγα λόγια, το αρχείο αυτό λειτουργεί ως ο κρίκος ανάμεσα στη διεπαφή χρήστη (UI) και στη βάση δεδομένων, κάνοντας δυνατή την πραγματική καταγραφή νέων εγκαταστάσεων στην εφαρμογή. Ο κώδικας του αρχείου παρατίθεται στο Παράρτημα B5.

### 2.6.6 Το αρχείο `src/ui/RegionFilter.tsx`

Το αρχείο `src/ui/RegionFilter.tsx` υλοποιεί το drop-down μενού επιλογής Περιφέρειας, μέσω του οποίου ο χρήστης μπορεί να φιλτράρει τα δεδομένα που προβάλλονται στον χάρτη. Ουσιαστικά, επιτρέπει την αλληλεπίδραση με το γεωγραφικό πεδίο "region" που είναι αποθηκευμένο στη βάση για κάθε εγκατάσταση ΑΠΕ. Έτσι, αντί να εμφανίζονται όλες οι καταχωρήσεις ταυτόχρονα, ο χρήστης μπορεί να επιλέγει μία από τις 13 Περιφέρειες της χώρας και να βλέπει μόνο τις εγκαταστάσεις που βρίσκονται σε αυτήν. Το component είναι υλοποιημένο ως ελαφρύ στοιχείο UI με χρήση React hooks, ώστε να ενημερώνει άμεσα την κατάσταση (state) κάθε φορά που ο χρήστης αλλάζει επιλογή.

Η διασύνδεση του `RegionFilter.tsx` με τα υπόλοιπα αρχεία είναι κρίσιμη για την ομαλή λειτουργία της εφαρμογής: το επιλεγμένο φίλτρο περνάει σαν prop στο `InstallationsLayer.tsx`, το οποίο φροντίζει να φορτώνει από το Supabase μόνο τις εγκαταστάσεις που ανήκουν στην τρέχουσα Περιφέρεια. Παράλληλα, ενσωματώνεται μέσα στο `MapView.tsx`, έτσι ώστε να βρίσκεται πάνω στο interface του χάρτη δίπλα στα βασικά κουμπιά πλοήγησης. Με αυτό τον τρόπο, το αρχείο αυτό λειτουργεί ως διαμεσολαβητής ανάμεσα στο UI και το layer των δεδομένων, δίνοντας στον χρήστη τη δυνατότητα να εστιάζει γρήγορα στη γεωγραφική περιοχή που τον ενδιαφέρει. Ο κώδικας του αρχείου παρατίθεται στο Παράρτημα Β6.

### 2.6.7 Το αρχείο `src/supabaseClient.ts`

Το αρχείο `src/supabaseClient.ts` αποτελεί ένα θεμελιώδες τμήμα της αρχιτεκτονικής της εφαρμογής, καθώς είναι υπεύθυνο για τη σύνδεση του front-end περιβάλλοντος με τη βάση δεδομένων PostgreSQL/PostGIS μέσω της πλατφόρμας Supabase. Η κύρια λειτουργία του είναι η αρχικοποίηση και διαμόρφωση του client που επιτρέπει την επικοινωνία της εφαρμογής με το Supabase API, εξασφαλίζοντας την πρόσβαση σε δεδομένα, την εισαγωγή νέων εγγραφών και την ενημέρωση ή διαγραφή υπαρχουσών.

Στο αρχείο πραγματοποιείται η εισαγωγή της βιβλιοθήκης `@supabase/supabase-js` και η δημιουργία ενός αντικειμένου `supabase`, το οποίο βασίζεται στις μεταβλητές περιβάλλοντος `VITE_SUPABASE_URL` και `VITE_SUPABASE_ANON_KEY`. Μέσω αυτής της ρύθμισης, το `supabaseClient` λειτουργεί ως κεντρικό σημείο επικοινωνίας για όλη την εφαρμογή, επιτρέποντας ασφαλή και τυποποιημένη πρόσβαση στη βάση δεδομένων. Με τον τρόπο αυτόν,

όλες οι κλήσεις προς το Supabase (π.χ. αναζήτηση, εισαγωγή ή ενημέρωση δεδομένων) γίνονται με συνεπή και ελεγχόμενο τρόπο.

Η λειτουργία του αρχείου διασυνδέεται στενά με τα υπόλοιπα modules της εφαρμογής. Συνεργάζεται με το InstallationsLayer.tsx, το οποίο χρησιμοποιεί τον supabaseClient για να ανακτήσει τις εγγραφές των ενεργειακών εγκαταστάσεων από τον πίνακα installations, και με το NewInstallation.tsx, που βασίζεται στον ίδιο μηχανισμό για την εισαγωγή νέων εγγραφών στη βάση. Επιπλέον, το RegionFilter.tsx αξιοποιεί το Supabase client για την εκτέλεση ερωτημάτων με φίλτρα, ώστε να εμφανίζονται μόνο οι εγκαταστάσεις συγκεκριμένων περιφερειών, ενώ το AreasLayer.tsx μπορεί να χρησιμοποιεί παρόμοια λογική για την άντληση ορίων ή περιοχών ενδιαφέροντος.

Παράλληλα, ο supabaseClient λειτουργεί και ως ενδιάμεσος μηχανισμός αυθεντικοποίησης χρηστών, όταν ενεργοποιείται το Supabase Auth, προσφέροντας δυνατότητες σύνδεσης και ελέγχου πρόσβασης στα δεδομένα. Ο ρόλος του είναι, επομένως, διπλός: αφενός εξασφαλίζει την τεχνική διασύνδεση της React εφαρμογής με το backend, και αφετέρου παρέχει τις βάσεις για την ασφάλεια και τη διαχείριση συνεδριών χρηστών.

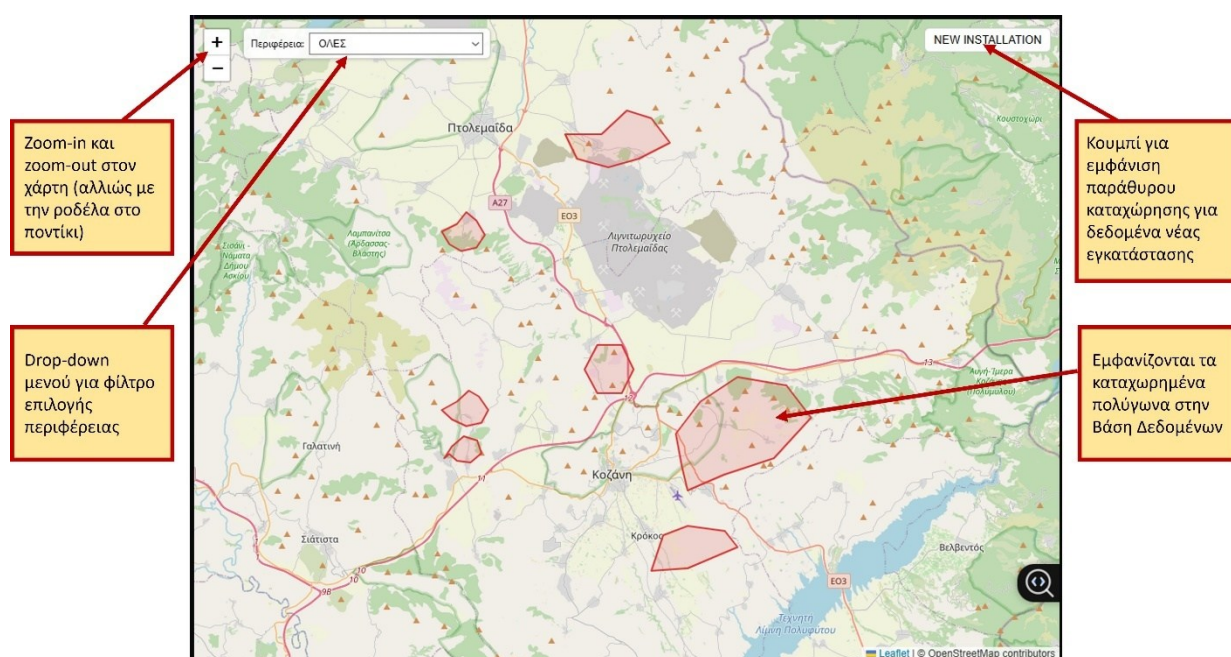
Συνολικά, το αρχείο supabaseClient.ts λειτουργεί ως ο πυρήνας επικοινωνίας μεταξύ του front-end (React/TypeScript) και της υποδομής αποθήκευσης δεδομένων (PostgreSQL/PostGIS). Μέσω αυτού, οι επιμέρους λειτουργίες της εφαρμογής μπορούν να ανταλλάσσουν δεδομένα σε πραγματικό χρόνο με τρόπο αποδοτικό και ασφαλές. Ο κώδικας του αρχείου παρατίθεται στο Παράρτημα Β7.

## 2.7. Η εμφάνιση και η λειτουργικότητα της αναρτημένης εφαρμογής

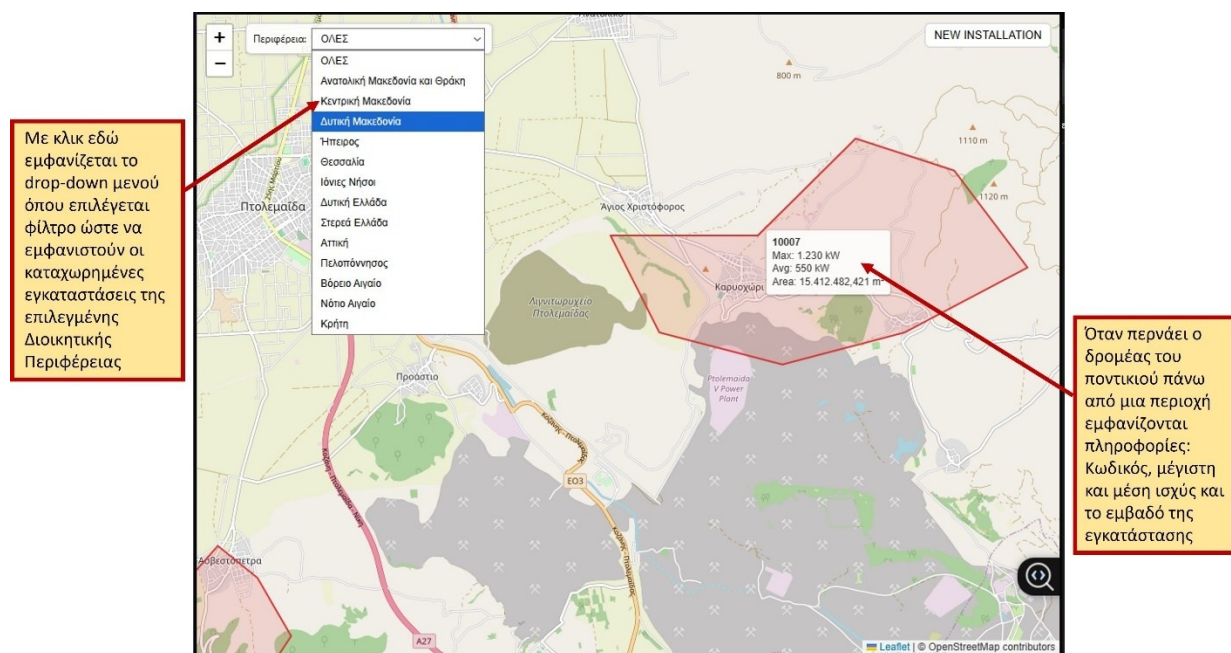
Η εφαρμογή έχει αναρτηθεί στην διεύθυνση <https://web-gis-res.netlify.app>.

Στην εφαρμογή έχουν εισαχθεί μερικά εικονικά δεδομένα για να διαπιστωθεί η ορθή λειτουργικότητά της στο αρχικό στάδιο της χρήσης της.

Τα μέρη της εφαρμογής και η λειτουργικότητά τους επισημαίνονται στις παρακάτω εικόνες:



Εικόνα 2.2. Τα μέρη στην αρχική σελίδα της εφαρμογής



Εικόνα 2.3. Drop-down φίλτρου Περιφέρειας και εμφάνιση πληροφοριών σε πολύγωνο καταχωρημένης εγκατάστασης

Στις παρακάτω εικόνες φαίνεται η διαδικασία καταχώρησης των δεδομένων για το πολύγωνο μιας νέας εγκατάστασης με εικονικά δεδομένα που θα δημιουργήσουν μια νέα πολυγωνική επιφάνεια που θα αντιπροσωπεύει την νέα καταχώρηση:

Σε αυτά τα πεδία εισαγωγής γίνεται η καταχώρηση των στοιχείων της νέας εγκατάστασης. (Κωδικός, Περιφέρεια από drop-down, μέγιστη και μέση ισχύς. Επίσης καταχωρείται το πλήθος των κορυφών του πολυγώνου της εγκατάστασης.)

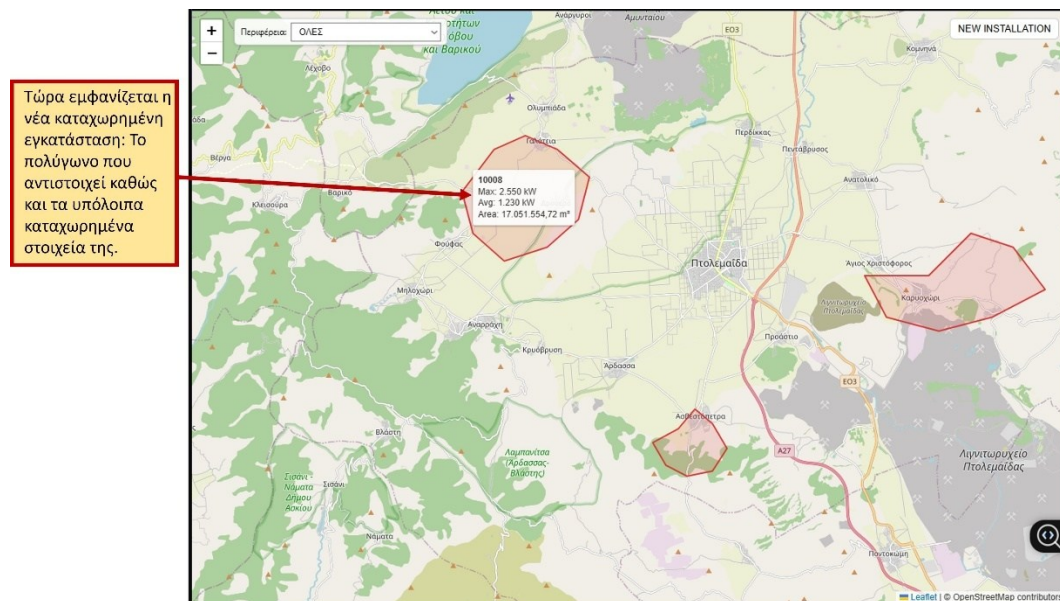
Έχει πατηθεί το κουμπί για εμφάνιση παραθύρου καταχώρησης νέας εγκατάστασης

Εδώ εισάγονται οι συντεταγμένες των κορυφών του πολυγώνου της εγκατάστασης σε μορφή λίστας.

Όταν πατηθεί το κουμπί αυτό μεταφέρονται οι συντεταγμένες στα πλαίσια εισαγωγής κάτω.

Με το κουμπί ENTER γίνεται η καταχώρηση των δεδομένων της νέας εγκατάστασης στην Βάση Δεδομένων.

Εικόνα 2.4. Το παράθυρο εισαγωγής των δεδομένων της Νέας Εγκατάστασης



Εικόνα 2.5. Το πολύγωνο και τα δεδομένα της Νέας Εγκατάστασης εμφανιζόμενα πάνω στον χάρτη της εφαρμογής

Η τελική μορφή της εφαρμογής παρουσιάζει έναν διαδραστικό χάρτη, ο οποίος καταλαμβάνει το μεγαλύτερο μέρος της οθόνης, εξασφαλίζοντας την ευκρινή απεικόνιση των γεωχωρικών δεδομένων. Στο επάνω αριστερό τμήμα του χάρτη βρίσκονται τα κουμπιά ελέγχου πλοήγησης,

επιτρέποντας στον χρήστη να πραγματοποιεί μεγέθυνση (zoom in) ή σμίκρυνση (zoom out) με ευκολία, καθώς και να μετακινείται ομαλά πάνω στον χάρτη. Δίπλα τους εμφανίζεται το κουμπί επιλογής περιφέρειας (Region Filter), μέσω του οποίου ο χρήστης μπορεί να φιλτράρει τις απεικονιζόμενες εγκαταστάσεις ανά γεωγραφική περιφέρεια της χώρας. Η επιλογή αυτή προκαλεί άμεση ανανέωση του περιεχομένου του χάρτη, ώστε να εμφανίζονται μόνο οι εγκαταστάσεις που ανήκουν στην επιλεγμένη περιοχή.

Στο επάνω δεξιό μέρος του χάρτη βρίσκεται το κουμπί “New Installation”, το οποίο ενεργοποιεί τη διαδικασία δημιουργίας νέας εγγραφής. Μόλις ο χρήστης το επιλέξει, εμφανίζεται ένα πλαίσιο εισαγωγής δεδομένων (form) στο οποίο συμπληρώνει τις βασικές πληροφορίες για τη νέα εγκατάσταση, όπως κωδικός (code), μέγιστη και μέση ισχύ (power\_max, power\_avg) και επιλέγει πολυγωνική περιοχή ή σημείο τοποθέτησης πάνω στον χάρτη. Η εφαρμογή υπολογίζει αυτόματα το εμβαδόν του πολυγώνου (σε m<sup>2</sup>) μέσω του PostGIS και το εμφανίζει στο κάτω μέρος του πλαισίου. Μόλις πατηθεί το πλήκτρο Enter, τα δεδομένα αποστέλλονται στη βάση Supabase και η νέα εγκατάσταση καταχωρίζεται άμεσα.

Η εφαρμογή υποστηρίζει άμεση αλληλεπίδραση μεταξύ χρήστη και χάρτη, προσφέροντας δυνατότητες όπως η οπτική επαλήθευση της καταχώρησης και η δυναμική ενημέρωση των επιπέδων (layers). Κάθε φορά που προστίθεται μια νέα εγκατάσταση, το πολύγωνο ή το σημείο της εμφανίζεται με σαφή οριοθέτηση και χαρακτηριστικό χρωματισμό. Με αυτό τον τρόπο, ο χρήστης έχει άμεση ανατροφοδότηση για τη θέση και τα όρια της νέας καταχώρησης. Επιπλέον, κάθε πολυγωνική περιοχή ή σημείο διαθέτει αναδυόμενο πλαίσιο πληροφοριών (popup) που εμφανίζει τα βασικά στοιχεία της εγκατάστασης, όπως ισχύ, περιφέρεια και ημερομηνία εισαγωγής.

Η χρήση του Leaflet εξασφαλίζει ομαλή απεικόνιση και απόκριση του χάρτη, επιτρέποντας την ταυτόχρονη απεικόνιση πολλαπλών γεωμετρικών αντικειμένων χωρίς καθυστέρηση. Οι γεωμετρικές πληροφορίες (GeoJSON) ανακτώνται σε πραγματικό χρόνο από τη βάση Supabase και προβάλλονται μέσω των αρχείων InstallationsLayer.tsx και AreasLayer.tsx, τα οποία ελέγχουν ξεχωριστά τα επίπεδα εγκαταστάσεων και περιοχών. Η αλληλεπίδραση μεταξύ αυτών των επιπέδων γίνεται δυνατή μέσω του MapView.tsx, που λειτουργεί ως ο κεντρικός ενορχηστρωτής της χαρτογραφικής απεικόνισης.

Η αισθητική της διεπαφής σχεδιάστηκε ώστε να είναι λιτή και λειτουργική, με έμφαση στην ευχρηστία. Τα κουτιά πληροφοριών (text boxes) διαθέτουν ημιδιαφανές φόντο, ώστε να επιτρέπουν την προβολή του υποκείμενου χάρτη χωρίς να διακόπτουν την οπτική συνέχεια. Η γραμματοσειρά και τα χρώματα έχουν επιλεγεί για υψηλή αναγνωσιμότητα, ενώ τα εργαλεία πλοήγησης τοποθετήθηκαν με τέτοιο τρόπο ώστε να μη συγκρούονται με τα στοιχεία ελέγχου ή τα επίπεδα του Leaflet. Έτσι, η εμπειρία χρήσης παραμένει ευχάριστη και εργονομικά ομαλή σε κάθε ανάλυση οθόνης.

Τέλος, η εφαρμογή είναι ανεβασμένη και λειτουργική στο διαδίκτυο μέσω της πλατφόρμας Netlify, η οποία συνδέεται αυτόματα με το αποθετήριο του GitHub. Κάθε αλλαγή στον πηγαίο κώδικα προκαλεί αυτόματο build και ανανέωση της ζωντανής εφαρμογής, εξασφαλίζοντας ότι η πιο πρόσφατη έκδοση είναι πάντοτε διαθέσιμη στο κοινό. Η διαδικασία αυτή υποστηρίζει τη συνεχή ανάπτυξη (CI/CD) και επιτρέπει στους χρήστες να έχουν πάντα πρόσβαση στη λειτουργική έκδοχή του συστήματος Web-GIS για τις εγκαταστάσεις ΑΠΕ, επιδεικνύοντας έτσι την εξελισσόμενη πρακτική ωριμότητα και τεχνολογική πληρότητα της εφαρμογής.

### **3. Συμπεράσματα – Περιορισμοί – Μελλοντικές προεκτάσεις**

#### **3.1. Συμπεράσματα**

- Η παρούσα πτυχιακή εργασία απέδειξε τη δυνατότητα ανάπτυξης ενός πλήρως λειτουργικού συστήματος Web-GIS με χρήση σύγχρονων τεχνολογιών ανοικτού λογισμικού. Η εφαρμογή ενσωματώνει με επιτυχία τα εργαλεία React, TypeScript, Vite, Leaflet, Supabase και PostgreSQL/PostGIS, παρέχοντας έναν αποδοτικό και ευέλικτο τρόπο απεικόνισης, ανάλυσης και διαχείρισης γεωχωρικών δεδομένων.
- Η δημιουργία ενός ενιαίου περιβάλλοντος οπτικοποίησης εγκαταστάσεων Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ) συνέβαλε ουσιαστικά στη βελτίωση της κατανόησης του ενεργειακού αποτυπώματος μιας περιοχής. Μέσα από τον δυναμικό χάρτη, οι χρήστες μπορούν να εντοπίζουν τις εγκαταστάσεις, να συγκρίνουν ισχύ και έκταση, και να λαμβάνουν άμεσα συμπεράσματα για τη γεωγραφική κατανομή της ενεργειακής δραστηριότητας.
- Η επιλογή του Supabase ως backend επέτρεψε τη χρήση μιας cloud-based υποδομής με δυνατότητα αποθήκευσης, ελέγχου πρόσβασης και real-time ενημέρωσης των δεδομένων. Αυτό καθιστά την εφαρμογή πιο αξιόπιστη και επεκτάσιμη σε σχέση με παραδοσιακές τοπικές λύσεις.
- Η διασύνδεση του PostGIS με το περιβάλλον Leaflet προσέφερε τη δυνατότητα εκτέλεσης χωρικών ερωτημάτων και αποδοτικής προβολής γεωμετρικών αντικειμένων στον χάρτη. Η σωστή αξιοποίηση των τύπων δεδομένων geometry και jsonb αποδείχθηκε κρίσιμη για τη δυναμική απεικόνιση πολυγώνων και σημείων.
- Η ανάπτυξη με React και TypeScript ενίσχυσε την επεκτασιμότητα και την ασφάλεια του κώδικα, επιτρέποντας αυστηρό έλεγχο τύπων και εύκολη επαναχρησιμοποίηση συστατικών (components). Η χρήση του Vite συνέβαλε σημαντικά στη βελτίωση των χρόνων μεταγλώττισης και δοκιμών.

- Το σύστημα καταφέρνει να γεφυρώσει την τεχνολογική και ενεργειακή διάσταση, αποτελώντας ενδεχομένως ένα ισχυρό εργαλείο για τη λήψη αποφάσεων σε επίπεδο τοπικής αυτοδιοίκησης και ενεργειακού σχεδιασμού.
- Η λειτουργικότητα φιλτραρίσματος ανά περιφέρεια, καθώς και η δυνατότητα προσθήκης νέων εγκαταστάσεων με γεωγραφικό εντοπισμό, δείχνουν πως η εφαρμογή μπορεί να αξιοποιηθεί τόσο σε εκπαιδευτικά όσο και σε επαγγελματικά περιβάλλοντα.
- Η εμπειρία χρήσης βελτιώθηκε με την προσθήκη διεπαφών φιλικών προς τον χρήστη, όπως drop-down επιλογείς, πληροφοριακά παράθυρα (pop-ups) και άμεση ενημέρωση του χάρτη χωρίς ανανέωση της σελίδας.
- Η αρχιτεκτονική της εφαρμογής είναι πλήρως αρθρωτή (modular), γεγονός που επιτρέπει την εύκολη συντήρηση και επέκταση, ενώ η φιλοξενία στο Netlify και ο συγχρονισμός με GitHub διασφαλίζουν συνεχή ανάπτυξη (continuous deployment).
- Συνολικά, η εργασία απέδειξε ότι ένα Web-GIS σύστημα για ΑΠΕ μπορεί να αναπτυχθεί με σύγχρονες τεχνολογίες front-end και back-end, παραμένοντας ταυτόχρονα αποδοτικό, ευέλικτο και ανοικτό σε μελλοντικές επεκτάσεις.

### 3.2. Περιορισμοί

- Η εφαρμογή αναπτύχθηκε ως Proof of Concept (PoC), με σκοπό να επιδείξει τις βασικές λειτουργίες ενός Web-GIS συστήματος και όχι να αποτελέσει ένα πλήρες εμπορικό προϊόν.
- Οι λειτουργίες επικύρωσης δεδομένων (validation) είναι περιορισμένες, με αποτέλεσμα ορισμένες εγγραφές να μπορούν να αποθηκευτούν χωρίς αυστηρό έλεγχο ορθότητας ή πληρότητας.
- Η αποθήκευση δεδομένων βασίζεται στην anon key του Supabase, που δεν παρέχει πλήρεις μηχανισμούς ασφαλείας ή role-based authorization, γεγονός που περιορίζει τη χρήση της εφαρμογής σε δημόσια ή παραγωγικά περιβάλλοντα.

- Οι χάρτες Leaflet δεν διαθέτουν ενσωματωμένο μηχανισμό caching, κάτι που μπορεί να οδηγήσει σε καθυστερήσεις φόρτωσης σε περιοχές με χαμηλό εύρος ζώνης.
- Η απεικόνιση πολυγώνων βασίζεται σε δεδομένα GeoJSON χωρίς εξειδικευμένη τοπολογική ανάλυση — δεν πραγματοποιούνται, δηλαδή, υπολογισμοί επικάλυψης, απόστασης ή buffer.
- Το σύστημα δεν περιλαμβάνει ακόμη προηγμένες λειτουργίες αναζήτησης (π.χ. χωρικά queries με SQL συνδυασμούς) ούτε μηχανισμό διαχείρισης χρηστών ή ρόλων.
- Οι επιδόσεις της εφαρμογής δεν έχουν αξιολογηθεί σε μεγάλα datasets εφόσον οι δοκιμές πραγματοποιήθηκαν με περιορισμένο όγκο δεδομένων για σκοπούς επίδειξης.
- Τέλος, η διεπαφή χρήστη (UI) παραμένει σε βασικό επίπεδο και θα μπορούσε να εμπλουτιστεί με περισσότερες επιλογές, στατιστικά εργαλεία και δυναμικά διαγράμματα.

### 3.3. Μελλοντικές προεκτάσεις

- Η πρώτη προτεραιότητα αφορά την ενσωμάτωση μηχανισμών ελέγχου και επικύρωσης δεδομένων, ώστε να διασφαλιστεί η ακεραιότητα και η αξιοπιστία των καταχωρήσεων.
- Μπορεί να προστεθεί διαχείριση χρηστών με διαφορετικά δικαιώματα πρόσβασης (admin, editor, viewer) μέσω της υποστήριξης Auth API του Supabase.
- Η χρήση ανάλυσης χωρικών δεδομένων (π.χ. proximity analysis, heatmaps, buffers) θα επιτρέψει την εξαγωγή πιο ουσιαστικών συμπερασμάτων για τη χωρική κατανομή των ΑΠΕ.
- Η ενσωμάτωση διαδραστικών γραφημάτων (με βιβλιοθήκες όπως Recharts ή D3.js) θα μπορούσε να βελτιώσει την παρουσίαση στατιστικών αποτελεσμάτων.

- Η σύνδεση με εξωτερικά APIs, όπως αυτά του Υπουργείου Περιβάλλοντος ή του ΔΕΔΔΗΕ, θα επιτρέψει την αυτόματη ενημέρωση των δεδομένων εγκαταστάσεων.
- Μια μελλοντική επέκταση θα μπορούσε να περιλαμβάνει ανάλυση ενεργειακής αποδοτικότητας με βάση τα δεδομένα ισχύος και επιφάνειας των εγκαταστάσεων, καθώς και δυναμικούς δείκτες (KPIs - Key Performance Indicator, που στα ελληνικά αποδίδεται ως Βασικός Δείκτης Απόδοσης ή Δείκτης Κρίσιμης Επίδοσης). Οι KPIs είναι μετρήσιμες τιμές που χρησιμοποιούνται για να αξιολογηθεί ο βαθμός επιτυχίας ενός οργανισμού, έργου ή συστήματος ως προς την επίτευξη συγκεκριμένων στόχων. Στο πλαίσιο εφαρμογών Web-GIS, οι KPIs μπορεί να περιλαμβάνουν:
  - Τον χρόνο απόκρισης του συστήματος.
  - Τον αριθμό ενεργών χρηστών ή αιτημάτων σε πραγματικό χρόνο.
  - Τον αριθμό εγκαταστάσεων ΑΠΕ που καταχωρούνται ανά μήνα.
  - Την ακρίβεια ή πληρότητα των γεωχωρικών δεδομένων.
- Τέλος, η ανάπτυξη μιας mobile-friendly ή PWA εκδοχής της εφαρμογής θα ενισχύσει τη χρηστικότητα και την πρόσβαση σε φορητές συσκευές, καθιστώντας το Web-GIS εργαλείο ευρύτερα διαθέσιμο. Το PWA είναι ακρωνύμιο του όρου Progressive Web Application, που στα ελληνικά αποδίδεται ως Προοδευτική Διαδικτυακή Εφαρμογή. Μια PWA είναι ένας τύπος web εφαρμογής που συνδυάζει τα πλεονεκτήματα των ιστοσελίδων και των native (εγκατεστημένων) εφαρμογών. Δηλαδή, είναι μια εφαρμογή που λειτουργεί μέσα από έναν browser, αλλά μοιάζει και συμπεριφέρεται σαν κανονική εφαρμογή κινητού ή υπολογιστή. (A. Russell et al., 2016) <sup>[112]</sup>. Η PWA εκδοχή της εφαρμογής θα επιτρέπει στον χρήστη να:
  - ανοίγει τον χάρτη και να βλέπει δεδομένα ακόμη και χωρίς σύνδεση,
  - αποθηκεύει προσωρινά τις τελευταίες προβολές χαρτών,
  - λαμβάνει ειδοποιήσεις για νέες ενεργειακές εγκαταστάσεις ή ενημερώσεις,
  - και να έχει την εφαρμογή στην αρχική οθόνη του κινητού σαν “native app”.

## Βιβλιογραφία

- [1] M. F. Goodchild, "Geographic Information System (GIS)," in *International Encyclopedia of Human Geography*, R. Kitchin and N. Thrift, Eds. Oxford: Elsevier, 2009, pp. 350–354, doi:10.1016/B978-008044910-4.00413-9.
- [2] R. Avtar, P. Kumar, A. Oono, C. Saraswat, T. Chaminda, and T. N. Hinh, "Exploring Renewable Energy Resources Using Remote Sensing and GIS," *Resources*, vol. 8, no. 3, p. 149, 2019, doi:10.3390/resources8030149.
- [3] Tethys, "A review of GIS and techno-economic tools for renewable energy," *Tethys*, 2023. [Online]. Available: <https://tethys.pnnl.gov>. [Accessed: Sep. 29, 2025].
- [4] A. Bieda, "Towards a Renewable Energy Source Cadastre," *Energies*, vol. 14, no. 23, p. 8095, 2021, doi:10.3390/en14238095.
- [5] E. Chatzigeorgiou and G. Martinopoulos, "Solar cadastre for assessment of near-zero energy districts," *IOP Conference Series: Earth and Environmental Science*, vol. 1196, p. 012003, 2023, doi:10.1088/1755-1315/1196/1/012003.
- [6] World Bank/ESMAP, *Global Solar Atlas*, 2019. [Online]. Available: <https://globalsolaratlas.info>. [Accessed: Sep. 29, 2025].
- [7] World Bank/ESMAP, *Global Wind Atlas*, 2019. [Online]. Available: <https://globalwindatlas.info>. [Accessed: Sep. 29, 2025].
- [8] Enel Green Power, "GIS systems in renewables," *Enel Green Power stories*, 2023. [Online]. Available: <https://www.enelgreenpower.com>. [Accessed: Sep. 29, 2025].
- [9] M. M. Bertsiou, G. Arampatzis, N. Giannopoulos, and C. Topaloglou, "Optimal Site Selection for Wind and Solar Parks in Karpathos," *ISPRS International Journal of Geo-Information*, vol. 14, no. 3, p. 125, 2025, doi:10.3390/ijgi14030125.
- [10] A. Vaswani et al., "Attention is All You Need," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [11] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv preprint*, arXiv:2010.11929, 2020.
- [12] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in Vision: A Survey," *ACM Computing Surveys*, vol. 54, no. 10s, pp. 1–41, 2022, doi:10.1145/3505244.
- [13] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, 2021, pp. 10012–10022, doi:10.1109/ICCV48922.2021.00986.
- [14] H. Touvron et al., "Training data-efficient image transformers & distillation through attention," in *Proc. 38th Int. Conf. Machine Learning (ICML)*, 2021, pp. 10347–10357.
- [15] T. Antic and T. Capuder, "A geographic information system-based modelling, analysing and visualising of low voltage networks: The potential of demand time-shifting in the power quality improvement," *arXiv preprint*, arXiv:2310.00382, 2023.
- [16] UN-GGIM, *Greece National Spatial Data Infrastructure Country Report*, 2015. [Online]. Available: <https://ggim.un.org>. [Accessed: Sep. 29, 2025].
- [17] "Open energy system databases," *Wikipedia*, 2025. [Online]. Available: [https://en.wikipedia.org/wiki/Open\\_energy\\_system\\_databases](https://en.wikipedia.org/wiki/Open_energy_system_databases). [Accessed: Sep. 29, 2025].
- [18] "Participatory GIS," *Wikipedia*, 2025. [Online]. Available: [https://en.wikipedia.org/wiki/Participatory\\_GIS](https://en.wikipedia.org/wiki/Participatory_GIS). [Accessed: Sep. 29, 2025].
- [19] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

- [20] P. Neis and D. Zielstra, "Recent developments and future trends in volunteered geographic information research: The case of OpenStreetMap," *Future Internet*, vol. 6, no. 1, pp. 76–106, 2014.
- [21] R. Avtar et al., "Exploring Renewable Energy Resources Using Remote Sensing and GIS," *Resources*, vol. 8, no. 3, p. 149, 2019.
- [22] V. Agafonkin, *Leaflet.js Documentation*, 2011. [Online]. Available: <https://leafletjs.com>. [Accessed: Aug. 22, 2025].
- [23] D. Chatzigeorgiou et al., "Solar cadastre for assessment of near-zero energy districts," *Sustainability*, vol. 15, no. 7, p. 6123, 2023.
- [24] M. Peterson, *Online Maps with APIs and WebServices*. Springer, 2012.
- [25] E. Hazzard, *OpenLayers 2.10 Beginner's Guide*. Packt Publishing, 2011.
- [26] UN-GGIM, *Greece National Spatial Data Infrastructure Country Report*. United Nations Committee of Experts on Global Geospatial Information Management, 2015.
- [27] M. M. Bertsiou et al., "Optimal Site Selection for Wind and Solar Parks in Karpathos," *ISPRS International Journal of Geo-Information*, vol. 14, no. 3, p. 125, 2025.
- [28] Google, *Project Sunroof*, 2017. [Online]. Available: <https://www.google.com/get/sunroof>. [Accessed: Aug. 22, 2025].
- [29] S. Shekhar and S. Chawla, *Spatial Databases: A Tour*. Prentice Hall, 2003.
- [30] P. Rigaux, M. Scholl, and A. Voisard, *Spatial Databases: With Application to GIS*. Morgan Kaufmann, 2001.
- [31] R. Obe and L. Hsu, *PostGIS in Action*, 1st ed. Manning Publications, 2011.
- [32] S. Steiniger and A. J. S. Hunter, "The 2012 free and open source GIS software map – A guide to facilitate research, development, and adoption," *Computers, Environment and Urban Systems*, vol. 39, pp. 136–150, 2013.
- [33] R. Avtar et al., "Exploring Renewable Energy Resources Using Remote Sensing and GIS," *Resources*, vol. 8, no. 3, p. 149, 2019.
- [34] K. Chodorow, *MongoDB: The Definitive Guide*, 2nd ed. O'Reilly Media, 2013.
- [35] K. Banker, *MongoDB in Action*, 2nd ed. Manning Publications, 2016.
- [36] J. Pokorný, "NoSQL databases: a step to database scalability in web environment," *International Journal of Web Information Systems*, vol. 12, no. 1, pp. 56–82, 2016.
- [37] C. Huang, J. Wang, Y. Li, and Y. Xu, "Big data driven fault diagnosis in power systems," *Renewable and Sustainable Energy Reviews*, vol. 76, pp. 928–935, 2017.
- [38] A. S. Tanenbaum and M. Steen, *Distributed Systems: Principles and Paradigms*, 2nd ed. Prentice Hall, 2007.
- [39] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*, 4th ed. Addison-Wesley, 2005.
- [40] Z. R. Peng and M. H. Tsou, *Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks*. Wiley, 2003.
- [41] T. Zhang, M. Goodchild, and M. Batty, *GIS and the City*. CRC Press, 2012.
- [42] M. Peterson, *Online Maps with APIs and WebServices*. Springer, 2012.
- [43] S. Steiniger and A. J. S. Hunter, "The 2012 free and open source GIS software map," *Computers, Environment and Urban Systems*, vol. 39, pp. 136–150, 2013.
- [44] G. Percivall, "OGC standards and geospatial interoperability," *Communications of the ACM*, vol. 51, no. 7, pp. 118–124, 2008.
- [45] OGC, *OGC Web Services Common Standard*. Open Geospatial Consortium, 2015.
- [46] R. Avtar et al., "Exploring Renewable Energy Resources Using Remote Sensing and GIS," *Resources*, vol. 8, no. 3, p. 149, 2019.

- [47] C. Yang et al., "Cloud computing and the integration of geospatial services and applications," *International Journal of Digital Earth*, vol. 4, no. 4, pp. 257–259, 2011.
- [48] R. Fielding, *Architectural Styles and the Design of Network-based Software Architectures*, Ph.D. dissertation, University of California, Irvine, 2000.
- [49] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.
- [50] M. Goodchild, "Citizens as sensors: the world of volunteered geography," *GeoJournal*, vol. 69, no. 4, pp. 211–221, 2007.
- [51] R. Avtar et al., "Exploring Renewable Energy Resources Using Remote Sensing and GIS," *Resources*, vol. 8, no. 3, p. 149, 2019.
- [52] World Bank/ESMAP, *Global Solar Atlas and Global Wind Atlas*. The World Bank Group, 2019.
- [53] S. Cros, L. Wald, T. Huld, and R. Müller, *The European Solar Radiation Atlas*. Springer, 2004.
- [54] P. Neis and D. Zielstra, "Recent developments and future trends in volunteered geographic information research: The case of OpenStreetMap," *Future Internet*, vol. 6, no. 1, pp. 76–106, 2014.
- [55] Google, *Project Sunroof*. Google Inc., 2017.
- [56] ΥΠΕΝ, *Ηλεκτρονικός Χάρτης ΑΠΕ*. Υπουργείο Περιβάλλοντος και Ενέργειας, 2020.
- [57] D. Chatzigeorgiou et al., "Solar cadastre for assessment of near-zero energy districts," *Sustainability*, vol. 15, no. 7, p. 6123, 2023.
- [58] R. Rodríguez et al., "Eolica 2.0: Web-based GIS system for wind energy assessment in Spain," *Renewable Energy*, vol. 75, pp. 640–649, 2015.
- [59] M. Šúri, T. Huld, E. Dunlop, and H. Ossenbrink, "Potential of solar electricity generation in the European Union member states," *Solar Energy*, vol. 81, no. 10, pp. 1295–1305, 2007.
- [60] EMD International, *WindPRO Software*. Aalborg, 2019.
- [61] C. Huang, J. Wang, Y. Li, and Y. Xu, "Big data driven fault diagnosis in power systems," *Renewable and Sustainable Energy Reviews*, vol. 76, pp. 928–935, 2017.
- [62] A. Dobos, "PVWatts Version 5 Manual," National Renewable Energy Laboratory (NREL), 2014.
- [63] A. Lapostolle et al., "BDPV: A participatory database of PV installations in France," *Energy Procedia*, vol. 83, pp. 245–253, 2015.
- [64] GIZ, *India Solar Resource Atlas*. Deutsche Gesellschaft für Internationale Zusammenarbeit, 2015.
- [65] Geoscience Australia, *AREMI Portal*. Government of Australia, 2016.
- [66] CSIR, *South Africa Renewable Energy Atlas*. Council for Scientific and Industrial Research, 2018.
- [67] Natural Resources Canada, *RETScreen Clean Energy Management Software*. Government of Canada, 2019.
- [68] EMHIREs, *European Meteorological High-resolution RES time series*. Joint Research Centre, 2016.
- [69] I. Gonzalez Aparicio et al., "EMHIREs dataset for wind and solar power generation," *Renewable Energy*, vol. 114, pp. 63–73, 2016.
- [70] M. Goodchild, "Citizens as sensors: the world of volunteered geography," *GeoJournal*, vol. 69, no. 4, pp. 211–221, 2007.
- [71] R. Avtar, P. Kumar, A. Oono, C. Saraswat, T. Chaminda, and T. N. Hinh, "Exploring Renewable Energy Resources Using Remote Sensing and GIS," *Resources*, vol. 8, no. 3, p. 149, 2019, doi: 10.3390/resources8030149.
- [72] A. Bieda, "Towards a Renewable Energy Source Cadastre," *Energies*, vol. 14, no. 23, p. 8095, 2021, doi: 10.3390/en14238095.

- [73] Enel Green Power, "GIS systems in renewables," *Enel Green Power stories*, 2023. [Online]. Available: <https://www.enelgreenpower.com>. [Accessed: Sep. 29, 2025].
- [74] M. F. Goodchild, "Geographic Information System (GIS)," in *International Encyclopedia of Human Geography*, R. Kitchin and N. Thrift, Eds. Oxford: Elsevier, 2009, pp. 350–354, doi: 10.1016/B978-008044910-4.00413-9.
- [75] *Open energy system databases*, Wikipedia, 2025. [Online]. Available: [https://en.wikipedia.org/wiki/Open\\_energy\\_system\\_databases](https://en.wikipedia.org/wiki/Open_energy_system_databases). [Accessed: Sep. 29, 2025].
- [76] *Participatory GIS*, Wikipedia, 2025. [Online]. Available: [https://en.wikipedia.org/wiki/Participatory\\_GIS](https://en.wikipedia.org/wiki/Participatory_GIS). [Accessed: Sep. 29, 2025].
- [77] Tethys, "A review of GIS and techno-economic tools for renewable energy," *Tethys*, 2023. [Online]. Available: <https://tethys.pnnl.gov>. [Accessed: Sep. 29, 2025].
- [78] UN-GGIM, *Greece National Spatial Data Infrastructure Country Report*. United Nations Committee of Experts on Global Geospatial Information Management, 2015. [Online]. Available: <https://ggim.un.org>. [Accessed: Sep. 29, 2025].
- [79] World Bank/ESMAP, *Global Solar Atlas*. The World Bank Group, 2019. [Online]. Available: <https://globalsolaratlas.info>. [Accessed: Sep. 29, 2025].
- [80] World Bank/ESMAP, *Global Wind Atlas*. The World Bank Group, 2019. [Online]. Available: <https://globalwindatlas.info>. [Accessed: Sep. 29, 2025].
- [81] M. F. Goodchild, "Geographic Information System (GIS)," in *International Encyclopedia of Human Geography*, R. Kitchin and N. Thrift, Eds. Oxford: Elsevier, 2009, pp. 350–354, doi: 10.1016/B978-008044910-4.00413-9.
- [82] G. Percivall, "OGC standards and geospatial interoperability," *Communications of the ACM*, vol. 51, no. 7, pp. 118–124, 2008.
- [83] M. F. Goodchild, "Citizens as sensors: the world of volunteered geography," *GeoJournal*, vol. 69, no. 4, pp. 211–221, 2007.
- [84] UN-GGIM, *Greece National Spatial Data Infrastructure Country Report*. United Nations Committee of Experts on Global Geospatial Information Management, 2015. [Online]. Available: <https://ggim.un.org>. [Accessed: Sep. 29, 2025].
- [85] E. Ries, *The Lean Startup: How Constant Innovation Creates Radically Successful Businesses*. New York: Crown Publishing, 2011.
- [86] D. Dutta, D. Vandermeer, and K. Zhu, "Business innovation and proof of concept in IT projects," *MIS Quarterly Executive*, vol. 16, no. 1, pp. 23–35, 2017.
- [87] A. Croll and B. Yoskovitz, *Lean Analytics: Use Data to Build a Better Startup Faster*. Sebastopol, CA: O'Reilly Media, 2013.
- [88] S. Blank and B. Dorf, *The Startup Owner's Manual: The Step-by-Step Guide for Building a Great Company*. Pescadero, CA: K&S Ranch, 2012.
- [89] H. H. Olsson and J. Bosch, "Towards Continuous Customer Validation: A Conceptual Model for Combining Qualitative Customer Feedback with Quantitative Customer Observation," in *Proc. Int. Conf. Agile Software Development*, 2015, pp. 142–157.
- [90] D. Griffith, *React 17 Design Patterns and Best Practices*. Packt Publishing, 2021.
- [91] G. Bierman, M. Abadi, and M. Torgersen, *Understanding TypeScript*. Microsoft Research, 2014.
- [92] E. You, *Vite: Next Generation Frontend Tooling*. 2021. [Online]. Available: <https://vitejs.dev/>. [Accessed: Oct. 1, 2025].
- [93] V. Agafonkin, *Leaflet: An open-source JavaScript library for mobile-friendly interactive maps*. 2011. [Online]. Available: <https://leafletjs.com>. [Accessed: Oct. 1, 2025].
- [94] R. O. Obe and L. S. Hsu, *PostGIS in Action*. Manning Publications, 2011.

- [95] Supabase, *Supabase Documentation*. 2023. [Online]. Available: <https://supabase.com/docs>. [Accessed: Oct. 1, 2025].
- [96] J. Loeliger and M. McCullough, *Version Control with Git*. O'Reilly Media, 2012.
- [97] Netlify, *Netlify Documentation*. 2023. [Online]. Available: <https://docs.netlify.com/>. [Accessed: Oct. 1, 2025].
- [98] S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to Build High-Performance Network Programs," *IEEE Internet Computing*, vol. 14, no. 6, pp. 80–83, 2010.
- [99] T. Bray, *The JavaScript Object Notation (JSON) Data Interchange Format*, RFC 7159, IETF, 2014. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc7159>. [Accessed: Oct. 1, 2025].
- [100] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta, "Comparison of JSON and XML Data Interchange Formats: A Case Study," in *Proc. 2009 IEEE Int. Conf. Computer Applications and Industrial Electronics*, 2009, pp. 157–162.
- [101] H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, and T. Schaub, *The GeoJSON Format*, RFC 7946, IETF, 2016. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc7946>. [Accessed: Oct. 1, 2025].
- [102] Baumann, P. (2018). *Web Coverage Service (WCS) – Interface Standard*. Open Geospatial Consortium.
- [103] De la Beaujardière, J. (2006). *OpenGIS Web Map Service (WMS) Implementation Specification*. Open Geospatial Consortium.
- [104] Goodchild, M. F. (2009). *Geographic Information Systems*. In: International Encyclopedia of Human Geography. Elsevier.
- [105] OGC (2021). *OpenGIS Web Feature Service (WFS) 3.0 Standard*. Open Geospatial Consortium.
- [106] OGC (2023). *About the Open Geospatial Consortium (OGC)*. Available at: <https://www.ogc.org>
- [107] Copernicus (2023). *Copernicus Open Access Hub: Spatial Data Access Services*. European Space Agency.
- [108] World Bank / ESMAP (2019). *Global Solar Atlas*. World Bank Group.
- [109] Z. Guo *et al.*, "TransPV: Refining photovoltaic panel detection accuracy through a vision transformer-based deep learning model," *Applied Energy*, vol. 355, p. 122282, 2024.
- [110] K. Ioannou *et al.*, "Automatic Detection of Photovoltaic Farms Using Satellite Imagery and Convolutional Neural Networks," *Sustainability*, vol. 13, no. 9, p. 5323, 2021.
- [111] H. Al Hashimi and C. Alwahaishi, "A Critical Review of the Use of Spikes in Agile Software Development," in *Proc. ICSEA 2019*, pp. 155–162, 2019.
- [112] A. Russell and A. Berriman, "Progressive Web Apps: Escaping Tabs Without Losing Our Soul," *Google Developers Conference*, 2016.

## ΠΑΡΑΡΤΗΜΑ Α: Παράθεση SQL με SCHEMA της Β.Δ.

create db.sql: Κώδικας SQL για δημιουργία schema της Βάσης Δεδομένων

```
1  -- =====
2  -- DROP & CLEAN
3  -- =====
4  DROP VIEW IF EXISTS public.v_installations_geojson;
5  DROP FUNCTION IF EXISTS public.trg_installations_area();
6  DROP TRIGGER IF EXISTS trg_installations_area_biu ON public.installations;
7  DROP FUNCTION IF EXISTS public.api_insert_installation(text,numeric,numeric,text,text);
8  DROP TABLE IF EXISTS public.installations;
9
10 -- =====
11 -- ENUM περιφερειών (δημιουργείται αν δεν υπάρχει ήδη)
12 -- =====
13 DO $$
14 BEGIN
15     IF NOT EXISTS (SELECT 1 FROM pg_type WHERE typname = 'gr_region') THEN
16         CREATE TYPE gr_region AS ENUM (
17             'Ανατολική Μακεδονία και Θράκη',
18             'Κεντρική Μακεδονία',
19             'Δυτική Μακεδονία',
20             'Ηπειρος',
21             'Θεσσαλία',
22             'Ιόνιες Νήσοι',
23             'Δυτική Ελλάδα',
24             'Στερεά Ελλάδα',
25             'Αττική',
26             'Πελοπόννησος',
27             'Βόρειο Αιγαίο',
28             'Νότιο Αιγαίο',
29             'Κρήτη'
30         );
31     END IF;
32 END$$;
33
34 -- =====
35 -- ΠΙΝΑΚΑΣ installations
36 -- =====
37 CREATE TABLE public.installations (
38     id          bigserial PRIMARY KEY,
39     code        text NOT NULL UNIQUE,
40     power_max   numeric,
41     power_avg   numeric,
42     region      gr_region NOT NULL,
43     geom        geometry(Polygon, 4326) NOT NULL,
44     area_m2     double precision NOT NULL DEFAULT 0,
45     created_at  timestamptz NOT NULL DEFAULT now()
46 );
47
48 -- Indexes
49 CREATE INDEX IF NOT EXISTS idx_installations_geom ON public.installations USING GIST (
geom);
50 CREATE INDEX IF NOT EXISTS idx_installations_region ON public.installations (region);
51
52 -- =====
53 -- TRIGGER: area_m2 = ST_Area(geography) (ακρίβεια σε m²)
54 -- =====
55 CREATE OR REPLACE FUNCTION public.trg_installations_area() RETURNS trigger
56 LANGUAGE plpgsql AS $$
57 BEGIN
58     IF NEW.geom IS NOT NULL THEN
59         NEW.area_m2 := ST_Area(NEW.geom::geography);
60     ELSE
61         NEW.area_m2 := 0;
62     END IF;
63     RETURN NEW;
64 END;
65 $$;
66
67 CREATE TRIGGER trg_installations_area_biu
68 BEFORE INSERT OR UPDATE OF geom ON public.installations
```

```

69 FOR EACH ROW EXECUTE FUNCTION public.trg_installations_area();
70
71 -- =====
72 -- RPC: insert + RETURNING id (region text -> cast σε gr_region)
73 -- =====
74 CREATE OR REPLACE FUNCTION public.api_insert_installation(
75     code      text,
76     power_avg numeric,
77     power_max numeric,
78     region    text,
79     wkt       text
80 )
81 RETURNS bigint
82 LANGUAGE plpgsql
83 SECURITY DEFINER
84 SET search_path = public
85 AS $$
86 DECLARE
87     new_id bigint;
88     is_valid boolean;
89 BEGIN
90     -- έλεγχος τιμής region έναντι του ENUM
91     SELECT EXISTS (
92         SELECT 1
93         FROM unnest(enum_range(NULL::gr_region)) AS t(val)
94         WHERE t.val::text = region
95     ) INTO is_valid;
96
97     IF NOT is_valid THEN
98         RAISE EXCEPTION 'Invalid region: % (must match gr_region)', region USING ERRCODE=
99         '22P02';
100     END IF;
101
102     INSERT INTO public.installations (code, power_max, power_avg, region, geom)
103     VALUES (code, power_max, power_avg, region::gr_region, ST_GeomFromText(wkt, 4326))
104     RETURNING id INTO new_id;
105
106     RETURN new_id;
107 END;
108 $$;
109
110 GRANT EXECUTE ON FUNCTION public.api_insert_installation(text,numeric,numeric,text,text)
111 TO anon, authenticated;
112
113 -- =====
114 -- VIEW για το frontend (MONO area_m2)
115 -- =====
116 CREATE OR REPLACE VIEW public.v_installations_geojson AS
117 SELECT
118     i.id,
119     jsonb_build_object(
120         'type', 'Feature',
121         'id',   i.id,
122         'geometry', ST_AsGeoJSON(i.geom)::jsonb,
123         'properties', jsonb_build_object(
124             'id',      i.id,
125             'code',    i.code,
126             'region',  i.region::text,
127             'power_max', i.power_max,
128             'power_avg', i.power_avg,
129             'area_m2', i.area_m2
130         ) AS feature
131 FROM public.installations i;
132
133 GRANT SELECT ON public.v_installations_geojson TO anon, authenticated;
134
135 -- refresh PostgREST cache
136 NOTIFY pgrst, 'reload schema';

```

## ΠΑΡΑΡΤΗΜΑ Β: Παράθεση κώδικα της εφαρμογής

### ΠΑΡΑΡΤΗΜΑ Β1. Ο κώδικας του αρχείου src\map\InstallationsLayer.tsx

```
1  import { useEffect, useRef } from "react";
2  import { useMap } from "react-leaflet";
3  import L from "leaflet";
4  import type { Feature, FeatureCollection, Geometry } from "geojson";
5  import { supabase } from "../supabaseClient";
6
7  type Props = {
8    /** Κλειδί που αλλάζει σε κάθε εισαγωγή/ενημέρωση ώστε να ξαναγίνεται fetch */
9    refreshKey?: number;
10   /** "ALL" ή συγκεκριμένο όνομα Περιφέρειας (στα ελληνικά) για φιλτράρισμα */
11   regionFilter?: string;
12 };
13
14 /**
15  * To component InstallationsLayer:
16  * - Φορτώνει από τη ΒΔ όλα τα πολύγωνα (installations)
17  * - Εμφανίζει τα πολύγωνα στον Leaflet χάρτη
18  * - Υποστηρίζει tooltip με πληροφορίες (code, ισχύς, εμβαδό)
19  * - Υποστηρίζει φίλτρο ανά Περιφέρεια
20  */
21 export default function InstallationsLayer({ refreshKey, regionFilter }: Props) {
22   const map = useMap(); // Πρόσβαση στον Leaflet χάρτη
23   const layerRef = useRef<L.GeoJSON | null>(null); // Αναφορά στο GeoJSON layer
24
25   useEffect(() => {
26     let cancelled = false; // Σημαία για cleanup όταν γίνει unmount
27
28     // Δημιουργία (ή reset) του GeoJSON layer
29     if (!layerRef.current) {
30       layerRef.current = L.geoJSON(undefined, {
31         style: () => ({
32           color: "#d33", // Κόκκινο περίγραμμα
33           weight: 2, // Πάχος γραμμής
34           opacity: 1, // Ορατότητα γραμμής
35           fillOpacity: 0.15, // Διαφάνεια γεμίσματος
36         }),
37         // Εφαρμογή tooltip για κάθε feature (πολύγωνο)
38         onEachFeature: (f: Feature, l: L.Layer) => {
39           const p: any = (f.properties ?? {}) as any;
40
41           // Ανάγνωση properties με fallback σε "-"
42           const code = p?.code ?? "-";
43           const max = p?.power_max ?? p?.powerMax ?? "-";
44           const avg = p?.power_avg ?? p?.powerAvg ?? "-";
45           const area = p?.area_m2 ?? p?.area ?? "-";
46
47           // HTML για το tooltip με σωστές μονάδες (kW, m²)
48           const html =
49             `<div style="line-height:1.3">` +
50             `<div><b>${code}</b></div>` +
51             `<div>Max: ${Number(max).toLocaleString("el-GR")} kW</div>` +
52             `<div>Avg: ${Number(avg).toLocaleString("el-GR")} kW</div>` +
53             `<div>Area: ${Number(area).toLocaleString("el-GR")} m²</div>` +
54             `</div>`;
55
56           // Σύνδεση tooltip στο πολύγωνο
57           (l as any).bindTooltip(html, {
58             direction: "center",
59             opacity: 0.9,
60             sticky: false,
61           });
62         },
63       }).addTo(map);
64     } else {
65       // Αν υπάρχει ήδη, καθάρισε τα δεδομένα
66       layerRef.current.clearLayers();
67     }
68
69     const layer = layerRef.current;
```

```

70
71 // Συνάρτηση που φέρνει δεδομένα από Supabase
72 async function fetchData() {
73   // Query από το view v_installations_geojson
74   let q = supabase.from("v_installations_geojson").select("feature");
75
76   // Αν ζητήθηκε φίλτρο Περιφέρειας
77   if (regionFilter && regionFilter !== "ALL") {
78     // Χρησιμοποιούμε JSONB contains για να φιλτράρουμε στο properties.region
79     q = (q as any).contains("feature->properties", { region: regionFilter });
80   }
81
82   const { data, error } = await q;
83
84   if (cancelled) return; // Αν έχει γίνει unmount, βγες
85
86   if (error) {
87     console.error("Supabase load error:", error);
88     return;
89   }
90
91   // Μετατροπή αποτελεσμάτων σε FeatureCollection
92   const features: Feature<Geometry>[] = (data ?? [])
93     .map((r: any) => r.feature as Feature<Geometry>)
94     .filter((f) => f && f.geometry);
95
96   const fc: FeatureCollection = {
97     type: "FeatureCollection",
98     features,
99   };
100
101   // Προσθήκη δεδομένων στο layer
102   layer?.addData(fc as any);
103
104   // Προσπάθεια αυτόματου zoom στο νέο σύνολο
105   try {
106     const b = layer?.getBounds();
107     if (b && b.isValid()) {
108       map.fitBounds(b.pad(0.15)); // μικρό "περιθώριο" γύρω από τα polygons
109     }
110   } catch {
111     // Αν κάτι πάει στραβά, αγνόησέ το
112   }
113 }
114
115 // Κάλυψε τη συνάρτηση
116 fetchData();
117
118 // Cleanup όταν αποσυνδεθεί το component
119 return () => {
120   cancelled = true;
121   // Αν θέλαμε πλήρη αφαίρεση layer (πχ σε αλλαγή σελίδας):
122   // if (layerRef.current) {
123   //   map.removeLayer(layerRef.current);
124   //   layerRef.current = null;
125   // }
126 };
127 }, [map, refreshKey, regionFilter]); // To effect ξανατρέχει σε αλλαγή refreshKey ή
regionFilter
128
129 return null; // Δεν κάνει render κάτι στο DOM, μόνο στον Leaflet χάρτη
130 }
131

```

## ΠΑΡΑΡΤΗΜΑ Β2. Ο κώδικας του αρχείου src\map\MapView.tsx

```
1 // src/map/MapView.tsx
2 // -----
3 // Κύριο component του χάρτη. Ρυθμίζει το Leaflet <MapContainer>,
4 // εμφανίζει το dropdown φίλτρου Περιφέρειας, το κουμπί "NEW INSTALLATION",
5 // και «δένει» το layer των εγκαταστάσεων (InstallationsLayer).
6 // Διατηρεί επίσης ένα refreshKey ώστε να ξαναφορτώνεται το layer
7 // αμέσως μετά από μια νέα εισαγωγή (INSERT).
8 // -----
9
10 import { MapContainer, TileLayer, ZoomControl } from "react-leaflet";
11 import "leaflet/dist/leaflet.css";
12 import { useState } from "react";
13 import InstallationsLayer from "../InstallationsLayer";
14 import NewInstallation from "../ui/NewInstallation";
15
16 // Οι επιλογές που φαίνονται στο UI (ετικέτες στα ελληνικά)
17 const UI_REGIONS = [
18   "ΟΛΕΣ",
19   "Ανατολική Μακεδονία και Θράκη",
20   "Κεντρική Μακεδονία",
21   "Δυτική Μακεδονία",
22   "Ηπειρος",
23   "Θεσσαλία",
24   "Ιόνιες Νήσοι",
25   "Δυτική Ελλάδα",
26   "Στερεά Ελλάδα",
27   "Αττική",
28   "Πελοπόννησος",
29   "Βόρειο Αιγαίο",
30   "Νότιο Αιγαίο",
31   "Κρήτη",
32 ] as const;
33
34 /**
35  * Μετατροπή της επιλογής του UI σε τιμή που καταλαβαίνει το layer/API.
36  * - "ΟΛΕΣ" => "ALL" (δηλ. χωρίς φίλτρο)
37  * - οποιαδήποτε άλλη τιμή περνάει αυτούσια (ίδιο string)
38  */
39 function toRegionFilter(value: string): string {
40   return value === "ΟΛΕΣ" ? "ALL" : value;
41 }
42
43 export default function MapView() {
44   // Χρησιμοποιείται σαν «σήμα» ανανέωσης στο InstallationsLayer.
45   // Κάθε φορά που αλλάζει, το layer κάνει εκ νέου fetch τα δεδομένα.
46   const [refreshKey, setRefreshKey] = useState(0);
47
48   // Τιμή dropdown όπως φαίνεται στον χρήστη (ελληνική ετικέτα)
49   const [regionUI, setRegionUI] = useState<string>("ΟΛΕΣ");
50
51   // Εμφάνιση/απόκρυψη του modal φόρμας για νέα εγκατάσταση
52   const [showForm, setShowForm] = useState(false);
53
54   // «Μετάφραση» της UI επιλογής σε φίλτρο που περνάει στο layer
55   const regionFilter = toRegionFilter(regionUI);
56
57   // Αύξηση «κλειδιού» για να κάνουμε trigger re-render / refetch του layer
58   const bump = () => setRefreshKey((k) => k + 1);
59
60   return (
61     <div style={{ position: "relative", width: "100%", height: "100vh" }}>
62       {/----- Dropdown Περιφέρειας (πάνω-αριστερά) -----
63        Τοποθετείται λίγο δεξιότερα από τα zoom controls για να μη τα καλύπτει.
64        pointerEvents: "auto" ώστε να είναι κλικ-άξιμο πάνω από τον χάρτη.
65        ----- */}
66       <div
67         style={{
68           position: "absolute",
69           top: 10,
```

```

70     left: 58, // λίγο δεξιά από τα +/- του Leaflet
71     zIndex: 1000,
72     background: "rgba(255,255,255,0.9)",
73     border: "1px solid #ccc",
74     borderRadius: 6,
75     padding: "4px 8px",
76     boxShadow: "0 1px 3px rgba(0,0,0,0.15)",
77     pointerEvents: "auto",
78   })
79 >
80 <label style={{ fontSize: 12, marginRight: 6 }}>Περιφέρεια:</label>
81 <select
82   value={regionUI}
83   onChange={(e) => setRegionUI(e.target.value)}
84   style={{ fontSize: 12, padding: "4px 6px" }}
85 >
86   {UI_REGIONS.map((r) => (
87     <option key={r} value={r}>
88       {r}
89     </option>
90   ))}
91 </select>
92 </div>
93
94 {/} ----- Κουμπί NEW INSTALLATION (πάνω-δεξιά) ----- */}
95 <div
96   style={{
97     position: "absolute",
98     top: 10,
99     right: 10,
100    zIndex: 1000,
101    pointerEvents: "auto",
102  }}
103 >
104   <button
105     onClick={() => setShowForm(true)}
106     style={{
107       padding: "6px 10px",
108       borderRadius: 6,
109       border: "1px solid #ccc",
110       background: "#fff",
111       cursor: "pointer",
112     }}
113   >
114     NEW INSTALLATION
115   </button>
116 </div>
117
118 {/} ----- Χάρτης Leaflet ----- */}
119 <MapContainer
120   center={[40.3, 21.8]} // αρχικό κέντρο (περιοχή Κοζάνης)
121   zoom={10}
122   minZoom={5}
123   maxZoom={18}
124   zoomControl={false} // θα βάλουμε το ZoomControl χειροκίνητα (πάνω-αριστερά)
125   style={{ width: "100%", height: "100%" }}
126 >
127   {/} Εμφάνιση των +/- επάνω-αριστερά (προεπιλογή Leaflet) */}
128   <ZoomControl position="topleft" />
129
130   {/} Υπόβαθρο OpenStreetMap */}
131   <TileLayer
132     attribution='&copy; OpenStreetMap contributors'
133     url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
134   />
135
136   {/} Layer εγκαταστάσεων.
137   - regionFilter: "ALL" ή ονομασία Περιφέρειας (ελληνικά)
138   - refreshKey: αυξάνεται κάθε φορά που δημιουργείται νέα εγγραφή,

```

```

139         προκαλώντας refetch ώστε να εμφανιστεί αμέσως το νέο πολύγωνο. */}
140     <InstallationsLayer
141         refreshKey={refreshKey}
142         regionFilter={regionFilter}
143     />
144 </MapContainer>
145
146     { /* ----- Modal Νέας Εγκατάστασης ----- */ }
147     { showForm && (
148         <NewInstallation
149             // Κλείσιμο φόρμας (χωρίς submit)
150             onClose={() => setShowForm(false)}
151             // Μετά το επιτυχές INSERT:
152             // 1) κλείνουμε το modal
153             // 2) αυξάνουμε το refreshKey για να ξαναφορτωθεί το layer
154             onCreate={() => {
155                 setShowForm(false);
156                 bump();
157             }}
158         />
159     ) }
160 </div>
161 );
162 }
163

```

## ΠΑΡΑΡΤΗΜΑ Β3. Ο κώδικας του αρχείου src\map\AreasLayer.tsx

```
1 import { useEffect } from "react";
2 import { useMap } from "react-leaflet";
3 import L from "leaflet";
4
5 // Ορισμός του endpoint της Supabase για το RPC (fn_areas_bbox)
6 // που επιστρέφει τα πολύγωνα "areas" τα οποία εμπίπτουν μέσα στο τρέχον bounding box
7 // του χάρτη
8 const API = import.meta.env.VITE_SUPABASE_URL + "/rest/v1/rpc/fn_areas_bbox";
9 const API_KEY = import.meta.env.VITE_SUPABASE_ANON_KEY;
10
11 export default function AreasLayer() {
12   // Απόκτηση του αντικειμένου map από το React-Leaflet
13   const map = useMap();
14
15   useEffect(() => {
16     // Δημιουργία ξεχωριστού "pane" ώστε τα layers με τα areas
17     // να μην παρεμβάλλονται ή να «κλέβουν» events από άλλα layers (π.χ. draw controls)
18     if (!map.getPane("areasPane")) {
19       map.createPane("areasPane");
20       const pane = map.getPane("areasPane");
21       if (pane) pane.style.zIndex = "250"; // Ρύθμιση σειράς εμφάνισης (zIndex)
22     }
23
24     // Δημιουργία Leaflet GeoJSON layer για την απεικόνιση των περιοχών (areas)
25     const layer = L.geoJSON(undefined, {
26       pane: "areasPane", // Το pane στο οποίο θα μπει το layer
27       interactive: false, // Δεν είναι διαδραστικά τα πολυγωνικά features
28       style: () => ({ weight: 2, opacity: 1, fillOpacity: 0.15 }), // Στυλ για τα
29       // πολύγωνα
30       pointToLayer: (_f, latlng) =>
31         // Αν τυχόν επιστραφεί σημείο αντί για πολύγωνο, εμφανίζεται ως marker (χωρίς
32         // interaction)
33         L.marker(latlng, { interactive: false, keyboard: false }),
34       onEachFeature: (f, l) => {
35         // Προσθήκη popup με το όνομα της περιοχής, αν υπάρχει
36         const p = (f.properties ?? {}) as any;
37         const n = p?.name ?? "-";
38         l.bindPopup(`<b>${n}</b>`);
39       }
40     }).addTo(map);
41
42     // Συνάρτηση που καλεί το RPC της Supabase και φορτώνει τα δεδομένα των areas
43     const loadData = async () => {
44       // Λήψη του τρέχοντος bounding box του χάρτη (όρια θέασης)
45       const b = map.getBounds();
46       const body = {
47         minx: b.getWest(),
48         miny: b.getSouth(),
49         maxx: b.getEast(),
50         maxy: b.getNorth()
51       };
52       try {
53         // Αποστολή POST request στο Supabase RPC
54         const resp = await fetch(API, {
55           method: "POST",
56           headers: {
57             "Content-Type": "application/json",
58             apikey: API_KEY,
59             Authorization: `Bearer ${API_KEY}` // Χρήση API key για authorization
60           },
61           body: JSON.stringify(body) // Στέλνουμε το bounding box ως JSON
62         });
63         if (!resp.ok) {
64           // Αν αποτύχει η κλήση στο RPC εμφανίζεται μήνυμα στο console
65           console.error("Supabase RPC error:", await resp.text());
66           return;
67         }
68         // Ανάγνωση FeatureCollection JSON και προσθήκη στο layer
69         const fc = await resp.json();
70       } catch (error) {
71         console.error("Error loading areas:", error);
72       }
73     };
74     loadData();
75   });
76 }
```

```

67     layer.clearLayers().addData(fc as any);
68   } catch (err) {
69     console.error("Network error", err); // Σφάλμα δικτύου
70   }
71 };
72
73 // Αρχικό φόρτωμα δεδομένων
74 loadData();
75
76 // Επαναφόρτωση δεδομένων όταν αλλάξει η θέαση του χάρτη (moveend)
77 map.on("moveend", loadData);
78
79 // Custom event "reload-areas" για επαναφόρτωση
80 map.on("reload-areas", loadData);
81
82 // Cleanup: αποσύνδεση listeners και αφαίρεση layer όταν το component καταστραφεί
83 return () => {
84   map.off("moveend", loadData);
85   map.off("reload-areas", loadData);
86   map.removeLayer(layer);
87 };
88 }, [map]);
89
90 // Το component δεν εμφανίζει κάποιο JSX, απλά χειρίζεται Leaflet layers
91 return null;
92 }
93

```

## ΠΑΡΑΡΤΗΜΑ Β4. Ο κώδικας του αρχείου src\map\DrawGeometries.tsx

```
1 import { useEffect } from "react";
2 import { useMap } from "react-leaflet";
3 import L from "leaflet";
4 import "leaflet-draw";
5
6 // Σταθερά που δείχνει στο Netlify Function για δημιουργία νέας περιοχής (area)
7 const CREATE_API = "/.netlify/functions/create-area";
8
9 export default function DrawGeometries() {
10   // Παίρνουμε το αντικείμενο του Leaflet χάρτη από το React-Leaflet
11   const map = useMap();
12
13   useEffect(() => {
14     // Δημιουργούμε ένα FeatureGroup (μια ομάδα γεωμετρικών features) και το προσθέτουμε
15     // στο χάρτη
16     const drawnItems = new L.FeatureGroup().addTo(map);
17
18     // Δημιουργία του εργαλείου σχεδίασης (Draw Control)
19     const drawControl = new (L.Control as any).Draw({
20       draw: {
21         marker: true, // Επιτρέπει σχεδίαση σημείων (markers)
22         polygon: {
23           showArea: true, // Εμφανίζει το εμβαδόν του πολυγώνου κατά τη σχεδίαση
24           allowIntersection: false, // Δεν επιτρέπει πολυγωνικές γραμμές να τέμνονται
25           finishOnDoubleClick: false // Αποτρέπει να «κλείσει» το πολύγωνο με διπλό κλικ
26         },
27         polyline: false, // Δεν επιτρέπεται γραμμή
28         rectangle: false, // Δεν επιτρέπεται παραλληλόγραμμο
29         circle: false, // Δεν επιτρέπεται κύκλος
30         circlemarker: false // Δεν επιτρέπεται circle marker
31       },
32       edit: {
33         featureGroup: drawnItems, // Ομάδα γεωμετριών που μπορούν να επεξεργαστούν
34         edit: false, // Δεν επιτρέπεται επεξεργασία
35         remove: false // Δεν επιτρέπεται διαγραφή
36       }
37     });
38
39     // Όταν ξεκινάει το σχέδιο, απενεργοποιούμε το zoom με διπλό κλικ
40     const onDrawStart = () => map.doubleClickZoom.disable();
41     // Όταν τελειώνει το σχέδιο, ξαναενεργοποιούμε το zoom με διπλό κλικ
42     const onDrawStop = () => map.doubleClickZoom.enable();
43
44     // Σύνδεση των παραπάνω events
45     map.on((L as any).Draw.Event.DRAWSTART, onDrawStart);
46     map.on((L as any).Draw.Event.DRAWSTOP, onDrawStop);
47
48     // Προσθήκη του DrawControl στο χάρτη
49     map.addControl(drawControl);
50
51     // Event handler που ενεργοποιείται όταν δημιουργηθεί μια νέα γεωμετρία
52     const onCreated = async (e: any) => {
53       const layer = e.layer as L.Layer; // Παίρνουμε το layer που δημιουργήθηκε
54       drawnItems.addLayer(layer); // Προσθέτουμε τη γεωμετρία στο FeatureGroup
55       const feature = (layer as any).toGeoJSON(); // Μετατροπή της γεωμετρίας σε GeoJSON
56
57       try {
58         // Κλήση στο Netlify Function για αποθήκευση στο backend
59         const resp = await fetch(CREATE_API, {
60           method: "POST",
61           headers: { "Content-Type": "application/json" },
62           body: JSON.stringify({
63             // Ανάλογα τον τύπο γεωμετρίας βάζουμε το όνομα
64             name: feature.geometry?.type === "Point" ? "Νέο σημείο" : "Νέο πολύγωνο",
65             properties: { source: "manual" }, // Βοηθητική ιδιότητα για το πως
66             // δημιουργήθηκε
67             geometry: feature.geometry // Η γεωμετρία (Point ή Polygon)
68           })
69         );
70       }
71     };
72   });
73 }
```

```

68     if (!resp.ok) {
69         // Αν αποτύχει το request εμφανίζουμε σφάλμα στο console
70         console.error("create-area failed:", await resp.text());
71     } else {
72         // Ειδοποιούμε το χάρτη να επαναφορτώσει τα areas
73         map.fire("reload-areas");
74     }
75 } catch (err) {
76     // Διαχείριση σφαλμάτων δικτύου
77     console.error("Network error saving area:", err);
78 } finally {
79     // Αφαιρούμε προσωρινά τη γεωμετρία από το drawnItems ώστε να μην μείνει διπλή
80     drawnItems.removeLayer(layer);
81 }
82 };
83
84 // Σύνδεση του event CREATED με το onCreate
85 map.on((L as any).Draw.Event.CREATED, onCreate);
86
87 // Καθαρισμός (cleanup) όταν καταστραφεί το component
88 return () => {
89     map.off((L as any).Draw.Event.CREATED, onCreate);
90     map.off((L as any).Draw.Event.DRAWSTART, onDrawStart);
91     map.off((L as any).Draw.Event.DRAWSTOP, onDrawStop);
92     map.removeControl(drawControl); // Αφαίρεση του draw control
93     map.removeLayer(drawnItems);    // Αφαίρεση του FeatureGroup
94     try { map.doubleClickZoom.enable(); } catch {}
95 };
96 }, [map]); // Εξαρτάται μόνο από το αντικείμενο map
97
98 // Δεν επιστρέφουμε UI, μόνο side-effects
99 return null;
100 }
101

```

## ΠΑΡΑΡΤΗΜΑ Β5. Ο κώδικας του αρχείου src/ui/NewInstallation.tsx

```
1 // src/ui/NewInstallation.tsx
2 // -----
3 // Φόρμα "Νέα Εγκατάσταση": επιτρέπει στον χρήστη να εισάγει
4 // κωδικό, ισχείς (μέγιστη/μέση), περιφέρεια και συντεταγμένες
5 // κορυφών (lon lat) για πολύγωνο. Παρέχει:
6 // - Parse επικολλημένου CSV (lon,lat ανά γραμμή) με κόμμα ή τελεία
7 // - Προεπισκόπηση polygon πάνω στον Leaflet χάρτη πριν το submit
8 // - Υποβολή σε Netlify Function (create-installation) με σώμα:
9 //   { code, power_max, power_avg, region, coords, wkt }
10 // - Αισιόδοξη απεικόνιση: αποστέλλει προσωρινό Feature στο Layer,
11 //   ώστε να μην χάνεται το πολύγωνο μέχρι να γίνει reload.
12 // - Μετακινούμενο dialog (drag) ως προς την οθόνη (position: fixed).
13 // -----
14
15 import React, { useEffect, useMemo, useRef, useState } from "react";
16 import ReactDOM from "react-dom";
17 import * as L from "leaflet";
18
19 // Τύπος για ένα ζεύγος συντεταγμένων που κρατιέται ως κείμενο
20 type Pair = { lon: string; lat: string };
21
22 // Οι 13 διοικητικές περιφέρειες της Ελλάδας (σταθερή λίστα)
23 const REGIONS = [
24   "Ανατολική Μακεδονία και Θράκη",
25   "Κεντρική Μακεδονία",
26   "Δυτική Μακεδονία",
27   "Ηπειρος",
28   "Θεσσαλία",
29   "Ιόνιες Νήσοι",
30   "Δυτική Ελλάδα",
31   "Στερεά Ελλάδα",
32   "Αττική",
33   "Πελοπόννησος",
34   "Βόρειο Αιγαίο",
35   "Νότιο Αιγαίο",
36   "Κρήτη",
37 ] as const;
38
39 // Λιγοστά inline styles για απλότητα (χωρίς εξωτερικό CSS)
40 const labelStyle: React.CSSProperties = { fontSize: 12, opacity: 0.7 };
41 const grid2Cols: React.CSSProperties = {
42   display: "grid",
43   gridTemplateColumns: "140px 1fr",
44   gap: 8,
45 };
46 const btnBar: React.CSSProperties = { display: "flex", gap: 8, marginTop: 6 };
47 // Το κουμπί NEW INSTALLATION καρφωμένο πάνω δεξιά
48 const fixedButton: React.CSSProperties = {
49   position: "fixed",
50   right: 12,
51   top: 12,
52   zIndex: 1000,
53   padding: "8px 12px",
54 };
55
56 // Κύριο component
57 export default function NewInstallation({
58   onCreate,
59   onClose,
60 }: {
61   // Καλείται αφού ολοκληρωθεί επιτυχώς η υποβολή (για refresh Layer)
62   onCreate?: () => void;
63   // Αν θες να κάνεις κάτι όταν κλείνει το dialog
64   onClose?: () => void;
65 }) {
66   // Αναφορές στο <dialog> και στο "header" του για drag
67   const dlgRef = useRef<HTMLDialogElement>(null);
68   const headerRef = useRef<HTMLDivElement>(null);
69 }
```

```

70 // Κατάσταση ανοίγματος/κλεισίματος modal
71 const [open, setOpen] = useState(false);
72
73 // Καταστάσεις για τα πεδία φόρμας
74 const [code, setCode] = useState("");
75 const [powerMax, setPowerMax] = useState<string>(""); // σε kW
76 const [powerAvg, setPowerAvg] = useState<string>(""); // σε kW
77 const [region, setRegion] = useState<(typeof REGIONS)[number]>("Αυτική Μακεδονία");
78
79 // Πλήθος κορυφών & ζεύγη (lon,lat) που θα συμπληρώσει ο χρήστης
80 const [vertices, setVertices] = useState<number>(3);
81 const [pairs, setPairs] = useState<Pair[]>([
82   { lon: "", lat: "" },
83   { lon: "", lat: "" },
84   { lon: "", lat: "" },
85 ]);
86
87 // Πεδίο ελεύθερου κειμένου για επικόλληση CSV (lon,lat ανά γραμμή)
88 const [csv, setCsv] = useState("");
89
90 // Κρατάμε το προσωρινό polygon της προεπισκόπησης για να το
91 // αφαιρούμε από τον χάρτη όταν χρειάζεται.
92 const previewLayer = useRef<L.Polygon | null>(null);
93
94 // Συγχρονισμός open <-> HTMLDialogElement API
95 useEffect(() => {
96   const d = dlgRef.current;
97   if (!d) return;
98   if (open && !d.open) d.showModal();
99   if (!open && d.open) d.close();
100 }, [open]);
101
102 // Κάνει το dialog "drag and drop" ως προς την οθόνη (όχι το DOM container)
103 useEffect(() => {
104   const header = headerRef.current;
105   const dlg = dlgRef.current;
106   if (!header || !dlg) return;
107
108   let sx = 0, sy = 0, ox = 0, oy = 0, dragging = false;
109
110   const onDown = (e: MouseEvent) => {
111     dragging = true;
112     sx = e.clientX;
113     sy = e.clientY;
114     const rect = dlg.getBoundingClientRect();
115     ox = rect.left;
116     oy = rect.top;
117     e.preventDefault();
118   };
119   const onMove = (e: MouseEvent) => {
120     if (!dragging) return;
121     const dx = e.clientX - sx;
122     const dy = e.clientY - sy;
123     dlg.style.left = `${ox + dx}px`;
124     dlg.style.top = `${oy + dy}px`;
125     dlg.style.margin = "0";
126     dlg.style.position = "fixed";
127   };
128   const onUp = () => (dragging = false);
129
130   header.addEventListener("mousedown", onDown);
131   window.addEventListener("mousemove", onMove);
132   window.addEventListener("mouseup", onUp);
133   return () => {
134     header.removeEventListener("mousedown", onDown);
135     window.removeEventListener("mousemove", onMove);
136     window.removeEventListener("mouseup", onUp);
137   };
138 }, []);

```

```

139
140 // Αν αλλάξει το πλήθος κορυφών, προσαρμόζουμε το pairs[]
141 useEffect(() => {
142   setPairs((prev) => {
143     const next = [...prev];
144     if (vertices > prev.length) {
145       for (let i = prev.length; i < vertices; i++) next.push({ lon: "", lat: "" });
146     } else if (vertices < prev.length) {
147       next.length = Math.max(0, vertices);
148     }
149     return next;
150   });
151 }, [vertices]);
152
153 // Μετατροπή από ζεύγη lon/lat -> WKT POLYGON
154 const toWKT = (pp: Pair[]) => {
155   // Μετατροπή σε αριθμούς & φιλτράρισμα μη έγκυρων
156   const nums = pp
157     .map((p) => [Number(p.lon), Number(p.lat)] as [number, number])
158     .filter(([lon, lat]) => isFinite(lon) && isFinite(lat));
159   if (nums.length < 3) return "";
160   // Δόμηση δακτυλίου "lon lat, lon lat, ..."
161   const ring = nums.map(([lon, lat]) => `${lon} ${lat}`);
162   // Κλείσιμο πολυγώνου (τελευταίο = πρώτο) αν δεν είναι ήδη κλειστό
163   if (ring[0] !== ring[ring.length - 1]) ring.push(ring[0]);
164   return `POLYGON((${ring.join(", ")}))`;
165 };
166
167 // Ζωγραφίζει προσωρινά στο χάρτη (Leaflet) την προεπισκόπηση polygon
168 const previewFrom = (pp: Pair[] = pairs) => {
169   const map = (window as any).__leafletMap as L.Map | undefined;
170   if (!map) return;
171   try {
172     // Καθάρισε παλιό preview, αν υπάρχει
173     if (previewLayer.current) {
174       map.removeLayer(previewLayer.current);
175       previewLayer.current = null;
176     }
177     // Μετατροπή σε [lat,lon] για Leaflet
178     const pts = pp
179       .map((p) => [Number(p.lat), Number(p.lon)] as [number, number])
180       .filter(([la, lo]) => isFinite(la) && isFinite(lo));
181     if (pts.length < 3) return;
182
183     const poly = L.polygon(pts, { weight: 2, opacity: 1, fillOpacity: 0.15 });
184     poly.addTo(map);
185     previewLayer.current = poly;
186
187     // Zoom στο πολύγωνο αν υπάρχουν έγκυρα bounds
188     const b = poly.getBounds();
189     if (b.isValid()) map.fitBounds(b, { padding: [20, 20] });
190   } catch (e) {
191     console.error("Preview error:", e);
192   }
193 };
194
195 // Όταν κλείνει το modal, καθάρισε τυχόν preview από τον χάρτη
196 useEffect(() => {
197   if (open) return;
198   const map = (window as any).__leafletMap as L.Map | undefined;
199   if (map && previewLayer.current) {
200     map.removeLayer(previewLayer.current);
201     previewLayer.current = null;
202   }
203 }, [open]);
204
205 // Παράρτημα CSV κειμένου (δέχεται "21.77 40.35" ή "21,77 40,35" κ.λπ.)
206 const parseCsv = () => {
207   const lines = csv

```

```

208     .split(/\r?\n/)
209     .map((l) => l.trim())
210     .filter(Boolean);
211
212     const parsed: Pair[] = [];
213     for (const line of lines) {
214         // Αποδεχόμαστε ; , κενά ως διαχωριστές, και μετατρέπουμε κόμμα -> τελεία
215         const parts = line.replace(/;/g, " ").replace(/,/g, ".").split(/\s+/);
216         if (parts.length < 2) continue;
217         const lon = parts[0],
218             lat = parts[1];
219         if (isFinite(Number(lon)) && isFinite(Number(lat))) {
220             parsed.push({ lon, lat });
221         }
222     }
223
224     if (parsed.length >= 3) {
225         // Ρύθμισε αυτόματα πλήθος κορυφών και τα input boxes
226         setVertices(parsed.length);
227         setPairs(parsed);
228         // Προεπισκόπηση
229         previewFrom(parsed);
230     } else {
231         alert("Χρειάζονται τουλάχιστον 3 ζεύγη (lon lat) για πολύγωνο.");
232     }
233 };
234
235 // Υποβολή στον server (Netlify Function)
236 const onSubmit = async () => {
237     if (!code.trim()) return alert("Δώσε κωδικό εγκατάστασης.");
238
239     // Μετατροπή των inputs σε αριθμητικά και φιλτράρισμα μη έγκυρων
240     const nums = pairs
241         .map((p) => ({ lon: Number(p.lon), lat: Number(p.lat) }))
242         .filter((p) => isFinite(p.lon) && isFinite(p.lat));
243     if (nums.length < 3) return alert("Χρειάζονται τουλάχιστον 3 έγκυρα σημεία.");
244
245     // Δημιουργία WKT για αποστολή στο backend (PostGIS θα υπολογίσει area_m2)
246     const wkt = toWKT(pairs);
247
248     // Σώμα αιτήματος: κωδικός/ισχύεις σε kW, περιφέρεια, coords & wkt
249     const body = {
250         code: code.trim(),
251         power_max: Number(powerMax || 0),
252         power_avg: Number(powerAvg || 0),
253         region,
254         coords: nums.map(({ lon, lat }) => [lon, lat] as [number, number]),
255         wkt,
256     };
257
258     try {
259         const res = await fetch("/.netlify/functions/create-installation", {
260             method: "POST",
261             headers: { "Content-Type": "application/json" },
262             body: JSON.stringify(body),
263         });
264         const text = await res.text();
265         if (!res.ok) throw new Error(text);
266
267         // -----
268         // (1) ΑΙΣΙΟΔΟΣΗ ΑΠΕΙΚΟΝΙΣΗ:
269         // Δημιουργούμε προσωρινό GeoJSON Feature (Polygon)
270         // για άμεση εμφάνιση στο Layer, ώστε να μην "εξαφανίζεται"
271         // το πολύγωνο τη στιγμή του submit.
272         // -----
273         const tmpFeature = {
274             type: "Feature" as const,
275             id: `tmp-${Date.now()}`,
276             geometry: {

```

```

277     type: "Polygon" as const,
278     coordinates: [...nums.map((n) => [n.lon, n.lat]), [nums[0].lon, nums[0].lat
279     ]],
280     properties: {
281       code: body.code,
282       region: body.region,
283       power_max: body.power_max, // kW
284       power_avg: body.power_avg, // kW
285       area_m2: null, // το πραγματικό έρχεται από τη ΒΔ μετά το reload
286       __optimistic__: true,
287     },
288   };
289   // Ενημέρωση του Layer μέσω custom event
290   window.dispatchEvent(new CustomEvent("optimistic-installation", { detail:
tmpFeature }));

291
292   // -----
293   // (2) Ζήτησε από το ανώτερο component να κάνει refresh
294   // ώστε να φέρει τα πραγματικά δεδομένα από τη ΒΔ.
295   // -----
296   onCreate?();
297
298   // -----
299   // (3) Καθάρισμα & κλείσιμο modal
300   // -----
301   setOpen(false);
302   onClose?();
303
304   // Αφαίρεση preview από τον χάρτη
305   const map = (window as any).__leafletMap as L.Map | undefined;
306   if (map && previewLayer.current) {
307     map.removeLayer(previewLayer.current);
308     previewLayer.current = null;
309   }
310
311   // Reset πεδίων
312   setCode("");
313   setPowerMax("");
314   setPowerAvg("");
315   setCsv("");
316   setVertices(3);
317   setPairs([{ lon: "", lat: "" }, { lon: "", lat: "" }, { lon: "", lat: "" }]);
318 } catch (e: any) {
319   alert("Αποτυχία καταχώρησης: " + (e?.message ?? e));
320 }
321 };
322
323 // Πού θα γίνει render το modal (portal στο <body>)
324 const portalTarget = useMemo(() => document.body, []);
325
326 // Το UI της φόρμας (με διάλογο)
327 const form = (
328   <
329     /* Σταθερό κουμπί για άνοιγμα της φόρμας */
330     <button style={fixedButton} onClick={() => setOpen(true)}>
331       NEW INSTALLATION
332     </button>
333
334     /* To modal dialog */
335     <dialog ref={dlgRef} style={{ width: 560, maxWidth: "95vw" }}>
336       /* "Κεφαλίδα" που χρησιμεύει ως λαβή μετακίνησης (drag) */
337       <div
338         ref={headerRef}
339         style={{ cursor: "move", margin: "-8px -8px 8px", padding: "8px", background:
340           "#f0f0f0" }}
341       >
342         <b>New Installation</b>
343       </div>

```

```

343
344      /* Κύρια φόρμα σε grid 2 στηλών */
345      <div style={grid2Cols}>
346          <label>Code</label>
347          <input value={code} onChange={ (e) => setCode(e.target.value)} />
348
349          <label>Vertices (≥3)</label>
350          <input
351              type="number"
352              min={3}
353              value={vertices}
354              onChange={ (e) => setVertices(Math.max(3, Number(e.target.value || 3)))}
355          />
356
357          <label>Region</label>
358          <select value={region} onChange={ (e) => setRegion(e.target.value as any)}>
359              {REGIONS.map((r) => (
360                  <option key={r} value={r}>
361                      {r}
362                  </option>
363              ))}
364          </select>
365
366          /* ΣΗΜΑΝΤΙΚΟ: οι μονάδες πλέον είναι kW (όχι kWh) */
367          <label>Power Max (kW)</label>
368          <input inputMode="decimal" value={powerMax} onChange={ (e) => setPowerMax(e.target.value)} />
369
370          <label>Power Avg (kW)</label>
371          <input inputMode="decimal" value={powerAvg} onChange={ (e) => setPowerAvg(e.target.value)} />
372      </div>
373
374      /* Οδηγία για το CSV / επικόλληση συντεταγμένων */
375      <div style={{ marginTop: 10, ...labelStyle }}>
376          Επικόλληση lon,lat ανά γραμμή (δέχεται τελείες ή κόμματα ως δεκαδικά).
377      </div>
378      <textarea
379          value={csv}
380          onChange={ (e) => setCsv(e.target.value)}
381          rows={6}
382          style={{ width: "100%", marginTop: 6 }}
383      />
384
385      /* Κουμπιά χειρισμού CSV / προεπισκόπησης */
386      <div style={btnBar}>
387          <button onClick={parseCsv}>Parse CSV → Preview on map</button>
388          <button onClick={() => setCsv("")}>Clear</button>
389          <button onClick={() => previewFrom()}>Preview from inputs</button>
390      </div>
391
392      /* Πλέγμα input boxes για τα ζεύγη (lon, lat) */
393      <div style={{ marginTop: 10, fontWeight: 600 }}>Coordinates (lon, lat)</div>
394      <div style={{ display: "grid", gridTemplateColumns: "1fr 1fr", gap: 6 }}>
395          {pairs.map((p, i) => (
396              <div key={i} style={{ display: "contents" }}>
397                  <input
398                      placeholder="Longitude"
399                      value={p.lon}
400                      onChange={ (e) => {
401                          const v = e.target.value;
402                          setPairs((prev) => {
403                              const next = [...prev];
404                              next[i] = { ...next[i], lon: v };
405                              return next;
406                          });
407                  </input>
408              </div>
409          );

```

```

410         placeholder="Latitude"
411         value={p.lat}
412         onChange={(e) => {
413             const v = e.target.value;
414             setPairs((prev) => {
415                 const next = [...prev];
416                 next[i] = { ...next[i], lat: v };
417                 return next;
418             });
419         }}
420     />
421 </div>
422     )}
423 </div>
424
425     {/* Τελικές ενέργειες */}
426     <div style={{ display: "flex", justifyContent: "flex-end", gap: 8, marginTop: 12 }}>
427         <button onClick={() => setOpen(false)}>CANCEL</button>
428         <button onClick={onSubmit}>ENTER</button>
429     </div>
430 </dialog>
431 </>
432 );
433
434 // Απόδοση του dialog στο <body> μέσω portal
435 return ReactDOM.createPortal(form, portalTarget);
436 }
437

```

## ΠΑΡΑΡΤΗΜΑ Β6. Ο κώδικας του αρχείου src/ui/RegionFilter.tsx

```
1 // src/ui/RegionFilter.tsx
2
3 import { useEffect, useState } from "react";
4
5 /**
6  * ΛΙΣΤΑ ΠΕΡΙΦΕΡΕΙΩΝ (σταθερή)
7  * -----
8  * - Δηλώνεται ως const assertion (`as const`) ώστε οι τιμές
9  *   να είναι literal types (π.χ. "Αττική" αντί για string),
10  *   κάτι που βοηθάει το TypeScript σε αυστηρότερο έλεγχο.
11  * - Το πρώτο στοιχείο "- Όλες -" σημαίνει "χωρίς φίλτρο".
12  */
13 const REGIONS = [
14   "- Όλες -",
15   "Ανατολική Μακεδονία και Θράκη",
16   "Κεντρική Μακεδονία",
17   "Δυτική Μακεδονία",
18   "Ηπειρος",
19   "Θεσσαλία",
20   "Ιόνιες Νήσοι",
21   "Δυτική Ελλάδα",
22   "Στερεά Ελλάδα",
23   "Αττική",
24   "Πελοπόννησος",
25   "Βόρειο Αιγαίο",
26   "Νότιο Αιγαίο",
27   "Κρήτη",
28 ] as const;
29
30 /**
31  * RegionFilter
32  * -----
33  * - Εμφανίζει ένα dropdown για επιλογή διοικητικής περιφέρειας.
34  * - Αποθηκεύει την επιλογή σε global χώρο (window.__selectedRegion)
35  *   για να είναι προσβάσιμη από άλλα components (π.χ. InstallationsLayer).
36  * - Με κάθε αλλαγή, εκπέμπει custom event "region-changed" στο window,
37  *   ώστε τα layers να ανανεώνουν τα δεδομένα τους.
38  */
39 export default function RegionFilter() {
40   /**
41    * ΤΟΠΙΚΗ ΚΑΤΑΣΤΑΣΗ ΕΠΙΛΟΓΗΣ
42    * - Αρχικοποιούμε από πιθανή προηγούμενη τιμή στο window.__selectedRegion,
43    *   αλλιώς προεπιλογή "- Όλες -".
44    */
45   const [value, setValue] = useState<string>(() => {
46     const v = (window as any).__selectedRegion as string | undefined;
47     return v ?? "- Όλες -";
48   });
49
50   /**
51    * SIDE EFFECT ΣΕ ΚΑΘΕ ΑΛΛΑΓΗ ΤΙΜΗΣ
52    * - Γράφουμε την επιλογή σε window.__selectedRegion.
53    * - Αν είναι "- Όλες -", θέτουμε null για να σημαίνει "κανένα φίλτρο".
54    * - Εκπέμπουμε event "region-changed" ώστε όποιος «ακούει» (π.χ. InstallationsLayer)
55    *   να κάνει refetch με το νέο φίλτρο.
56    */
57   useEffect(() => {
58     (window as any).__selectedRegion = value === "- Όλες -" ? null : value;
59     window.dispatchEvent(new Event("region-changed"));
60   }, [value]);
61
62   /**
63    * UI
64    * - Τοποθετούμε το κουτί σε fixed θέση πάνω αριστερά, αλλά
65    *   λίγο πιο δεξιά από τα Leaflet zoom controls για να μην επικαλύπτεται.
66    * - Προσαρμόζεις `left`/`top` αν χρειαστεί.
67    */
68   return (
69     <div
```

```

70     style={{
71       position: "fixed", // σταθερή σελίδα (όχι σχετική με το χάρτη), ώστε να μην
72       «πηδάει»
73       top: 10,
74       left: 60, // λίγο δεξιότερα από τα +/- του Leaflet
75       zIndex: 1000,
76       background: "white",
77       padding: 8,
78       borderRadius: 6,
79       boxShadow: "0 2px 8px rgba(0,0,0,0.15)",
80       fontSize: 14,
81     }}
82   >
83     { /* Ετικέτα/Τίτλος */ }
84     <div style={{ marginBottom: 4, fontWeight: 600 }}>Περιφέρεια</div>
85
86     { /* Dropdown επιλογής περιφέρειας */ }
87     <select
88       value={value}
89       onChange={(e) => setValue(e.target.value)}
90       style={{ minWidth: 260, padding: 6 }}
91       aria-label="Φίλτρο περιφέρειας"
92     >
93       {REGIONS.map((r) => (
94         <option key={r} value={r}>
95           {r}
96         </option>
97       ))}
98     </select>
99   </div>
100   );
101 }

```

## ΠΑΡΑΡΤΗΜΑ Β7. Ο κώδικας του αρχείου src\supabaseClient.ts

```
1
2 // src/supabaseClient.ts
3 import { createClient } from "@supabase/supabase-js";
4
5 const url = import.meta.env.VITE_SUPABASE_URL as string;
6 const anon = import.meta.env.VITE_SUPABASE_ANON_KEY as string;
7
8 export const supabase = createClient(url, anon);
9
```

