

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Πτυχιακή Εργασία

**Παρουσίαση μιας βάσης δεδομένων με περιγραφές τόπων σε νοητό χάρτη
τοπίου**

Ξενοφών Κουρτζής

Π2015119

Επιβλέπουσα: Άννα Σωτηροπούλου

22 ΙΟΥΝΙΟΥ 2022

Επιβλέπουσα

Άννα Σωτηροπούλου, ΕΔΙΠ,
Ιόνιο Πανεπιστήμιο

Τριμελής Επιτροπή

Κάτια – Λήδα Κερμανίδου, Αναπληρώτρια καθηγήτρια
Ιόνιο Πανεπιστήμιο

Φοίβος Μυλωνάς, Αναπληρωτής Καθηγητής
Ιόνιο Πανεπιστήμιο

Η συγγραφή αυτής της πτυχιακής αφιερώνεται στην οικογένεια μου η
οποία με στήριξε καθ' όλη την διάρκεια της ακαδημαϊκής μου
πορείας.

Περίληψη

Η παρούσα πτυχιακή εργασία εστιάζει στην αξιοποίηση της ηλεκτρονικής πληροφορίας και των δεδομένων της σε μορφές χρήσιμες για τον άνθρωπο και την κοινωνία. Σε συνδυασμό με την χρήση ηλεκτρονικών βάσεων δεδομένων μπορούμε πλέον να χρησιμοποιήσουμε οποιοδήποτε είδος πληροφορίας για την εξόρυξη συμπερασμάτων, δεικτών, οπτικών μέσων και λοιπών στοιχείων προς όφελος μας με μεγαλύτερη ευκολία, ερευνητική ακρίβεια και λεπτομέρεια. Σκοπός της εργασίας αυτής είναι μέσω της χρήσης μίας βάσεως αρχαιολογικών δεδομένων του αρχαίου Λαυρίου, να απεικονιστεί σε χάρτες όλη η χρήσιμη πληροφορία και να γίνει ευκολότερη κατανόηση της. Με αυτό τον τρόπο ο μέσος αναγνώστης των χαρτών θα μπορεί με μία ματιά να ξεναγηθεί σε περιοχές της αρχαιότητας, να αποκτήσει μία σφαιρική αντίληψη των τότε Δήμων της Αττικής αλλά και να ανακαλύψει την θέση μεταλλείων και άλλων χώρων της εποχής. Η παρούσα εργασία στοχεύει αρχικά, στην χρήση της καταγραφής και της μαζικής οργάνωσης των αρχαιολογικών δεδομένων. Δεύτερον προσεγγίζει την χερσόνησο του Αρχαίου Λαυρίου, της σημερινής Αττικής, μέσω νοητικών χαρτών σύμφωνα με επιγραφές που χρονολογούνται στον 4ο αιώνα π.Χ οι οποίες καταγράφουν τις μισθώσεις μεταλλείων από την Αθηναϊκή πολιτεία σε πολίτες της. Το σύστημα αυτό μπορεί να αποτελέσει ένα σημαντικό εργαλείο για τους αρχαιολόγους και να βελτιώσει την ερευνητική διαδικασία στο πλαίσιο της συνεχούς εξέλιξης ολόκληρου του κλάδου. Για την υλοποίηση της εργασίας χρησιμοποιήθηκε μία βάση δεδομένων σχετική με το αρχαίο Λαύριο, καθώς επίσης βιβλιοθήκες και εργαλεία της γλώσσας προγραμματισμού Python για την δημιουργία των χαρτών. Τέλος, μέσω των νοητικών χαρτών που αναπτύχθηκαν, δίνεται ξανά ζωή στους αρχαιολόγους χώρους και δήμους της εποχής.

Λέξεις κλειδιά: Αρχαιολογία, Ανασκαφή, Έρευνα, Διεπαφή Χρήστη, Βάσεις Δεδομένων, Επιγραφές, Νοητικοί Χάρτες, Αρχαίο Λαύριο, Αρχαίος Δήμος, Python, Μεταλλεία

Περιεχόμενα

Περίληψη	4
Δομή Εργασίας	8
Κεφάλαιο 1: Εισαγωγή	9
Κεφάλαιο 2: Εργαλεία και Μεθοδολογία	13
2.1 Μεθοδολογία	13
2.2 Wamp	14
2.3 phpMyAdmin	14
2.4 Anaconda	14
2.5 Python και Βιβλιοθήκες Python	15
2.6 Βάση δεδομένων	20
2.7 Σύνδεση Python με MySQL Server	21
Κεφάλαιο 3: Οπτικοποίηση δεδομένων και απεικόνιση σε χάρτες	24
3.1 Εξαγωγή και μορφοποίηση δεδομένων	24
3.2 Ομαδοποίηση δεδομένων και μετασχηματισμός για χαρτογράφηση	26
3.3 Δημιουργία γεωγραφικού χάρτη με τη χρήση του Folium	26
Κεφάλαιο 4 Απεικόνιση βάσης δεδομένων	28
4.1 Αποτελέσματα κώδικα	28
4.1.2 Απεικόνιση Χαρτών	37
Κεφάλαιο 5 Επεκτάσεις και συμπεράσματα	45
Παράτημα Α'	47
Παράτημα Β'	51
Παράτημα Γ'	56
Βιβλιογραφία	73

Πίνακας Εικόνων

ΕΙΚΟΝΑ 1: ΛΗΨΗ WAMP SERVER (1)	56
ΕΙΚΟΝΑ 2: ΛΗΨΗ WAMP SERVER (2)	56
ΕΙΚΟΝΑ 3:ΕΠΙΛΟΓΗ ΓΛΩΣΣΑΣ WAMP SERVER.....	57
ΕΙΚΟΝΑ 4:ΕΓΚΑΤΑΣΤΑΣΗ WAMP SERVER (1).....	57
ΕΙΚΟΝΑ 5:ΕΓΚΑΤΑΣΤΑΣΗ WAMP SERVER (2).....	58
ΕΙΚΟΝΑ 6:ΕΓΚΑΤΑΣΤΑΣΗ WAMP SERVER (3).....	59
ΕΙΚΟΝΑ 7:ΕΓΚΑΤΑΣΤΑΣΗ WAMP SERVER (4).....	60
ΕΙΚΟΝΑ 8:ΕΓΚΑΤΑΣΤΑΣΗ WAMP SERVER (5).....	61
ΕΙΚΟΝΑ 9:ΕΓΚΑΤΑΣΤΑΣΗ WAMP SERVER (6).....	62
ΕΙΚΟΝΑ 10:ΕΓΚΑΤΑΣΤΑΣΗ WAMP SERVER (7).....	63
ΕΙΚΟΝΑ 11:ΕΓΚΑΤΑΣΤΑΣΗ WAMP SERVER (8).....	64
ΕΙΚΟΝΑ 12:ΟΛΟΚΛΗΡΩΣΗ ΕΓΚΑΤΑΣΤΑΣΗΣ WAMP SERVER.....	65
ΕΙΚΟΝΑ 13:ΣΥΝΔΕΣΗ ΣΤΗΝ PHPMYADMIN	66
ΕΙΚΟΝΑ 14:ΔΗΜΙΟΥΡΓΙΑ ΚΕΝΗΣ ΒΑΣΗΣ (1)	67
ΕΙΚΟΝΑ 15:ΔΗΜΙΟΥΡΓΙΑ ΚΕΝΗΣ ΒΑΣΗΣ (2)	67
ΕΙΚΟΝΑ 16:ΔΗΜΙΟΥΡΓΙΑ ΚΕΝΗΣ ΒΑΣΗΣ (3)	68
ΕΙΚΟΝΑ 17:IMPORT ΤΗΣ ΒΑΣΗΣ ΣΤΗΝ ΚΕΝΗ ΒΑΣΗ	68
ΕΙΚΟΝΑ 18:ΟΛΟΚΛΗΡΩΣΗ ΔΗΜΙΟΥΡΓΙΑΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	69
ΕΙΚΟΝΑ 19:ΛΗΨΗ ANACODA DISTRIBUTION.....	69
ΕΙΚΟΝΑ 20:ΕΓΚΑΤΑΣΤΑΣΗ ANACODA (1)	70
ΕΙΚΟΝΑ 21:ΕΓΚΑΤΑΣΤΑΣΗ ANACODA (2)	70
ΕΙΚΟΝΑ 22:ΕΓΚΑΤΑΣΤΑΣΗ ANACODA (3)	71
ΕΙΚΟΝΑ 23:ΕΓΚΑΤΑΣΤΑΣΗ ANACODA (4)	71
ΕΙΚΟΝΑ 24:ΕΚΚΙΝΗΣΗ JUPITER NOTEBOOK	72
ΕΙΚΟΝΑ 25:ΕΚΚΙΝΗΣΗ JUPYTER NOTEBOOK (2).....	72
ΕΙΚΟΝΑ 26:ΕΓΚΑΤΑΣΤΑΣΗ ΒΙΒΛΙΟΘΗΚΩΝ ΚΑΙ MODULES	28
ΕΙΚΟΝΑ 27:DATAFRAME DATA	29
ΕΙΚΟΝΑ 28:DATAFRAME DICTIONARY	30
ΕΙΚΟΝΑ 29:ΕΝΩΣΗ DATAFRAME(DATA) ΜΕ DATAFRAME(DICTIONARY)	30
ΕΙΚΟΝΑ 30:ΠΕΡΙΕΧΟΜΕΝΟ ΑΠΟ ΤΟ DATAFRAME DICTIONARY A(1).....	31
ΕΙΚΟΝΑ 31:ΠΕΡΙΕΧΟΜΕΝΟ ΑΠΟ ΤΟ DATAFRAME DICTIONARY A(2).....	32
ΕΙΚΟΝΑ 32:ΠΕΡΙΕΧΟΜΕΝΟ ΑΠΟ ΤΟ DATAFRAME ONITI(1)	33
ΕΙΚΟΝΑ 33:ΠΕΡΙΕΧΟΜΕΝΟ ΑΠΟ ΤΟ DATAFRAME ONITI(2)	34
ΕΙΚΟΝΑ 34:ΠΕΡΙΕΧΟΜΕΝΟ ΑΠΟ ΤΟ DATAFRAME DICTIONARY B	35
ΕΙΚΟΝΑ 35:ΠΕΡΙΕΧΟΜΕΝΟ ΑΠΟ ΤΟ DATAFRAME EDAPH	36
ΕΙΚΟΝΑ 36:ΑΠΟΤΕΛΕΣΜΑΤΑ MAP1(ΟΝΟΜΑ ΔΗΜΟΥ) (1).....	37
ΕΙΚΟΝΑ 37:ΑΠΟΤΕΛΕΣΜΑΤΑ MAP1(ΟΝΟΜΑ ΔΗΜΟΥ) (2).....	37
ΕΙΚΟΝΑ 38:ΑΠΟΤΕΛΕΣΜΑΤΑ MAP1(ΟΝΟΜΑ ΔΗΜΟΥ) (3).....	38
ΕΙΚΟΝΑ 39:ΑΠΟΤΕΛΕΣΜΑΤΑ MAP1(ΟΝΟΜΑ ΔΗΜΟΥ) (4).....	38
ΕΙΚΟΝΑ 40:ΑΠΟΤΕΛΕΣΜΑΤΑ MAP2(ΟΝΟΜΑ ΩΝΗΤΗ) (1).....	39
ΕΙΚΟΝΑ 41:ΑΠΟΤΕΛΕΣΜΑΤΑ MAP2(ΟΝΟΜΑ ΩΝΗΤΗ) (2).....	39
ΕΙΚΟΝΑ 42:ΑΠΟΤΕΛΕΣΜΑΤΑ MAP2(ΟΝΟΜΑ ΩΝΗΤΗ) (3).....	40
ΕΙΚΟΝΑ 43:ΑΠΟΤΕΛΕΣΜΑΤΑ MAP2(ΟΝΟΜΑ ΩΝΗΤΗ) (4).....	40
ΕΙΚΟΝΑ 44:ΑΠΟΤΕΛΕΣΜΑΤΑ MAP3(ΟΝΟΜΑ ΚΑΙ ΕΙΔΟΣ ΜΕΤΑΛΛΕΙΟΥ) (1).....	41
ΕΙΚΟΝΑ 45:ΑΠΟΤΕΛΕΣΜΑΤΑ MAP3(ΟΝΟΜΑ ΚΑΙ ΕΙΔΟΣ ΜΕΤΑΛΛΕΙΟΥ) (2).....	41
ΕΙΚΟΝΑ 46:ΑΠΟΤΕΛΕΣΜΑΤΑ MAP3(ΟΝΟΜΑ ΚΑΙ ΕΙΔΟΣ ΜΕΤΑΛΛΕΙΟΥ) (3).....	42

ΕΙΚΟΝΑ 47:ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΑΡ3(ΟΝΟΜΑ ΚΑΙ ΈΙΔΟΣ ΜΕΤΑΛΛΕΙΟΥ) (4).....	42
ΕΙΚΟΝΑ 48:ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΑΡ3(ΟΝΟΜΑ ΕΔΑΦΟΥΣ ΚΑΙ ΓΑΙΟΚΤΗΜΟΝΑ) (1)	43
ΕΙΚΟΝΑ 49:ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΑΡ3(ΟΝΟΜΑ ΕΔΑΦΟΥΣ ΚΑΙ ΓΑΙΟΚΤΗΜΟΝΑ) (2)	43
ΕΙΚΟΝΑ 50:ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΑΡ3(ΟΝΟΜΑ ΕΔΑΦΟΥΣ ΚΑΙ ΓΑΙΟΚΤΗΜΟΝΑ) (3)	43
ΕΙΚΟΝΑ 51:ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΑΡ3(ΟΝΟΜΑ ΕΔΑΦΟΥΣ ΚΑΙ ΓΑΙΟΚΤΗΜΟΝΑ) (4)	44

Δομή Εργασίας

Παρακάτω αναφέρεται περιληπτικά το περιεχόμενο του κάθε κεφαλαίου:

1. Το πρώτο κεφάλαιο αποτελεί εισαγωγή στο περιεχόμενο της εργασίας και περιλαμβάνει κάποια εισαγωγικά στοιχεία, απαραίτητα για την κατανόηση του περιεχομένου της.
2. Το δεύτερο κεφάλαιο αναφέρει αναλυτικά τη μεθοδολογία και τα εργαλεία που χρησιμοποιήθηκαν καθώς και τα βήματα για την εγκατάσταση τους.
3. Το τρίτο κεφάλαιο περιγράφει τους λόγους επιλογής της γλώσσας προγραμματισμού Python, τις βιβλιοθήκες που χρησιμοποιήθηκαν καθώς και τους τρόπους σύνδεσής της με τον εξυπηρετητή.
4. Το τέταρτο κεφάλαιο καταγράφει τον κώδικα που αναπτύχθηκε για τη δημιουργία των χαρτών, τα βήματα και την επεξήγησή τους, καθώς επίσης γίνεται παρουσίαση των χαρτών.
5. Το πέμπτο κεφάλαιο παρουσιάζει τα συμπεράσματα της εργασίας.

Κεφάλαιο 1: Εισαγωγή

Η πολιτιστική κληρονομιά της Χώρας μας από αρχαιοτάτων χρόνων ήταν και είναι αδιαμφισβήτητο διαχρονικό αγαθό. Αναγκαίο είναι να δίνεται έμφαση και προαγωγή στις αρχαιογνωστικές επιστήμες, καθώς οδηγεί πρωτίστως στην βελτίωση του ηθικού εξοπλισμού του ανθρώπου, στη συνειδητοποιημένη μορφή του ‘άριστου μέτρου’, αλλά και την ορθολογική προσέγγιση της ισορροπημένης τεχνολογικής ανάπτυξης σε αρμονία με το περιβάλλον, και την αυτογνωσιακή πρόοδο. Χρήσιμη εφαρμογή που αφορά τη συλλογή, επεξεργασία και ανάλυση πληροφοριών σχετικά με διάφορα αρχαία αντικείμενα και αρχαιολογικούς χώρους, είναι το ευρύ πεδίο των αρχαιογεωφυσικών διασκοπήσεων στο οποίο χαρτογραφείται το υπέδαφος και εντοπίζονται αρχαιολογικά ευρήματα. Στη συνέχεια, τα ευρήματα αυτά οπτικοποιούνται για την ανάδειξη του αρχαιολογικού γεωγραφικού χώρου με τη βοήθεια των συστημάτων γεωγραφικών πληροφοριών (GIS). Ως εκ τούτου ύψιστης σημασίας είναι η διατήρηση της πληροφορίας στον χρόνο, ώστε κάθε φορά η γνώση να οικοδομείται πάνω σε υπάρχοντα ερευνητικά δεδομένα (Ιωάννης Λυριτζή, 2022). Στην παρούσα πτυχιακή εργασία γίνεται χρήση και σημασιολογική απεικόνιση πληροφοριών από επιγραφές πολιτών που ήταν αναρτημένες στην αρχαία Αθηναϊκή Αγορά σε μορφή χρήσιμη για τους αρχαιολόγους. Οι επιγραφές αφορούν πληροφορίες σχετικές με ιδιοκτησίες μεταλλείων στη χερσόνησο στην οποία βρισκόταν το Αρχαίο Λαύριο. Δημιουργούνται χάρτες στους οποίους απεικονίζονται πληροφορίες της περιόδου του 4ου αιώνα π.Χ. Τοποθετούνται σε χάρτες της σημερινής περιόδου όλα τα αρχαιολογικά δεδομένα που βρίσκονται αποθηκευμένα σε μία βάση δεδομένων, και με αυτόν τον τρόπο καταδεικνύεται περίτρανα η καθομολογούμενη και πασιφανής αξία τους.

Πιο συγκεκριμένα η εργασία αυτή εστιάζει στη δημιουργία νοητικών χαρτών, με χρήση βάσης αρχαιολογικών δεδομένων, στοχεύοντας στην οπτικοποίηση της πληροφορίας και σε μία ψηφιακή ξενάγηση των αρχαιολογικών πληροφοριών. Ο εκάστοτε παρατηρητής του κάθε διαφορετικού χάρτη, έχει τη δυνατότητα μέσα σε λίγα δευτερόλεπτα να αποκτήσει σφαιρική αντίληψη του χώρου, γνωρίζοντας

ταυτόχρονα λεπτομέρειες για τις περιοχές που μελετά. Επίσης, οι χάρτες μπορούν να χρησιμοποιηθούν από αρχαιολόγους οι οποίοι θέλουν να τακτοποιήσουν τη συλλεγμένη πληροφορία με τρόπο που θα τους επιτρέπει να έχουν συνεχώς πρόσβαση σε αυτή και να βγάζουν γρηγορότερα συμπεράσματα. Μέσω της βάσης δεδομένων του συστήματος, ο χρήστης μπορεί να έχει πρόσβαση στην αποθηκευμένη πληροφορία. Όταν όμως αυτή η πληροφορία εισάγεται σε ένα χάρτη, τότε αυτόματα ζωντανεύει και αναγεννιέται κάθε μικρή αποθηκευμένη λεπτομέρεια. Έτσι ο χρήστης της παρούσας εργασίας, μπορεί να βγάλει συμπεράσματα σχετικά με την κοινωνία του αρχαίου Λαυρίου και τον γεωγραφικό χώρο που αυτό εκτεινόταν. Βλέπουμε λοιπόν πώς μπορούν να συνεργαστούν η επιστήμη της αρχαιολογίας με αυτή της επιστήμης της πληροφορικής. Το σύνολο της αποθηκευμένης πληροφορίας σε συνδυασμό με την κατάλληλη απεικόνισή της μπορεί να συντελέσει στην αποτελεσματικότερη έρευνα και το έργο των αρχαιολόγων. Αντικειμενικός σκοπός της εργασίας αποτέλεσε, εξ αρχής, η χρήση τεχνολογίας για την ανάπτυξη ρεαλιστικών δισδιάστατων αναπαραστάσεων, οι οποίες παρέχουν στο χρήστη αλληλεπιδραστικές δυνατότητες.

Οι νοητικοί χάρτες, γνωστοί και ως γνωστικοί χάρτες, είναι χρήσιμοι για τους ιστορικούς επειδή βοηθούν στην οπτικοποίηση και κατανόηση των χωρικών σχέσεων και του πολιτισμικού πλαισίου των γεγονότων του παρελθόντος (Edsall et al., n.d.). Οι νοητικοί χάρτες είναι οι νοητικές αναπαραστάσεις του γεωγραφικού χώρου που έχουν οι άνθρωποι στο μυαλό τους και μπορούν να επηρεάσουν την αντίληψη και την κατανόησή τους για τον κόσμο γύρω τους.

Δημιουργώντας νοητικούς χάρτες ιστορικών γεγονότων, οι ιστορικοί μπορούν να αποκτήσουν γνώσεις για τις χωρικές διαστάσεις του παρελθόντος, συμπεριλαμβανομένων των τοποθεσιών βασικών γεγονότων. Αυτοί οι χάρτες μπορούν επίσης να βοηθήσουν στον εντοπισμό προτύπων και τάσεων που μπορεί να μην είναι άμεσα εμφανείς από γραπτά αρχεία ή άλλους τύπους αποδεικτικών στοιχείων.

Επιπλέον, οι νοητικοί χάρτες μπορούν να βοηθήσουν στην παροχή μιας πιο λεπτομερούς κατανόησης των ιστορικών γεγονότων, λαμβάνοντας υπόψη τους πολιτιστικούς και ψυχολογικούς παράγοντες, που διαμόρφωσαν τις αντιλήψεις και τις συμπεριφορές των ανθρώπων στο παρελθόν. Αυτό μπορεί να είναι ιδιαίτερα σημαντικό για την κατανόηση του πώς διαφορετικές ομάδες ανθρώπων βίωσαν ιστορικά γεγονότα και πώς αυτές οι εμπειρίες επηρεάστηκαν από πολιτιστικούς, κοινωνικούς και πολιτικούς παράγοντες.

Συνολικά, οι συγκεκριμένοι νοητικοί χάρτες είναι ένα χρήσιμο εργαλείο για τους ιστορικούς επειδή συμβάλλουν στην δημιουργία μιας χωρικής και πολιτισμικής προοπτικής στη μελέτη του παρελθόντος. Επιπλέον παρέχουν νέες γνώσεις για τις πολύπλοκες σχέσεις και τα πρότυπα που διαμόρφωσαν τα ιστορικά γεγονότα. Ένας νοητικός χάρτης θα μπορούσε να δείχνει τις διαδρομές που ακολουθούν οι έμποροι, οι στρατοί ή οι εξερευνητές, κάτι που μπορεί να βοηθήσει στον εντοπισμό της εξάπλωσης αγαθών, ιδεών και τεχνολογιών σε διαφορετικές περιοχές και πολιτισμούς.

Εκτός από την παροχή μιας χωρικής προοπτικής, οι νοητικοί χάρτες μπορούν επίσης να βοηθήσουν ρίχνοντας «φως» στο πολιτιστικό πλαίσιο των ιστορικών γεγονότων. Για παράδειγμα, μπορούν να αποκαλύψουν πώς οι διαφορετικοί πολιτισμοί έβλεπαν τον κόσμο και τη θέση τους μέσα σε αυτόν, κάτι που μπορεί να προσφέρει πολύτιμες γνώσεις για τις πεποιθήσεις, τις αξίες και τις στάσεις των ανθρώπων στο παρελθόν. Αυτό είναι ιδιαίτερα σημαντικό για την κατανόηση των κινήτρων και των συμπεριφορών των ανθρώπων σε διαφορετικούς πολιτισμούς και περιοχές και μπορεί να συμβάλλει στην παροχή μιας πιο ιδιαίτερης και πολιτισμικά ευαίσθητης άποψης της ιστορίας.

Οι νοητικοί χάρτες είναι χρήσιμοι για τον εντοπισμό προτύπων και τάσεων που μπορεί να μην είναι άμεσα εμφανείς από άλλα είδη στοιχείων, όπως γραπτά αρχεία ή αρχαιολογικά ευρήματα. Οπτικοποιώντας τις χωρικές σχέσεις μεταξύ διαφορετικών γεγονότων και τόπων, οι ιστορικοί μπορούν πιο εύκολα να

εντοπίσουν συσχετισμούς, σχέση αιτίας-αποτελέσματος και άλλα μοτίβα που μπορούν να παρέχουν νέες γνώσεις.

Τέλος, οι νοητικοί χάρτες μπορούν επίσης να χρησιμοποιηθούν για τη διερεύνηση των ψυχολογικών και συναισθηματικών διαστάσεων των ιστορικών γεγονότων. Για παράδειγμα, μπορούν να αποκαλύψουν τους τρόπους με τους οποίους βίωσαν και αντιλήφθηκαν οι άνθρωποι στο παρελθόν τα διάφορα μέρη και τα γεγονότα και αυτό φυσικά θα μπορούσε να προσφέρει πολύτιμες γνώσεις για τις στάσεις, τους φόβους και τις φιλοδοξίες των ανθρώπων σε διαφορετικούς πολιτισμούς και χρονικές περιόδους.

Κεφάλαιο 2: Εργαλεία και Μεθοδολογία

2.1 Μεθοδολογία

Για την εκπόνηση της εργασίας χρησιμοποιήθηκαν διάφορα εργαλεία ανοιχτού κώδικα. Αρχικά έπρεπε να γίνει σύνδεση με τη σχετική βάση δεδομένων. Γι' αυτόν τον σκοπό χρησιμοποιήθηκε το εργαλείο phpMyAdmin το οποίο υποστηρίζει ένα ευρύ φάσμα λειτουργιών για διαχείριση βάσεων δεδομένων, πινάκων, στηλών και δίνει τη δυνατότητα επεξεργασίας και πρόσβασης σε πολλούς χρήστες. Μέσω αυτού του εργαλείου έγινε η σύνδεση με το σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων MySQL. Για να μην χρειαστεί να γίνουν ξεχωριστές εγκαταστάσεις των εργαλείων που αναφέρθηκαν, χρησιμοποιήθηκε το WAMP, ένα εργαλείο που λειτουργεί ως εικονικός διακομιστής και μας επιτρέπει να δοκιμάσουμε τις δυνατότητες όλων των εργαλείων που μας προσφέρει χωρίς καθυστερήσεις.

Σχετικά με την απεικόνιση των δεδομένων σε χάρτες χρησιμοποιήθηκε η Anacoda Distribution, η οποία είναι η πιο δημοφιλής πλατφόρμα διανομής Python ανοιχτού κώδικα. Μέσω αυτής έγινε χρήση του Jupyter Notebook, το οποίο είναι μία διαδικτυακή εφαρμογή για τη δημιουργία και την κοινή χρήση υπολογιστικών εγγράφων.

Η οπτικοποίηση των γεωχωρικών δεδομένων είναι ένα από τα πιο ενδιαφέροντα πράγματα που μπορεί να κάνει κανείς με τα δεδομένα του, ειδικά εάν τα δεδομένα αυτά περιέχουν στήλες που μπορούν να αντιστοιχιστούν απευθείας σε τοποθεσίες στον χάρτη, όπως στην περίπτωση μας. Για να γίνει αυτή η οπτικοποίηση χρησιμοποιήθηκε η πλέον κατάλληλη γλώσσα προγραμματισμού για τέτοια προβλήματα, η Python. Η Python διαθέτει αρκετές βιβλιοθήκες για οπτικοποίηση δεδομένων, όπως οι Matplotlib, οι οποίες διευκολύνουν τους επιστήμονες δεδομένων και τους προγραμματιστές να δημιουργούν διαδραστικά και οπτικά ελκυστικά γραφικά. Επιπλέον, η Python έχει μια μεγάλη κοινότητα που συμβάλλει στην ανάπτυξη και την υποστήριξή της, παρέχοντας πληθώρα πόρων και εργαλείων για ανάλυση και οπτικοποίηση δεδομένων. Τέλος

ενσωματώνεται καλά με άλλες δημοφιλείς βιβλιοθήκες ανάλυσης δεδομένων και μηχανικής μάθησης, καθιστώντας την ένα ισχυρό εργαλείο για τη δημιουργία οπτικοποιήσεων που συνδυάζουν διαφορετικές πηγές δεδομένων και μοντέλα

2.2 Wamp

Για την σύνδεση με τη βάση δεδομένων χρησιμοποιήθηκαν κάποια εργαλεία. Αρχικά έγινε εγκατάσταση του WAMP Server ώστε να υπάρχει πρόσβαση στο phpMyAdmin και στην MySQL. Η εγκατάσταση είναι δωρεάν και υπάρχει αναλυτικές οδηγίες στην ιστοσελίδα του WAMP Server όπως και στο παράρτημα Γ στο πρώτο μέρος (Bourdon, 2022).

2.3 phpMyAdmin

Για να καταφέρουμε να συνδεθούμε στην βάση δεδομένων που δημιουργήθηκε πρέπει να ακολουθήσουμε τα βήματα από το παράρτημα Γ και στην συνέχεια να επιλέξουμε την βάση που θέλουμε. Προϋπόθεση για την εμφάνιση της βάσης στις επιλογές είναι να βρισκόμαστε σε κοινό δίκτυο με τον δημιουργό της, τον Χαράλαμπο Κωνσταντή. Στην συγκεκριμένη περίπτωση αυτό δεν ήταν εφικτό. Έτσι, η δημιουργία της βάσης έγινε από την αρχή μέσω της χρήσης του κώδικα που δημιουργήθηκε κατά την εκπόνηση της πτυχιακής εργασίας του κύριου Κωνσταντή. Στο παράρτημα Γ υπάρχει η αναλυτική εγκατάσταση με εικόνες.

2.4 Anacoda

Για την απεικόνιση των δεδομένων σε χάρτες χρησιμοποιήθηκε η Anaconda Distribution και κατ' επέκταση έγινε χρήση του Jupyter Notebook (Bechhofer et al., 2013). Το Anaconda είναι ένα one-stop-shop για πακέτα Python και R, που λύνει πολλά προβλήματα που σχετίζονται με τη χρήση και τη διαχείριση λογισμικού ανοιχτού κώδικα. Τα περισσότερα έργα στο οικοσύστημα λογισμικού ανοιχτού κώδικα Python (OSS) εξαρτώνται από άλλα έργα για να λειτουργήσουν και αυτό σχηματίζει μια αλυσίδα προαπαιτούμενων απαιτήσεων που αλλιώς ονομάζονται «εξαρτήσεις». Τα έργα ανοιχτού κώδικα ελέγχονται ανεξάρτητα από συντηρητές σε όλο τον κόσμο που εκδίδουν ενημερώσεις ασύγχρονα. Επομένως, ένα εργαλείο που διαχειρίζεται πακέτα, αναγνωρίζει την

αρχιτεκτονική του υλικού και διατηρεί τα έργα ξεχωριστά είναι απαραίτητο, προκειμένου να διασφαλιστεί η σωστή λειτουργικότητα. Το Jupyter Notebook είναι μια διαδικτυακή εφαρμογή ανοικτού κώδικα που επιτρέπει την δημιουργία και την διανομή εγγράφων που περιέχουν ζωντανό κώδικα(ο ζωντανός κώδικας είναι η ικανότητα να τρέχει και να εκτελείτε ο κώδικας στο απευθείας στο browser χωρίς να χρειάζεται κάπου η αποθήκευση του για να τρέξει ξεχωριστά), εξισώσεις, απεικονίσεις και αφηγηματικό κείμενο. Οι χρήσεις του περιλαμβάνουν καθαρισμό και μετασχηματισμό δεδομένων, αριθμητική προσομοίωση, στατιστική μοντελοποίηση, οπτικοποίηση δεδομένων, μηχανική μάθηση κτλ. Το Jupyter Notebook (πρώην IPython Notebooks) είναι ένα διαδικτυακό διαδραστικό υπολογιστικό περιβάλλον για τη δημιουργία εγγράφων (Rolon-Mérette et al., 2020).

2.5 Python και Βιβλιοθήκες Python

Python (Python Software Foundation, 2011) είναι γλώσσα προγραμματισμού υψηλού επιπέδου, ανοικτού πηγαίου κώδικα (open source) και γενικής χρήσης. Είναι εύκολη στην εκμάθηση (απλούστερη από τις C, C++, Java) και παρέχει ισχυρές δυνατότητες σε αρχάριους αλλά και σε έμπειρους προγραμματιστές. Ένα από τα κύρια χαρακτηριστικά της είναι η αντικειμενοστρέφεια. Αναπτύχθηκε από τον Guido van Rossum, στις αρχές της δεκαετίας '90 ως διάδοχος της γλώσσας ABC και το όνομά της προέρχεται από την ομάδα κωμικών Monty Python. Η βασική ιστοσελίδα της κοινότητας γύρω από τη γλώσσα βρίσκεται στην ηλεκτρονική διεύθυνση <http://python.org> (Python Software Foundation). Η Python χρησιμοποιείται, διεθνώς, ως εκπαιδευτική γλώσσα αλλά και στην ανάπτυξη σημαντικών εφαρμογών με την ίδια επιτυχία. Εκδόσεις της είναι διαθέσιμες για μια ευρεία γκάμα από λειτουργικά συστήματα, συμπεριλαμβανομένων των Windows, Unix/Linux, Mac OS X κλπ. Είναι διαθέσιμη μέσω ενσωματωμένων βιβλιοθηκών, δηλαδή, μπορεί να υλοποιηθεί σημαντική λειτουργικότητα χωρίς να χρειαστεί να αναζητηθούν επεκτάσεις, όπως μεταφορικά λέγεται στα αγγλικά, 'batteries are included' (Μηχανικών et al., n.d.).

Οι Βιβλιοθήκες python που χρησιμοποιήθηκαν στην εργασία είναι οι NumPy, Pandas, Matplotlib, folium και geopy.

Η πρώτη και ίσως πιο γνωστή βιβλιοθήκη που χρησιμοποιήθηκε είναι η “pandas”. Μεταξύ άλλων η συγκεκριμένη βιβλιοθήκη παρέχει την δημιουργία των DataFrames, χρήσιμα για χειρισμό δεδομένων με ενσωματωμένο ευρετήριο . Ένα DataFrame στη βιβλιοθήκη του Pandas είναι ένας δισδιάστατος , με δυνατότητα αλλαγής μεγέθους και ετερογενής δομή πίνακας δεδομένων με επισημασμένους άξονες (γραμμές και στήλες). Έχει σχεδιαστεί για να χειρίζεται δεδομένα με τρόπο παρόμοιο με ένα υπολογιστικό φύλλο, αλλά με πιο προηγμένες δυνατότητες .

Τα DataFrames είναι η πιο συχνά χρησιμοποιούμενη δομή δεδομένων στα Panda και χρησιμοποιούνται για την αποθήκευση και τον χειρισμό δεδομένων από διάφορες πηγές, συμπεριλαμβανομένων αρχείων CSV, υπολογιστικών φύλλων Excel και βάσεων δεδομένων. Επιτρέπουν να εκτελούνται λειτουργίες στα δεδομένα, όπως φιλτράρισμα, ομαδοποίηση, συγκέντρωση και μετατροπή, καθώς και να πραγματοποιεί διάφορες στατιστικές αναλύσεις. Τα DataFrames υποστηρίζουν επίσης τιμές που λείπουν, διαφορετικούς τύπους δεδομένων στην ίδια στήλη και λειτουργίες συγχώνευσης και ένωσης.

Επίσης παρέχει εργαλεία για την ανάγνωση και την εγγραφή δεδομένων διαφορετικών μορφών όπως αρχεία CSV και κειμένου, βάσεις δεδομένων SQL και άλλα. Η Python με Pandas χρησιμοποιείται σε μεγάλη ποικιλία ακαδημαϊκών και εμπορικών τομέων, όπως στις Νευροεπιστήμες, στην Στατιστική, τη Διαφήμιση, την Ανάλυση Ιστού (The Python Library Reference Release 3.11.1 Guido van Rossum and the Python Development Team, 2023), (Capital, 2022).

Η NumPy είναι μια δημοφιλής βιβλιοθήκη Python για αριθμητικούς υπολογισμούς, συμπεριλαμβανομένης όπως υποστήριξης για ένα ισχυρό αντικείμενο πίνακα N-διάστατων. Ο πρωταρχικός στόχος του NumPy είναι να παρέχει γρήγορους και αποτελεσματικούς υπολογισμούς πίνακα για αριθμητικές πράξεις, συμπεριλαμβανομένης όπως γραμμικής άλγεβρας, όπως δημιουργίας τυχαίων αριθμών και των μετασχηματισμών Fourier (Oliphant, 2006).

Μερικά από τα βασικά χαρακτηριστικά του NumPy περιλαμβάνουν:

1. Πίνακες N-διάστατων: Το NumPy παρέχει ένα γρήγορο και αποτελεσματικό αντικείμενο πίνακα N-διαστάσεων, το οποίο χρησιμοποιείται ως η βασική δομή δεδομένων για όπως όπως βιβλιοθήκες και εργαλεία στο επιστημονικό οικοσύστημα Python.
2. Μαθηματικές συναρτήσεις: Το NumPy περιλαμβάνει μεγάλο αριθμό μαθηματικών συναρτήσεων για την εκτέλεση πράξεων σε πίνακες, συμπεριλαμβανομένης όπως γραμμικής άλγεβρας, των στατιστικών συναρτήσεων και των τριγωνομετρικών συναρτήσεων.
3. Μετάδοση: Το NumPy παρέχει υποστήριξη για μετάδοση συστοιχιών διαφορετικών σχημάτων, που επιτρέπει να εκτελούνται λειτουργίες βάσει στοιχείων σε συστοιχίες διαφορετικών μεγεθών.
4. Ενσωμάτωση με όπως βιβλιοθήκες: Το NumPy χρησιμοποιείται ευρέως στην επιστημονική κοινότητα Python και είναι ενσωματωμένο με όπως όπως βιβλιοθήκες, συμπεριλαμβανομένων των Pandas, Matplotlib και SciPy, μεταξύ άλλων (Johansson, 2018).

Συνολικά, η NumPy είναι μια ισχυρή βιβλιοθήκη για αριθμητικούς υπολογισμούς στην Python και παρέχει μεγάλο εύρος επιστημονικής, μηχανικής και ανάλυσης δεδομένων. Είτε χρησιμοποιείται σε μεγάλους πίνακες, εκτελεί πολύπλοκες μαθηματικές πράξεις είτε απαιτείται για έναν γρήγορο και αποτελεσματικό τρόπο για να εκτελεστούν αριθμητικοί υπολογισμοί, το NumPy παρέχει μια βολική και ευέλικτη λύση (Varoquaux Emmanuelle Gouillart Olaf Vahtras Pierre de Buyl Scipy et al., n.d.).

Το Matplotlib είναι μια βιβλιοθήκη 2D σχεδίασης για την Python που παρέχει έναν βολικό και ισχυρό τρόπο δημιουργίας όπως φάσματος στατικών, κινούμενων και διαδραστικών απεικονίσεων. Είναι μια από όπως πιο ευρέως χρησιμοποιούμενες βιβλιοθήκες οπτικοποίησης στην επιστημονική κοινότητα όπως Python και χρησιμοποιείται για ένα μεγάλο φάσμα εφαρμογών, όπως

οπτικοποίηση δεδομένων, επιστημονική γραφική παράσταση και ανάλυση δεδομένων.

Μερικά από τα βασικά χαρακτηριστικά του Matplotlib περιλαμβάνουν:

1. Προσαρμοσσιμο: Το Matplotlib παρέχει ένα ευρύ φάσμα επιλογών προσαρμογής, επιτρέποντάς να ελεγχθεί η εμφάνιση και η αίσθηση των απεικονίσεων του χρήστη , από το συνολικό στυλ έως τις λεπτομέρειες μεμονωμένων στοιχείων πλοκής.
2. Διαδραστικότητα: Το Matplotlib παρέχει διαδραστικά εργαλεία για εξερεύνηση και χειρισμό οπτικοποιήσεων, όπως ζουμ, μετατόπιση και σχολιασμό.
3. Συμβατότητα: Το Matplotlib είναι συμβατό με άλλες βιβλιοθήκες και εργαλεία, συμπεριλαμβανομένων των NumPy, Pandas και Seaborn, καθιστώντας εύκολη την ενσωμάτωση στη ροή εργασιών ανάλυσης και οπτικοποίησης δεδομένων.
4. Έξοδος ποιότητας δημοσίευσης: Το Matplotlib παρέχει αποτελέσματα υψηλής ποιότητας για έντυπη και διαδικτυακή δημοσίευση, επιτρέποντας να δημιουργηθούν εικόνες και γραφές ποιότητας έκδοσης με ευκολία.

Το Matplotlib παρέχει έναν ισχυρό και ευέλικτο τρόπο οπτικοποίησης δεδομένων στην Python και είναι ένα απαραίτητο εργαλείο για πολλούς επιστήμονες και ερευνητές δεδομένων. Είτε δημιουργούνται απλές πλοκές είτε σύνθετες απεικονίσεις, το Matplotlib παρέχει τα εργαλεία και τη λειτουργικότητα που χρειάζεται για να ολοκληρωθεί η δουλειά(ben Root & Packt Publishing, 2016).

Το Folium είναι μια βιβλιοθήκη Python που παρέχει έναν απλό τρόπο οπτικοποίησης γεωχωρικών δεδομένων. Είναι χτισμένο στη βιβλιοθήκη JavaScript Leaflet και επιτρέπει την δημιουργία διαδραστικών χαρτών με μια ποικιλία από δείκτες, αναδυόμενα παράθυρα και άλλες δυνατότητες.

Μερικά από τα βασικά χαρακτηριστικά του Folium περιλαμβάνουν:

1. Εύκολο στη χρήση: Το Folium παρέχει μια απλή και διαισθητική διεπαφή για τη δημιουργία χαρτών και την οπτικοποίηση γεωχωρικών δεδομένων στην Python.
2. Διαδραστικοί: Οι χάρτες Folium είναι διαδραστικοί, επιτρέποντάς την μετακίνηση, μεγέθυνση και εξερεύνηση των δεδομένων. Μπορεί επίσης να προστεθεί μια ποικιλία από δείκτες, αναδυόμενα παράθυρα και άλλες δυνατότητες καθιστώντας τα δεδομένα πιο ενημερωτικά και ελκυστικά.
3. Προσαρμόσιμο: Το Folium παρέχει ένα ευρύ φάσμα επιλογών προσαρμογής, επιτρέποντας στον χρήστη να ελέγχει την εμφάνιση και την αίσθηση των χαρτών, από το συνολικό στυλ έως τις λεπτομέρειες μεμονωμένων δεικτών και αναδυόμενων παραθύρων.
4. Ενσωμάτωση με άλλες βιβλιοθήκες: Το Folium είναι χτισμένο πάνω από το Leaflet, το οποίο είναι μια ευρέως χρησιμοποιούμενη βιβλιοθήκη ανοιχτού κώδικα για χαρτογράφηση Ιστού και είναι συμβατό με ένα ευρύ φάσμα άλλων βιβλιοθηκών και εργαλείων, συμπεριλαμβανομένων των Pandas, GeoPandas και Matplotlib, μεταξύ άλλων.

Το Folium είναι μια χρήσιμη βιβλιοθήκη για τη δημιουργία διαδραστικών χαρτών και την οπτικοποίηση γεωχωρικών δεδομένων στην Python. Είτε για την δημιουργία απλών χαρτών για την επικοινωνία των δεδομένων είτε σύνθετες οπτικοποιήσεις για εξερεύνηση και ανάλυση των δεδομένων, το Folium παρέχει μια βολική και ισχυρή λύση (Unknown, n.d.).

Η Geopy είναι μια βιβλιοθήκη Python για γεωκωδικοποίηση και εργασία με γεωγραφικά δεδομένα. Η γεωκωδικοποίηση είναι η διαδικασία μετατροπής μιας φυσικής διεύθυνσης ή τοπωνυμίου σε ένα σύνολο γεωγραφικών συντεταγμένων (γεωγραφικό πλάτος και μήκος). Το Geopy παρέχει μια απλή και εύχρηστη διεπαφή για εργασία με γεωκωδικοποίηση και γεωγραφικά δεδομένα στην Python και διευκολύνει την ενσωμάτωση της γεωκωδικοποίησης στις ροές εργασιών ανάλυσης και οπτικοποίησης δεδομένων σας.

Μερικά από τα βασικά χαρακτηριστικά του Geopy περιλαμβάνουν:

1. Υποστήριξη για πολλαπλές υπηρεσίες γεωκωδικοποίησης: Η Geopy παρέχει υποστήριξη για ένα ευρύ φάσμα υπηρεσιών γεωκωδικοποίησης, συμπεριλαμβανομένης της υπηρεσίας OpenStreetMap Nominatim, των Χαρτών Google και άλλων.
2. Εύκολο στη χρήση: Το Geopy παρέχει μια απλή και διαισθητική διεπαφή για εργασία με γεωγραφικά δεδομένα, καθιστώντας εύκολη τη μετατροπή διευθύνσεων σε γεωγραφικές συντεταγμένες και αντίστροφα.
3. Ενσωμάτωση με άλλες βιβλιοθήκες: Το Geopy είναι συμβατό με άλλες δημοφιλείς βιβλιοθήκες στο οικοσύστημα Python, συμπεριλαμβανομένων των NumPy, Pandas και Matplotlib, καθιστώντας εύκολη την ενσωμάτωση της γεωκωδικοποίησης στις ροές εργασιών ανάλυσης δεδομένων και οπτικοποίησης.

Το Geopy διευκολύνει την εργασία με γεωγραφικά δεδομένα στην Python, είτε γίνεται γεωκωδικοποίηση διευθύνσεων, είτε εκτελεί χωρική ανάλυση είτε δημιουργεί απεικονίσεις γεωχωρικών δεδομένων. Είναι ένα απαραίτητο εργαλείο για πολλούς επιστήμονες δεδομένων και ερευνητές που εργάζονται με γεωχωρικά δεδομένα (GeoPy Documentation Release 2.3.0 GeoPy Contributors, 2022).

2.6 Βάση δεδομένων

Για τη νοητική απεικόνιση των αρχαιολογικών πληροφοριών χρησιμοποιήθηκε μία σχετική βάση δεδομένων, η οποία δημιουργήθηκε στα πλαίσια της πτυχιακής εργασίας του φοιτητή Χαράλαμπου Κωνσταντή του Ιονίου Πανεπιστημίου με επιβλέπουσα καθηγήτρια την κυρία Άννα Σωτηροπούλου.

Η πληροφορία που είναι αποθηκευμένη στη βάση δεδομένων αναφέρεται στην εποχή του Χαλκού το 3200 π.Χ και πιο συγκεκριμένα στην περιοχή του Λαυρίου, Αττικής. (Κωνσταντής, 2022)

Τα μεταλλεία διατίθονταν προς εκμετάλλευση μόνο σε εύπορους Αθηναίους πολίτες για προκαθορισμένο χρονικό διάστημα και με συγκεκριμένο ποσοστό του πολίτη επί των κερδών. Αποσπασματικές επιγραφές πολιτών βρέθηκαν σε

ανασκαφές που έγιναν επί της Αθηναϊκής Αγοράς. Αποτελούσαν επίσημα έγγραφα του Αθηναϊκού κράτους και ανάμεσά τους διακρίνουμε και επιγραφές που αφορούν μισθώσεις των μεταλλείων του Αρχαίου Λαυρίου από την πολιτεία προς τους πολίτες για εκμετάλλευση. Αυτές είχαν συγκεκριμένη μορφή και τις περισσότερες φορές ανέφεραν: το όνομα του μεταλλείου, το είδος του, τον δήμο που άνηκε, το όνομα του μισθωτή και τον δήμο που άνηκε και το αντίτιμο που πλήρωνε αυτός για τη μίσθωση του μεταλλείου. Επίσης ανέφεραν μια περιγραφή της τοποθεσίας που βρισκόταν το μεταλλείο, εστιάζοντας στα περίχωρα που συνόρευαν με αυτό. Αυτό δημιούργησε την δυνατότητα να μπορούν να δημιουργηθούν κατάλογοι ονοματισμένων ορυχείων με τη θέση αυτών να περιγράφεται με ένα σύστημα αναφοράς στα γεωμορφολογικά και ανθρωπογενή χαρακτηριστικά με τα οποία αυτά συνορεύουν στα τέσσερα σημεία του ορίζοντα (Βορράς, Νότος, Ανατολή, Δύση). Τα χαρακτηριστικά μεταξύ άλλων μπορεί να είναι ονόματα τοποθεσιών ή άλλων γεωγραφικών χαρακτηριστικών, όπως ποτάμια, θάλασσες και βουνά ή ακόμα ιδιοκτησίες τρίτων. (Κωνσταντής, 2022)

2.7 Σύνδεση Python με MySQL Server

Υπάρχουν πολλοί τρόποι για να γίνει μια σύνδεση python με την MySQL. Στην συνέχεια παραθέτω 3 τρόπους που μπορεί αυτό να επιτευχθεί :

1. Χρήση της ενσωματωμένης βιβλιοθήκης mysql-connector-python: Αυτή είναι μια βιβλιοθήκη που παρέχεται από τη MySQL και επιτρέπει την σύνδεση σε έναν διακομιστή MySQL χρησιμοποιώντας Python. Παρέχει μια απλή και απλή διεπαφή για τη σύνδεση, την εκτέλεση ερωτημάτων και τη διαχείριση συναλλαγών.
2. Χρήση της βιβλιοθήκης pymysql: Πρόκειται για μια καθαρή εφαρμογή Python ενός πελάτη MySQL που επιτρέπει την σύνδεση σε έναν διακομιστή MySQL και να αλληλεπιδράσει μαζί του χρησιμοποιώντας Python. Παρέχει παρόμοια διεπαφή με τη βιβλιοθήκη mysql-connector-python, αλλά δεν διατηρείται από τη MySQL και μπορεί να έχει διαφορετικά χαρακτηριστικά και χαρακτηριστικά απόδοσης (Code.Hub, 2022).
3. Χρήση ενός Object Relational Mapper (ORM) όπως η SQLAlchemy: Αυτή είναι μια βιβλιοθήκη τρίτου μέρους που παρέχει μια διεπαφή υψηλού

επιπέδου για αλληλεπίδραση με βάσεις δεδομένων SQL στην Python. Αφαιρεί τις υποκείμενες δηλώσεις SQL και σας επιτρέπει να αλληλεπίδραση με τη βάση δεδομένων χρησιμοποιώντας αντικείμενα Python. Αυτό μπορεί να απλοποιήσει τον κώδικά σας και να τον διευκολύνει στη διατήρησή του.

Ο τρόπος που χρησιμοποιήθηκε σε αυτήν την πτυχιακή για την σύνδεση είναι ο δεύτερος. Αναλυτικότερα η Pymysql είναι μια βιβλιοθήκη Python που παρέχει έναν απλό τρόπο σύνδεσης και αλληλεπίδρασης με μια βάση δεδομένων MySQL. Είναι μια καθαρή εφαρμογή Python του πρωτοκόλλου MySQL, που σημαίνει ότι δεν απαιτεί εξωτερικές βιβλιοθήκες ή εργαλεία (Pajankar, 2021).

Μερικά από τα χαρακτηριστικά του Pymysql περιλαμβάνουν:

1. Συμβατότητα: Το pymysql είναι συμβατό με Python 2.7 και Python 3.x. Λειτουργεί σε όλα τα δημοφιλή λειτουργικά συστήματα, συμπεριλαμβανομένων των Windows, macOS και Linux.
2. Σύνδεση: Για να συνδεθεί σε μια βάση δεδομένων MySQL χρησιμοποιώντας pymysql, πρέπει να δοθούν πολλές πληροφορίες, όπως το όνομα κεντρικού υπολογιστή του διακομιστή, το όνομα χρήστη, τον κωδικό πρόσβασης και το όνομα της βάσης δεδομένων στην οποία θέλει ο χρήστης να συνδεθεί. Μπορεί να χρησιμοποιήσει τη συνάρτηση connect() από τη βιβλιοθήκη pymysql για να δημιουργηθεί μια σύνδεση.
3. Δρομείς: Αφού δημιουργηθεί μια σύνδεση, μπορεί ο χρήστης να χρησιμοποιήσει έναν δρομέα για να εκτελεί εντολές SQL και να ανακτήσει αποτελέσματα. Ο δρομέας είναι ένα αντικείμενο που αντιπροσωπεύει έναν δρομέα βάσης δεδομένων, ο οποίος χρησιμοποιείται για την εκτέλεση εντολών SQL και την ανάκτηση αποτελεσμάτων.
4. Συναλλαγές: Η pymysql υποστηρίζει συναλλαγές, οι οποίες επιτρέπουν την εκτέλεση πολλαπλών δηλώσεων SQL ως μια ενιαία, ατομική λειτουργία. Αυτό μπορεί να είναι χρήσιμο για τη διασφάλιση ότι η βάση δεδομένων παραμένει σε συνεπή κατάσταση ακόμη και αν παρουσιαστεί σφάλμα κατά την εκτέλεση μιας πολύπλοκης λειτουργίας.

5. Κρυπτογράφηση SSL: Εάν πρέπει να διασφαλιστεί ότι η σύνδεση με τον διακομιστή MySQL είναι ασφαλής, η pymysql υποστηρίζει κρυπτογράφηση SSL.
6. Προετοιμασμένες δηλώσεις: Η pymysql υποστηρίζει προετοιμασμένες δηλώσεις, οι οποίες επιτρέπουν να εκτελεί ο χρήστης την ίδια πρόταση SQL πολλές φορές με διαφορετικές παραμέτρους. Οι προετοιμασμένες δηλώσεις μπορούν να βελτιώσουν την απόδοση των λειτουργιών της βάσης δεδομένων και επίσης να βοηθήσουν στην αποτροπή επιθέσεων SQL injection.

Το pymysql συνδέει τον χρήστη σε έναν διακομιστή MySQL, για να εκτελέσει ερωτήματα SQL και να ανακτήσει τα αποτελέσματα. Μπορεί επίσης να χρησιμοποιήσει το pymysql για να εκτελέσει πιο περίπλοκες λειτουργίες, όπως τη δημιουργία πινάκων, την εισαγωγή δεδομένων, την ενημέρωση δεδομένων και τη διαγραφή δεδομένων.

Η pymysql είναι μια δημοφιλής βιβλιοθήκη για εργασία με βάσεις δεδομένων MySQL στην Python και είναι μια καλή επιλογή ως μια αξιόπιστη και εύχρηστη βιβλιοθήκη για σύνδεση και εργασία με βάσεις δεδομένων MySQL.

Κεφάλαιο 3: Οπτικοποίηση δεδομένων και απεικόνιση σε χάρτες

3.1 Εξαγωγή και μορφοποίηση δεδομένων

Στόχος ήταν να γίνει απεικόνιση των δεδομένων της βάσης σε χάρτες, ανάλογα με την σημασία τους, ώστε να μπορούν να χρησιμοποιηθούν και να είναι εύκολα κατανοητά σε όλους. Η Βάση δεδομένων είναι αποθηκευμένη σε έναν MySQL Server, η οποία δημιουργήθηκε από τον Κωνσταντή Χ.. Αρχικά έγινε προσπέλαση της βάσης αυτής χρησιμοποιώντας έναν κέρσορα για να προσπελαστούν τα δεδομένα. Η βιβλιοθήκη που χρησιμοποιήθηκε για την σύνδεση με την βάση δεδομένων είναι η pymysql. Σε ένα αντικείμενο με όνομα «conn» αποθηκεύτηκαν οι παράμετροι που χρειάστηκαν για να παραχθούν τα αποτελέσματα . Πιο συγκεκριμένα το όνομα του χρήστη καθορίστηκε ως «root», ο κωδικός παρέμεινε κενός και ο εξυπηρετητής σαν «localhost». Το τελευταίο, σημαίνει πως η βάση δεδομένων τρέχει στο ίδιο μηχάνημα με τον κώδικα που γράφτηκε. Στην συνέχεια δημιουργήθηκε ένας κέρσορας με το όνομα cursor . Έπειτα, δημιουργήθηκε ένα CVS αρχείο τοπικά ώστε να φορτωθούν μέσα οι συντεταγμένες των Γεωγραφικών Πινάκων που δόθηκαν, στη συγκεκριμένη περίπτωση από τον πίνακα dimoi. Η δημιουργία του csv αρχείου έγινε με την χρήση του πίνακα «dimoi» αφού πρώτα αποσπάστηκε μόνο η πληροφορία που ήταν αναγκαία για δημιουργία των «dataframes». Αφού προσπελάστηκαν τα δεδομένα από το CSV αρχείο που δημιουργήθηκε, μέσω της λειτουργίας «pd.read_csv» που ανήκει στην pandas βιβλιοθήκη, φορτώθηκαν σε ένα pandas dataframe το οποίο ονομάστηκε «data». Με την χρήση του κέρσορα που αναφέρθηκε προηγουμένως έτρεξαν κάποια SQL ερωτήματα χρησιμοποιώντας δεδομένα διαφορετικών πινάκων. Το πρώτο SQL ερώτημα έφερε τα μοναδικά ονόματα δήμων, τα ονόματα μεταλλείων, τα είδη μεταλλείων, τα id δήμων και τα id ειδών μεταλλείου αφού φιλτραρίστηκαν όσα id μεταλλείων ήταν κενά . Αποθηκεύτηκε και εκτελέστηκε το ερώτημα αφού χρησιμοποίησε τον κέρσορα και με την μέθοδο της Python, «fetchall()» μέσω της οποίας ανακτήθηκαν τα δεδομένα σε πλειάδες. Η μεταβλητή η οποία είχε την πληροφορία που αναφέρθηκε, ονομάστηκε «dataA». Το επόμενο βήμα στον

κώδικα ήταν να γραφτεί ένας βρόχος «for» που επαναλαμβάνεται πάνω από τα δεδομένα της μεταβλητής «dataA» και προσθέτει τις τιμές πέντε λιστών σε ένα dictionary αντίστοιχα στην καθεμία, με το όνομα «dictionary». Κάθε λίστα αντιστοιχίζεται σε μία κολώνα του ερωτήματος SQL που αναλύθηκε παραπάνω.

Στην Python τα dictionaries είναι μία συλλογή από ζευγάρια κλειδιών-τιμών που αντιστοιχίζονται σε μία συγκεκριμένη τιμή. Δεν έχουν σειρά ή δείκτες αλλά κλειδιά. Τα κλειδιά αυτά πρέπει να είναι μοναδικά. Ένα dictionary έχει την μορφή:

```
d = {'key1': 'value1', 'key2': 'value2', 'key3': 'value3'}
```

Επόμενο βήμα ήταν να δημιουργηθεί ένα Pandas Dataframe από το dictionary και να ενωθεί με ένα δεύτερο dataframe που ονομάζεται «data» πάνω στις τιμές της κολώνας "Όνομα Δήμου" από το dictionary και της κολώνας "onoma" του «data» dataframe. Το αποτέλεσμα αυτής της ένωσης αποθηκεύτηκε σε ένα τρίτο Dataframe με το όνομα df_cd.

Με παρόμοια λογική εκτελέστηκαν και τα υπόλοιπα ερωτήματα και φορτώθηκαν σε pandas dataframes. Στο δεύτερο ερώτημα SQL πάρθηκαν όλα τα στοιχεία του πίνακα «onitis» και όσα είχαν κοινά id Δήμων από τον πίνακα «b-n-a-d». Το ερώτημα εκτελέστηκε χρησιμοποιώντας τη μέθοδο "cursor.execute" και το αποτέλεσμα αποθηκεύτηκε στη μεταβλητή "datab". Στη συνέχεια, η μέθοδος "fetchall" καλέστηκε από τον κέρσορα για να ανακτήσει όλες τις σειρές του συνόλου των αποτελεσμάτων. Όπως και προηγουμένως, δημιουργήθηκε ένα λεξικό με όνομα "dictionaryA" το οποίο περιείχε τιμές σε λίστες. Η πληροφορία αποθηκεύτηκε στη μεταβλητή "datab". Έπειτα, έγινε η μετατροπή σε Pandas dataframe χρησιμοποιώντας την μέθοδο "pd.DataFrame" και το dataframe αυτό ενώθηκε με το dataframe "data" χρησιμοποιώντας την μέθοδο "pd.merge" επάνω στις κολώνες "ID Δήμων" και των δύο πλαισίων. Το τελικό dataframe αποθηκεύτηκε σε μία μεταβλητή με το όνομα «df_oniti».

Το τρίτο SQL ερώτημα επέλεξε τις μοναδικές τιμές των ονομάτων δήμων, των id δήμων από τους πίνακες «edafh» και «b-n-a-d» ενώ φιλτραρίστηκαν όσα id δήμων δεν περιείχαν τιμές. Με την ίδια διαδικασία δημιουργήθηκε ένα dictionary για την κάθε λίστα και ανακτήθηκαν οι τιμές των κολωνών Όνομα Εδάφους –

Γαιοκτήμονας και Id Δήμου. Το αποτέλεσμα αποθηκεύτηκε σε ένα Pandas dataframe με όνομα «df_edafh» αφού πρώτα έγινε η ένωση των dataframe «data» και dictionaryB πάνω στα id των δήμων.

3.2 Ομαδοποίηση δεδομένων και μετασχηματισμός για χαρτογράφηση

Το dataframe που αναλύθηκε παραπάνω με το όνομα “data” περιείχε πληροφορίες οι οποίες χρησιμοποιήθηκαν στην συνέχεια για την απεικόνιση τους στους χάρτες. Χρειάστηκε να γίνουν ομαδοποιήσεις στις κολώνες που περιείχε το dataframe και πιο συγκεκριμένα ομαδοποιήθηκαν οι κολώνες 'lon', 'lat', και 'onoma', ενώ επίσης υπολογίστηκε το μέγεθος της κάθε ομάδας. Η μεταβλητή start_location περιείχε την πληροφορία με τα ονόματα των δήμων και προστέθηκε σε αυτή και μία νέα κολώνα με το όνομα Map. Το ίδιο συμβαίνει και για την ομαδοποίηση του «df_cd» που περιέχει τις κολώνες 'lon', 'lat', 'Όνομα Μεταλλείου' και 'Είδος Μεταλλείου'. Επίσης σε αυτό το σημείο προστέθηκε μία νέα κολώνα με το όνομα Map. Η μεταβλητή «metalleia» περιέχουν το «Είδος Μεταλλείου». Με τον ίδιο ακριβώς τρόπο δημιουργήθηκαν και οι μεταβλητές onitis που περιέχει την πληροφορία για το όνομα του ωνήτη, η μεταβλητή «edafh» που περιέχει την πληροφορία «Όνομα Εδάφους – Γαιοκτήμονας».

Τέλος, μετατράπηκαν και τα τέσσερα dataframes start_location, metalleia, onitis, και edafh σε numpy arrays. Τα numpy arrays είναι μια δομή δεδομένων στη βιβλιοθήκη numpy της Python. Χρησιμοποιούνται για να αποθηκεύουν δεδομένα ίδιου τύπου με συμπαγές και αποτελεσματικό τρόπο. Χρησιμεύουν για μαθηματικές διεργασίες και είναι πολύ γρηγορότερα από άλλες λίστες στην Python.

3.3 Δημιουργία γεωγραφικού χάρτη με τη χρήση του Folium

Στο συγκεκριμένο σημείο γίνεται η οπτική αναπαράσταση των δεδομένων σε χάρτες χρησιμοποιώντας τη βιβλιοθήκη Folium. Αρχικά δημιουργείται η βάση του χάρτη παίρνοντας την μέση τιμή των συντεταγμένων που έχουν φορτωθεί

στο «start_location» και ορίζεται η μεγέθυνση του χάρτη. Στη συνέχεια διαβάζεται κάθε γραμμή του «start_location» και δημιουργείται ένα κόκκινο κυκλικό σημάδι με ακτίνα ανάλογη με το γεωγραφικό πλάτος του σημείου δεδομένων και το προσθέτει στον χάρτη. Επιπλέον, για κάθε γραμμή, δημιουργείται ένας αντίστοιχος πράσινος δείκτης με το όνομα της τοποθεσίας και προστίθεται στον χάρτη. Όταν ο κέρσορας του χρήστη πλανάται πάνω από τα σημάδια που βλέπει στον χάρτη, τότε εμφανίζεται η πληροφορία την οποία θα δει αφού πατήσει στα αντίστοιχα εικονίδια.

Για την δημιουργία των χαρτών χρησιμοποιήθηκαν οι παρακάτω μέθοδοι:

- folium.Map: δημιουργεί έναν αρχικό χάρτη με δεδομένη τοποθεσία και συγκεκριμένο βαθμό μεγέθυνσης
- np.mean: υπολογίζει τον μέσο όρο ενός συνόλου αριθμών
- folium.CircleMarker: δημιουργεί έναν κυκλικό δείκτη στο χάρτη
- folium.Marker: δημιουργεί έναν δείκτη στο χάρτη
- add_to: προσθέτει ένα δείκτη ή άλλα αντικείμενα στο χάρτη
- shape: επιστρέφει το σχήμα ενός πίνακα, δηλαδή τον αριθμό των γραμμών και στηλών

Ο πρώτος χάρτης που δημιουργήθηκε απεικονίζει τα ονόματα των δήμων όπως καταγράφονται από την βάση δεδομένων του Κωνσταντή Χ. Με κόκκινο χρώμα παρατηρείται η γενική έκταση του κάθε δήμου.

Στον δεύτερο χάρτη η πληροφορία που απεικονίζεται είναι το Όνομα του Ωνήτη. Ο τρίτος απεικονίζει το Όνομα και το Είδος κάθε Μεταλλείου και τέλος, ο τέταρτος χάρτης απεικονίζει το Όνομα Εδάφους και τον αντίστοιχο Γαιοκτήμονα.

Κεφάλαιο 4 Απεικόνιση βάσης δεδομένων

4.1 Αποτελέσματα κώδικα

Στην εικόνα 26 είναι η εγκατάσταση συγκεκριμένων modules και βιβλιοθηκών

```
!pip install pymysql
!pip install folium
!pip install geopy
import sys
import pymysql
import sys
import folium
from folium.plugins import MarkerCluster
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Requirement already satisfied: pymysql in c:\users\xeno\anaconda3\lib\site-packages (1.0.2)
Requirement already satisfied: folium in c:\users\xeno\anaconda3\lib\site-packages (0.12.1.post1)
Requirement already satisfied: Jinja2>=2.9 in c:\users\xeno\anaconda3\lib\site-packages (from folium) (2.11.3)
Requirement already satisfied: branca>=0.3.0 in c:\users\xeno\anaconda3\lib\site-packages (from folium) (0.5.0)
Requirement already satisfied: numpy in c:\users\xeno\anaconda3\lib\site-packages (from folium) (1.21.5)
Requirement already satisfied: requests in c:\users\xeno\anaconda3\lib\site-packages (from folium) (2.27.1)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\xeno\anaconda3\lib\site-packages (from Jinja2>=2.9->folium) (2.0.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\xeno\anaconda3\lib\site-packages (from requests->folium) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\xeno\anaconda3\lib\site-packages (from requests->folium) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\xeno\anaconda3\lib\site-packages (from requests->folium) (1.26.9)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\xeno\anaconda3\lib\site-packages (from requests->folium) (2021.10.8)
Requirement already satisfied: geopy in c:\users\xeno\anaconda3\lib\site-packages (2.2.0)
Requirement already satisfied: geographiclib<2,>=1.49 in c:\users\xeno\anaconda3\lib\site-packages (from geopy) (1.52)

Εικόνα 1:Εγκατάσταση βιβλιοθηκών και modules

Στην εικόνα 27 είναι η δημιουργία του dataframe data το οποίο περιέχει το csv με τις συντεταγμένες.

```
In [10]: data = pd.read_csv('/Users/xeno/Desktop/ptyxiakh/Lon_Lat.csv', delimiter=';')
pd.DataFrame(data)
```

Out[10]:

	onoma	id_dimou	lon	lat
0	Σουνίου	1.0	37.661312	24.019108
1	Θορικήων	2.0	37.735490	24.047895
2	Παιανία	3.0	37.960288	23.849142
3	Σφηπτός	4.0	37.906028	23.846839
4	Σιφνίων	5.0	38.507830	22.505062
5	Κυδαντίδες	6.0	38.050612	23.893657
6	Κεραμεικός	7.0	37.980824	23.716659
7	Αίγιο	8.0	38.192015	22.111738
8	Λαμπτρες	15.0	37.872488	23.836559
9	Πιπθεύς	9.0	38.030182	23.868208
10	Αναγυρούς	10.0	37.841040	23.814605
11	Αιξώνη	11.0	37.858803	23.771528
12	Κύθηρος	12.0	37.904356	23.983576
13	Λαύριο	13.0	37.708572	24.052906
14	Ευώνυμον	14.0	37.927525	23.728610
15	Αλωπεκή	16.0	37.957398	23.727528
16	Παλλήνη	17.0	38.003034	23.893731
17	Βήσα	18.0	37.727745	23.991124
18	Γαργηπτός	19.0	38.025354	23.863835
19	Μελίτη	20.0	37.989602	23.721925
20	Αμφιάλη	21.0	37.968800	23.597929
21	Ανάγυρος	22.0	37.841040	23.814605
22	Ανακαία	23.0	38.122438	23.736048
23	Ανάφλυστος	24.0	37.732571	23.937260
24	Συπαλλητός	25.0	38.060060	23.760187
25	NaN	NaN	NaN	NaN

Εικόνα 2: dataframe data

Στην εικόνα 28 συναντάμε την δημιουργία ενός dataframe dictionary που περιέχει τις στήλες : Όνομα Δήμου, Όνομα Μεταλλείου, Είδος Μεταλλείου, Id Δήμου και Id Είδος Μεταλλείου

```
In [11]: query='select DISTINCT a.`Όνομα Δήμου`,b.`Όνομα Μεταλλείου`,c.`Είδος Μεταλλείου`,a.`Id Δήμου`,c.`Id Είδους Μεταλλείου` ;
r=cursor.execute(query)
dataA=cursor.fetchall()

counter=0
dictionary={"Όνομα Δήμου":[], "Όνομα Μεταλλείου":[], "Είδος Μεταλλείου":[], "Id Δήμου":[], "Id Είδους Μεταλλείου":[]}
for d in dataA:
    dictionary["Όνομα Δήμου"].append(d[0])
    dictionary["Όνομα Μεταλλείου"].append(d[1])
    dictionary["Είδος Μεταλλείου"].append(d[2])
    dictionary["Id Δήμου"].append(d[3])
    dictionary["Id Είδους Μεταλλείου"].append(d[4])
    counter+=1
pd.DataFrame(dictionary)
```

```
Out[11]:
```

	Όνομα Δήμου	Όνομα Μεταλλείου	Είδος Μεταλλείου	Id Δήμου	Id Είδους Μεταλλείου
0	Θορικήων	Αρτεμισιακών	Ανασάξιμον	2	1
1	Ανάφλυστος	Φάνειον	Ανασάξιμον	24	1
2	Λαύριο	Ευδότηιον	Ανασάξιμον	13	1
3	Λαύριο	Ερμιαϊκόν	Εργάσιμον	13	2
4	Ανάφλυστος	Αρτεμισιακών	Παλαιόν Ανασάξιμον	24	4
5	Ανάφλυστος	Ποσειδονειακόν	Παλαιόν Ανασάξιμον	24	4
6	Ανάφλυστος	Απολλωνιακόν	Παλαιόν Ανασάξιμον	24	4
7	Λόφο Βαμβιδέω	Ηρωικόν	Παλαιόν Ανασάξιμον	43	4
8	Παιανία	kala	Συγκεχωρημένον	3	5

Εικόνα 3: dataframe dictionary

Στην εικόνα 29 είναι η απεικόνιση της ενώσεις των δυο dataframe της εικόνας 27(dataframe(data)) και της εικόνας 28 (dataframe (dictionary))

```
In [12]: df_cd = pd.merge(pd.DataFrame(dictionary), pd.DataFrame(data), how='left', left_on = 'Όνομα Δήμου', right_on = 'onoma',
df_cd
```

```
Out[12]:
```

	Όνομα Δήμου	Όνομα Μεταλλείου	Είδος Μεταλλείου	Id Δήμου	Id Είδους Μεταλλείου	onoma	id_dimou	lon	lat
0	Θορικήων	Αρτεμισιακών	Ανασάξιμον	2	1	Θορικήων	2.0	37.735490	24.047895
1	Ανάφλυστος	Φάνειον	Ανασάξιμον	24	1	Ανάφλυστος	24.0	37.732571	23.937260
2	Λαύριο	Ευδότηιον	Ανασάξιμον	13	1	Λαύριο	13.0	37.708572	24.052906
3	Λαύριο	Ερμιαϊκόν	Εργάσιμον	13	2	Λαύριο	13.0	37.708572	24.052906
4	Ανάφλυστος	Αρτεμισιακών	Παλαιόν Ανασάξιμον	24	4	Ανάφλυστος	24.0	37.732571	23.937260
5	Ανάφλυστος	Ποσειδονειακόν	Παλαιόν Ανασάξιμον	24	4	Ανάφλυστος	24.0	37.732571	23.937260
6	Ανάφλυστος	Απολλωνιακόν	Παλαιόν Ανασάξιμον	24	4	Ανάφλυστος	24.0	37.732571	23.937260
7	Λόφο Βαμβιδέω	Ηρωικόν	Παλαιόν Ανασάξιμον	43	4	NaN	NaN	NaN	NaN
8	Παιανία	kala	Συγκεχωρημένον	3	5	Παιανία	3.0	37.960288	23.849142

Εικόνα 4: Ένωση dataframe(data) με dataframe(dictionary)

Στην εικόνα 30 και 31 είναι τα στοιχεία που μας δίνει το dataframe dictionaryA

	Id Ωνήτη	Όνομα Ωνήτη	Πατρώνυμο Ωνήτη	Id Δήμου	Id Επιγραφής
0	1	Ευετίων	Αυτοκλείδου	4.0	None
1	2	Αισιμίδης	Διοφάνους	19.0	None
2	3	Τιμοκλείδης	Ισοκράτους	26.0	None
3	4	Επικράτης	Ισοκράτους	14.0	None
4	5	Ευπόλεμος	None	1.0	None
5	6	Αγνόθεος	None	NaN	None
6	7	Καλλιμέδων	Καλλικράτου	27.0	None
7	8	Ευφημίδης	Κηφισόδωρου	28.0	None
8	9	Καλλίας	None	4.0	None
9	10	Επιτέλης	None	29.0	None
10	11	Φείδιππος	None	9.0	None
11	12	Κλεώνυμος	Φιλοχάρους	30.0	None
12	13	Θρασύλοχος	None	10.0	None
13	14	Τελέσαρχος	None	11.0	None
14	15	Θρασύλοχος	None	22.0	None
15	16	Κηφισόδοτος	None	31.0	None
16	17	Δήμων	None	32.0	None
17	18	Καλλίας	None	15.0	None
18	19	Καλλίας	None	4.0	None
19	20	Κηφισοφών	None	33.0	None
20	21	Επικλέης	None	4.0	None
21	22	Κηφισόδοτος	None	31.0	None
22	23	Ευθυκράτης	Αντιδότου	34.0	None
23	24	Αλεξίμαχος	None	35.0	None
24	25	Χαιρέδημος	None	36.0	None
25	26	Ευετίων	Αυτοκλείδου	4.0	None
26	27	Κλεώνυμος	Φιλοχάρους	30.0	None
27	28	Ευθυκράτης	Αντιδότου	34.0	None
28	29	Αρκεσίλο	None	20.0	None
29	30	Μενέξενος	Μεταξένου	38.0	None
30	31	Λυσανίας	Λυσικλέους	40.0	None

Εικόνα 5:Περιεχόμενο από το Dataframe dictionaryA(1)

31	32	Αριστοτέλης	Οψιάδου	41.0	None
32	33	Ίσανδρος	Στρατοκλέους	42.0	None
33	34	Αρκεσίλο	None	20.0	None
34	35	Θεόδωρος	Ολυμπίχου	20.0	None
35	37	Τιμοκλείδης	Υψηχίδου	26.0	None
36	38	Επικράτης	Ισοκράτους	14.0	None
37	39	Ευπόλεμος	None	1.0	None
38	40	Καλλίας	None	4.0	None
39	41	Επιτελής	None	29.0	None
40	42	Φείδιππος	None	9.0	None
41	43	Θρασύλοχος	None	22.0	None
42	44	Τελέσαρχος	None	11.0	None
43	45	Χαιρέδημος	None	36.0	None
44	46	Λυσανίας	Λυσικλέους	40.0	None
45	47	Στρατοκλέους	None	42.0	None
46	48	Θεόδωρος	None	20.0	None

Εικόνα 6:Περιεχόμενο από το Dataframe dictionaryA(2)

Στην εικόνα 32 και 33 είναι το περιεχόμενο του dataframe oniti το οποίο δημιουργήθηκε από την ένωση του dataframe dictionaryA και dataframe data

	Id Ωνήτη	Όνομα Ωνήτη	Πατρώνυμο Ωνήτη	Id Δήμου	Id Επιγραφής	ονομα	id_διμου	lon	lat
0	1	Ευετίων	Αυτοκλείδου	4.0	None	Σφηττός	4.0	37.906028	23.846839
1	2	Αισιμίδης	Διοφάνους	19.0	None	Γαργηττός	19.0	38.025354	23.863835
2	3	Τιμοκλείδης	Ισοκράτους	26.0	None	NaN	NaN	NaN	NaN
3	4	Επικράτης	Ισοκράτους	14.0	None	Ευώνυμον	14.0	37.927525	23.728610
4	5	Ευπόλεμος	None	1.0	None	Σουνίου	1.0	37.661312	24.019108
5	6	Αγνόθεος	None	NaN	None	NaN	NaN	NaN	NaN
6	7	Καλλιμέδων	Καλλικράτου	27.0	None	NaN	NaN	NaN	NaN
7	8	Ευφημίδης	Κηφισόδωρου	28.0	None	NaN	NaN	NaN	NaN
8	9	Καλλίας	None	4.0	None	Σφηττός	4.0	37.906028	23.846839
9	10	Επιτέλης	None	29.0	None	NaN	NaN	NaN	NaN
10	11	Φεΐδιππος	None	9.0	None	Πιπθεύς	9.0	38.030182	23.868208
11	12	Κλεώνυμος	Φιλοχάρους	30.0	None	NaN	NaN	NaN	NaN
12	13	Θρασύλοχος	None	10.0	None	Αναγυρούς	10.0	37.841040	23.814605
13	14	Τελέσαρχος	None	11.0	None	Αιζώνη	11.0	37.858803	23.771528
14	15	Θρασύλοχος	None	22.0	None	Ανάγυρος	22.0	37.841040	23.814605
15	16	Κηφισόδοτος	None	31.0	None	NaN	NaN	NaN	NaN
16	17	Δήμων	None	32.0	None	NaN	NaN	NaN	NaN
17	18	Καλλίας	None	15.0	None	Λαμπτρες	15.0	37.872488	23.836559
18	19	Καλλίας	None	4.0	None	Σφηττός	4.0	37.906028	23.846839
19	20	Κηφισοφών	None	33.0	None	NaN	NaN	NaN	NaN
20	21	Επικλέης	None	4.0	None	Σφηττός	4.0	37.906028	23.846839
21	22	Κηφισόδοτος	None	31.0	None	NaN	NaN	NaN	NaN
22	23	Ευθυκράτης	Αντιδότου	34.0	None	NaN	NaN	NaN	NaN
23	24	Αλεξίμαχος	None	35.0	None	NaN	NaN	NaN	NaN
24	25	Χαιρέδημος	None	36.0	None	NaN	NaN	NaN	NaN
25	26	Ευετίων	Αυτοκλείδου	4.0	None	Σφηττός	4.0	37.906028	23.846839
26	27	Κλεώνυμος	Φιλοχάρους	30.0	None	NaN	NaN	NaN	NaN
27	28	Ευθυκράτης	Αντιδότου	34.0	None	NaN	NaN	NaN	NaN
28	29	Αρκεσίλο	None	20.0	None	Μελίτη	20.0	37.989602	23.721925
29	30	Μενέξενος	Μεταξένου	38.0	None	NaN	NaN	NaN	NaN
30	31	Λυσανίας	Λυσικλέους	40.0	None	NaN	NaN	NaN	NaN

Εικόνα 7: Περιεχόμενο από το Dataframe oniti(1)

31	32	Αριστοτέλης	Οψιάδου	41.0	None	NaN	NaN	NaN	NaN
32	33	Ίσανδρος	Στρατοκλέους	42.0	None	NaN	NaN	NaN	NaN
33	34	Αρκεσίλο	None	20.0	None	Μελίτη	20.0	37.989602	23.721925
34	35	Θεόδωρος	Ολυμπίχου	20.0	None	Μελίτη	20.0	37.989602	23.721925
35	37	Τιμοκλείδης	Υψηχίδου	26.0	None	NaN	NaN	NaN	NaN
36	38	Επικράτης	Ισοκράτους	14.0	None	Ευώνυμον	14.0	37.927525	23.728610
37	39	Ευπάλεμος	None	1.0	None	Σουνίου	1.0	37.661312	24.019108
38	40	Καλλίας	None	4.0	None	Σφηπτός	4.0	37.906028	23.846839
39	41	Επιτελής	None	29.0	None	NaN	NaN	NaN	NaN
40	42	Φεΐδιππος	None	9.0	None	Πιπθεύς	9.0	38.030182	23.868208
41	43	Θρασύλοχος	None	22.0	None	Ανάγυρος	22.0	37.841040	23.814605
42	44	Τελέσαρχος	None	11.0	None	Αιξώνη	11.0	37.858803	23.771528
43	45	Χαιρέδημος	None	36.0	None	NaN	NaN	NaN	NaN
44	46	Λυσανίας	Λυσικλέους	40.0	None	NaN	NaN	NaN	NaN
45	47	Στρατοκλέους	None	42.0	None	NaN	NaN	NaN	NaN
46	48	Θεόδωρος	None	20.0	None	Μελίτη	20.0	37.989602	23.721925

Εικόνα 8:Περιεχόμενο από το Dataframe oniti(2)

Στην εικόνα 34 συναντάμε τα στοιχεία που εμπεριέχει το dataframe dictionary B

	Όνομα Εδάφους - Γαιοκτήμονας	Id Δήμου
0	E-2 Νικίας	6
1	E-4 Κλεόκριτος	8
2	E-5 Λεύκιος	1
3	E-6 Καλλίας	4
4	E-7 Διοκλής	9
5	E-8 Δημόστρατος	12
6	E-9 Διοπείθης	14
7	E-10 Καλλίας	16
8	E-13 Τελέσων	1
9	E-14 Επικράτης	17
10	E-15 Καλλίας	15
11	E-18 Πύρρακος	8
12	E-20 Φανόστρατος	19
13	E-24 Διότιμος	14
14	E-25 Αισχίνης	2
15	E-26 Ευθυκράτος	21
16	E-30 Διφίλου	9
17	E-31 Μειξιάδης	1
18	E-32 Ευπόλεμος	1

Εικόνα 9:Περιεχόμενο από το Dataframe dictionaryB

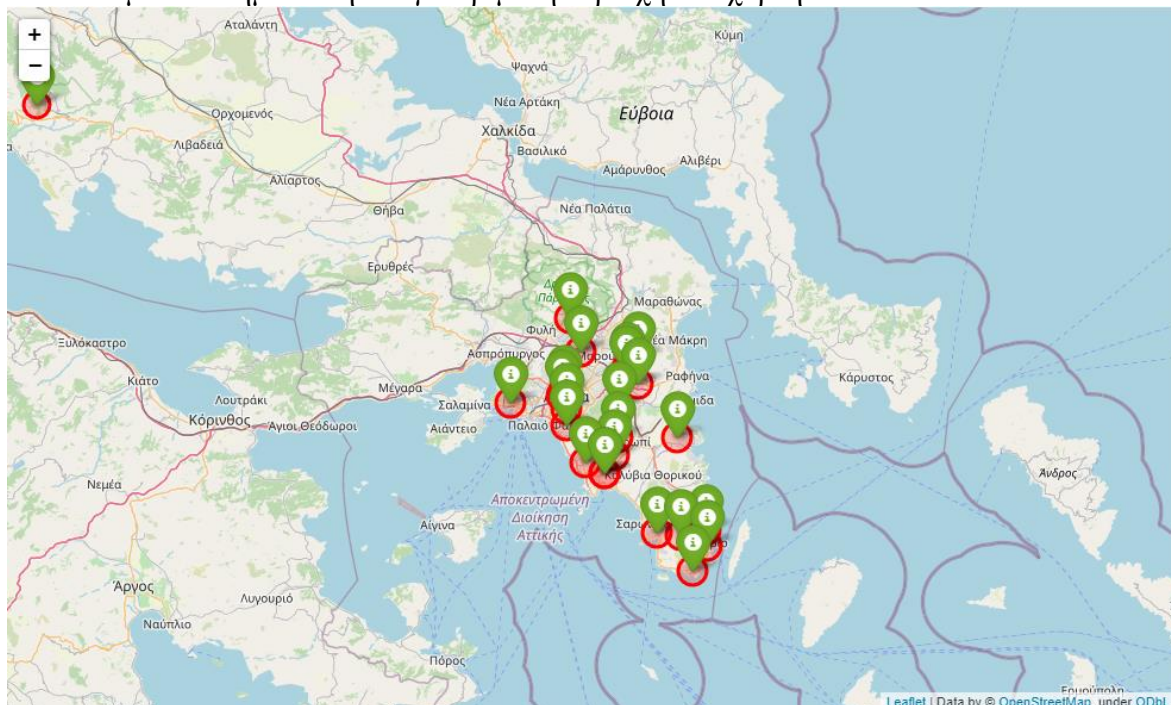
Η εικόνα 35 έχει τα στοιχεία του dataframe edafh το οποίο δημιουργήθηκε από την ένωση του dataframe dictionaryb και του dataframe data

	Όνομα Εδάφους - Γαιοκτήμονας	Id Δήμου	ονομα	id_dimou	lon	lat
0	E-2 Νικίας	6	Κυδαντίδες	6.0	38.050612	23.893657
1	E-4 Κλεόκριτος	8	Αίγιο	8.0	38.192015	22.111738
2	E-5 Λεύκιος	1	Σουνίου	1.0	37.661312	24.019108
3	E-6 Καλλίας	4	Σφηπτός	4.0	37.906028	23.846839
4	E-7 Διοκλές	9	Πιπθεύς	9.0	38.030182	23.868208
5	E-8 Δημόστρατος	12	Κύθηρος	12.0	37.904356	23.983576
6	E-9 Διοπείθης	14	Ευώνυμον	14.0	37.927525	23.728610
7	E-10 Καλλίας	16	Αλωπεκή	16.0	37.957398	23.727528
8	E-13 Τελέσων	1	Σουνίου	1.0	37.661312	24.019108
9	E-14 Επικράτης	17	Παλλήνη	17.0	38.003034	23.893731
10	E-15 Καλλίας	15	Λαμπτρες	15.0	37.872488	23.836559
11	E-18 Πύρρακος	8	Αίγιο	8.0	38.192015	22.111738
12	E-20 Φανόστρατος	19	Γαργηπτός	19.0	38.025354	23.863835
13	E-24 Διότιμος	14	Ευώνυμον	14.0	37.927525	23.728610
14	E-25 Αισχίνης	2	Θορικήων	2.0	37.735490	24.047895
15	E-26 Ευθυκράτος	21	Αμφιάλη	21.0	37.968800	23.597929
16	E-30 Διφίλου	9	Πιπθεύς	9.0	38.030182	23.868208
17	E-31 Μειξιάδης	1	Σουνίου	1.0	37.661312	24.019108
18	E-32 Ευπόλεμος	1	Σουνίου	1.0	37.661312	24.019108

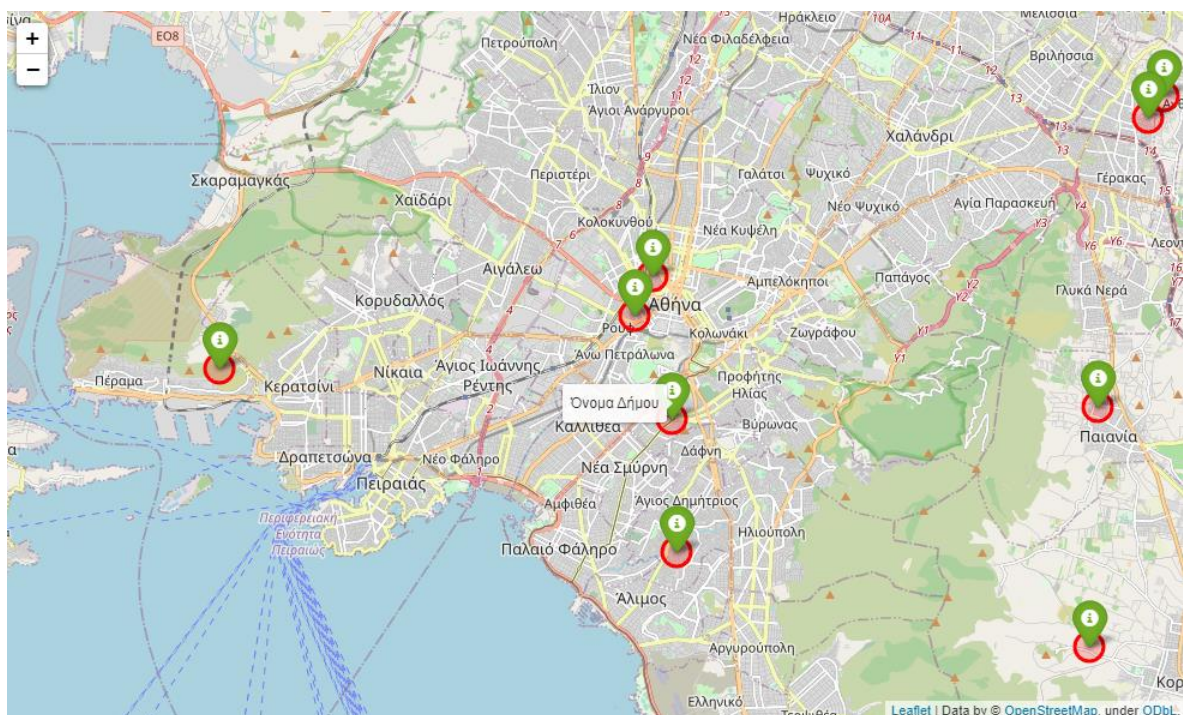
Εικόνα 10:Περιεχόμενο από το Dataframe edafh

4.1.2 Απεικόνιση Χαρτών

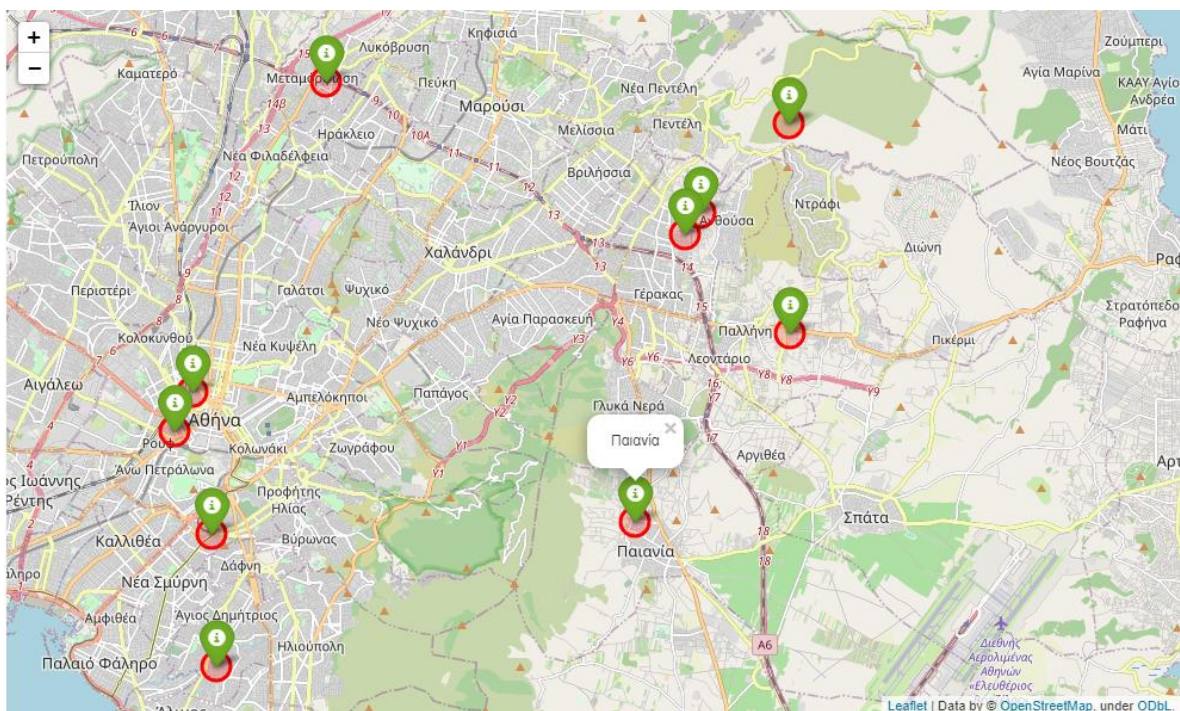
Στις εικόνες από 36 μέχρι 39 απεικονίζετε το map 1 έχει της πινέζες που του έχουμε ορίσει. Σε αυτόν τον χάρτη όταν το επιλεγθεί μια πινέζα εμφανίζεται το όνομα του δήμου στην συγκεκριμένη περιοχή του χάρτη.



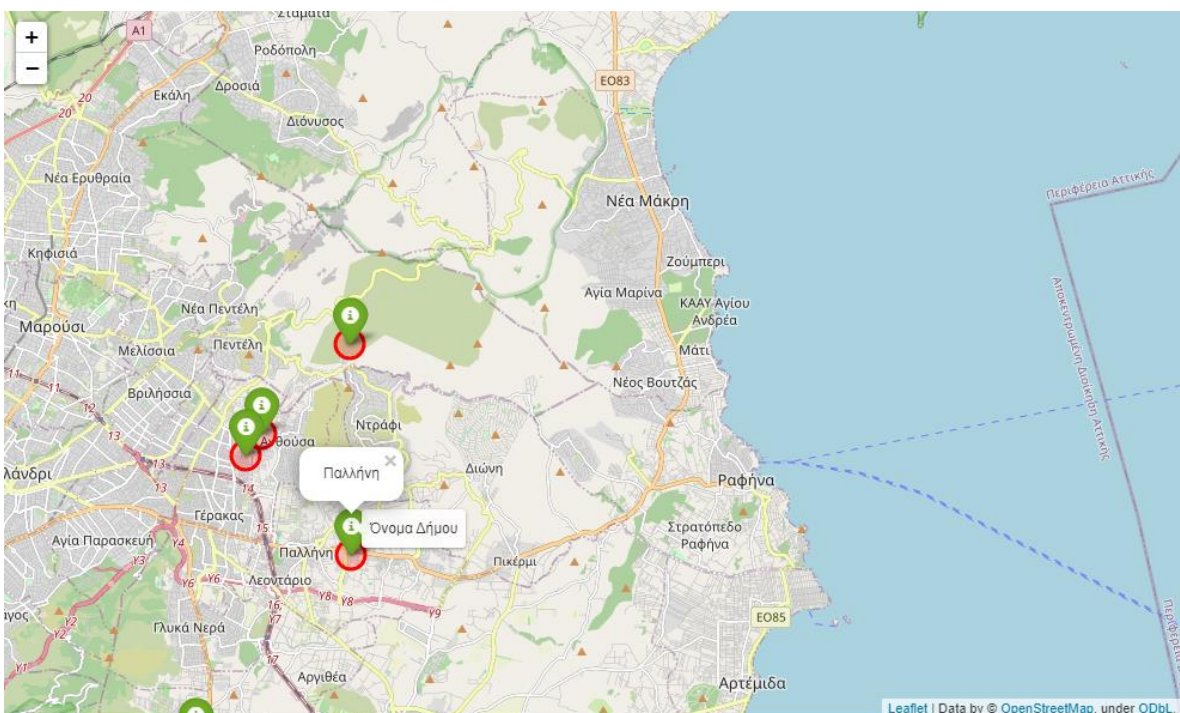
Εικόνα 11: Αποτελέσματα map1(όνομα δήμου) (1)



Εικόνα 12: Αποτελέσματα map1(όνομα δήμου) (2)

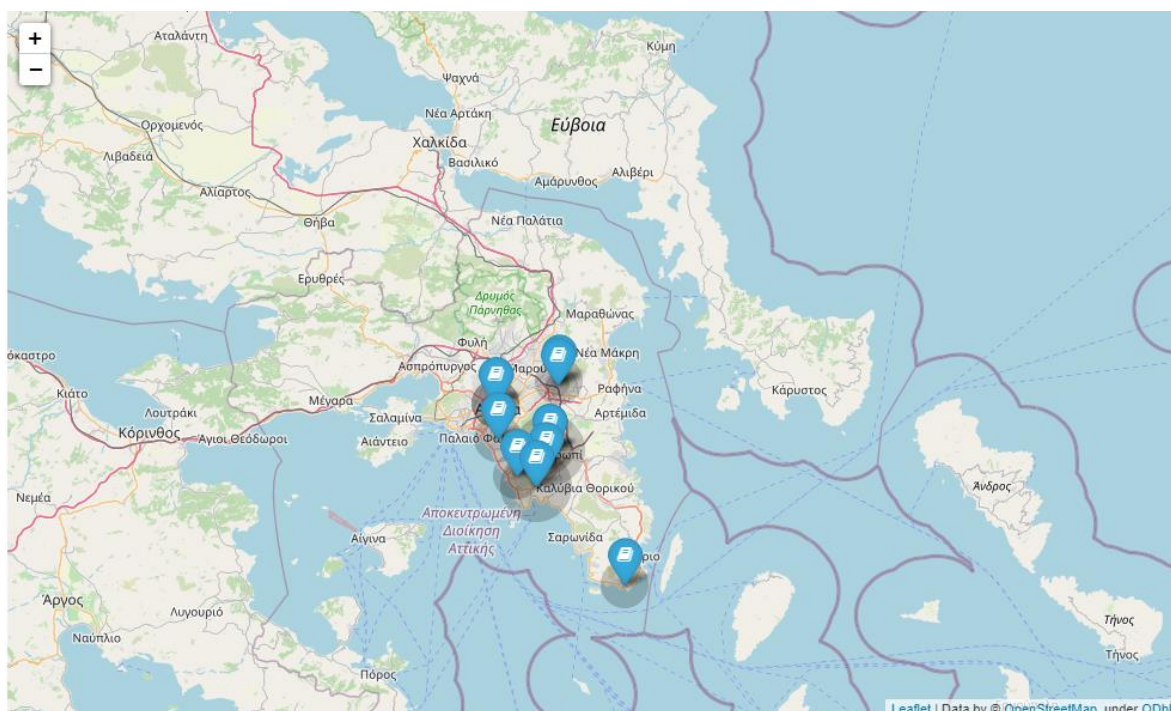


Εικόνα 13: Αποτελέσματα *map1*(όνομα δήμου) (3)

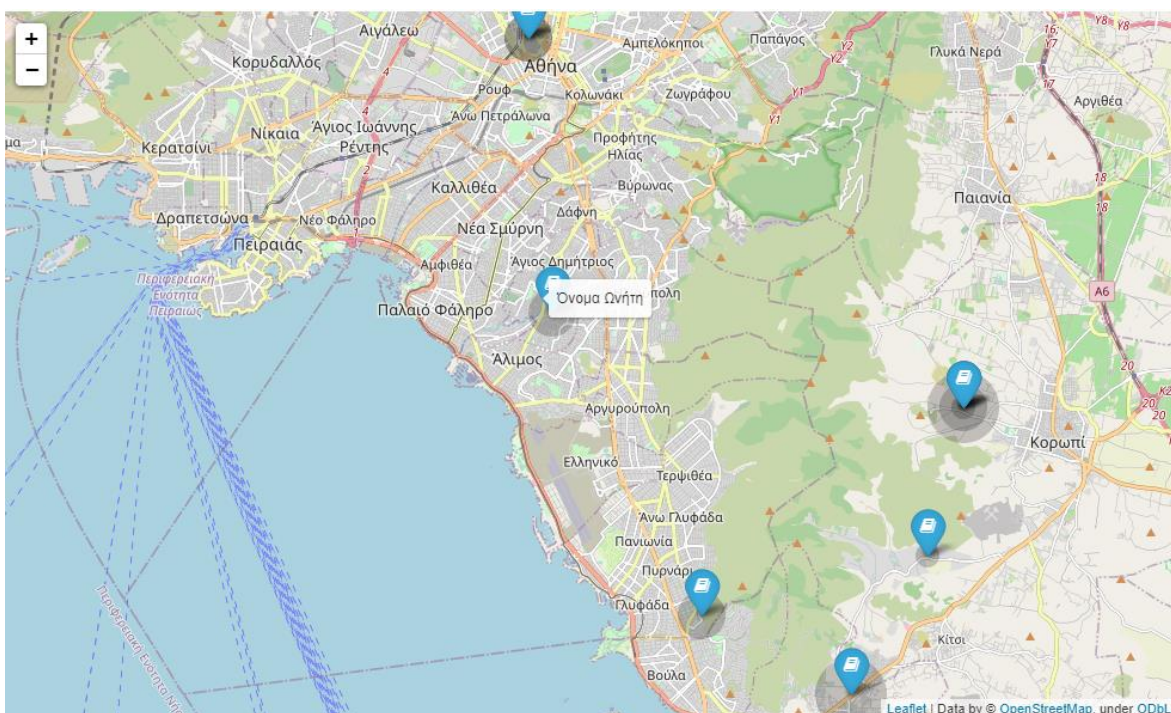


Εικόνα 14: Αποτελέσματα *map1*(όνομα δήμου) (4)

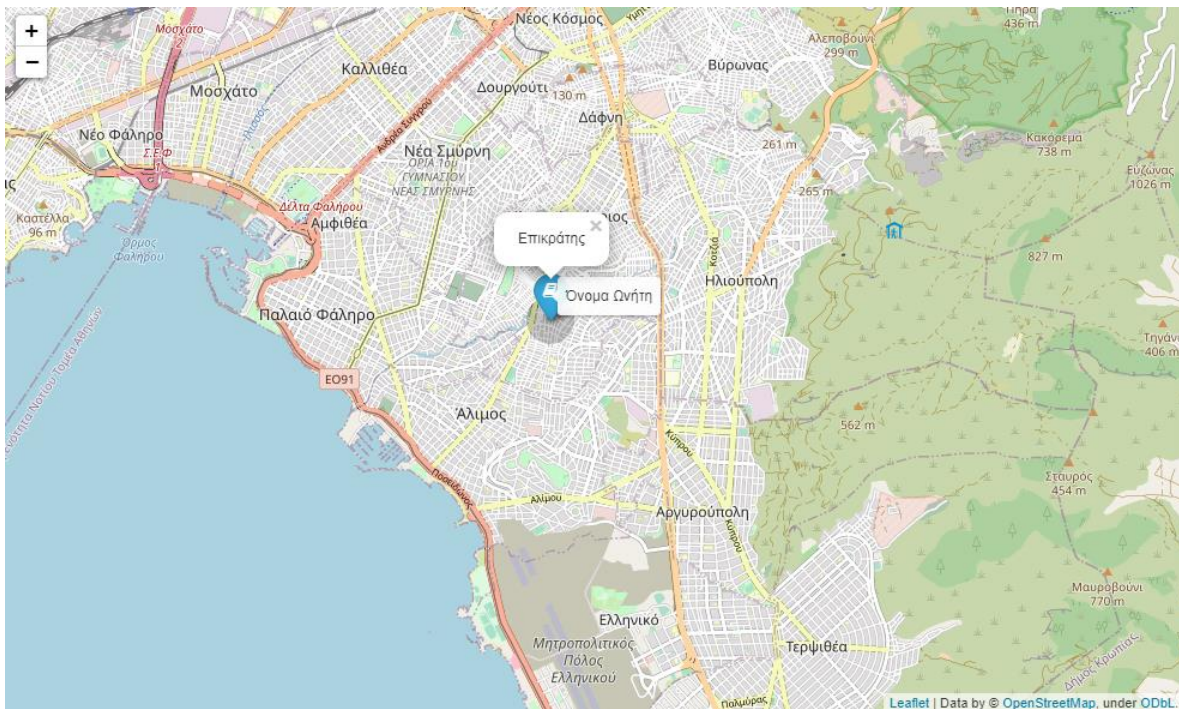
Στις εικόνες από 40 μέχρι 43 απεικονίζετε το map 2 με της πινέζες που του έχουμε ορίσει. Σε αυτόν τον χάρτη όταν το επιλεχθεί μια πινέζα εμφανίζεται το όνομα του ωνήτη στην συγκεκριμένη περιοχή του χάρτη.



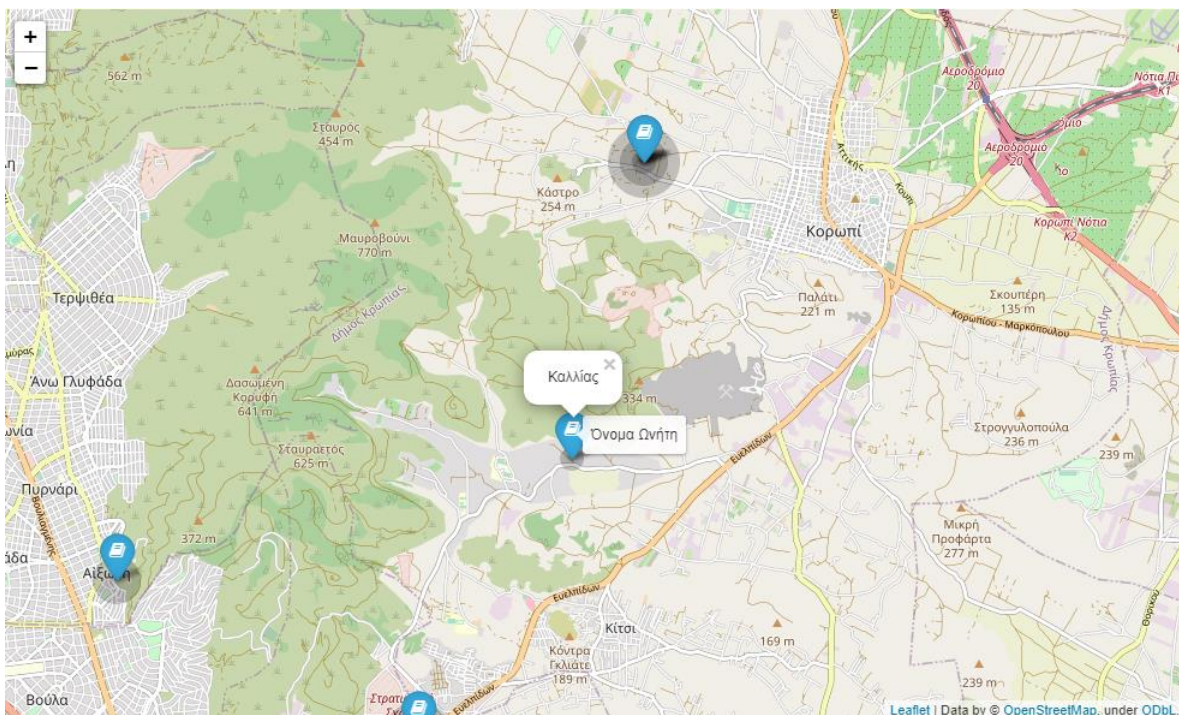
Εικόνα 15: Αποτελέσματα map2(Όνομα Ωνήτη) (1)



Εικόνα 16: Αποτελέσματα map2(Όνομα Ωνήτη) (2)

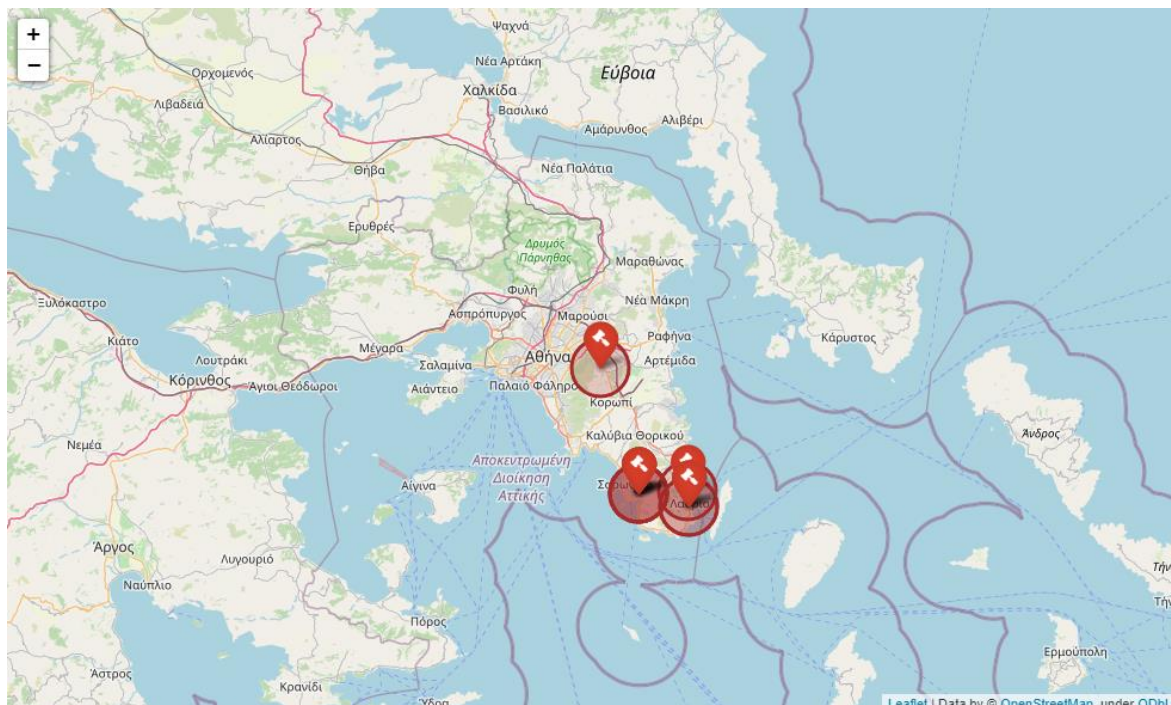


Εικόνα 17: Αποτελέσματα map2(Όνομα Ωνήτη) (3)

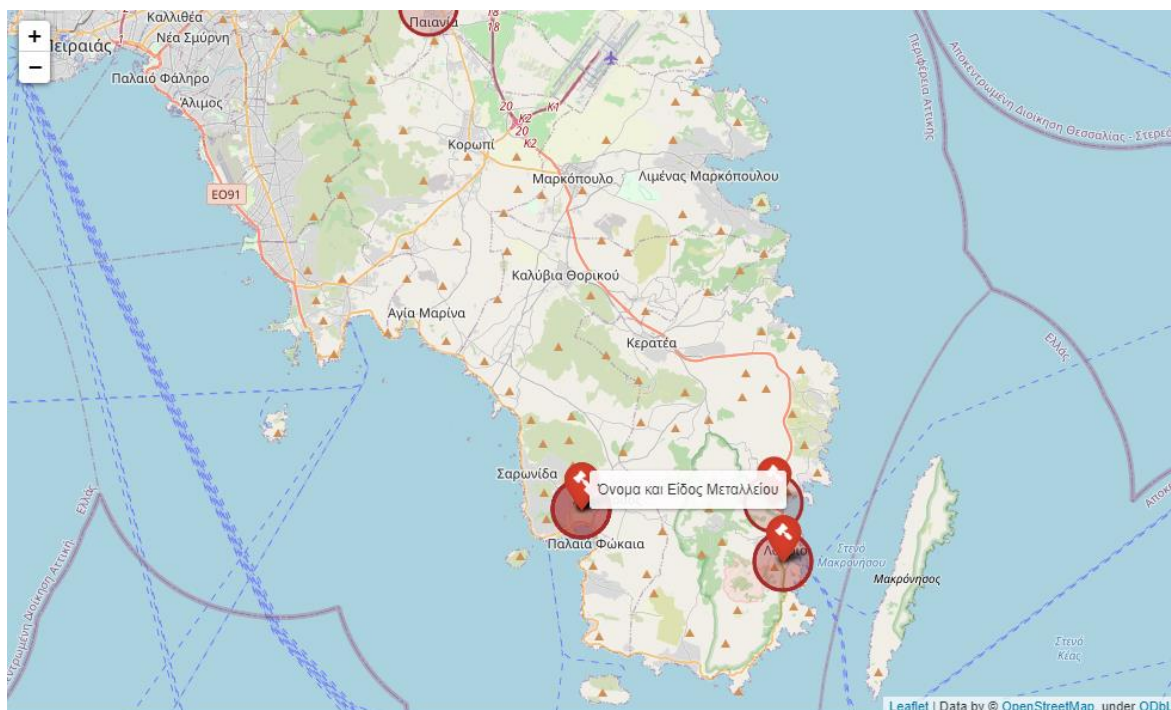


Εικόνα 18: Αποτελέσματα map2(Όνομα Ωνήτη) (4)

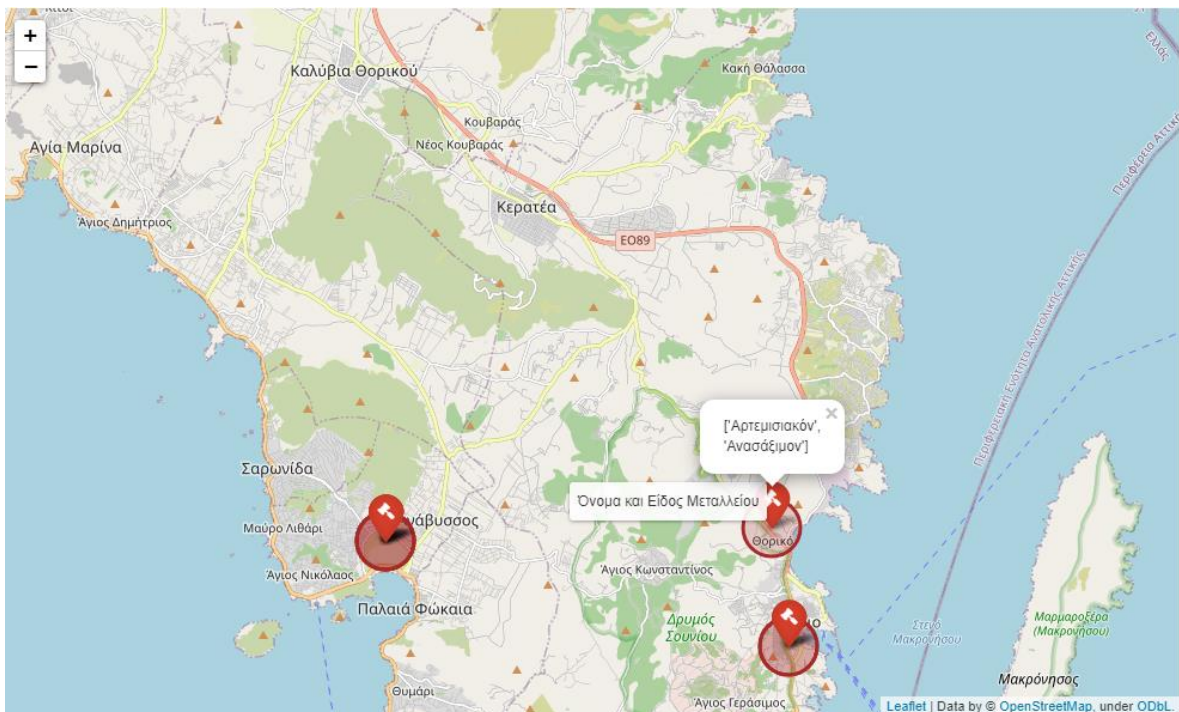
Στις εικόνες από 44 μέχρι 47 απεικονίζετε το map 3 με της πινέζες που του έχουμε ορίσει. Σε αυτόν τον χάρτη όταν το επιλεγθεί μια πινέζα εμφανίζεται το είδος του μεταλλείου στην συγκεκριμένη περιοχή του χάρτη.



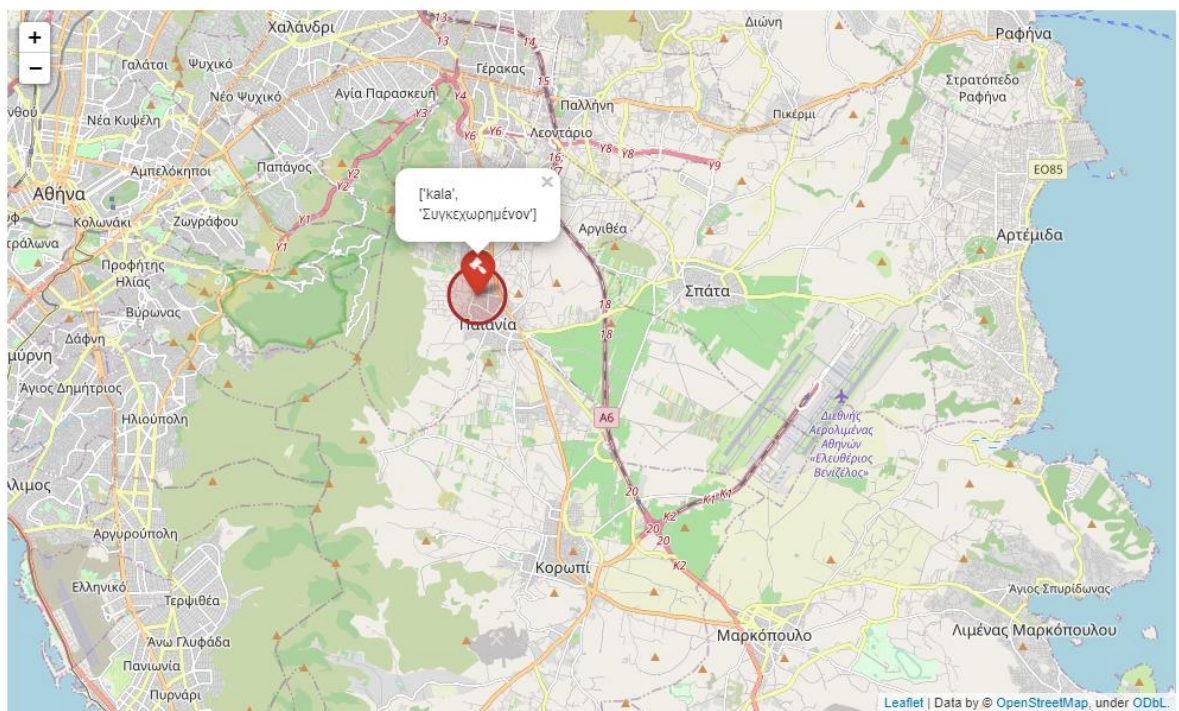
Εικόνα 19: Αποτελέσματα map3(Όνομα και Είδος μεταλλείου) (1)



Εικόνα 20: Αποτελέσματα map3(Όνομα και Είδος μεταλλείου) (2)

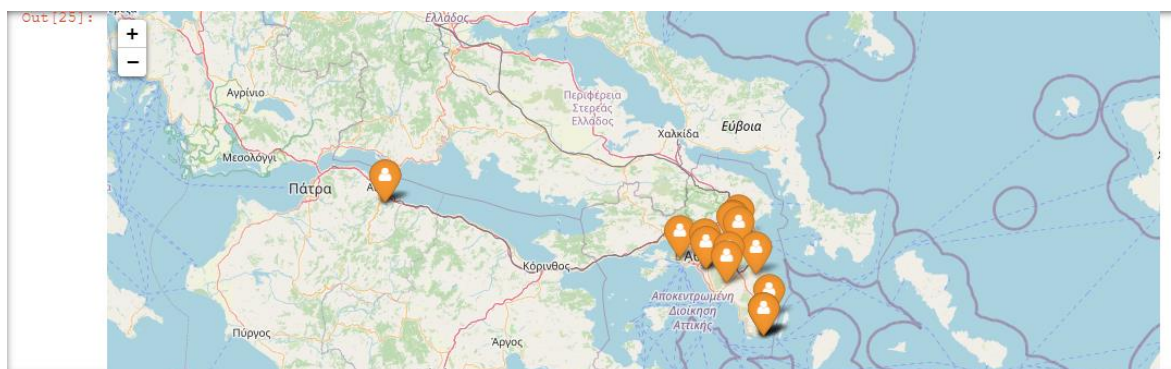


Εικόνα 21: Αποτελέσματα map3 (Όνομα και Είδος μεταλλείου) (3)

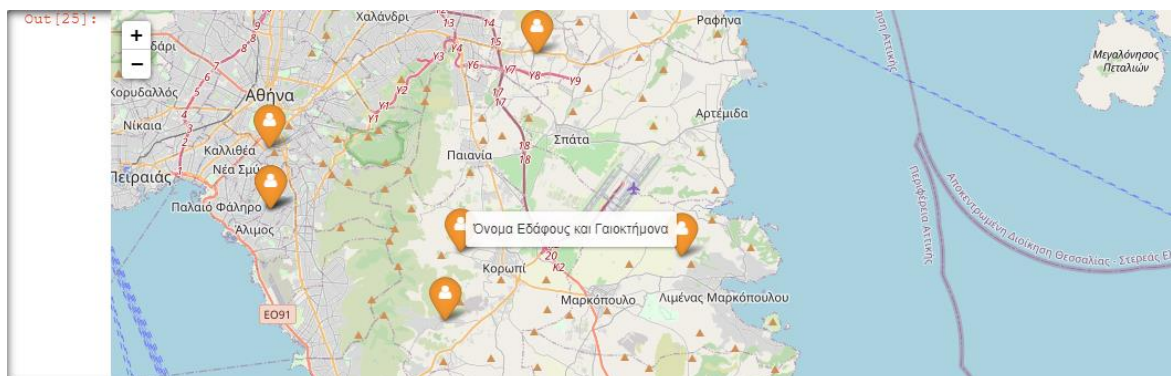


Εικόνα 22: Αποτελέσματα map3 (Όνομα και Είδος μεταλλείου) (4)

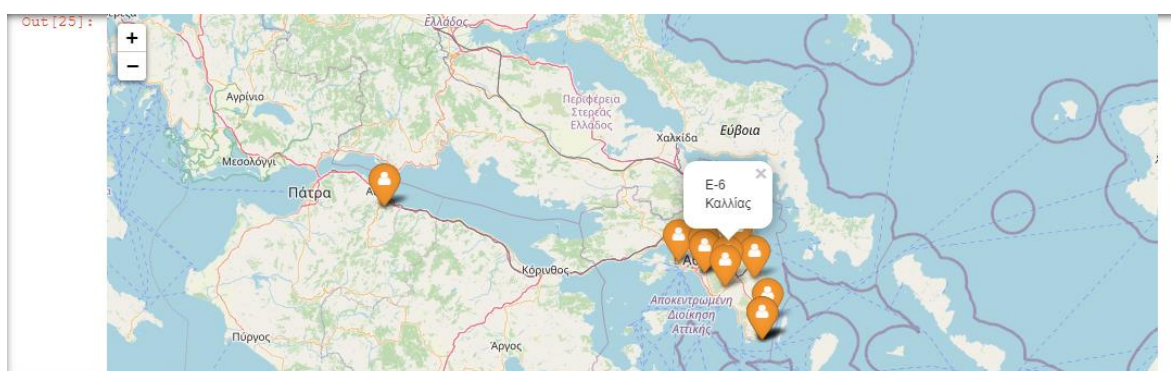
Στις εικόνες από 48 μέχρι 51 απεικονίζετε το map 4 με της πινέζες που του έχουμε ορίσει. Σε αυτόν τον χάρτη όταν το επιλεγθεί μια πινέζα εμφανίζεται το όνομα του εδάφους και του γαιοκτήμονα στην συγκεκριμένη περιοχή του χάρτη.



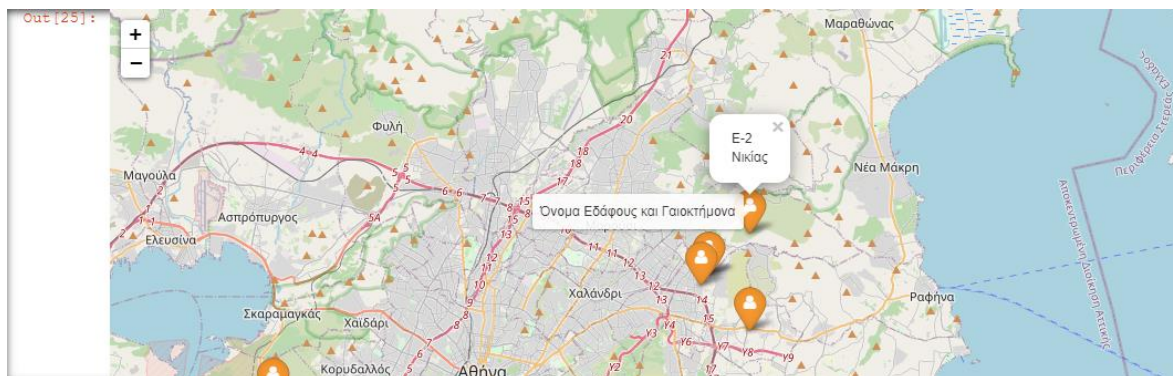
Εικόνα 23: Αποτελέσματα map3(Όνομα Εδάφους και Γαιοκτήμονα) (1)



Εικόνα 24: Αποτελέσματα map3(Όνομα Εδάφους και Γαιοκτήμονα) (2)



Εικόνα 25: Αποτελέσματα map3(Όνομα Εδάφους και Γαιοκτήμονα) (3)



Εικόνα 26: Αποτελέσματα map3 (Όνομα Εδάφους και Γαιοκτήμονα) (4)

Κεφάλαιο 5 Επεκτάσεις και συμπεράσματα

Στην εργασία αυτή, αξιολογήθηκαν οι πιθανές επεκτάσεις και τα συμπεράσματα του έργου συμπεριλαμβανομένης της ενσωμάτωσης δεδομένων για την αρχαία Αθήνα.

Αρχικά, η εργασία βασίζεται σε ένα συγκεκριμένο σύνολο δεδομένων, και θα μπορούσε να επεκταθεί ώστε να συμπεριλάβει επιπλέον πηγές με στόχο να αυξηθεί το βάθος των πληροφοριών που αναπαράγονται στις οπτικοποιήσεις. Για παράδειγμα, ενσωματώνοντας δεδομένα σχετικά με την δημογραφία της αρχαίας Αθήνας, όπως το πληθυσμιακό μέγεθος και τη διανομή, αυτά θα παρέχουν μια πιο συνολική κατανόηση της πόλης και των αρχαιολογικών ανακαλύψεών της.

Ένα ακόμη ιδανικό πρόσθετο θα ήταν η χρονική πληροφορία, η οποία θα επέτρεπε στους χρήστες να παρακολουθούν την κατανομή των αρχαιολογικών ανακαλύψεων με βάση τον χρόνο. Με τον τρόπο αυτό θα παρέχονται πληροφορίες για την ανάπτυξη και την εξέλιξη της αρχαίας Αθήνας, καθώς και για τον τρόπο που διαφορετικά μέρη της πόλης αλλάζουν και χρησιμοποιούνται κατά τα χρόνια.

Μια άλλη ιδέα μπορεί να είναι να συμπεριλαμβάνονται τεχνικές ανάλυσης γεωπλεγματικών δεδομένων για την εξερεύνηση των σχέσεων μεταξύ των τοποθεσιών των αρχαιολογικών ανακαλύψεων και παραγόντων, όπως η τοπογραφία και οι διαδρομές μεταφοράς.

Η εργασία μπορεί, επίσης, να επεκταθεί με τη χρήση επιπλέον πληροφοριών από άλλες πηγές, όπως ιστορικοί χάρτες, ιστορικά περιοδικά, και ανασκαφικά έργα. Η συγκέντρωση των πληροφοριών από αυτές τις πηγές θα μπορούσε να διευκολύνει τη δημιουργία πληρέστερης και πιο ολοκληρωμένης εικόνας του ιστορικού χώρου.

Μια επιπλέον χρήση της εργασίας είναι αυτή για την αναπαράσταση της εξέλιξης των ανασκαφικών έργων στην περιοχή της Αθήνας. Η χρήση των οικονομικών και πολιτιστικών δεδομένων μαζί με τα αποτελέσματα των ανασκαφικών έργων θα μπορούσε να δείξει την εξέλιξη των αρχαιολογικών σταθμών στην περιοχή κατά το χρόνο και να αποκαλύψει περισσότερες πτυχές της ιστορίας.

Επιπροσθέτως, η εργασία μπορεί να συνδεθεί και με άλλα έργα για την ιστορία και την αρχαιολογία, όπως π.χ. με έργα που έχουν γίνει σε άλλες περιοχές της Ελλάδας, ώστε να σχηματιστεί μια πλήρης και συνολική εικόνα της ιστορίας και της αρχαιολογίας της χώρας.

Σημαντική μπορεί να είναι η συμβολή της εργασίας και στην εκπαίδευση των μαθητών στην αρχαιολογία και την ιστορία. Μέσω της χρήσης των ενσωματωμένων εικονικών σχεδιαγραμμάτων, οι μαθητές μπορούν να εντοπίσουν και να συγκρίνουν τις ανακαλύψεις σε πολλές περιοχές της Αθήνας, για να μάθουν περισσότερα για την ιστορία και την αρχαιολογία της περιοχής. Με τη χρήση γεωγραφικών επιστημονικών εργαλείων και μετασχηματισμού των δεδομένων σε εικονικές παραστάσεις, μπορεί να προσφέρεται μια πιο εντυπωσιακή και ενδιαφέρουσα παρουσίαση των αρχαιολογικών ανακαλύψεων στην περιοχή της Αθήνας.

Τέλος, η εργασία συνιστά μια σημαντική βάση για μελλοντικές ερευνητικές δραστηριότητες, όπως η εκτέλεση περαιτέρω έρευνας στην αρχαιολογία και την ιστορία της Αθήνας, ή τη δημιουργία παρόμοιων εργασιών για άλλες αρχαίες πόλεις ή περιοχές.

Παράλληλα, είναι εφικτό να συνδυάσουμε τα δεδομένα μας με άλλες πηγές δεδομένων, όπως π.χ. στατιστικά στοιχεία για την οικονομία και την πολιτιστική ζωή της αρχαίας Αθήνας, ώστε να αποκτήσουμε μια πιο συνολική εικόνα της πόλης και των ανακαλύψεών της.

Παράτημα Α΄

Οι παρακάτω πίνακες προέρχονται από την πτυχιακή : *Επιγραφές πολιτών και μεταλλείων: αποθήκευση και σημασιολογική απεικόνιση τους στο διαδίκτυο του Κωνσταντής, Χ. (2022).*

dimoi			
Στήλη	Τύπος Δεδομένων	Πρόσθετα	Περιγραφή Στήλης
Id Δήμου	integer	Primary Key, Auto Increment	Μοναδικό αναγνωριστικό κάθε Δήμου.
Όνομα Δήμου	varchar		Όνομα Δήμου
Όνομα Δημοτικού	varchar		Κάθε δημοτικό αποτελούσε μια ευρύτερη περιοχή από τον Δήμο, όπως οι περιφερειακές ενότητες σήμερα.
Συντομογραφία	varchar		Συντομογραφία του Δήμου
Τριττύα	varchar		Διοικητική υποδιαίρεση της φυλής
Φυλή	varchar		Ονομασία της φυλής του δήμου
Αριθμός Βουλευτών	varchar		Πλήθος βουλευτών
Παρατηρήσεις	varchar		Πεδίο εισαγωγής επιπλέον παρατηρήσεων/ χαρακτηριστικών
B-N-A-Δ Id	integer	Foreign Key Unique	Μοναδικό αναγνωριστικό κάθε δήμου.

metalleia			
Στήλη	Τύπος Δεδομένων	Πρόσθετα	Περιγραφή Στήλης
Id Μεταλλείου	integer	Primary Key, Auto Increment	Μοναδικό αναγνωριστικό κάθε μεταλλείου
Όνομα Μεταλλείου	varchar		Όνομα μεταλλείου
Id Ωνητή	integer	Foreign Key	Μοναδικό αναγνωριστικό ωνήτη του μεταλλείου
Id Δήμου	integer	Foreign Key	Μοναδικό αναγνωριστικό δήμου που ανήκει το μεταλλείο
Είδος Μεταλλείου	varchar	Foreign Key	Υπάρχουν συγκεκριμένα είδη μεταλλείων. Κάθε μεταλλείο ανήκει σε ένα είδος.
Id Επιγραφής	integer	Foreign Key	Μοναδικό αναγνωριστικό της επιγραφής στην οποία αναφέρεται το μεταλλείο
Παρατηρήσεις	varchar		Πεδίο εισαγωγής επιπλέον παρατηρήσεων/ χαρακτηριστικών
B-N-A-Δ Id	integer	Foreign Key Unique	Μοναδικό αναγνωριστικό κάθε μεταλλείου. Ο λόγος ύπαρξης του γίνεται κατανοητός στην περιγραφή του πίνακα 'Λίστα B-N-A-Δ' και το κεφάλαιο 'Δυσκολίες-Επίλυση'.

onitis			
Στήλη	Τύπος Δεδομένων	Πρόσθετα	Περιγραφή Στήλης
Id Ωνητή	integer	Primary Key, Auto Increment	Μοναδικό αναγνωριστικό κάθε ωνητή. Ωνητής σημαίνει μισθωτής στην αρχαία Ελληνική γλώσσα.
Όνομα Ωνητή	varchar		Όνομα ωνητή
Πατρώνυμο Ωνητή	varchar		Πατρώνυμο Ωνητή
Id Δήμου	integer	Foreign Key	Μοναδικό αναγνωριστικό δήμου που ανήκει ο ωνητής
Χρονολόγηση	varchar		Αναφέρεται στην εποχή ή τη χρονολογία κατά την οποία έζησε ή μίσθωσε ο ωνητής κάποιο μεταλλείο
Id Επιγραφής	integer	Foreign Key	Μοναδικό αναγνωριστικό της επιγραφής στην οποία αναφέρεται ο ωνητής
Παρατηρήσεις	varchar		Πεδίο εισαγωγής επιπλέον παρατηρήσεων/ χαρακτηριστικών

eidh metalleiwn			
Στήλη	Τύπος Δεδομένων	Πρόσθετα	Περιγραφή Στήλης
Id Είδους Μεταλλείου	int	Primary Key	Μοναδικό αναγνωριστικό κάθε είδους μεταλλείου
Είδος Μεταλλείου	varchar		Το είδος του μεταλλείου. Πχ 'Ανασάξιμον'.

edafh			
Στήλη	Τύπος Δεδομένων	Πρόσθετα	Περιγραφή Στήλης
Id Εδάφους	integer	Primary Key, Auto Increment	Μοναδικό αναγνωριστικό κάθε εδάφους
Όνομα Εδάφους-Γαιοκτήμονα	varchar		Όνομα γαιοκτήμονα εδάφους
Id Τοπωνυμίου	integer	Foreign Key	Μοναδικό αναγνωριστικό του τοπωνυμίου που ανήκει κάθε έδαφος
Παρατηρήσεις	varchar		Πεδίο εισαγωγής επιπλέον παρατηρήσεων/ χαρακτηριστικών

b-n-a-d			
Στήλη	Τύπος Δεδομένων	Πρόσθετα	Περιγραφή Στήλης
B-N-A-Δ Id	integer	Primary Key, Auto Increment	Μοναδικό αναγνωριστικό κάθε στοιχείου που βρίσκεται στην λίστα.
B-N-A-Δ Όνομα	varchar		Όνομα στοιχείου
Id Δήμου B-N-A-Δ	integer	Foreign Key	Μοναδικό αναγνωριστικό δήμου στον οποίο ανήκει το στοιχείο
Όνομα Δήμου B-N-A-Δ	varchar		Ονομασία του δήμου

Παράτημα Β'

Σε αυτό το παράρτημα θα παραθέσω τον κώδικα που έγραψα με τα απαραίτητα σχόλια για να είναι κατανοητός.

#Δημιουργία του κοννέκτορα για να συνδεθούμε με τον Server και να διαβάσουμε την database.

```
conn=pymysql.connect(user='root',password='',host='localhost',database='ptyxiakh')
```

```
cursor=conn.cursor()
```

#Αποθηκεύουμε σε ένα dataframe το αρχείο με της συντεταγμένες.

```
data = pd.read_csv('/Users/xeno/Desktop/ptyxiakh/Lon_Lat.csv', delimiter=';')
```

```
pd.DataFrame(data)
```

#Φτιάχνουμε τον κέρσορα και κάνουμε fetchcall.

#Επειτα Βάζουμε το query σε ένα dataframe (dictionary).

```
query='select DISTINCT a.`Όνομα Δήμου`,b.`Όνομα Μεταλλείου`,c.`Είδος Μεταλλείου`,a.`Id Δήμου`,c.`Id Είδους Μεταλλείου` from dimoi a left join metalleia b on a.`Id Δήμου`=b.`Id Δήμου` left join `eidh metalleiwn` c on b.`Id Είδους Μεταλλείου`=c.`Id Είδους Μεταλλείου` where c.`Id Είδους Μεταλλείου` is not null ;'
```

```
r=cursor.
```

```
execute(query)
```

```
dataA=cursor.fetchall()
```

```
counter=0
```

```
dictionary={"Όνομα Δήμου":[],"Όνομα Μεταλλείου":[],"Είδος Μεταλλείου":[],"Id Δήμου":[],"Id Είδους Μεταλλείου":[]}
```

```
for d in dataA:
```

```
    dictionary["Όνομα Δήμου"].append(d[0])
```

```
    dictionary["Όνομα Μεταλλείου"].append(d[1])
```

```
    dictionary["Είδος Μεταλλείου"].append(d[2])
```

```
    dictionary["Id Δήμου"].append(d[3])
```

```
    dictionary["Id Είδους Μεταλλείου"].append(d[4])
```

```
    counter+=1
```

```
pd.DataFrame(dictionary)
```

#Ενωση αρχικού dataframe (με τις συντεταγμένες) με το dataframe που φτιάξαμε στο προηγούμενο κομμάτι του κώδικα.

```

df_cd = pd.merge(pd.DataFrame(dictionary), pd.DataFrame(data),
how='left', left_on = 'Όνομα Δήμου', right_on = 'onoma')
df_cd

#
queryb='select Distinct a.* from onitis a left join `b-n-a-d` b on b.`Id Δήμου`
b-n-a-d`= a.`Id Δήμου`;'
r=cursor.execute(queryb)
datab=cursor.fetchall()

counter=0
dictionaryA={"Id Ωνήτη":[ ],"Όνομα Ωνήτη":[ ],"Πατρώνυμο Ωνήτη":[
 ],"Id Δήμου":[ ],"Id Επιγραφής":[ ]}
for d in datab:
    dictionaryA["Id Ωνήτη"].append(d[0])
    dictionaryA["Όνομα Ωνήτη"].append(d[1])
    dictionaryA["Πατρώνυμο Ωνήτη"].append(d[2])
    dictionaryA["Id Δήμου"].append(d[3])
    dictionaryA["Id Επιγραφής"].append(d[4])
    counter+=1

pd.DataFrame(dictionaryA)

#
df_oniti = pd.merge(pd.DataFrame(dictionaryA), pd.DataFrame(data),
how='left', left_on = 'Id Δήμου', right_on = 'id_dimou')
Df_oniti

#
queryc='select distinct a.`Όνομα Εδάφους - Γαιοκτήμονας`, a.`Id Δήμου`
from edafh a left join `b-n-a-d` b on b.`Id Δήμου` b-n-a-d`= a.`Id Δήμου`
where a.`Id Δήμου` is not null ;'

r=cursor.execute(queryc)
datac=cursor.fetchall()

counter=0
dictionaryB={"Όνομα Εδάφους - Γαιοκτήμονας":[ ],"Id Δήμου":[ ]}

for d in datac:
    dictionaryB["Όνομα Εδάφους - Γαιοκτήμονας"].append(d[0])
    dictionaryB["Id Δήμου"].append(d[1])

    counter+=1

```

```
pd.DataFrame(dictionaryB)
```

```
#  
df_edafh = pd.merge(pd.DataFrame(dictionaryB), pd.DataFrame(data),  
how='left', left_on = 'Id Δήμου', right_on = 'id_dimou')  
df_edafh
```

#Γκρουπάρισμα τις κολώνες lat,lon μαζί με τις υπόλοιπες κολώνες που θα χρησιμοποιήσουμε από την ένωση των dataframe που δημιουργήσαμε νωρίτερα και δίνουμε όνομα στο γκρουπάρισμα μετατρέποντας το σε dataframe (πχ το start_location έχει πληροφορία Ονόματα Δήμων)

```
start_location = data.groupby(['lon','lat','onoma']).size().reset_index()
```

#Στην συνέχεια δημιουργούμε καινούργια κολώνα με όνομα map με την χρήσιμη πληροφορία(Στο metalleia βάζουμε “Είδος Μεταλλείου” η οποία είναι κολώνα από το dataframe dictionary)

```
start_location['Map'] = start_location['onoma'].astype(str)
```

```
metalleia=df_cd.groupby(['lon','lat','Όνομα Μεταλλείου','Είδος Μεταλλείου']).size().reset_index()  
metalleia['Map']=metalleia['Είδος Μεταλλείου'].astype(str)
```

```
onitis=df_oniti.groupby(['lon','lat','Όνομα Ωνήτη']).size().reset_index()  
onitis['Map']=onitis['Όνομα Ωνήτη'].astype(str)
```

```
edafh=df_edafh.groupby(['lon','lat','Όνομα Εδάφους - Γαιοκτήμονας']).size().reset_index()  
edafh['Map']=edafh['Όνομα Εδάφους - Γαιοκτήμονας'].astype(str)
```

#Μετατρέπουμε το dataframe σε NumPy array

```
start_location =np.asarray(start_location)
```

```
metalleia=np.asarray(metalleia)
```

```
onitis=np.asarray(onitis)
```

```
edafh=np.asarray(edafh)
```

#Δημιουργούμε έναν χάρτη εδώ ονομάζετε map1 και χρησιμοποιούμε την βιβλιοθήκη folium.Στο location θέτουμε τις συντεταγμένες στις οποίες θα

ξεκινήσει ο χάρτης μας και στο zoom_start επιλέγουμε το πόσο zoom θέλουμε να είναι ο χάρτης.

```
map1 = folium.Map(  
    location=[np.mean(start_location[:,0]), np.mean(start_location[:,1])],  
    zoom_start=10
```

```
)
```

#Για κάθε γραμμή με δεδομένα που βρίσκεις ζητάμε να τρέξει το αντίστοιχο folium.

```
for i in range(start_location.shape[0]):
```

#Στο CircleMarker δημιουργούμε ένα radius για το location που θέτουμε.

```
    folium.CircleMarker(  
        location=[start_location[i,0], start_location[i,1]],  
        radius=start_location[i,1]/2,  
        color = 'red',  
        fill=True).add_to(map1)
```

#Στο folium.Marker δημιουργούμε την πινέζα μας

```
    folium.Marker(  
        location=[start_location[i,0], start_location[i,1]],  
        popup = start_location[i,4],  
        tooltip='Όνομα Δήμου',  
        icon=folium.Icon(color='green',icon='info-sign')  
    ).add_to(map1)
```

```
map1
```

#Αντίστοιχα με τον παραπάνω χάρτη είναι και αυτός

```
map2 = folium.Map(  
  
    location=[np.mean(onitis[:,0]), np.mean(onitis[:,1])],  
    zoom_start=10
```

```
)
```

```
for i in range(onitis.shape[0]):
```

```
    folium.CircleMarker(  
        location=[onitis[i,0], onitis[i,1]],  
        radius=onitis[i,3]*10,  
        color = 'light',  
        fill=True).add_to(map2)  
    folium.Marker(  
        location=[onitis[i,0], onitis[i,1]],  
        popup = onitis[i,4],  
        tooltip='Όνομα Ωνήτη',  
        icon=folium.Icon(icon='fa-book', prefix='fa', color='blue')  
    ).add_to(map2)
```

map2

```
map3 = folium.Map(

    location=[np.mean(metalleia[:,0]), np.mean(metalleia[:,1])],
    zoom_start=10
)

for i in range(metalleia.shape[0]):
    folium.CircleMarker(
        location=[metalleia[i,0], metalleia[i,1]],
        radius=metalleia[i,1],
        color = 'brown',
        fill=True).add_to(map3)
    folium.Marker(
        location=[metalleia[i,0], metalleia[i,1]],
        popup = [metalleia[i,2], metalleia[i,3]],
        tooltip="Όνομα και Είδος Μεταλλείου",
        icon=folium.Icon(icon='fa-gavel', prefix='fa', color='red')
    ).add_to(map3)
map3

map4 = folium.Map(

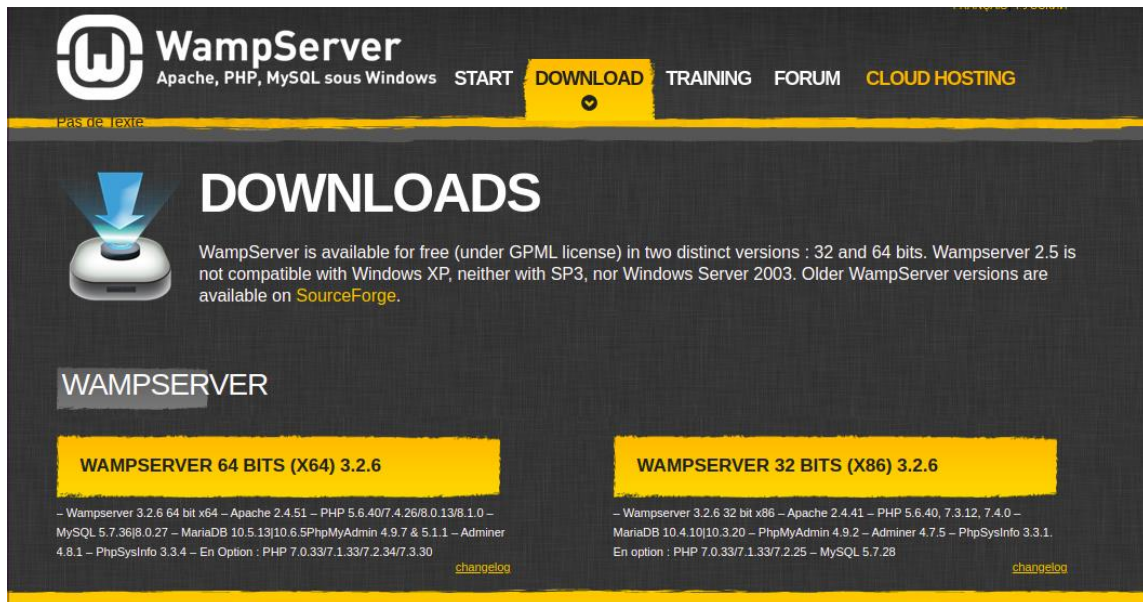
    location=[np.mean(edafh[:,0]), np.mean(edafh[:,1])],
    zoom_start=10
)

for i in range(edafh.shape[0]):

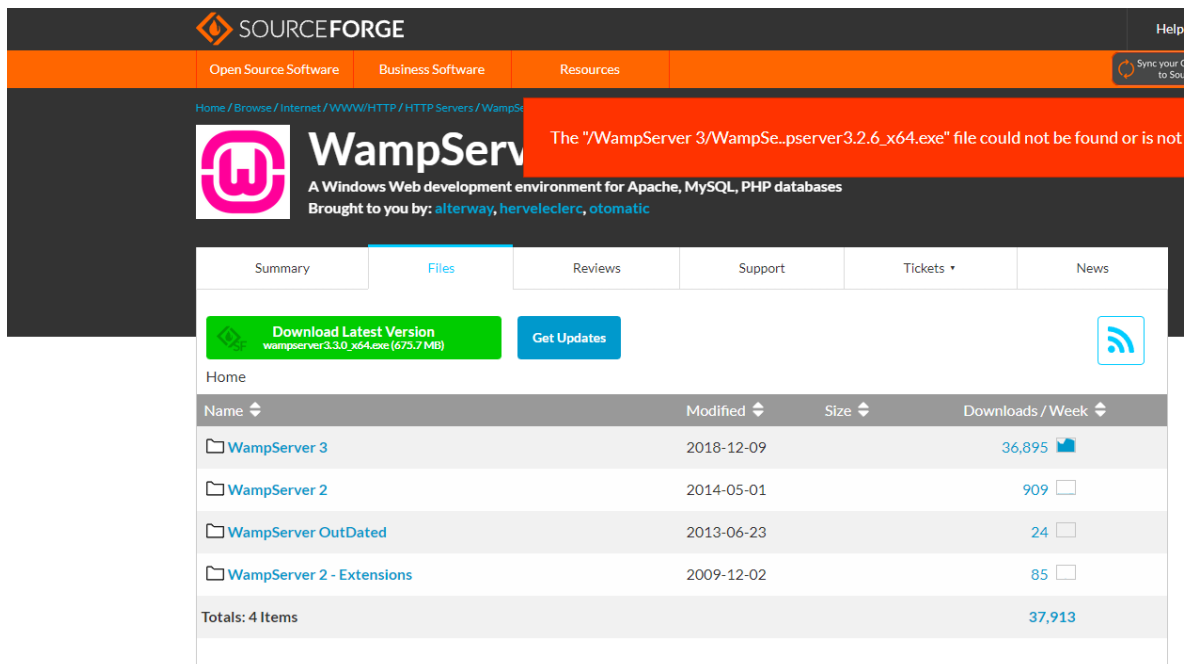
    folium.Marker(
        location=[edafh[i,0], edafh[i,1]],
        popup = edafh[i,2],
        tooltip="Όνομα Εδάφους και Γαιοκτήμονα",
        icon=folium.Icon(icon='fa-user', prefix='fa', color='orange')
    ).add_to(map4)
map4
```

Παράτημα Γ'

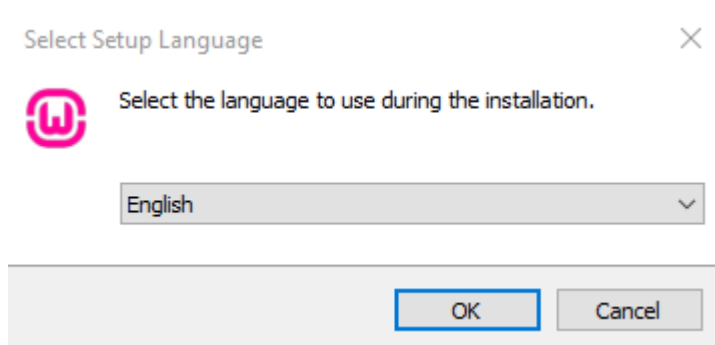
1)Wamp



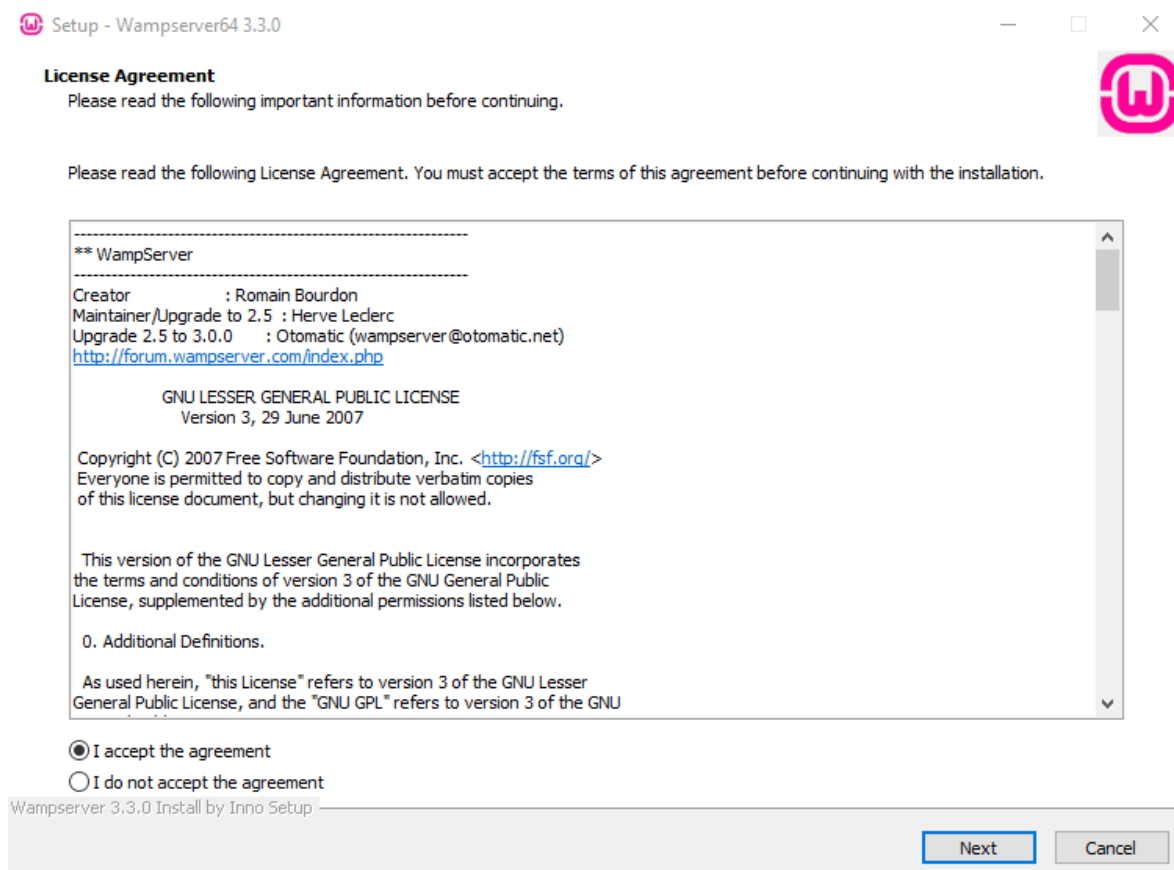
Εικόνα 27: Λήψη WAMP Server (1)



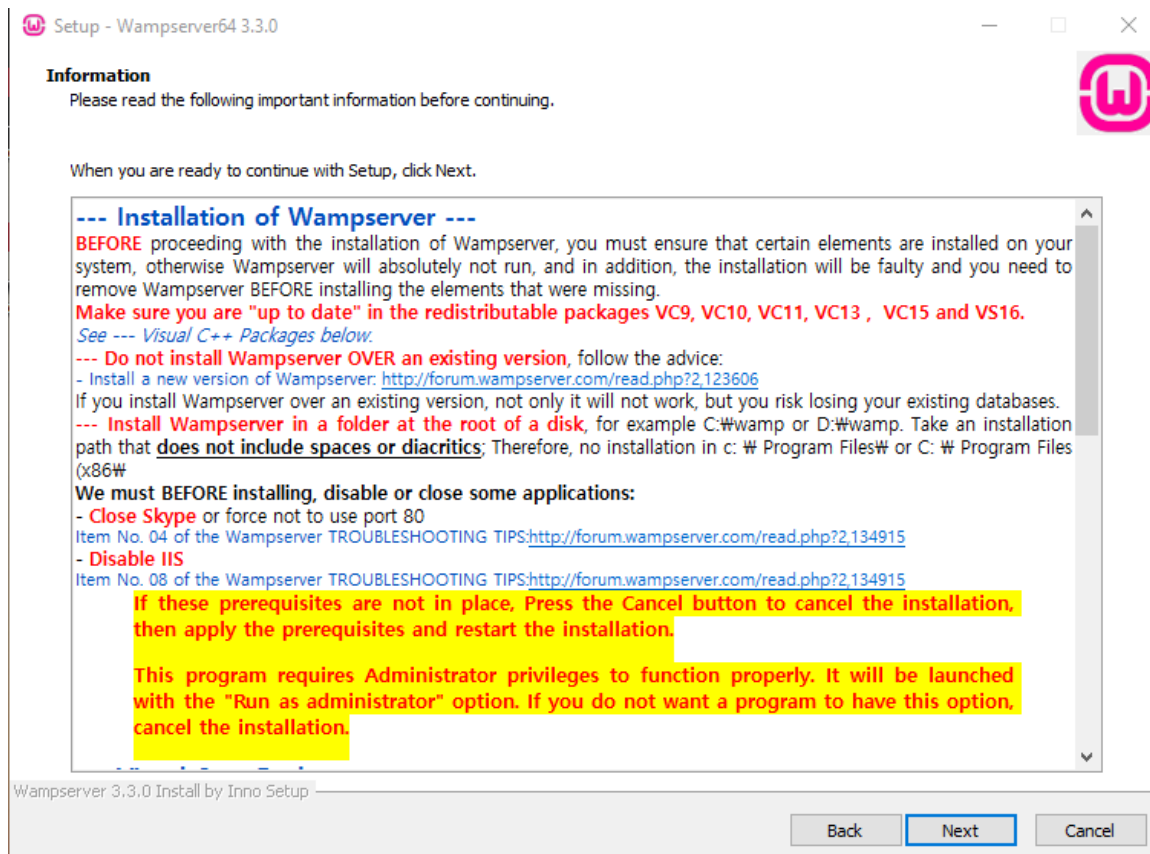
Εικόνα 28: Λήψη WAMP Server (2)



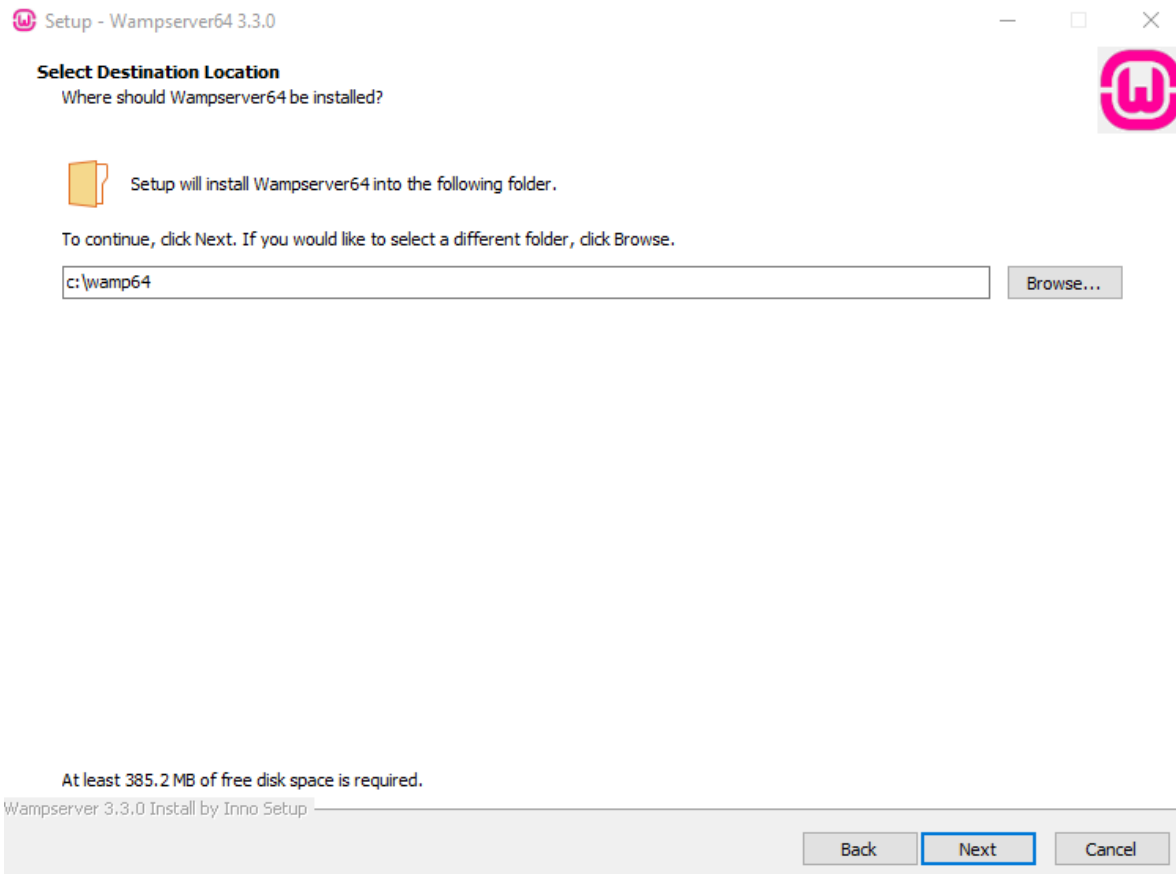
Εικόνα 29:Επιλογή Γλώσσας Wamp Server



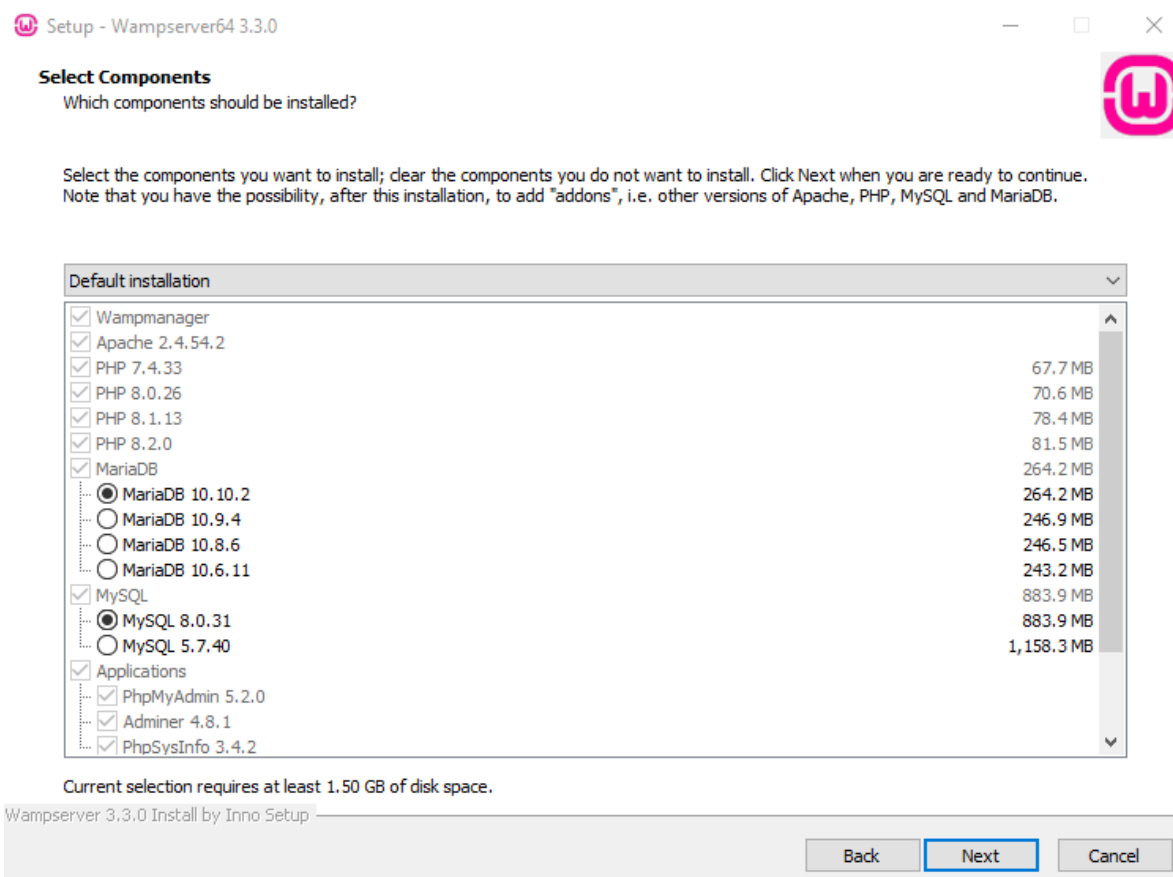
Εικόνα 30:Εγκατάσταση Wamp Server (1)



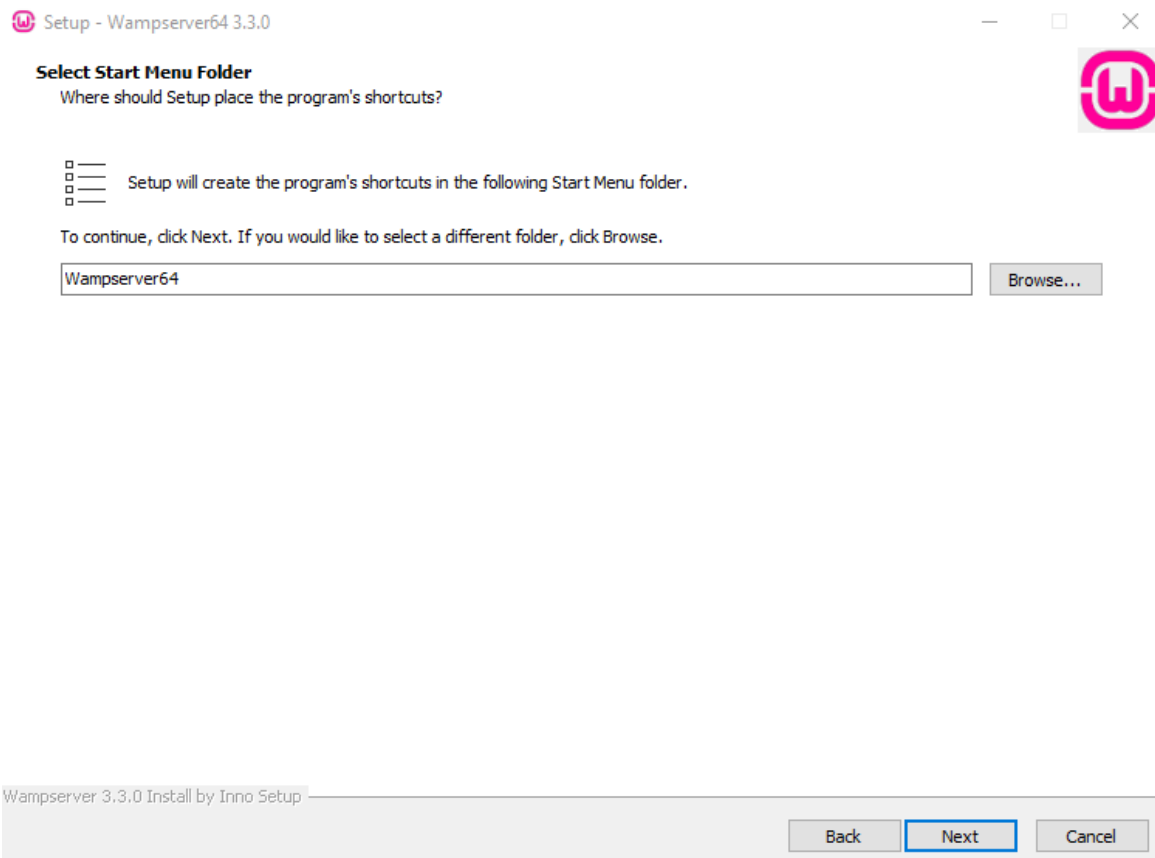
Εικόνα 31:Εγκατάσταση Wamp Server (2)



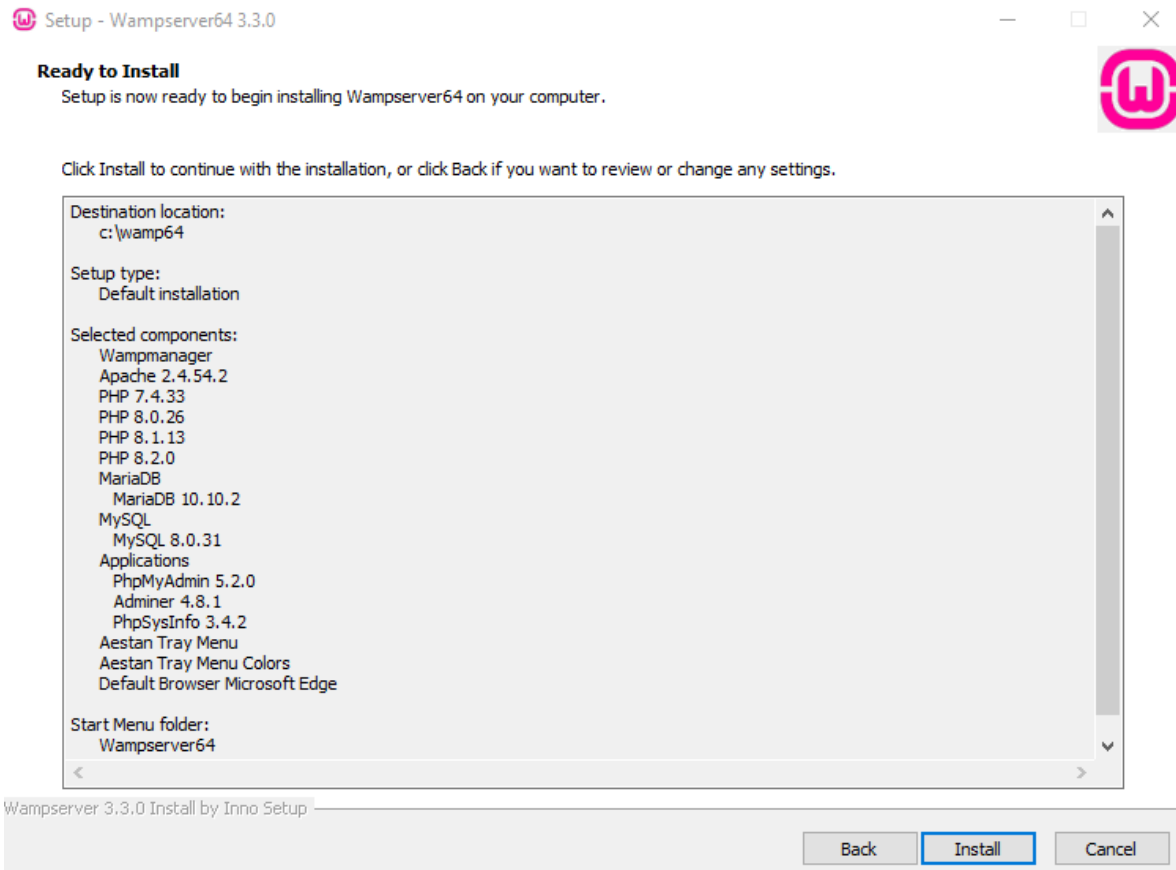
Εικόνα 32: Εγκατάσταση Wamp Server (3)



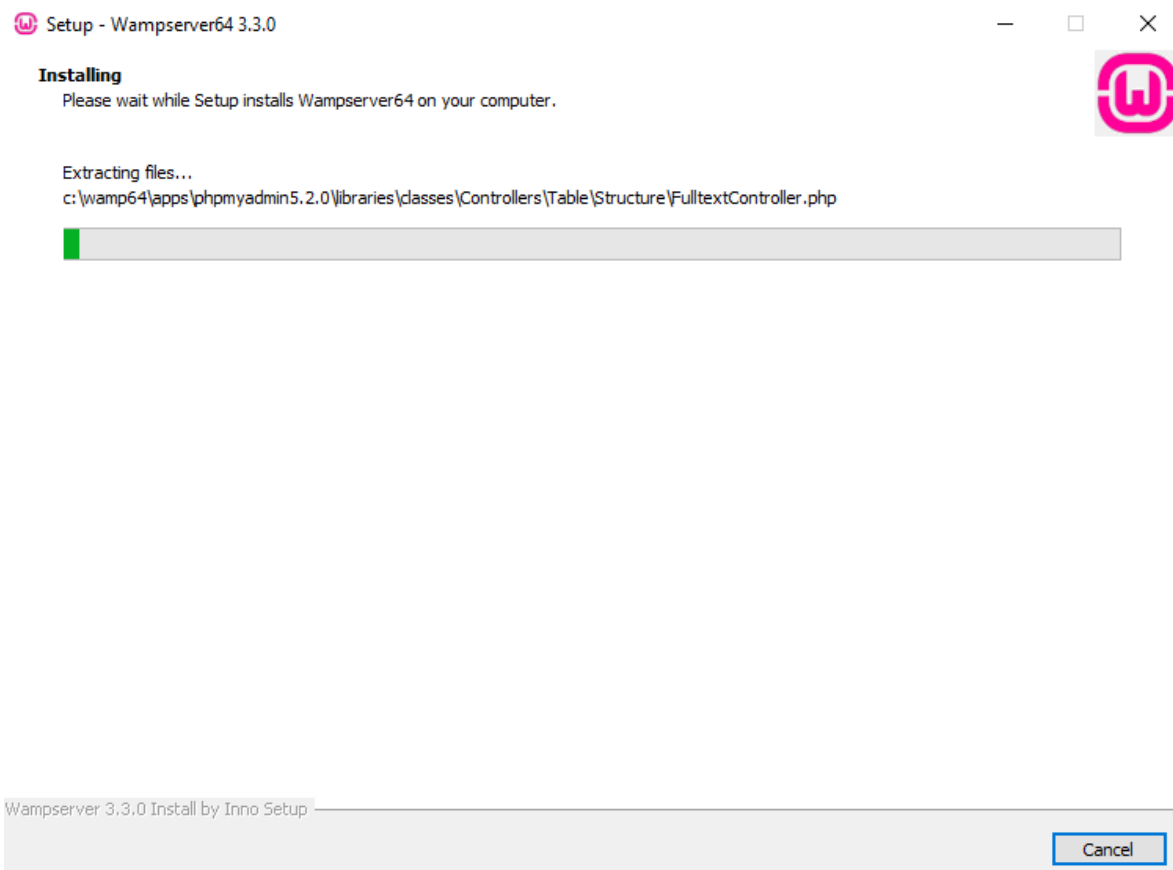
Εικόνα 33:Εγκατάσταση Wamp Server (4)



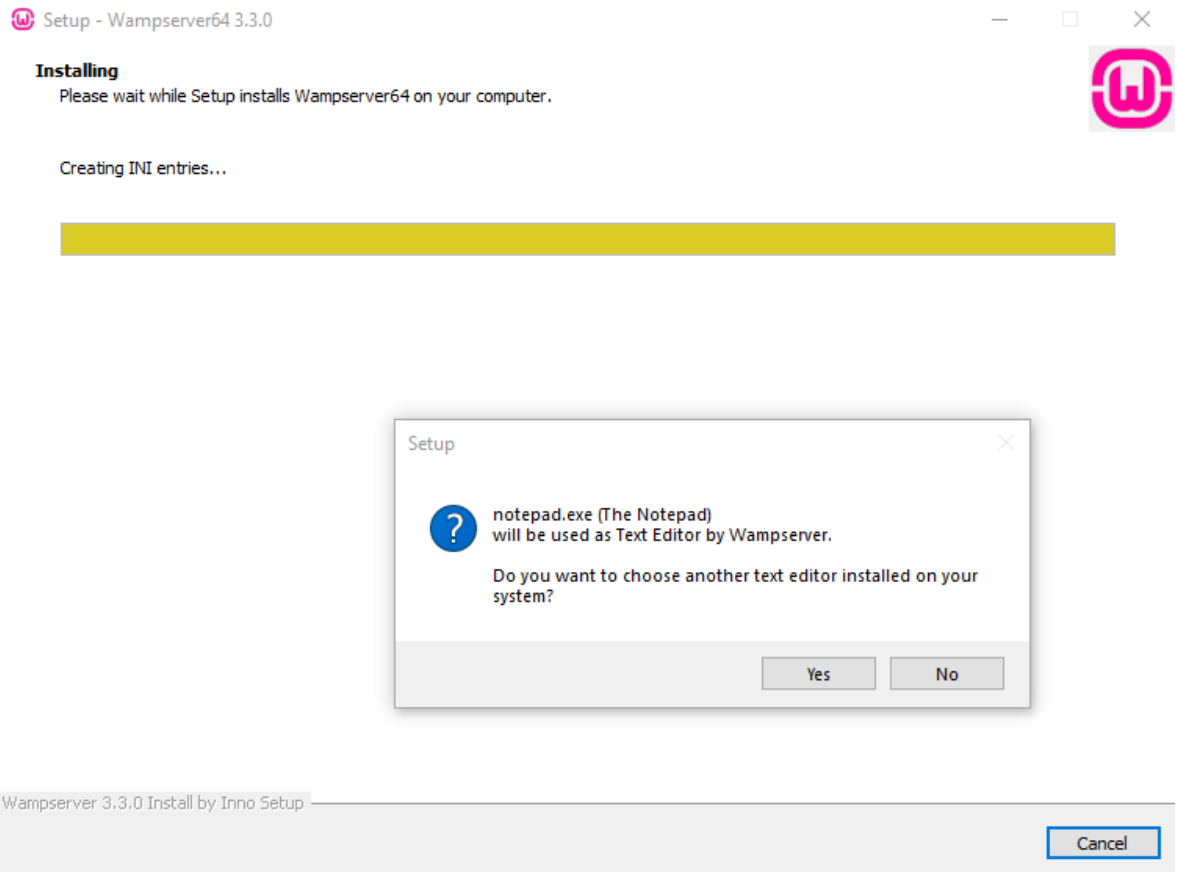
Εικόνα 34:Εγκατάσταση Wamp Server (5)



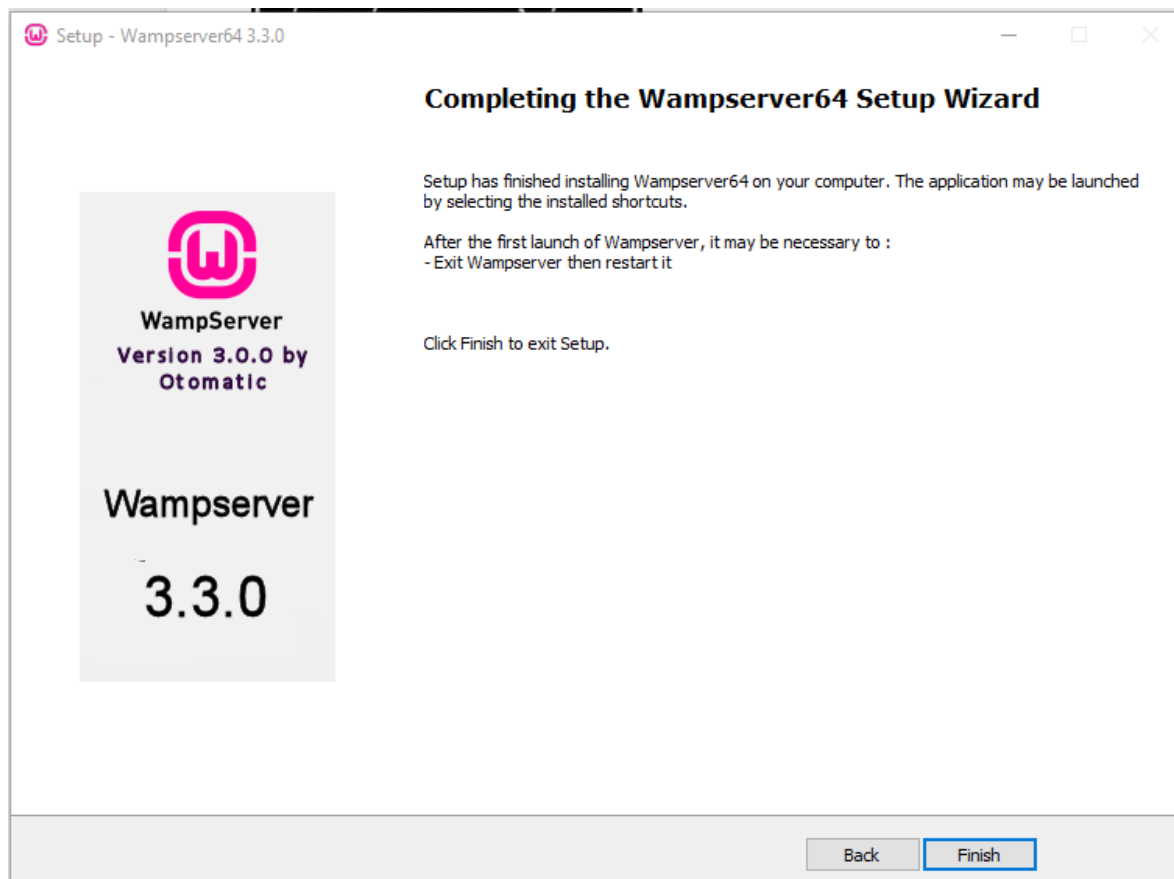
Εικόνα 35:Εγκατάσταση Wamp Server (6)



Εικόνα 36: Εγκατάσταση Wamp Server (7)



Εικόνα 37:Εγκατάσταση Wamp Server (8)



Εικόνα 38: Ολοκλήρωση εγκατάστασης Wamp Server

2)phpMyAdmin.



Καλωσήρθατε στο phpMyAdmin

Γλώσσα (Language)

Ελληνικά - Greek

Σύνδεση

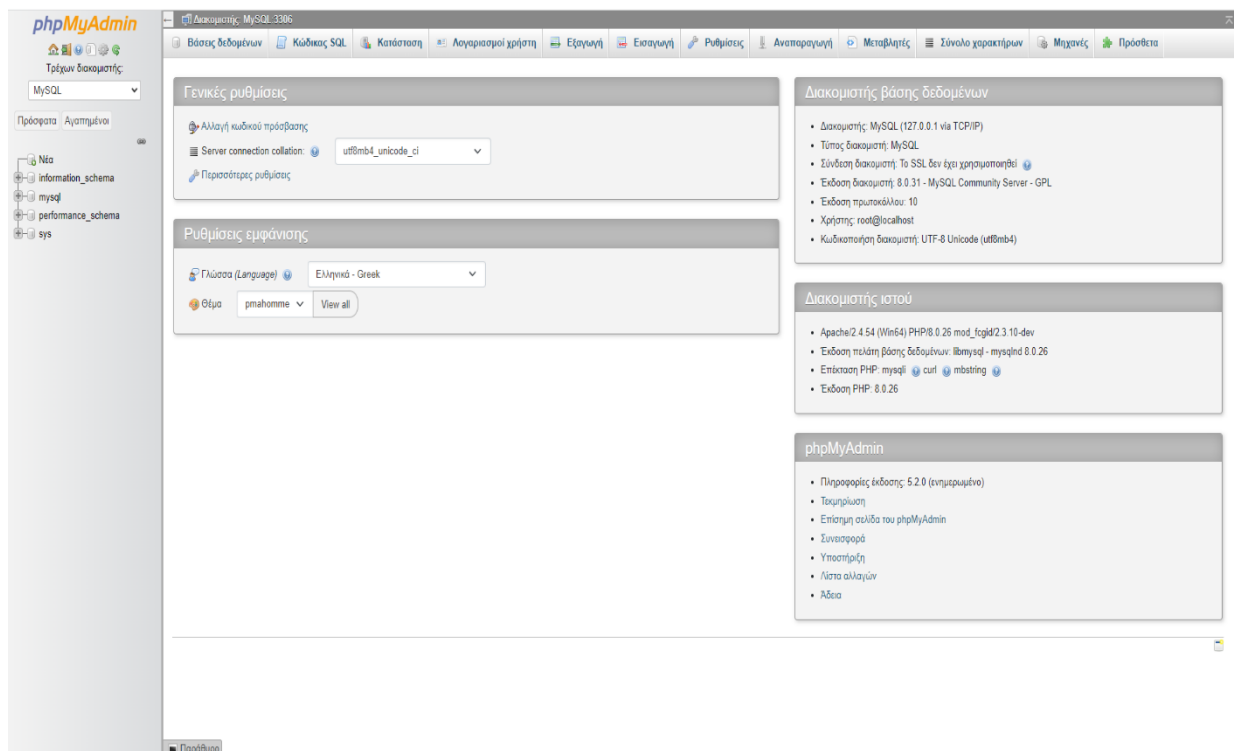
Όνομα χρήστη: root

Κωδικός πρόσβασης:

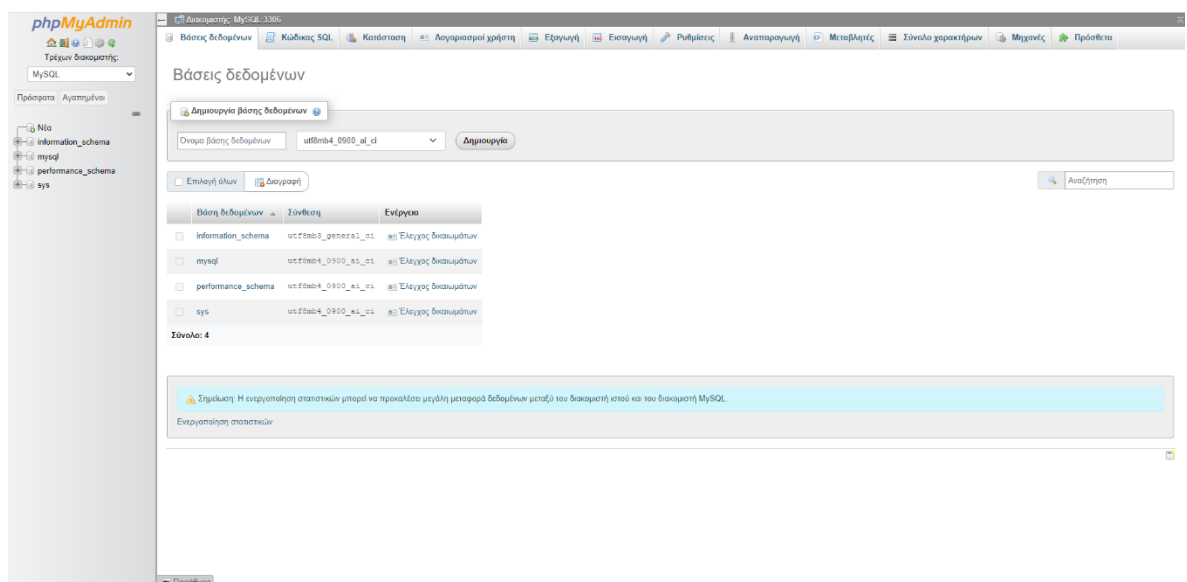
Server choice: MySQL

Σύνδεση

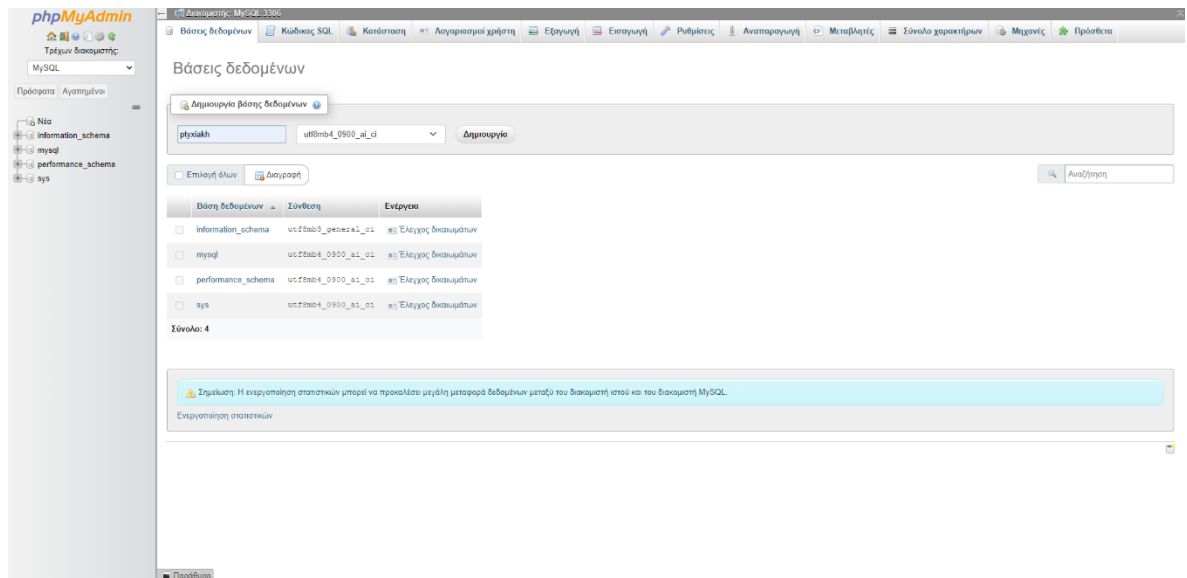
Εικόνα 39: Σύνδεση στην phpMyAdmin



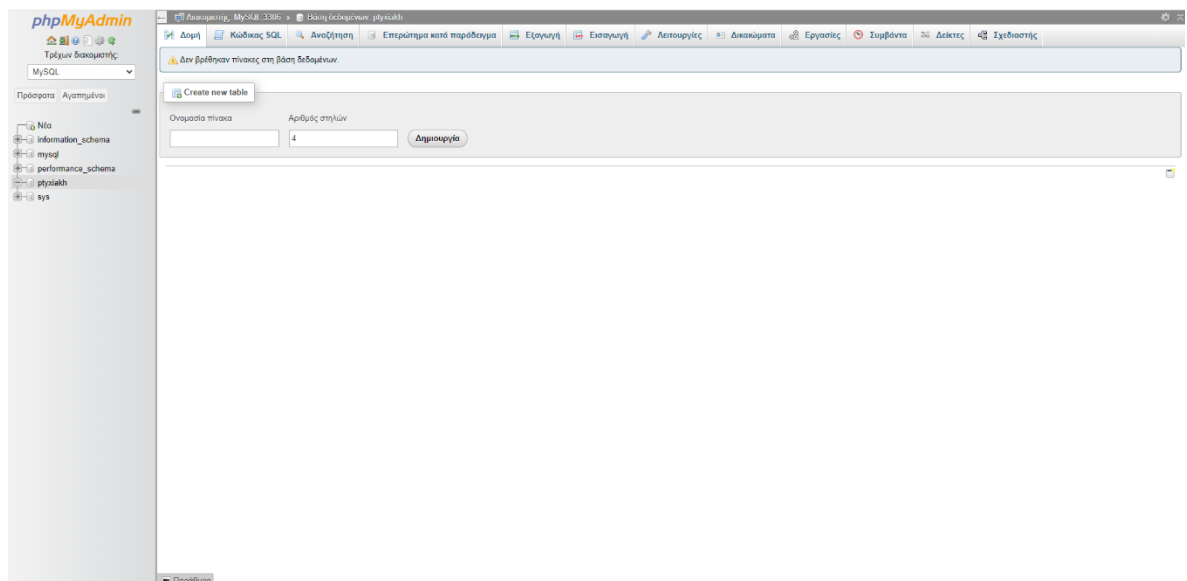
Εικόνα 40: Δημιουργία κενής βάσης (1)



Εικόνα 41: Δημιουργία κενής βάσης (2)



Εικόνα 42: Δημιουργία κενής βάσης (3)



Εικόνα 43: Import της βάσης στην κενή βάση

Πίνακας	Ενέργεια	Εγγραφή	Τύπος	Σύνθεση	Μέγεθος	Περίοδος
aktes	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	3	InnoDB greek_general_ci 45,0 KB
b.a.a.d	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	176	InnoDB greek_general_ci 32,0 KB
dimoi	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	43	InnoDB greek_general_ci 32,0 KB
edafis	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	36	InnoDB greek_general_ci 32,0 KB
eggrafes	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	43	InnoDB greek_general_ci 32,0 KB
eidh metallon	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	5	InnoDB greek_general_ci 16,0 KB
eidh takinomis	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	6	InnoDB greek_general_ci 16,0 KB
ergastria	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	59	InnoDB greek_general_ci 32,0 KB
exei_sthn_anatoli	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	14	InnoDB greek_general_ci 45,0 KB
exei_sthn_dushi	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	43	InnoDB greek_general_ci 32,0 KB
exei_sthn_borra	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	43	InnoDB greek_general_ci 32,0 KB
iera	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	43	InnoDB greek_general_ci 32,0 KB
klittees	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	50	InnoDB greek_general_ci 32,0 KB
loko	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	43	InnoDB greek_general_ci 32,0 KB
metallia	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	1	InnoDB greek_general_ci 45,0 KB
odoi	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	10	InnoDB greek_general_ci 32,0 KB
onidis	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	47	InnoDB greek_general_ci 32,0 KB
oroi	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	2	InnoDB greek_general_ci 45,0 KB
pediades	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	0	InnoDB greek_general_ci 45,0 KB
xoria	Περιήγηση	Αναζήτηση	Αδίσιασμα	Διαγραφή	1	InnoDB greek_general_ci 45,0 KB

Εικόνα 44:Ολοκλήρωση δημιουργίας της βάσης δεδομένων

3)Anacoda και Jupiter

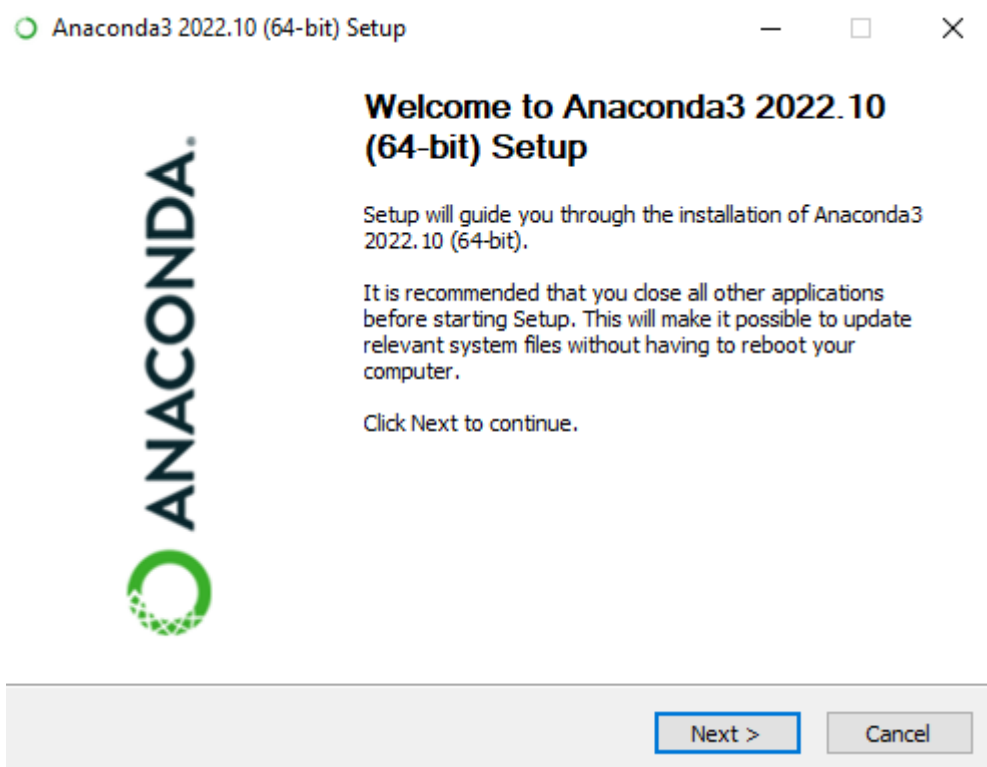
Individual Edition is now

ANACONDA DISTRIBUTION

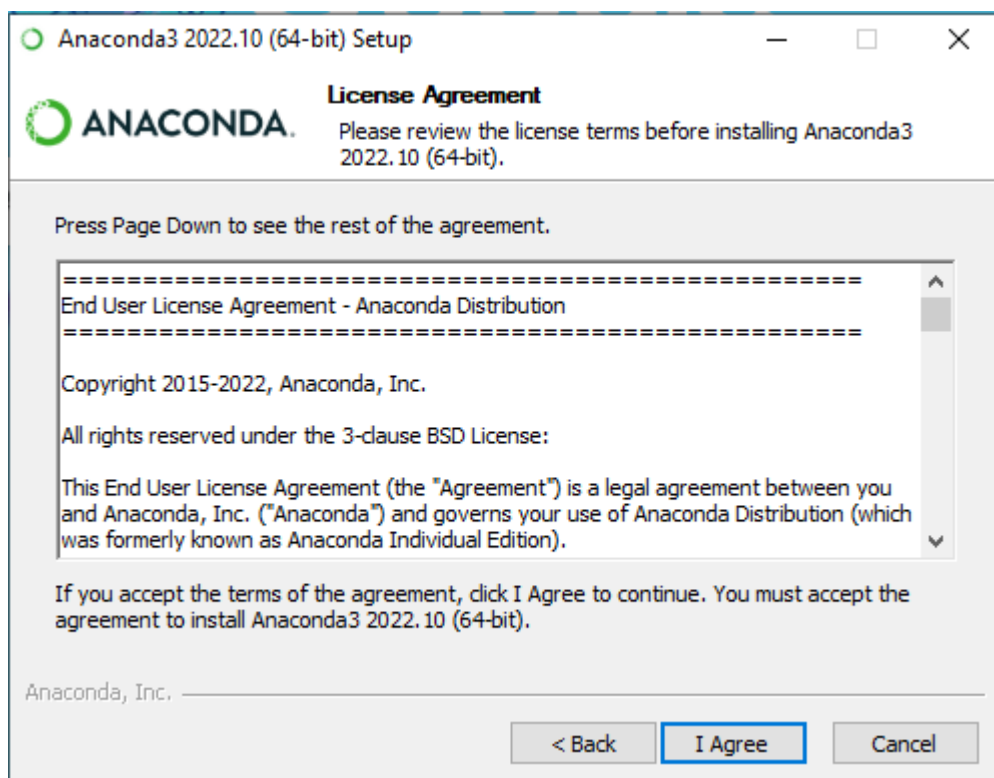
The world's most popular open-source Python distribution platform



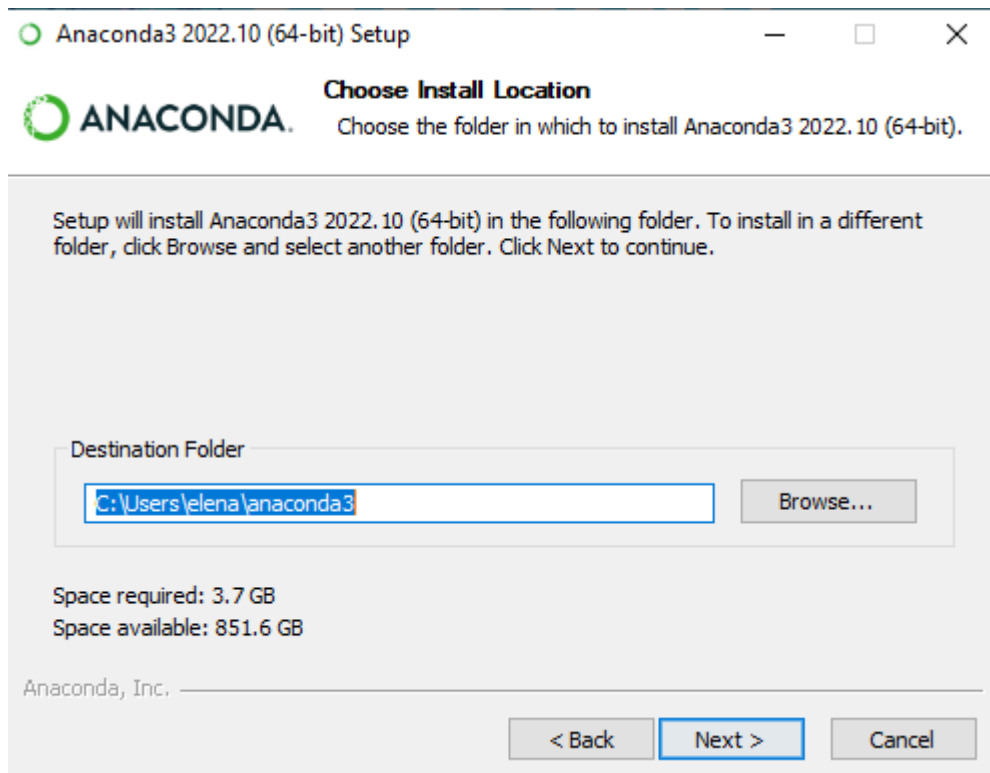
Εικόνα 45:Λήψη Anaconda Distribution



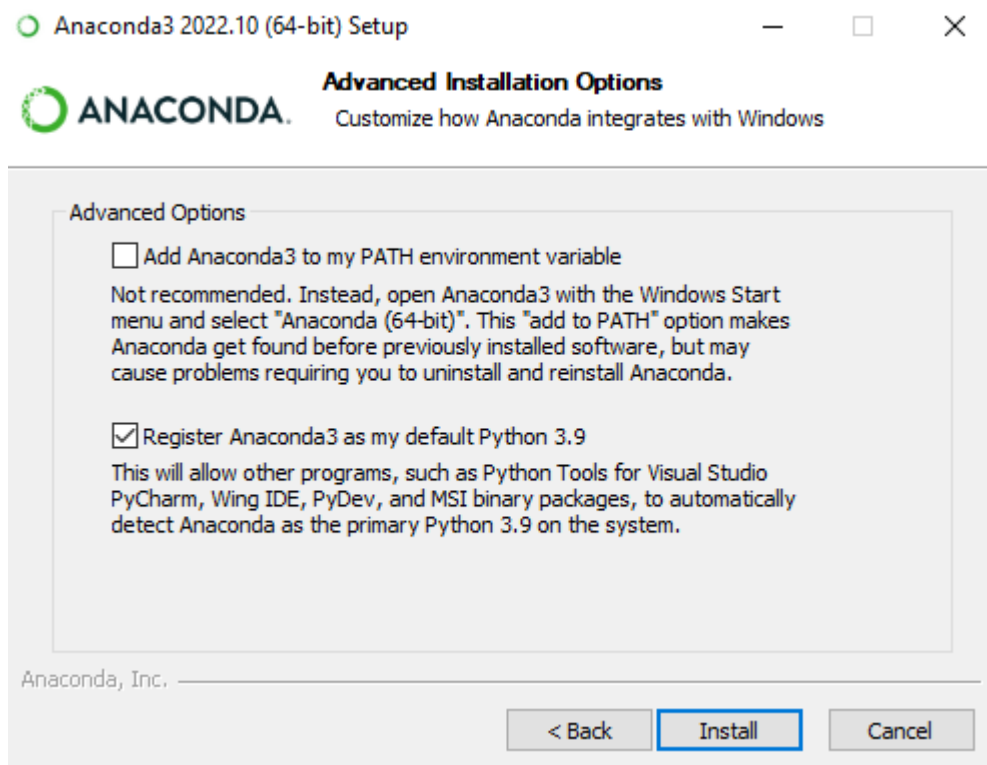
Εικόνα 46:Εγκατάσταση Anacoda (1)



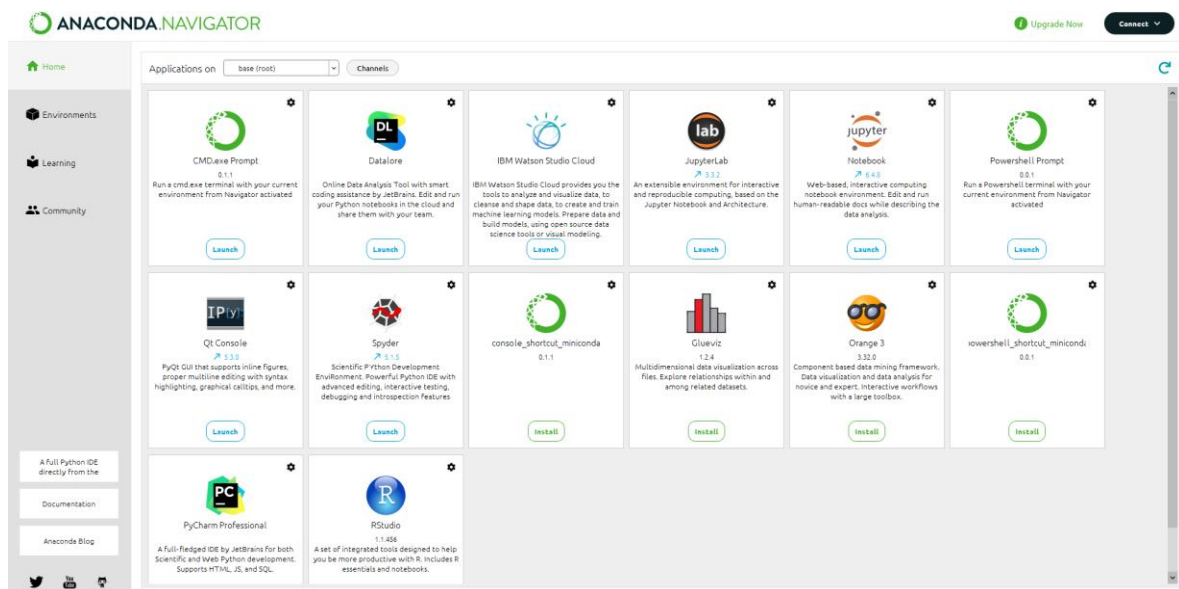
Εικόνα 47:Εγκατάσταση Anacoda (2)



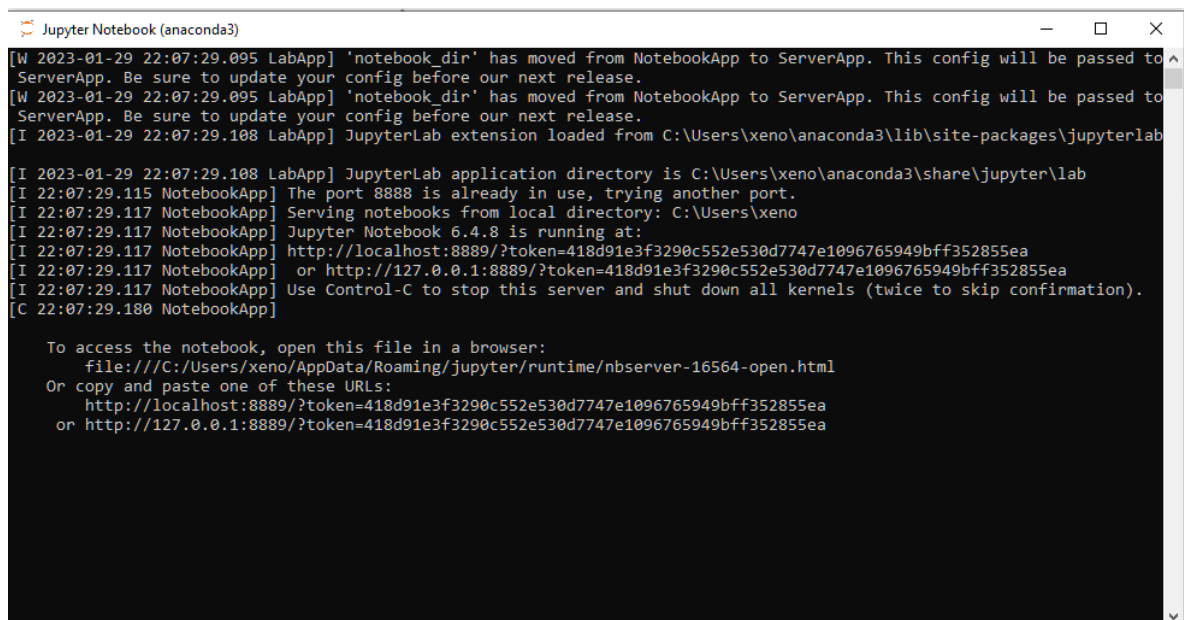
Εικόνα 48:Εγκατάσταση Anacoda (3)



Εικόνα 49:Εγκατάσταση Anacoda (4)



Εικόνα 50:Εκκίνηση Jupyter Notebook



Εικόνα 51:Εκκίνηση Jupyter Notebook (2)

Βιβλιογραφία

- Bechhofer, S., Buchan, I., de Roure, D., Missier, P., Ainsworth, J., Bhagat, J., Couch, P., Cruickshank, D., Delderfield, M., Dunlop, I., Gamble, M., Michaelides, D., Owen, S., Newman, D., Sufi, S., & Goble, C. (2013). Why linked data is not enough for scientists. *Future Generation Computer Systems*, 29(2), 599–611. <https://doi.org/10.1016/j.future.2011.08.004>
- ben Root, & Packt Publishing. (2016). *Data Visualization with Matplotlib and Python*.
- Bourdon, R. (2022). *WampServer*. <https://www.wampserver.com/en/>.
- Capital, A. (2022). *Pandas*. <https://pandas.pydata.org/about/index.html>.
- Code.Hub. (2022). *Data Science & Data Engineering Python and MS-SQL*.
- Edsall, R., Andrienko, G., Andrienko, N., & Battenfield, B. (n.d.). *The GIS Manual CHAPTER 42 Interactive Maps for Exploring Spatial Data*.
- GeoPy Documentation Release 2.3.0 GeoPy Contributors. (2022).
- Johansson, R. (2018). Numerical python: Scientific computing and data science applications with numpy, SciPy and matplotlib, Second edition. In *Numerical Python Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib, Second Edition*. Apress Media LLC. <https://doi.org/10.1007/978-1-4842-4246-9>
- Oliphant, T. E. (2006). *Guide to NumPy*. <http://www.trelgol.com>
- Pajankar, A. (2021). Practical Python Data Visualization. In *Practical Python Data Visualization*. Apress. <https://doi.org/10.1007/978-1-4842-6455-3>
- Rolon-Mérette, D., Ross, M., Rolon-Mérette, T., & Church, K. (2020). Introduction to Anaconda and Python: Installation and setup. *The Quantitative Methods for Psychology*, 16(5), S3–S11. <https://doi.org/10.20982/tqmp.16.5.s003>
- Simon Yuill and Harry Halpin. (2006). *Python*.
- The Python Library Reference Release 3.11.1 Guido van Rossum and the Python development team. (2023).
- Unknown. (n.d.). *Geographic Data Science with PySAL and the pydata stack*.
- Varoquaux Emmanuelle Gouillart Olaf Vahtras Pierre de Buyl Scipy, G., Varoquaux, G., Gouillart, E., Vahtras, O., de Buyl Christopher Burns, P., Chauve, A., Cimrman, R., Combelles Ralf Gommers, C., Espaze, A., Jędrzejewski-Szmek Valentin Haenel, Z., Hartmann, M., Ingold, G.-L., Pedregosa Didrik Pinte, F., Rougier, N. P., van den Bossche, J., Virtanen, P., & ot hers, man. (n.d.). *Python Matplotlib SciKits Numpy SciPy IPython*. www.scipy-lectures.org
- Ιωάννης Λυριτζή. (2022). *ΝΕΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΣΤΙΣ ΑΡΧΑΙΟΓΝΩΣΤΙΚΕΣ ΕΠΙΣΤΗΜΕΣ*.
- Κουρτζέλλης, Γ. (2011). *Εικονική αρχαιολογία και η συμβολή των τρισδιάστατων γραφικών στην αρχαιολογική έρευνα*.
- Κωνσταντής, Χ. (2022). *Επιγραφές πολιτών και μεταλλείων: αποθήκευση και σημασιολογική απεικόνιση τους στο διαδίκτυο*.
- Μηχανικών, Τ., Πανεπιστήμιο, Π., Νικολαου, Ι. Χ., Επιστήμης, Τ., & Κρήτης, Υ. Π. (n.d.). *Κ. ΜΑΓΚΟΥΤΗΣ ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΜΕ PYTHON Μια Προσέγγιση από την Πλευρά των Υπολογιστικών Συστημάτων*. www.kallipos.gr

