



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Οπτικοποίηση δεδομένων στον παγκόσμιο ιστό –
σε ιστοσελίδα**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του

Γεωργιάδη Κωνσταντίνου

(ΑΕΜ: 2013)

Επιβλέπων : Γεώργιος Τασκασαπλίδης

Ιδιότητα : Ε.ΔΙ.Π

Καστοριά Απρίλιος 2024



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Οπτικοποίηση δεδομένων στον παγκόσμιο ιστό –
σε ιστοσελίδα**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του

Γεωργιάδη Κωνσταντίνου

(ΑΕΜ: 2013)

Επιβλέπων : Γεώργιος Τασκασαπλίδης

Ιδιότητα: Ε.ΔΙ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12/4/2024

.....
Σινάτκας Ιωάννης
Καθηγητής

.....
Δημήτριος Ι. Βέργαδος
Καθηγητής

.....
Δημήτριος Φωτιάδης
Επίκουρος Καθηγητής

Καστοριά Απρίλιος 2024

Copyright © 2023 – Γεωργιάδης Κωνσταντίνος

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Δυτικής Μακεδονίας.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Ευχαριστίες

Θέλω να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κύριο Τασκασαπλιδή Γεώργιο για την πολύτιμη καθοδήγησή του και τις χρήσιμες συμβουλές που μου παρείχε κατά τη διάρκεια της εκπόνησης της εργασίας μου. Επίσης, εκφράζω τις ευχαριστίες μου στην οικογένεια μου που με έχει στηρίξει σημαντικά όλα αυτά τα χρόνια όπως επίσης και τους φίλους μου που με εμπύχωσαν με τα λόγια τους.

Περίληψη

Η οπτικοποίηση δεδομένων δεν είναι απλώς μία σύγχρονη ανακάλυψη στη στατιστική, αλλά έχει ρίζες που χρονολογούνται από παλαιότερες ιστορικές περιόδους.

Η ικανότητα οπτικής αναπαράστασης των δεδομένων αποτελεί ζωτικής σημασίας στην επιστημονική έρευνα και εφαρμόζεται ευρέως σε διάφορους τομείς.

Η οπτικοποίηση δεδομένων είναι η διαδικασία με την οποία γίνεται η γραφική απεικόνιση αφηρημένων πληροφοριών, με στόχο να γίνουν κατανοητές από τον αποδέκτη. Αυτή η διαδικασία περιλαμβάνει τη δημιουργία εικόνων και διαγραμμάτων, που απλοποιούν και παρουσιάζουν τα δεδομένα με έναν πιο συνοπτικό και κατανοητό τρόπο στον άνθρωπο. Τα γραφήματα βοηθούν στην μελέτη και στην παρατήρηση μεγάλων όγκων δεδομένων και επιτρέπουν στον θεατή να εξάγει συμπεράσματα με μεγαλύτερη ευκολία και να παρατηρεί σχέσεις και μοτίβα μεταξύ των δεδομένων.

Ένας από τους τρόπους που μπορεί να επιτευχθεί η οπτικοποίηση είναι μέσω του διαδικτύου. Η γραφική απεικόνιση των δεδομένων σε ιστοσελίδες είναι κάτι πολύ διαδεδομένο στις μέρες μας. Η οπτικοποίηση δεδομένων στο διαδίκτυο επιτυγχάνεται με ποικίλους τρόπους, ένας εκ των οποίων είναι η χρήση βιβλιοθηκών γραφημάτων της Javascript για την αναπαράσταση των δεδομένων.

Η βιβλιοθήκη Chart.js είναι μία από αυτές, επιτρέποντας στον προγραμματιστή να σχεδιάσει τα δικά του γραφήματα κάνοντας χρήση του Canvas API. Η δημιουργία μιας διαδικτυακής εφαρμογής με σκοπό την οπτικοποίηση απαιτεί συνεργασία μεταξύ των τεχνολογιών της ιστοσελίδας και του διακομιστή, καθώς και την ανάκτηση των δεδομένων από τον διακομιστή, τα οποία είναι απαραίτητα για τη δημιουργία των γραφημάτων.

Λέξεις Κλειδιά: Chart.js, Οπτικοποίηση Δεδομένων, HTML, CSS, Javascript, PHP, Canvas API, Web Application, XAMPP

Abstract

Data visualization is not just a modern discovery in statistics but has roots dating back to earlier historical periods.

The ability to visually represent data is a vital aspect of scientific research and is widely applied in a variety of fields.

Data visualization is the process by which abstract information is graphically represented to be understood by the recipient. This process involves the creation of images and diagrams that simplify and present data in a more concise and understandable way to humans. Graphs aid in the study and observation of large volumes of data and allow the viewer to draw conclusions more easily and observe relationships and patterns between data.

One of the ways that visualization can be achieved is through the internet. The graphical representation of data on web pages is something very common nowadays. Data visualization on the web is achieved in a variety of ways, one of which is the use of Javascript graph libraries to represent data.

The Chart.js library is one of them, allowing the developer to design their own charts by making use of the Canvas API. Creating a web application for the purpose of visualization requires collaboration between the website and server technologies, as well as retrieving the data from the server, which is necessary to create the charts.

Key Words: Chart.js, Data Visualization, HTML, CSS, Javascript, PHP, Canvas API, Web Application, XAMPP

Πίνακας Περιεχομένων

Εισαγωγή.....	1
1. Οπτικοποίηση Δεδομένων.....	3
1.1 Εισαγωγή	3
1.2 Βιβλιοθήκες Οπτικοποίησης Δεδομένων	4
1.2.1 Βιβλιοθήκη Chart.js.....	4
1.2.2 Βιβλιοθήκη D3.js.....	5
1.2.3 Βιβλιοθήκη Plotly.js	6
1.2.4 Βιβλιοθήκη Google Charts	6
1.3 Χρησιμότητα της Οπτικοποίησης Δεδομένων σε Διάφορους Τομείς	7
2. Τεχνολογίες Εφαρμογής	8
2.1 Η Ιστοσελίδα.....	9
2.1.1 HTML και CSS	9
2.1.2 Η Γλώσσα JavaScript	10
2.1.3 Canvas και Chart.js.....	10
2.2 Τοπικός Διακομιστής	11
2.2.1 Πακέτο Λογισμικού XAMPP	11
2.2.2 Apache HTTP Server	12
2.2.3 Σύστημα Διαχείρισης Βάσεων Δεδομένων MariaDB.....	12
2.2.4 Η Γλώσσα PHP	12
2.2.5 Η Γλώσσα Perl	13
2.3 Άλλες Τεχνολογίες	14
2.3.1 Visual Studio Code	14
2.3.2 Δεδομένα στο Διαδίκτυο	15
3. Υλοποίηση Εφαρμογής	16
3.1 Η Δομή της Εφαρμογής	16
3.2 Τα Αρχεία της Ιστοσελίδας	17
3.2.1 Η Επικεφαλίδα	18
3.2.2 Η Μπάρα Περιήγησης.....	18
3.2.3 Η Σελίδα index.php	20
3.2.4 Το Script Connection.php.....	20
3.2.5 Η Σελίδα Graphs.php	21
3.2.6 Το Script query.php.....	26

3.2.7	Η Σελίδα Create.php	27
3.2.8	Το Script selectInputs.js	34
3.2.9	Η Σελίδα Upload.php	35
3.2.10	Η Σελίδα About.php	38
3.2.11	Το Φύλλο Μορφοποίησης main.css	38
3.3	Εκτέλεση της Εφαρμογής	38
3.3.1	Η Τοποθεσία των Αρχείων	38
3.3.2	Control Panel του XAMPP	38
3.3.3	phpMyAdmin	39
3.3.4	Εκτέλεση της Εφαρμογής.....	40
	Συμπεράσματα.....	41
	Βιβλιογραφία	42
	Παράρτημα Κώδικα	46

Λίστα Εικόνων

Εικόνα 1 Γράφημα με την κίνηση των πλανητών από άγνωστο αστρονόμο του 10ου αιώνα πηγή: Funkhouser, H. G. (1936). “A note on a tenth century graph,” σελ. 261.....	3
Εικόνα 2 Δημιουργία γραφήματος ραντάρ με χρήση της Chart.js, https://www.chartjs.org/docs/latest/samples/other-charts/radar.html	4
Εικόνα 3 Απεικόνιση ραβδογράμματος με χρήση της D3.js, https://observablehq.com/@d3/stacked-to-grouped-bars?intent=fork	5
Εικόνα 4 Διάγραμμα περιγράμματος με χρήση της Plotly.js, https://plotly.com/javascript/contour-plots/	6
Εικόνα 5 Γράφημα γραμμής μέσω Google Charts, https://developers.google.com/chart/interactive/docs/gallery/linechart	7
Εικόνα 6 Πρότυπος κώδικας για την κατασκευή αντικειμένων γραφημάτων της Chart.js, https://www.chartjs.org/docs/latest/getting-started/	11
Εικόνα 7 Αναζήτηση επεκτάσεων στο Marketplace	15
Εικόνα 8 Η τοποθεσία των αρχείων της ιστοσελίδας.....	17
Εικόνα 9 Metadata και εισαγωγή εξωτερικών πόρων στο έγγραφο	18
Εικόνα 10 Ο HTML κώδικας πίσω από την μπάρα περιήγησης	19
Εικόνα 11 Οι κανόνες CSS για την μπάρα περιήγησης.....	19
Εικόνα 12 Το script connection.php	20
Εικόνα 13 Η εμφάνιση του Graphs.php στο πρόγραμμα περιήγησης	21
Εικόνα 14 Ορισμός της περιοχής καμβά και δημιουργία drop-down για την επιλογή γραφήματος	22
Εικόνα 15 Μετατροπή των δεδομένων σε μορφή JSON	23
Εικόνα 16 Η αρχικοποίηση των αντικειμένων γραφημάτων	23
Εικόνα 17 Αλλαγή τύπου γραφήματος μέσω της συνάρτησης changeShape()	24
Εικόνα 18 Επιλογή τύπου γραφήματος στην καρτέλα Graphs.....	25
Εικόνα 19 Ο κώδικας του query.php	26
Εικόνα 20 Εισαγωγή δεδομένων στα πεδία εισόδων στην καρτέλα Create	28
Εικόνα 21 Γράφημα Radar με βάση δεδομένα από το πληκτρολόγιο του χρήστη.....	28
Εικόνα 22 Η επιλογή του αριθμού των εισόδων του χρήστη.....	29

Εικόνα 23 Η απόκρυψη των αντικειμένων λίστας «li» στο αρχείο main.css	30
Εικόνα 24 Δημιουργία των κουμπιών Create και Reset.....	30
Εικόνα 25 Η συνάρτηση createButton()	31
Εικόνα 26 Δημιουργία CSS για την διαχείριση της εμφάνισης των κουμπιών.	32
Εικόνα 27 Οι λειτουργίες της συνάρτησης resetButton()	33
Εικόνα 28 Η συνάρτηση selectInput(selectNum)	34
Εικόνα 29 Η καρτέλα Upload	36
Εικόνα 30 Η συνάρτηση uploadFile και η χρήση της βιβλιοθήκης Sheet.js για την επεξεργασία του αρχείου Excel.	36
Εικόνα 31 Δημιουργία και αρχικοποίηση των πινάκων labelArray και numberArray μέσα στην συνάρτηση createButton.....	37
Εικόνα 32 Xampp Control Panel	39
Εικόνα 33 Δημιουργία Βάσης Δεδομένων στο phpMyAdmin	40

Εισαγωγή

Τα τελευταία χρόνια, έχει παρατηρηθεί μια εκρηκτική αύξηση στον όγκο των δεδομένων που δημιουργούνται, αποθηκεύονται και στην συνέχεια τίθενται σε επεξεργασία. Μεγάλοι οργανισμοί παράγουν δεδομένα σε καθημερινή βάση, γεγονόσ που οδηγεί σε σημαντική αύξηση του όγκου των δεδομένων που είναι διαθέσιμα στο διαδίκτυο. Αυτές οι μεγάλες ποσότητες δεδομένων δημιουργούν δυσκολίες για τους χρήστες που θέλουν να τα κατανοήσουν, να τα διερευνήσουν και να τα αξιοποιήσουν αποτελεσματικά. Στη σύγχρονη εποχή, οι υπολογιστές διαδραματίζουν καθοριστικό ρόλο στην επεξεργασία εκτεταμένων συνόλων δεδομένων. [1]

Η γραφική αναπαράσταση των δεδομένων μέσω της οπτικοποίησης επιτρέπει στον παρατηρητή να μελετήσει αυτά τα περίπλοκα δεδομένα πιο εύκολα και να διεξάγει συμπεράσματα για τις σχέσεις τους.

Είναι σύνηθες να πιστεύουμε ότι η οπτικοποίηση δεδομένων είναι μία σχετικά σύγχρονη ανακάλυψη στον τομέα της στατιστικής. Ωστόσο η ιστορία της οπτικοποίησης δεδομένων είναι πολύ πλούσια και έχει βαθιές ρίζες που χρονολογούνται από αρχαίες περιόδους. Ανάμεσα στα πρώτα επίσημα παραδείγματα ανήκουν τα γεωμετρικά διαγράμματα της αρχαίας Αιγύπτου περί το 200 π.Χ., καθώς και γραφήματα αστρονόμων του 10ου αιώνα που καταγράφουν την θέση των πλανητών. Μέχρι τον 16^ο αιώνα, εργαλεία και τεχνικές είχαν αναπτυχθεί για τη μέτρηση φυσικών και γεωγραφικών μεγεθών, σηματοδοτώντας την πρόοδο προς την οπτικοποίηση δεδομένων. Τον 18^ο αιώνα ο Σκωτσέζος κοινωνικός επιστήμονας William Playfair [1759-1823] εφηύρε το ραβδόγραμμα (bar chart), το γράφημα γραμμής (line graph), το διάγραμμα πίτας (pie chart) και το γράφημα κύκλου (circle graph) και θεωρείται ευρέως ως ο εφευρέτης των πιο διαδεδομένων γραφικών μορφών που χρησιμοποιούνται σήμερα. Ο χάρτης κουκκίδων του Dr. John Snow ανέδειξε το ρόλο των μολυσμένων πηγών νερού στην εξάπλωση της νόσου της χολέρας που χτύπησε το Λονδίνο το 1831. Από το 1975 και μετά, η οπτικοποίηση δεδομένων βίωσε γρήγορες εξελίξεις. Κάποιες από τις σημαντικότερες περιλαμβάνουν την ανάπτυξη διαδραστικών υπολογιστικών συστημάτων, καινοτόμες μεθόδους οπτικοποίησης για υψηλές διαστάσεις δεδομένων και την εφαρμογή των μεθόδων αυτών σε πολλαπλά προβλήματα. [2] [3] [4]

Σήμερα πλέον η γραφική αναπαράσταση δεδομένων μπορεί να θεαθεί παντού γύρω μας και ανθίζει ιδιαίτερα στο διαδίκτυο. Η οπτικοποίηση δεδομένων στο διαδίκτυο μπορεί να επιτευχθεί με διάφορους τρόπους. Ένας από αυτούς είναι η χρήση βιβλιοθηκών οπτικοποίησης, οι οποίες παρέχουν στον προγραμματιστή έτοιμο κώδικα, ο οποίος του επιτρέπει να σχεδιάσει ο ίδιος τα γραφήματα της αρεσκείας του και να τα ενσωματώσει στην ιστοσελίδα του.

Ο στόχος της διπλωματικής εργασίας είναι η οπτικοποίηση δεδομένων στο διαδίκτυο. Για τον σκοπό αυτόν, δημιουργήθηκε μία διαδικτυακή εφαρμογή, η οποία χρησιμοποιεί την βιβλιοθήκη Chart.js για την δημιουργία γραφημάτων πάνω σε μία ιστοσελίδα, ανακτώντας δεδομένα, τα οποία είναι αποθηκευμένα σε έναν τοπικό διακομιστή, ο οποίος στήθηκε με την βοήθεια του πακέτου λογισμικού ΧΑΜΡΡ.

Αρχικά, στο πρώτο κεφάλαιο περιγράφεται η οπτικοποίηση δεδομένων. Παρουσιάζονται επίσης κάποιες βιβλιοθήκες οπτικοποίησης με τα πλεονεκτήματα και τα μειονεκτήματα τους και τέλος η χρησιμότητα της οπτικοποίησης σε διάφορους κλάδους.

Στο δεύτερο κεφάλαιο περιγράφονται οι τεχνολογίες που χρησιμοποιήθηκαν για την δημιουργία της εφαρμογής.

Στο τρίτο κεφάλαιο γίνεται αναφορά στην δομή της εφαρμογής και αναλύεται λεπτομερώς ο κώδικας των αρχείων της εφαρμογής.

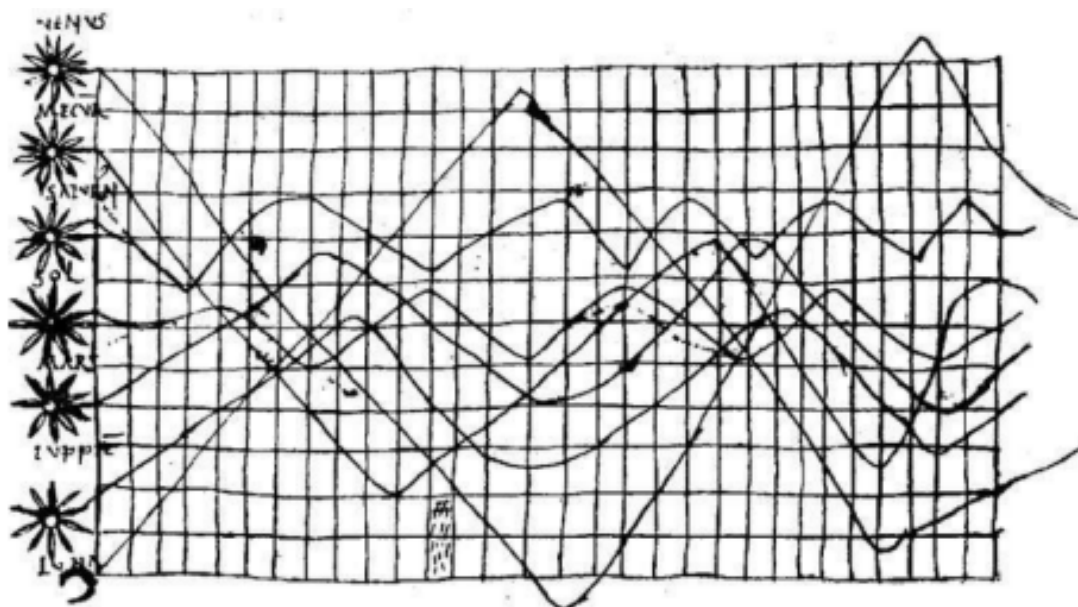
Τέλος, υπάρχουν τα συμπεράσματα που διεξάχθηκαν κατά την διάρκεια της εκπόνησης της διπλωματικής εργασίας καθώς επίσης και προτάσεις για μελλοντικές επεκτάσεις.

1. Οπτικοποίηση Δεδομένων

1.1 Εισαγωγή

Η οπτικοποίηση δεδομένων αποτελεί τη διαδικασία της γραφικής απεικόνισης αφηρημένων πληροφοριών με τρόπο τέτοιο ώστε να είναι κατανοητός από τον αποδέκτη. Συμπεριλαμβάνει την δημιουργία εικόνων και διαγραμμάτων που έχουν ως στόχο να παρουσιάζονται τα δεδομένα με έναν πιο συνοπτικό και κατανοητό τρόπο προς τον άνθρωπο, διότι το ανθρώπινο μάτι καταλαβαίνει πιο εύκολα μία οπτική αναπαράσταση των δεδομένων από την αριθμητική τους μορφή και είναι πιο απλό να παρατηρηθούν και να εξεταστούν μοτίβα και σχέσεις μεταξύ των δεδομένων και να διεξαχθούν συμπεράσματα.

Η ικανότητα οπτικής αναπαράστασης των δεδομένων είναι υψίστης σημασίας στην επιστημονική έρευνα. Η οπτικοποίηση δεδομένων χρησιμοποιείται επίσης σε διάφορους κλάδους όπως στην υγεία, τις πωλήσεις και το μάρκετινγκ, στον στρατό, στην εκπαίδευση, στην επιστήμη του data science καθώς επίσης και σε πολλούς άλλους τομείς. [5]



Εικόνα 1 Γράφημα με την κίνηση των πλανητών από άγνωστο αστρονόμο του 10ου αιώνα πηγή: Funkhouser, H. G. (1936). "A note on a tenth century graph," σελ. 261

1.2 Βιβλιοθήκες Οπτικοποίησης Δεδομένων

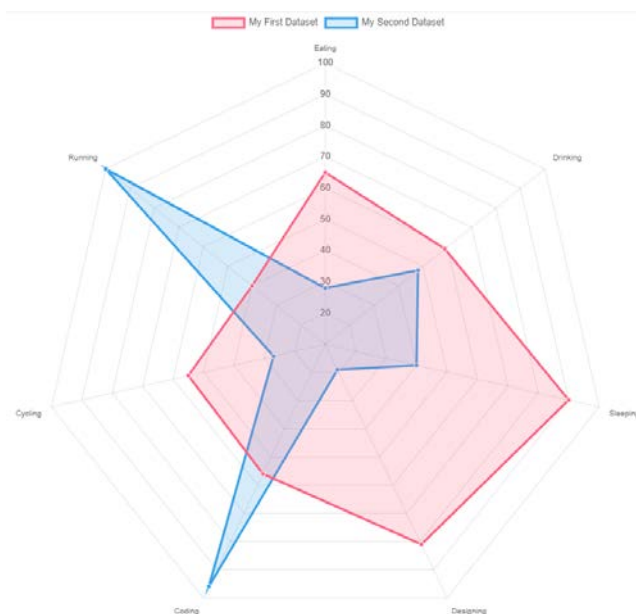
Στις μέρες μας η οπτικοποίηση δεδομένων στο διαδίκτυο είναι κάτι το πολύ σύνθητες. Ένας πολύ αποτελεσματικός τρόπος για να επιτευχθεί αυτό είναι με την χρήση έτοιμων βιβλιοθηκών που παρέχουν την δυνατότητα στον καθένα να δημιουργεί τα δικά του γραφήματα στην ιστοσελίδα του. Παρακάτω θα αναλυθούν συνοπτικά κάποιες από τις πιο δημοφιλείς βιβλιοθήκες οπτικοποίησης δεδομένων, συμπεριλαμβανομένης και της Chart.js η οποία χρησιμοποιήθηκε για την δημιουργία της εφαρμογής αυτής της εργασίας και θα αναλυθεί περεταίρω στα επόμενα κεφάλαια.

1.2.1 Βιβλιοθήκη Chart.js

Το πρώτο εργαλείο που θα αναλυθεί είναι η βιβλιοθήκη Chart.js της JavaScript. Η Chart.js είναι μια δωρεάν βιβλιοθήκη JavaScript ανοικτού κώδικα που χρησιμοποιείται για την οπτικοποίηση δεδομένων. Αυτή η βιβλιοθήκη εκμεταλλεύεται την ετικέτα HTML <canvas> για την απόδοση διαφορετικών τύπων διαγραμμάτων και είναι μία από τις πιο δημοφιλείς βιβλιοθήκες γραφικών παραστάσεων JavaScript στο Github. [6]

Ένα από τα πλεονεκτήματα της χρήσης του Canvas API είναι ότι είναι ένα πρότυπο που αναπτύχθηκε από το W3C εξασφαλίζοντας ομαλή λειτουργία με την HTML και την JavaScript. Είναι επίσης κατάλληλο για εφαρμογές έντονων γραφικών και υποστηρίζεται σε πλατφόρμες κινητών τηλεφώνων. [7]

Το πλεονέκτημα της Chart.js έγκειται στην απλή της χρήση, η οποία απαιτεί μόνο βασικές γνώσεις HTML, CSS και JavaScript, επιτρέποντας στον χρήστη να δημιουργήσει γρήγορα διαγράμματα με αυτήν. Ωστόσο η βιβλιοθήκη έχει περιορισμένες δυνατότητες και υποστηρίζει μέχρι στιγμής μόνο 9 διαφορετικούς τύπους γραφημάτων. [8] [6]



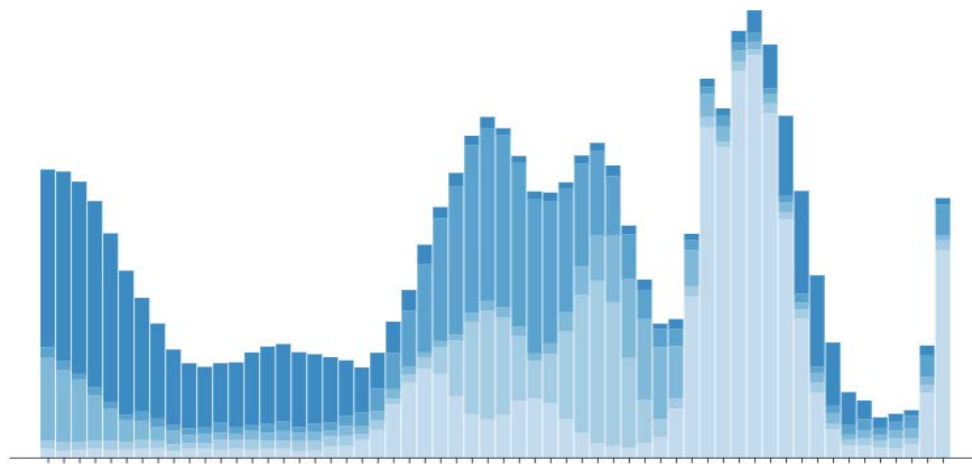
Εικόνα 2 Δημιουργία γραφήματος ραντάρ με χρήση της Chart.js, <https://www.chartjs.org/docs/latest/samples/other-charts/radar.html>

1.2.2 Βιβλιοθήκη D3.js

Η D3.js είναι επίσης άλλη μία βιβλιοθήκη ανοιχτού κώδικα της JavaScript, που χρησιμοποιείται για την οπτικοποίηση ψηφιακών δεδομένων σε γραφική και δυναμική μορφή. Είναι ένα ισχυρό εργαλείο που ακολουθεί τα πρότυπα του W3C και χρησιμοποιεί κοινές τεχνολογίες του διαδικτύου, που είναι το SVG ή HTML Canvas, JavaScript και CSS.

Σε αντίθεση με άλλες βιβλιοθήκες, το D3.js πέρα από βιβλιοθήκη οπτικοποίησης δεδομένων είναι και ένα σύνολο εργαλείων της JavaScript που βοηθούν στην επεξεργασία των δομών δεδομένων που είναι απαραίτητες για την δημιουργία οπτικοποιήσεων. Δεν παρέχει στον προγραμματιστή έτοιμα πρότυπα για την κατασκευή γραφημάτων. Ο προγραμματιστής πρέπει ο ίδιος να σχεδιάσει τα γραφήματα κάνοντας χρήση των τεχνολογιών SVG ή HTML Canvas. Ωστόσο αυτό έχει ως αποτέλεσμα μεγαλύτερη ελευθερία πάνω στα χαρακτηριστικά της οπτικοποίησης και επιτρέπει τον βαθύτερο και εκτενέστερο έλεγχο του τελικού οπτικού αποτελέσματος. [9]

Η βιβλιοθήκη αναπτύχθηκε ως απάντηση στην αυξανόμενη ανάγκη για ισχυρές και ευέλικτες οπτικοποιήσεις δεδομένων στο διαδίκτυο. Ο ευέλικτος χαρακτήρας της D3.js της επιτρέπει να προχωρήσει πέρα από την απλή δημιουργία διαγραμμάτων πίτας και γραμμικών γραφημάτων, επιτρέποντας στους προγραμματιστές να ενσωματώνουν σύνθετους χάρτες ή διαδραστικά διαγράμματα χωρίς να καταφεύγουν σε ξεπερασμένες τεχνολογίες όπως το Flash ή τα Java applets.

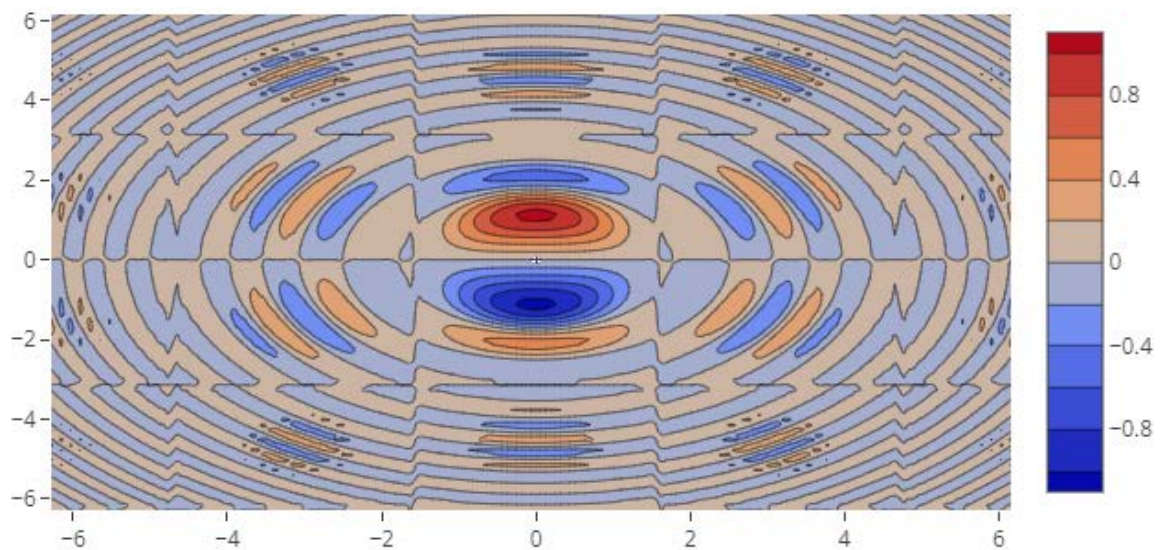


Εικόνα 3 Απεικόνιση ραβδογράμματος με χρήση της D3.js, <https://observablehq.com/@d3/stacked-to-grouped-bars?intent=fork>

1.2.3 Βιβλιοθήκη Plotly.js

Το Plotly.js είναι μια ισχυρή βιβλιοθήκη JavaScript ανοικτού κώδικα που χρησιμοποιείται για τη δημιουργία διαδραστικών γραφικών στο διαδίκτυο. Η βιβλιοθήκη αυτή προσφέρει πολλές δυνατότητες για τη δημιουργία όμορφων και εξατομικευμένων γραφημάτων, συμπεριλαμβανομένων γραφημάτων διασποράς, γραμμικών γραφικών, δημιουργία χαρτών, 3D γραφικών και πολλών άλλων.

Η Plotly.js υποστηρίζει διάφορες γλώσσες προγραμματισμού όπως JavaScript, R ή Python και μπορεί να ενσωματωθεί εύκολα σε ιστοσελίδες και εφαρμογές για την οπτικοποίηση δεδομένων. [10]



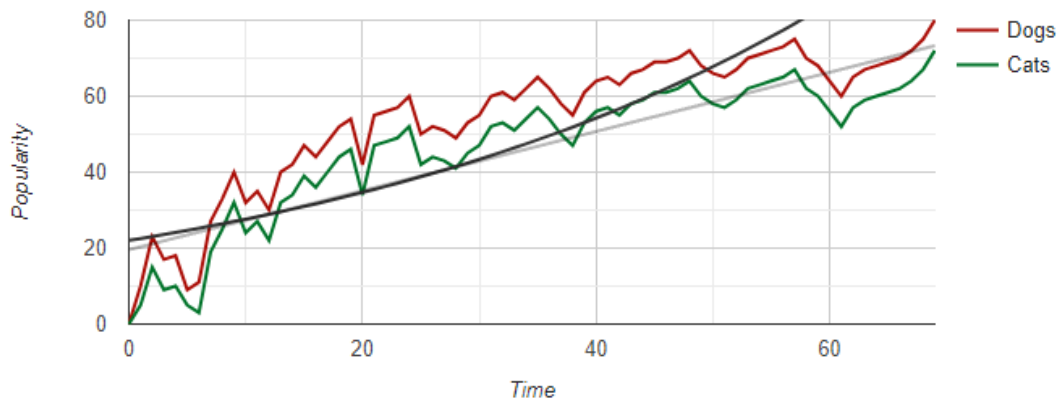
Εικόνα 4 Διάγραμμα περιγράμματος με χρήση της Plotly.js, <https://plotly.com/javascript/contour-plots/>

1.2.4 Βιβλιοθήκη Google Charts

Η Google Charts είναι μία βιβλιοθήκη οπτικοποίησης δεδομένων της Google βασισμένη στην JavaScript, η οποία επιτρέπει την δημιουργία διαδραστικών γραφημάτων. Παρέχει μία ποικιλία από απλά διδιάστατα γραφήματα. Είναι πολύ εύκολη στην χρήση της και μπορεί να χρησιμοποιηθεί από οποιαδήποτε ιστοσελίδα στο διαδίκτυο. Ο χρήστης που θέλει να χρησιμοποιήσει το Google Charts ενσωματώνει τα δεδομένα και τις παραμέτρους μορφοποίησης του γραφήματος που θέλει να δημιουργήσει σε ένα HTTP request στην ιστοσελίδα του, και η Google δημιουργεί και επιστρέφει μία εικόνα του γραφήματος σε μορφή PNG.

Τα διαγράμματα απεικονίζονται με τη χρήση τεχνολογίας HTML5/SVG για να παρέχουν συμβατότητα μεταξύ διαφορετικών προγραμμάτων περιήγησης καθώς και

φορητότητα μεταξύ διαφορετικών πλατφορμών σε iPhones, iPads και Android. [11]



Εικόνα 5 Γράφημα γραμμής μέσω Google Charts,
<https://developers.google.com/chart/interactive/docs/gallery/linechart>

1.3 Χρησιμότητα της Οπτικοποίησης Δεδομένων σε Διάφορους Τομείς

Η οπτικοποίηση δεδομένων αποτελεί ισχυρό εργαλείο που εφαρμόζεται με επιτυχία σε διάφορους τομείς, από την επιστήμη των δεδομένων(Data Science) και την υγεία μέχρι την επιχειρηματικότητα και τη δημόσια πολιτική. Η δυνατότητα να μετατρέπουμε πολύπλοκα δεδομένα σε οπτικές αναπαραστάσεις προσφέρει νέες διαστάσεις κατανόησης, επικοινωνίας και λήψης αποφάσεων.

Στην συνέχεια θα εξετάσουμε τη χρησιμότητα της οπτικοποίησης δεδομένων σε διάφορους τομείς και πώς συμβάλλει στην ενίσχυση της κατανόησης και της λήψης αποφάσεων.

Υγεία: Στον τομέα της υγείας θα μπορούσε ένας πίνακας που απεικονίζει το ιστορικό ενός ασθενή να οδηγήσει τον γιατρό σε διάφορα συμπεράσματα για την υγεία του ασθενή. Σε περιόδους έκτακτης ανάγκης, η οπτικοποίηση δεδομένων θα μπορούσε επίσης να παρέχει πίνακες που διευκολύνουν την ανάλυση της κατάστασης, με αποτέλεσμα ταχύτερο χρόνο αντίδρασης των ειδικών. [12]

Στρατός: Η ανάλυση δεδομένων είναι απολύτως αναγκαία στον στρατό για τη λήψη αποφάσεων που μπορεί να αποδειχθούν κρίσιμες. Η δυναμική οπτικοποίηση δεδομένων βελτιώνει την κατανόηση της γεωγραφίας και του κλίματος, βοηθώντας στην πιο αποτελεσματική προσέγγιση. Επιπλέον, η χρήση διαγραμμάτων επιτρέπει την ανάλυση του στρατιωτικού εξοπλισμού και τη λήψη αποφάσεων για αγορές ανάλογα με τις ανάγκες. [13]

Data Science: Ο στόχος της επιστήμης του Data Science είναι η επεξεργασία ή η εξαγωγή γνώσης από μεγάλα σύνολα δεδομένων. Η οπτικοποίηση δεδομένων

διαδραματίζει σημαντικό ρόλο σε αυτό, επιτρέποντας στους επιστήμονες να οπτικοποιούν τα δεδομένα τους και να βρίσκουν μοτίβα και σχέσεις μεταξύ των δεδομένων, τα οποία με την σειρά τους οδηγούν σε κάποια συμπεράσματα. [14]

Εμπόριο: Η οπτικοποίηση δεδομένων είναι εξαιρετικά χρήσιμη στην βιομηχανία του εμπορίου. Οι εταιρείες χρησιμοποιούν την οπτικοποίηση δεδομένων για να κατανοήσουν τη συμπεριφορά των πελατών και να προβλέπουν τις ανάγκες τους, προσφέροντας αντίστοιχα προϊόντα. Αυτά τα εργαλεία παρέχουν επίσης πληροφορίες σχετικά με την αναμενόμενη διακύμανση της ζήτησης των πελατών στους επόμενους μήνες και έτη. [15]

Εκπαίδευση: Τέλος στον τομέα της εκπαίδευσης με τη χρήση γραφημάτων, γραφικών και άλλων οπτικών αναπαραστάσεων, εκπαιδευτικά ιδρύματα μπορούν να αναλύσουν δεδομένα σχετικά με την απόδοση των μαθητών, τη συμπεριφορά των μαθητών και τις εκπαιδευτικές τάσεις. Η οπτικοποίηση επιτρέπει την ευκολότερη παρακολούθηση και αξιολόγηση των δεδομένων αυτών, επιτρέποντας στους εκπαιδευτικούς να αντλήσουν σημαντικές ερμηνείες από αυτά. Επιπλέον, η οπτικοποίηση μπορεί να βοηθήσει στον προσανατολισμό της διδασκαλίας και την προσαρμογή των εκπαιδευτικών προγραμμάτων στις ανάγκες των μαθητών, βελτιώνοντας έτσι την εκπαιδευτική διαδικασία γενικά. [13]

2. Τεχνολογίες Εφαρμογής

Στο παρόν κεφάλαιο, θα εξετάσουμε λεπτομερώς τις τεχνολογίες που χρησιμοποιήθηκαν για τη δημιουργία της εφαρμογής μας. Η εφαρμογή αυτή αποτελείται από δύο βασικά κομμάτια: το μέρος της ιστοσελίδας (webpage) και το μέρος του τοπικού διακομιστή (local web server). Σε αυτό το κεφάλαιο, θα εξετάσουμε κάθε ένα από τα επιμέρους κομμάτια και τις τεχνολογίες που τα απαρτίζουν και πώς συνεργάζονται αναμεταξύ τους για την δημιουργία και την λειτουργία της εφαρμογής αυτής της πτυχιακής εργασίας.

Αρχικά, θα εστιάσουμε στο πρώτο μέρος της εφαρμογής, όπου χρησιμοποιήθηκαν τεχνολογίες όπως HTML, CSS και JavaScript για την δημιουργία της ιστοσελίδας. Αυτές οι τεχνολογίες παίζουν καίριο ρόλο στην δημιουργία μίας διεπαφής φιλική προς τους χρήστες για την εφαρμογή και στην αποδοτική αλληλεπίδραση με τους χρήστες.

Στη συνέχεια, θα εξετάσουμε τον τομέα του τοπικού διακομιστή, όπου χρησιμοποιήσαμε τεχνολογίες όπως η SQL και η PHP καθώς και τον Apache HTTP Server που μας παρέχει το XAMPP. Αυτές οι τεχνολογίες επιτρέπουν την αποθήκευση και διαχείριση των δεδομένων της εφαρμογής και την επικοινωνία ανάμεσα στην ιστοσελίδα και την βάση δεδομένων, επιτρέποντας τη λειτουργία της εφαρμογής και την αποστολή δεδομένων προς και από το μέρος του προγράμματος περιήγησης.

2.1 Η Ιστοσελίδα

Όπως αναφέρθηκε προηγουμένως, σε αυτό το υποκεφάλαιο θα αναλυθούν οι τεχνολογίες της εφαρμογής του μέρους της ιστοσελίδας. Συγκεκριμένα πρόκειται για την δημιουργία μίας ιστοσελίδας, η οποία τρέχει σε ένα πρόγραμμα περιήγησης και παίζει τον ρόλο μίας φιλικής διεπαφής προς τον χρήστη. Σε αυτήν την ιστοσελίδα πραγματοποιείται η οπτικοποίηση δεδομένων και παρουσιάζονται τα δεδομένα με την μορφή γραφημάτων σε αυτήν, κάνοντας χρήση της βιβλιοθήκης **Chart.js** της JavaScript. Τα δεδομένα που χρησιμοποιούνται για την οπτικοποίηση είναι αποθηκευμένα στον τοπικό διακομιστή που μας παρέχει το πακέτο λογισμικού XAMPP, ωστόσο υπάρχει η δυνατότητα στον χρήστη να προσθέσει ο ίδιος δεδομένα στην ιστοσελίδα και να δημιουργήσει ο ίδιος τα δικά του γραφήματα.

2.1.1 HTML και CSS

Αρχικά θα εξετάσουμε τις τεχνολογίες που χρησιμοποιήθηκαν για την δημιουργία της ιστοσελίδας της εφαρμογής. Το πρώτο εργαλείο στο οποίο θα γίνει αναφορά είναι η HTML.

Η HTML (HyperText Markup Language) είναι η γλώσσα σήμανσης που χρησιμοποιείται για τη δημιουργία και τον σχεδιασμό ιστοσελίδων. Είναι το θεμέλιο κάθε ιστοσελίδας διότι χωρίζει σε τμήματα το περιεχόμενο της σελίδας και ορίζει την δομή της. Τα τμήματα στα οποία χωρίζεται το περιεχόμενο της σελίδας ονομάζονται στοιχεία HTML και το κάθε στοιχείο αποτελείται από δύο ετικέτες και το περιεχόμενο που παρευρίσκεται ενδιάμεσα. Η πρώτη έκδοση της δημιουργήθηκε από τον Tim Berners-Lee το 1993. Σήμερα πλέον χρησιμοποιείται η 5^η κύρια έκδοσή της, η HTML 5. [16]

Μολονότι μπορεί να ορίσει την δομή μίας σελίδας, η HTML δεν μπορεί να ορίσει την εμφάνιση της. Για αυτόν τον σκοπό αξιοποιείται η CSS (Cascading Style Sheets), η οποία είναι μία γλώσσα φύλλων στυλ, και η οποία χρησιμοποιείται για να περιγράψει την εμφάνιση ενός εγγράφου γραμμένο σε γλώσσα σήμανσης. Σε ένα έγγραφο CSS γράφονται κάποιοι κανόνες, οι οποίοι καθορίζουν τα στιλιστικά χαρακτηριστικά των στοιχείων HTML στα οποία εφαρμόζονται. Με τη βοήθεια της CSS, οι προγραμματιστές

μπορούν να δημιουργήσουν εντυπωσιακές ιστοσελίδες και να βελτιστοποιήσουν την εμπειρία του χρήστη.

2.1.2 Η Γλώσσα JavaScript

Η JavaScript (JS) αποτελεί μία προγραμματιστική γλώσσα script, η οποία χρησιμοποιείται συνήθως για τη δημιουργία διαδραστικών και δυναμικών ιστοσελίδων. Αν και η HTML και η CSS αφορούν τη δομή και την εμφάνιση μιας σελίδας, η JavaScript παρέχει τη δυνατότητα προγραμματισμού για την ενίσχυση της λειτουργικότητας της ιστοσελίδας, προσφέροντας ένα δυναμικό και ενεργό περιβάλλον για τον χρήστη. [17]

Μία δυναμική ιστοσελίδα επιτρέπει την αλλαγή του περιεχομένου και της εμφάνισής της κατά την διάρκεια της χρήσης της χωρίς την ανάγκη επαναφόρτωσης της. Επίσης η JavaScript δίνει την δυνατότητα διαχείρισης γεγονότων παρέχοντας τρόπους για την κατάλληλη αντιμετώπιση τους από την ιστοσελίδα. Τέλος είναι εφικτή η επικοινωνία μεταξύ της ιστοσελίδας μέσω της JavaScript με άλλες υπηρεσίες και τεχνολογίες κάνοντας χρήση των API τους.

2.1.3 Canvas και Chart.js

Το Canvas είναι ένα API που χρησιμοποιείται για τον σχεδιασμό γραφικών σε ιστοσελίδες. Το Canvas API επιτρέπει στον προγραμματιστή να ορίσει μία περιοχή πλαισίου καμβά στην ιστοσελίδα, κάνοντας χρήση της ετικέτας HTML <canvas>. Στην συνέχεια ο προγραμματιστής μπορεί να σχεδιάσει γραφικά στην περιοχή αυτήν χρησιμοποιώντας την γλώσσα JavaScript. Το Canvas επικεντρώνεται κυρίως στην παραγωγή δισδιάστατων γραφικών. Ωστόσο είναι εφικτή η παραγωγή τρισδιάστατων γραφικών στον καμβά, χρησιμοποιώντας το WebGL API. [18] [7]

Αναφέρθηκε προηγουμένως ότι η Chart.js είναι βιβλιοθήκη της JavaScript, η οποία χρησιμοποιείται για οπτικοποίηση δεδομένων εκμεταλλευόμενη το Canvas API. Εφόσον ο προγραμματιστής ορίσει έναν «καμβά», μπορεί στην συνέχεια να σχεδιάσει γραφήματα πάνω του, κάνοντας χρήση του έτοιμου κώδικα που παρέχει η βιβλιοθήκη Chart.js για την κατασκευή αντικειμένων(object) γραφημάτων.

Ο προγραμματιστής μπορεί να επισκεφθεί την ιστοσελίδα του Chart.js (<https://www.chartjs.org>). Εκεί δίνεται ο πρότυπος κώδικας και η μεθοδολογία για την κατασκευή ενός αντικειμένου γραφήματος. Ο ενδιαφερόμενος μπορεί να προσαρτήσει τον πρότυπο κώδικα στην δικιά του ιστοσελίδα, και ακολουθώντας την μεθοδολογία να φτιάξει τα δικά του γραφήματα με πολύ εύκολο και επεξηγηματικό τρόπο. Υπάρχει επίσης πλήρης ανάλυση του κώδικα για διάφορες ρυθμίσεις και παραμετροποιήσεις στην κατασκευή των αντικειμένων γραφημάτων.

Create a Chart

In this example, we create a bar chart for a single dataset and render it on an HTML page. Add this code snippet to your page:

```
<div>
  <canvas id="myChart"></canvas>
</div>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script>
  const ctx = document.getElementById('myChart');

  new Chart(ctx, {
    type: 'bar',
    data: {
      labels: ['Red', 'Blue', 'Yellow', 'Green', 'Purple', 'Orange'],
      datasets: [{
        label: '# of Votes',
        data: [12, 19, 3, 5, 2, 3],
        borderWidth: 1
      }]
    },
    options: {
      scales: {
        y: {
          beginAtZero: true
        }
      }
    }
  });
</script>
```

Εικόνα 6 Πρότυπος κώδικας για την κατασκευή αντικειμένων γραφημάτων της Chart.js, <https://www.chartjs.org/docs/latest/getting-started/>

2.2 Τοπικός Διακομιστής

Στην συνέχεια θα αναλυθεί το κομμάτι του τοπικού διακομιστή, ο οποίος στήθηκε με την βοήθεια των τεχνολογιών που μας παρέχει το XAMPP. Ο διακομιστής αποτελείται από τον Apache HTTP server, ο οποίος διαχειρίζεται αιτήματα HTTP, το MariaDB που διαχειρίζεται και οργανώνει τα δεδομένα καθώς και την PHP ως γλώσσα script στην πλευρά του διακομιστή.

2.2.1 Πακέτο Λογισμικού XAMPP

Το XAMPP είναι ένα δωρεάν πακέτο λογισμικού ανοικτού κώδικα, όπου αναπτύχθηκε από τους Apache Friends και περιλαμβάνει κάποια από τα πιο διαδεδομένα εργαλεία που χρησιμοποιούνται για την εγκατάσταση ενός web server. Το όνομα του προέρχεται από τα αρχικά των κύριων τεχνολογιών που το απαρτίζουν Cross-Platform (X), Apache HTTP Server(A), MariaDB(M), PHP(P) και Perl(P).

Το XAMPP παρέχει την δυνατότητα στον χρήστη της εγκατάστασης ενός τοπικού web server, δεδομένου όμως ότι παρέχει ίδιες τεχνολογίες με αυτές των πραγματικών εγκαταστάσεων web server στο διαδίκτυο, καθιστά την μετάβαση μίας εφαρμογής που χρησιμοποιεί τις υπηρεσίες του XAMPP σε έναν online web server πολύ εύκολη. Το

XAMPP χρησιμοποιείται από προγραμματιστές για την δημιουργία τοπικών web server για δοκιμαστικούς σκοπούς. Επίσης το μικρό του μεγέθους, τα εργαλεία που παρέχει και η φορητότητα το καθιστά κατάλληλο εργαλείο για μαθησιακούς σκοπούς. [19] [20]

2.2.2 Apache HTTP Server

Ο Apache HTTP Server είναι ένας δωρεάν HTTP server ανοικτού κώδικα που χρησιμοποιείται ευρέως σε παγκόσμιο επίπεδο. Ο σκοπός του είναι να παρέχει υπηρεσίες σύμφωνα με το πρωτόκολλο HTTP, επιτρέποντας τη μεταφορά ιστοσελίδων προς τους χρήστες που κάνουν αιτήσεις στα προγράμματα περιήγησης τους. Θεωρείται πολύ αξιόπιστος, σταθερός και εύκολος στην χρήση του και είναι ένας από τους πιο διαδεδομένους HTTP server στο διαδίκτυο. Αναπτύχθηκε αρχικά από μία μικρή ομάδα προγραμματιστών τον Απρίλιο του 1995. Σήμερα πλέον αναπτύσσεται και συντηρείται από προγραμματιστές ανά τον κόσμο, οι οποίοι επικοινωνούν αναμεταξύ τους για την λήψη αποφάσεων μέσω email, κάτω από την αιγίδα του Apache Software Foundation. [21] [22] [23]

Ο σκοπός του Apache HTTP Server στην ανάπτυξη της εφαρμογής αυτής της πτυχιακής εργασίας είναι η φιλοξενία των αρχείων που απαρτίζουν την ιστοσελίδα τοπικά στον διακομιστή.

2.2.3 Σύστημα Διαχείρισης Βάσεων Δεδομένων MariaDB

Το MariaDB είναι ένα σύστημα διαχείρισης βάσεων δεδομένων ανοικτού κώδικα που παρέχεται δωρεάν, είναι δηλαδή μια πλατφόρμα λογισμικού που διαχειρίζεται και οργανώνει τις βάσεις δεδομένων. Το MariaDB αναπτύχθηκε λόγω των ανησυχιών για το μέλλον της MySQL μετά την εξαγορά του από την Oracle το 2009. Δημιουργήθηκε από τον Michael Widenius, έναν από τους δημιουργούς του MySQL, με στόχο τη διατήρηση της ανοικτής φύσης του κώδικα του. Το MariaDB είναι βασισμένο στον κώδικα του MySQL και διατηρεί μεγάλα επίπεδα συμβατότητας μαζί του μέχρι και σήμερα. [24] [25]

Στην εφαρμογή αυτής της εργασίας, το MariaDB διαχειρίζεται και φιλοξενεί τα δεδομένα που χρησιμοποιούνται για την οπτικοποίηση. Η εισαγωγή αυτών των δεδομένων στο MariaDB αναλύεται στο υποκεφάλαιο 3.3.3

2.2.4 Η Γλώσσα PHP

Η PHP (Hypertext Preprocessor) είναι μία προγραμματιστική γλώσσα script ανοικτού κώδικα, η οποία χρησιμοποιείται επί των πλείστον για την ανάπτυξη εφαρμογών στο διαδίκτυο. Η PHP εκτελείται στην πλευρά του διακομιστή, εφόσον υπάρχει ένας PHP Parser εγκατεστημένος, επιτρέποντας τη δημιουργία δυναμικού περιεχομένου το οποίο παράγεται από τον κώδικα της PHP και στέλνεται στην συνέχεια στο πρόγραμμα

περιήγησης του χρήστη ως αρχείο HTML. Αυτό σημαίνει ότι με την χρήση της PHP δίνεται η δυνατότητα δημιουργίας μίας δυναμικής ιστοσελίδας, όπου για παράδειγμα μπορεί να εμφανίζει περιεχόμενο με βάσει τις προτιμήσεις που έχει ο χρήστης ή την ανάκτηση δεδομένων από μία βάση δεδομένων σε πραγματικό χρόνο χωρίς την ανάγκη επαναφόρτωσης της ιστοσελίδας.

Η φύση της PHP ως γλώσσα ανοικτού κώδικα επιτρέπει στους χρήστες να τη χρησιμοποιούν, να την τροποποιούν και να τη διανέμουν χωρίς περιορισμούς. Αυτό έχει οδηγήσει στην ευρεία δημοτικότητα της PHP και σε μια μεγάλη κοινότητα προγραμματιστών οι οποίοι συντελούν στην συνεχή ανάπτυξη και βελτίωσή της.

Η PHP χρησιμοποιείται για την αλληλεπίδραση μίας εφαρμογής διαδικτύου με μία βάση δεδομένων που βρίσκεται στην πλευρά του διακομιστή. Εφόσον ο κώδικας της PHP εκτελείται στην πλευρά του διακομιστή, ο χρήστης δεν μπορεί να έχει πρόσβαση ή να τροποποιήσει τον κώδικα, γεγονός που βελτιώνει την ασφάλεια και επιτρέπει στους προγραμματιστές να δημιουργούν εφαρμογές οι οποίες ανακτούν, προσθέτουν ή τροποποιούν δεδομένα σε μία βάση δεδομένων. [26] [27] [28]

Ο ρόλος και η χρησιμότητα της PHP στην εφαρμογή αυτής της πτυχιακής εργασίας είναι η αλληλεπίδραση της εφαρμογής με το σύστημα διαχείρισης βάσεων δεδομένων, MariaDB, που βρίσκεται στην πλευρά του τοπικού διακομιστή, ο οποίος έχει στηθεί με την βοήθεια των εργαλείων που παρέχονται από το πακέτο λογισμικού XAMPP. Αρχικά η PHP χρησιμοποιείται για την σύνδεση και την επικοινωνία της ιστοσελίδας με το MariaDB. Μετέπειτα μας επιτρέπει να ανακτηθούν δεδομένα από την βάση δεδομένων και να αποθηκευτούν προσωρινά σε έναν πίνακα. Τα δεδομένα του πίνακα μεταφράζονται σε μορφή JSON για να είναι κατανοητά από την Javascript μέσω της μεθόδου `json_encode` της PHP και στην συνέχεια στέλνονται και αποθηκεύονται σε έναν πίνακα της Javascript στην πλευρά του χρήστη. Τέλος αυτός ο πίνακας δεδομένων είναι η βάση που χρησιμοποιεί η βιβλιοθήκη Chart.js για την δημιουργία των γραφημάτων. Αυτή η διαδικασία αναλύεται περεταίρω και στο τρίτο κεφάλαιο.

2.2.5 Η Γλώσσα Perl

Η Perl είναι μία προγραμματιστική γλώσσα γενικού σκοπού η οποία δημιουργήθηκε από τον Larry Wall το 1987. Η Perl αρχικά αναπτύχθηκε ως μία γλώσσα κατάλληλη για την σάρωση αρχείων κειμένου, την εξαγωγή πληροφοριών από αυτό καθώς και την εκτύπωση αναφορών με βάση αυτές τις πληροφορίες. Με την πάροδο του χρόνου εξελίχθηκε σε μία γλώσσα γενικού σκοπού με ένα ευρύ φάσμα εργασιών, όπως η διαχείριση συστημάτων, η ανάπτυξη διαδραστικών εφαρμογών διαδικτύου, ο προγραμματισμός δικτύων, η ανάπτυξη γραφικών διεπαφών και πολλά άλλα. Οι

δυνατότητες της γλώσσας υποστηρίζουν τόσο δομημένο όσο και αντικειμενοστραφή προγραμματισμό. [29] [30]

Η γλώσσα προορίζεται να είναι περισσότερο πρακτική και λιγότερο όμορφη και κομψή. Επίσης είναι εύκολη στην εκμάθησή της και σχεδιάστηκε έτσι ώστε η σύνταξή της να είναι πιο κοντά στην ανθρώπινη γλώσσα παρά σε γλώσσα μηχανής.

Τέλος είναι μία από τις πιο φορητές γλώσσες προγραμματισμού και μπορεί να τρέξει σε πάνω από 70 λειτουργικά συστήματα. [31]

2.3 Άλλες Τεχνολογίες

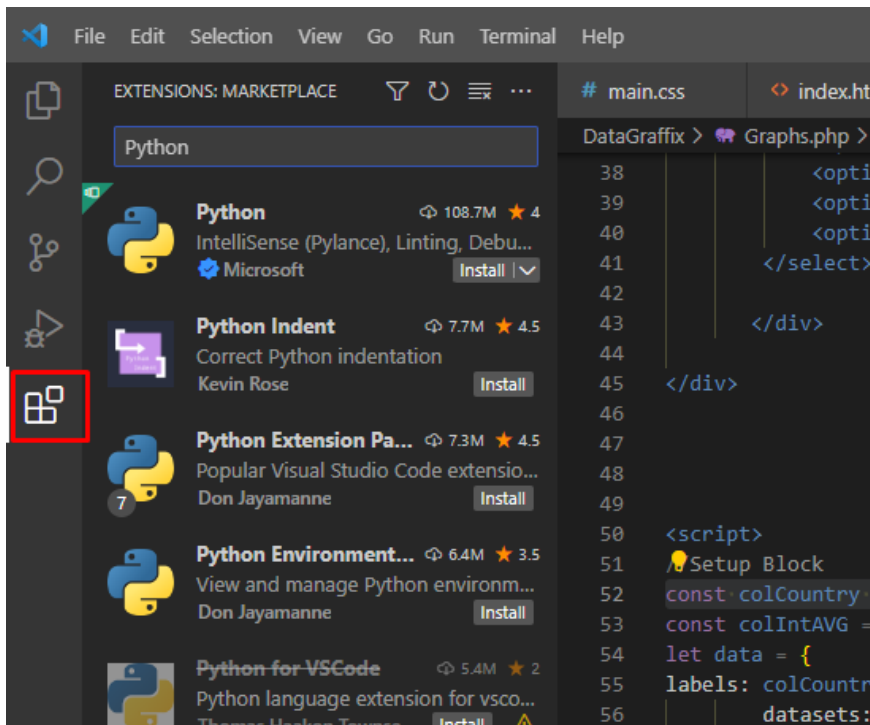
2.3.1 Visual Studio Code

Το περιβάλλον ανάπτυξης κώδικα στο οποίο γράφτηκε όλος ο κώδικας της εφαρμογής είναι το Visual Studio Code. Το VS Code είναι ένας δωρεάν text editor ανοικτού κώδικα που αναπτύσσεται και διανέμεται από την Microsoft.

Το Visual Studio Code παρέχει στους προγραμματιστές υποστήριξη για μία πληθώρα διαφορετικών γλωσσών προγραμματισμού μέσω των επεκτάσεων, τόσο της Microsoft όσο και third-party επεκτάσεις, οι οποίες είναι διαθέσιμες και άμεσα προσβάσιμες στο Marketplace εντός της εφαρμογής. (Εικόνα 7) [32] [33]

Το Visual Studio Code είναι πολύ ελαφρύ και έχει πολύ μικρές απαιτήσεις συστήματος, γεγονός που του επιτρέπει να τρέχει σχεδόν σε οποιοδήποτε υπολογιστικό σύστημα. Πέρα από αυτό είναι πολύ μικρό σε απαιτήσεις αποθηκευτικού χώρου και καταλαμβάνει μόλις 200 MB. Έπειτα είναι cross-platform και τρέχει σε Windows, macOS και Linux. [34]

Τέλος η επέκταση Live Share επιτρέπει την συνεργασία δύο ατόμων πάνω στον ίδιο κώδικα, δουλεύοντας ο καθένας από τον δικό του text editor και διατηρώντας ο καθένας τις ρυθμίσεις του. Οποιοσδήποτε από τους δύο έχει την δυνατότητα να ανοίξει αρχεία, να πλοηγηθεί και να επεξεργαστεί τον κώδικα και οι αλλαγές έχουν αντίκτυπο στο κοινό αρχείο σε πραγματικό χρόνο. [35]



Εικόνα 7 Αναζήτηση επεκτάσεων στο Marketplace

2.3.2 Δεδομένα στο Διαδίκτυο

Υπάρχουν πολλοί τρόποι να αποκτήσει κανείς δεδομένα μέσω του διαδικτύου. Ένας από αυτούς τους τρόπους είναι η απόκτηση δεδομένων μέσω ιστοσελίδων που παρέχουν υπηρεσίες διαμοίρασης δεδομένων.

Μία από αυτές τις ιστοσελίδες είναι το Kaggle, το οποίο είναι μία πλατφόρμα που ανήκει στην Google και είναι κατάλληλη για εκμάθηση για τους ενδιαφερόμενους πάνω στον τομέα της επιστήμης των δεδομένων και του Machine Learning. Η πλατφόρμα επιτρέπει στους χρήστες να δημοσιεύουν ολόκληρα σύνολα δεδομένων καθώς και να ψάξουν δημοσιευμένα δεδομένα από άλλους χρήστες και να τα κατεβάσουν.

Το Kaggle υποστηρίζει διάφορες μορφές δεδομένων προς δημοσίευση των χρηστών αλλά συνιστανται οι ανοικτές και προσβάσιμες μορφές δεδομένων όπως η CSV, JSON, SQLite και BigQuery. [36] [37]

3. Υλοποίηση Εφαρμογής

3.1 Η Δομή της Εφαρμογής

Στις μέρες μας υπάρχουν πάρα πολλοί ιστότοποι οι οποίοι χρησιμοποιούν κάποιου είδους οπτικοποίηση δεδομένων, για να παρέχουν στους χρήστες μία πιο εύπεπτη εικόνα των δεδομένων που υπάρχουν, κάνοντας χρήση κάποιων γραφημάτων, τα οποία βοηθούν τους χρήστες να βγάλουν πιο εύκολα συμπεράσματα μέσω μίας γραφικής απεικόνισης των δεδομένων και να παρατηρήσουν μοτίβα και πιθανές σχέσεις αναμεταξύ τους.

Ο κύριος σκοπός αυτής της πτυχιακής εργασίας είναι η ανάπτυξη μίας τέτοιας διαδικτυακής εφαρμογής, η οποία θα μπορεί να οπτικοποιεί κάποια δεδομένα, τα οποία βρίσκονται αποθηκευμένα στον τοπικό διακομιστή. Για να επιτευχθεί αυτό χρειάζεται η αρμονική συνεργασία διαφόρων τεχνολογιών, οι οποίες τρέχουν είτε στην πλευρά του χρήστη(client-side) είτε στην πλευρά του διακομιστή(server-side), επιτρέποντας έτσι την ομαλή επικοινωνία μεταξύ των δύο πλευρών.

Το πρώτο κομμάτι της εφαρμογής αποτελεί η ιστοσελίδα. Η επαφή που έχει ο χρήστης με την εφαρμογή είναι μέσω της ιστοσελίδας που εμφανίζεται στο πρόγραμμα περιήγησής του, η οποία χρησιμοποιείται κυρίως ως ένας καμβάς, πάνω στον οποίο πραγματοποιείται η οπτικοποίηση δεδομένων κάνοντας χρήση της βιβλιοθήκης Chart.js της JavaScript. Ταυτόχρονα η ιστοσελίδα είναι μία διεπαφή ανάμεσα στον χρήστη και την εφαρμογή, μέσω της οποίας ο χρήστης έχει την δυνατότητα να επιλέγει ο ίδιος το είδος του γραφήματος που θέλει να απεικονιστεί. Τέλος του δίνεται η δυνατότητα να πραγματοποιήσει ο ίδιος την δική του οπτικοποίηση παρέχοντας τα δικά του δεδομένα στην ιστοσελίδα.

Ύστερα, το δεύτερο κομμάτι της εφαρμογής περιλαμβάνει τον τοπικό διακομιστή. Συγκεκριμένα περιλαμβάνει τεχνολογίες που παρέχει το πακέτο λογισμικού XAMPP, οι οποίες δίνουν την δυνατότητα της εγκατάστασης ενός τοπικού διακομιστή, ο οποίος απαρτίζεται από τον Apache HTTP Server, το MariaDB και την PHP. Οι λειτουργίες αυτού του κομματιού της εφαρμογής είναι η φιλοξενία των απαραίτητων αρχείων της ιστοσελίδας στο σύστημα φακέλων του Apache Server καθώς και η διαχείριση των HTTP αιτημάτων από τον χρήστη. Ο PHP parser με την σειρά του εκτελεί οποιοδήποτε script της PHP υπάρχει στα αρχεία της ιστοσελίδας. Η PHP είναι επίσης υπεύθυνη για την σύνδεση της εφαρμογής με το σύστημα διαχείρισης βάσεων δεδομένων MariaDB καθώς και για την ανάκτηση δεδομένων από αυτήν, στα οποία θα βασιστεί μετέπειτα η εφαρμογή για την δημιουργία των γραφημάτων στην ιστοσελίδα.

3.2 Τα Αρχεία της Ιστοσελίδας

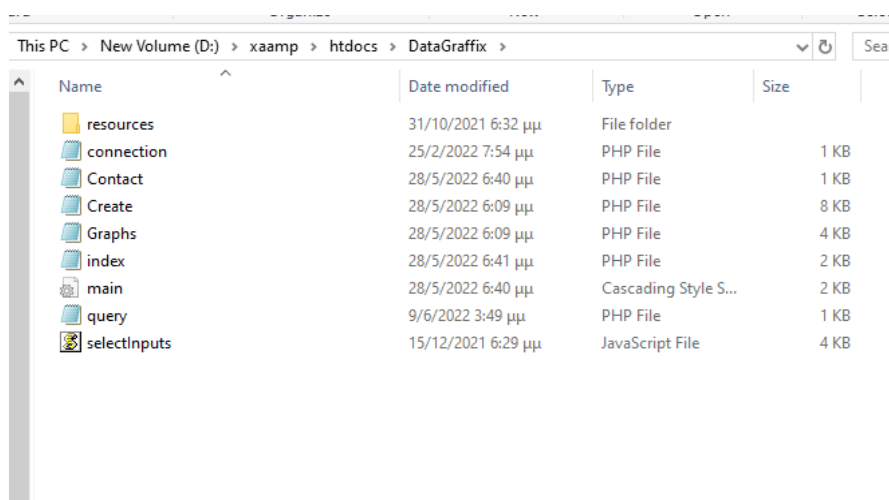
Όπως προαναφέρθηκε η ιστοσελίδα είναι μία διεπαφή ανάμεσα στον χρήστη και την εφαρμογή και είναι επίσης το μέσο πάνω στο οποίο πραγματοποιείται η οπτικοποίηση δεδομένων με την χρήση της βιβλιοθήκης Chart.js.

Η ιστοσελίδα αποτελείται από πέντε βασικά αρχεία PHP το **index.php**, **Graphs.php**, **Create.php**, **Upload.php** και **About.php** τα οποία καθορίζουν την δομή της και αποτελούνται κυρίως από κώδικα HTML. Το καθένα είναι προσβάσιμο από την μπάρα περιήγησης της ιστοσελίδας. Υπάρχει επίσης το αρχείο **main.css**, το οποίο εμπεριέχει κανόνες CSS που εφαρμόζονται για να μορφοποιήσουν την ιστοσελίδα στυλιστικά.

Το αρχείο **selectInput.js** είναι ένα αρχείο JavaScript που περιέχει την συνάρτηση **selectInput()**. Η συνάρτηση αυτή χρησιμοποιείται στο **Create.php**. Εκεί ο χρήστης έχει την δυνατότητα να προσθέτει ο ίδιος τις δικές του παραμέτρους και δεδομένα στην ιστοσελίδα και να φτιάχνει τα δικά του γραφήματα. Η συνάρτηση **selectInput()** βοηθάει να εμφανίζονται τόσα κενά πεδία όσα έχει επιλέξει ο χρήστης να εμφανιστούν, στα οποία μετέπειτα μπορεί να προσθέσει τα δεδομένα του.

Ύστερα υπάρχει το **connection.php**, το οποίο είναι ένα script γραμμένο σε PHP. Ο σκοπός του είναι να συνδέει την εφαρμογή με την βάση δεδομένων που υπάρχει στο στην πλευρά του διακομιστή. Υπάρχει επίσης το **query.php** το οποίο είναι ένα script υπεύθυνο για την ανάκτηση δεδομένων, κάνοντας ένα SQL query στην βάση. Στην συνέχεια αποθηκεύει αυτά τα ανακτημένα δεδομένα σε πίνακες.

Τέλος αυτά τα αρχεία είναι συγκεντρωμένα σε έναν φάκελο με την ονομασία της ιστοσελίδας στην τοποθεσία **\xampp\htdocs\DataGraffix** όπως φαίνεται στην εικόνα 8. Ο πλήρης κώδικας της εφαρμογής μπορεί να βρεθεί στο GitHub στην τοποθεσία: <https://github.com/morsWasTaken/DataGraffix>



Εικόνα 8 Η τοποθεσία των αρχείων της ιστοσελίδας

3.2.1 Η Επικεφαλίδα

Η επικεφαλίδα **<head>** της HTML συνήθως περιλαμβάνει πληροφορίες για το έγγραφο όπως τον τίτλο της σελίδας, συνδέσμους προς εξωτερικούς πόρους και metadata όπως φαίνεται στην Εικόνα 9. Οι πληροφορίες αυτές παρέχουν σημαντικά μεταδεδομένα που χρησιμοποιούνται από τα προγράμματα περιήγησης για την κατανόηση και την σωστή παρουσίαση του περιεχομένου μίας ιστοσελίδας.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <link href="main.css" rel="stylesheet" type="text/css">
6   <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.5.1/chart.min.js"></script>
7   <?php require "connection.php" ?>
8   <title>DataGraffix</title>
9 </head>
```

Εικόνα 9 Metadata και εισαγωγή εξωτερικών πόρων στο έγγραφο

Πριν από το **<head>**, στην γραμμή 1 η δήλωση **<!DOCTYPE html>** προσδιορίζει στο πρόγραμμα περιήγησης ότι ο τύπος του εγγράφου είναι HTML5 και το **<html lang="en">** υποδεικνύει ότι η γλώσσα του εγγράφου είναι η αγγλική.

Η ετικέτα **<meta charset="UTF-8">** καθορίζει την κωδικοποίηση χαρακτήρων που χρησιμοποιείται στην ιστοσελίδα και χρησιμοποιείται για να διασφαλιστεί η σωστή ερμηνεία των χαρακτήρων από τον πρόγραμμα περιήγησης.

Το **<link href="main.css" rel="stylesheet" type="text/css">** χρησιμοποιείται για να συνδέσει το έγγραφο HTML με το εξωτερικό αρχείο CSS με όνομα **main.css**, το οποίο χρησιμοποιείται για την μορφοποίηση της ιστοσελίδας.

Το **script** στην γραμμή 6 είναι απαραίτητο γιατί εισάγει τη βιβλιοθήκη Chart.js στην ιστοσελίδα. Αυτό επιτυγχάνεται κατεβάζοντας τον κώδικα της βιβλιοθήκης από το Cloudflare CDN και ενσωματώνοντας τον στη σελίδα.

Τέλος το **<?php require "connection.php" ?>** εισάγει το **script connection.php** στην ιστοσελίδα και η εντολή **require** απαιτεί την ομαλή εκτέλεση του. Σε περίπτωση που το **script** δεν βρεθεί ή αποτύχει να εκτελεστεί, τότε το πρόγραμμα θα σταματήσει και θα εμφανιστεί το αντίστοιχο μήνυμα σφάλματος.

3.2.2 Η Μπάρα Περιήγησης

Η μπάρα περιήγησης σε μια ιστοσελίδα είναι μία περιοχή που περιλαμβάνει συνήθως συνδέσμους ή κουμπιά που επιτρέπουν στους χρήστες να πλοηγηθούν σε διάφορες σελίδες ή τμήματα της ιστοσελίδας. Αποτελεί ένα σημαντικό στοιχείο κάθε

ιστοσελίδας, διότι παρέχει στους χρήστες την δυνατότητα περιήγησης σε διάφορα τμήματα της ιστοσελίδας και στο περιεχόμενό τους.

Ο κώδικας για την μπάρα περιήγησης βρίσκεται τόσο στο αρχείο **index.php**, όσο και στα **Graphs.php**, **Create.php** και **About.php**. Αυτό σημαίνει ότι σε οποιοδήποτε κομμάτι της ιστοσελίδας περιηγηθεί ο χρήστης, θα υπάρχει πάντα μία μπάρα περιήγησης στην οθόνη του που θα του επιτρέπει να επισκεφθεί ένα άλλο τμήμα της ιστοσελίδας.

```
14 | <div class="navbar">
15 |     <a href="index.php">Home</a>
16 |     <a href="Graphs.php">Graphs</a>
17 |     <a href="Create.php">Create</a>
18 |     <a href="About.php">About</a>
19 | </div>
```

Εικόνα 10 Ο HTML κώδικας πίσω από την μπάρα περιήγησης

Στον παραπάνω κομμάτι του κώδικα βλέπουμε τα HTML στοιχεία που αποτελούν την μπάρα περιήγησης. Από την γραμμή 15 ως 18 η ετικέτα **<a>** ορίζει κάποιους συνδέσμους, στους οποίους όταν θα κάνει κλικ ο χρήστης, ο σύνδεσμος θα τον οδηγήσει στο αντίστοιχο τμήμα της σελίδας που καθορίζεται από το **href=""**. Ύστερα η ετικέτα **<div class="navbar">** δημιουργεί ένα κοντέινερ κλάσης "navbar" που περιλαμβάνει αυτούς τους συνδέσμους και τους ομαδοποιεί κάτω από την ίδια κλάση. Με την βοήθεια αυτής της κλάσης είναι εφικτή η μορφοποίηση της μπάρας περιήγησης από τους κανόνες CSS του **main.css**.

```
17 | /* Navbar container */
18 | v .navbar {
19 |     overflow: hidden;
20 |     background-color: #333;
21 |     font-family: Arial;
22 | }
23 |
24 | /* Links inside the navbar */
25 | v .navbar a {
26 |     float: left;
27 |     font-size: 18px;
28 |     color: white;
29 |     text-align: center;
30 |     padding: 14px 16px;
31 |     text-decoration: none;
32 | }
```

Εικόνα 11 Οι κανόνες CSS για την μπάρα περιήγησης

Ένας από τους κανόνες που αξίζει να αναφερθεί είναι το **float: left;** το οποίο επιτρέπει στους συνδέσμους που αποτελούν την μπάρα περιήγησης να εμφανίζονται ο

ένας δίπλα στον άλλον, δίνοντας την εμφάνιση μίας οριζόντιας μπάρας περιήγησης. Επίσης το **text-decoration: none;** χρησιμοποιείται για να αφαιρεθεί οποιαδήποτε διακόσμηση του κειμένου. Αυτό έχει ως αποτέλεσμα στην περίπτωση της μπάρας περιήγησης να αφαιρεθεί η υπογράμμιση των συνδέσμων της.

3.2.3 Η Σελίδα index.php

Το αρχείο index.php αντιπροσωπεύει την αρχική σελίδα της εφαρμογής. Ο κώδικας που περιλαμβάνεται στο **index.php** εκτελείται όταν ο χρήστης επισκέπτεται για πρώτη φορά την ιστοσελίδα. Το **index.php** περιέχει ένα μήνυμα καλωσορίσματος και μία εικόνα ενός ραβδογράμματος.

3.2.4 Το Script Connection.php

Είναι σημαντικό πρώτα να γίνει μία επεξήγηση του **connection.php**. Όπως είδαμε προηγουμένως, στο <head> των αρχείων της ιστοσελίδας υπάρχει το script **<?php require "connection.php" ?>** μέσω του οποίου εισάγεται και απαιτείται η σωστή εκτέλεση του **connection.php**. Το **connection.php** είναι υπεύθυνο για την σύνδεση μεταξύ της εφαρμογής και της βάση δεδομένων χρησιμοποιώντας την διεπαφή PDO(PHP Data Objects).

```
1  <?php
2
3  // Definition of variables for the database connection
4  $username = "root";
5  $password = "";
6  $database = "internetavg";
7
8  // Attempt to create a PDO connection to the database
9  try {
10     $pdo = new PDO("mysql:host=localhost;database=$database", $username, $password);
11     // Setting error handling mode for PDO
12     $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
13 } catch(PDOException $e){
14     // If there is an error during connection, print error message
15     die("ERROR: Could not connect. " . $e->getMessage());
16 }
17
18 ?>
```

Εικόνα 12 Το script connection.php

Το script ορίζει αρχικά τα στοιχεία σύνδεσης στη βάση δεδομένων. Συγκεκριμένα, ορίζονται μεταβλητές για το username (όνομα χρήστη), το password (κωδικός πρόσβασης) και το database (όνομα της βάσης δεδομένων) και τις τιμές που παίρνουν αντίστοιχα.

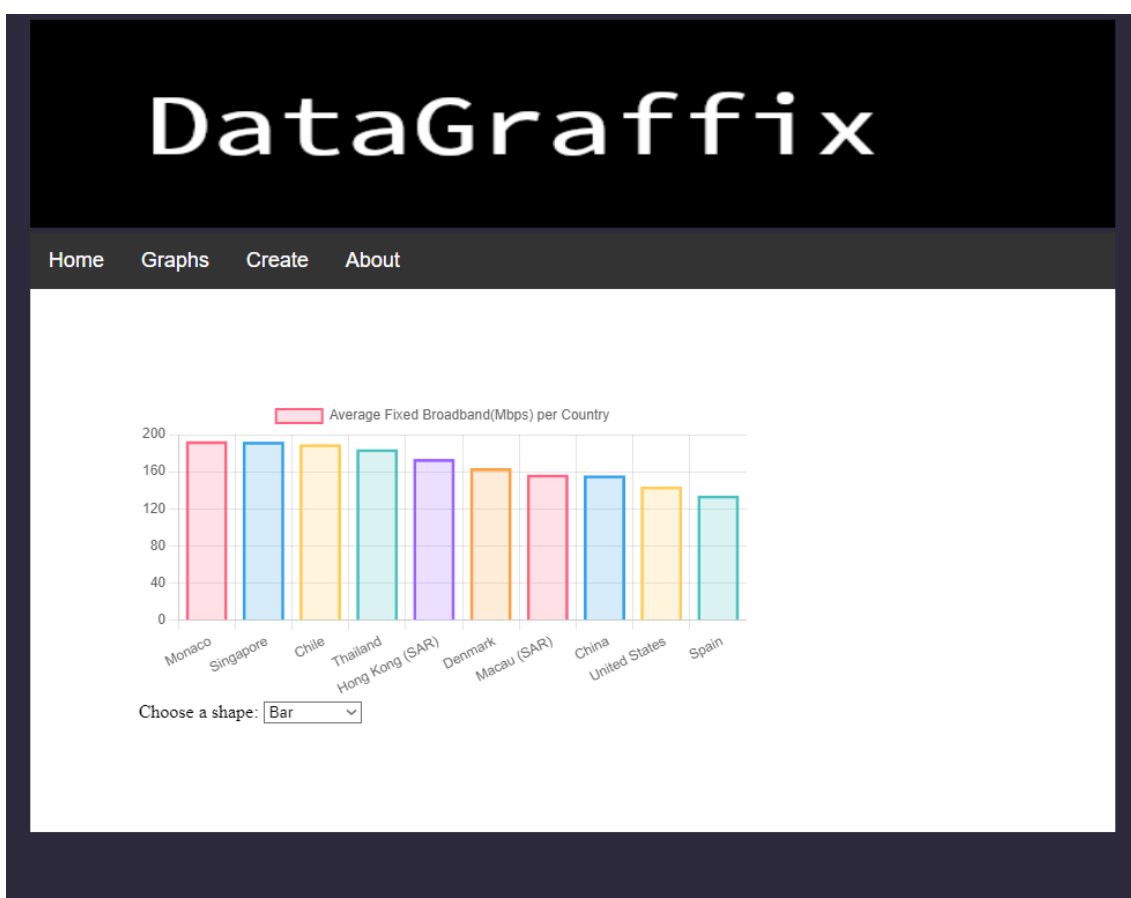
Στην συνέχεια χρησιμοποιείται η δομή try-catch μέσω της οποίας προσπαθεί να δημιουργήσει μία σύνδεση με την βάση δεδομένων. Αυτό πραγματοποιείται με την

μέθοδο **new PDO()**, η οποία δημιουργεί ένα αντικείμενο PDO που αντιπροσωπεύει τη σύνδεση προς τη βάση δεδομένων η οποία βρίσκεται στην MariaDB.

Τέλος στην γραμμή 10 δηλώνεται ότι εάν συμβεί οποιοδήποτε σφάλμα κατά την εκτέλεση ενός SQL ερωτήματος, το PDO θα εγείρει μια εξαίρεση τύπου PDOException για την καλύτερη αντιμετώπιση των σφαλμάτων. Εάν προκύψει κάποιο σφάλμα κατά τη σύνδεση, πιάνεται η εξαίρεση (PDOException) από το catch και στην συνέχεια, όπως φαίνεται στην γραμμή 12, εμφανίζεται ένα μήνυμα λάθους και τερματίζεται η εκτέλεση του προγράμματος.

3.2.5 Η Σελίδα Graphs.php

Το **Graphs.php** περιλαμβάνει τον κώδικα για το τμήμα της ιστοσελίδας που ασχολείται με την οπτικοποίηση των δεδομένων που ανακτώνται από την βάση δεδομένων στην πλευρά του τοπικού διακομιστή. Αυτό το μέρος της σελίδας παρουσιάζει στον χρήστη τα δεδομένα με τρόπο γραφικό και του δίνει την δυνατότητα να αλλάξει τον τύπο του γραφήματος που παρουσιάζονται τα δεδομένα μέσω ενός drop-down menu.



Εικόνα 13 Η εμφάνιση του Graphs.php στο πρόγραμμα περιήγησης

Ο κώδικας του **Graphs.php** αποτελείται κυρίως από δύο μεγάλα κομμάτια, το κομμάτι της HTML και ένα κομμάτι Javascript που είναι ενσωματωμένο μέσα στο έγγραφο. Τα δύο αυτά κομμάτια συνεργάζονται για να επιτευχθεί η οπτικοποίηση.

```
22     <?php include 'query.php';?>
23
24     <div class="graphs">
25
26         <div class="chartBox">
27             <canvas id="myChart" ></canvas>
28         </div>
29         <div class="dropdown">
30             <label for="shapes">Choose a shape:</label>
31             <select name="shapes" id="shapes" onchange="changeShape(this)">
32                 <option value="bar">Bar</option>
33                 <option value="line">Line</option>
34                 <option value="pie">Pie</option>
35                 <option value="doughnut">Doughnut</option>
36                 <option value="radar">Radar</option>
37                 <option value="polarArea">Polar Area</option>
38             </select>
39         </div>
40
41     </div>
42
43 </div>
```

Εικόνα 14 Ορισμός της περιοχής καμβά και δημιουργία drop-down για την επιλογή γραφήματος

Η εντολή στην γραμμή 22 χρησιμοποιείται για να εισάγει και να εκτελεστεί το script **query.php** με την χρήση της εντολής `include`. Όταν εκτελεστεί αυτό το script, θα ανακτηθούν δεδομένα από την βάση μέσω ενός query και θα αποθηκευτούν σε δύο πίνακες της PHP. Αυτά τα δεδομένα θα χρησιμοποιηθούν μετά για την δημιουργία του γραφήματος. Ο κώδικας του **query.php** θα αναλυθεί περαιτέρω παρακάτω.

Στην γραμμή 28 ορίζεται στο έγγραφο μία περιοχή κανβά με την ετικέτα **<canvas>**, πάνω στην οποία θα σχεδιαστεί το γράφημα.

Από την γραμμή 30 ως την γραμμή 39 φαίνεται η δημιουργία ενός drop-down με την ετικέτα "Choose a shape" που εμφανίζεται δίπλα του. Το μενού περιλαμβάνει τις επιλογές που έχει ο χρήστης για τον τύπο του γραφήματος που θέλει να απεικονιστεί. Στην γραμμή 32 έχει οριστεί να καλείται η συνάρτηση `changeShape()` κάθε φορά που ο χρήστης αλλάζει την επιλογή του στο drop-down. Η συνάρτηση αυτή χρησιμοποιείται για να αλλάξει τον τύπο του γραφήματος ανάλογα με την επιλογή του χρήστη στο drop-down.

Όπως αναφέρθηκε προηγουμένως το script **query.php** ανακτεί δεδομένα από την βάση και τα αποθηκεύει σε πίνακες της PHP. Αυτά τα δεδομένα δεν μπορούν να διαβαστούν σε αυτήν την μορφή από την Javascript. Έτσι χρησιμοποιείται η συνάρτηση

json_encode() της PHP για να μετατραπούν τα δεδομένα σε μορφή JSON την οποία μπορεί να διαβάσει η Javascript.

```
48 <script>
49 //Setup Block
50 const colCountry = <?php echo json_encode($colCountry); ?>;
51 const colIntAVG = <?php echo json_encode($colIntAVG); ?>; /,
```

Εικόνα 15 Μετατροπή των δεδομένων σε μορφή JSON

Όπως φαίνεται στην γραμμή 50 και 51, τα δεδομένα στους πίνακες \$colCountry και \$colIntAvg της PHP μετατρέπονται σε μορφή JSON και στην συνέχεια τα δεδομένα αποθηκεύονται στους αντίστοιχους πίνακες της Javascript. Ο πίνακας colCountry εμπεριέχει ονόματα από χώρες και ο colIntAvg την μέση ταχύτητα ευρυζωνικής σύνδεσης στο ίντερνετ των χωρών αυτών.

Στην συνέχεια θα αναλυθεί ο κώδικας που είναι απαραίτητος για την δημιουργία ενός αντικειμένου γραφήματος της Chart.js

```
52 let data = {
53   labels: colCountry, // Define labels for the chart using the Cou
54   datasets: [{
55     label: 'Average Fixed Broadband(Mbps) per Country',
56     data: colIntAVG, // Definition of the data for the c
57     backgroundColor: [
58       'rgba(255, 99, 132, 0.2)',
59       'rgba(54, 162, 235, 0.2)',
60       'rgba(255, 206, 86, 0.2)',
61       'rgba(75, 192, 192, 0.2)',
62       'rgba(153, 102, 255, 0.2)',
63       'rgba(255, 159, 64, 0.2)'
64     ],
65     borderColor: [
66       'rgba(255, 99, 132, 1)',
67       'rgba(54, 162, 235, 1)',
68       'rgba(255, 206, 86, 1)',
69       'rgba(75, 192, 192, 1)',
70       'rgba(153, 102, 255, 1)',
71       'rgba(255, 159, 64, 1)'
72     ],
73     borderWidth: 2
74   }]
75 };
76
77 //Configuration Block
78 let config = {
79   type: "bar", // Define the type of chart (bar chart)
80   data, // Pass the data object to the chart
81   options: {
82     scales: {
83       y: {
84         beginAtZero: true
85       }
86     }
87   }
88 }
```

Εικόνα 16 Η αρχικοποίηση των αντικειμένων γραφημάτων

Στην αρχή δημιουργείται ένα αντικείμενο με το όνομα `data` που εμπεριέχει όλη την αρχικοποίηση των δεδομένων για τα γραφήματα μέσα του. Στην γραμμή 53, στην θέση των ετικετών του γραφήματος(`label`) προστίθενται τα δεδομένα του πίνακα `colCountry`, δηλαδή τα ονόματα των χωρών. Αυτό θα έχει ως αποτέλεσμα να αντιστοιχεί κάθε τιμή του γραφήματος σε μία ετικέτα. Επίσης στην γραμμή 55 ορίζεται μία ετικέτα η οποία λειτουργεί ως ο τίτλος του γραφήματος. Στην γραμμή 56 ορίζονται τα αριθμητικά δεδομένα του γραφήματος από τον πίνακα `colIntAvg`. Στην συνέχεια υπάρχουν κάποιες ρυθμίσεις για τα χρώματα του γραφήματος.

Ύστερα δημιουργείται ένα άλλο αντικείμενο με όνομα `config` που εμπεριέχει την ρύθμιση του τύπου του γραφήματος σε `"bar"` καθώς επίσης και το αντικείμενο `data` που δημιουργήθηκε προηγουμένως που αντιπροσωπεύει τα δεδομένα των γραφημάτων. Το αντικείμενο `config` είναι πλέον αρχικοποιημένο με τα κατάλληλα δεδομένα και μπορεί να χρησιμοποιηθεί για την κατασκευή αντικειμένων γραφήματος.

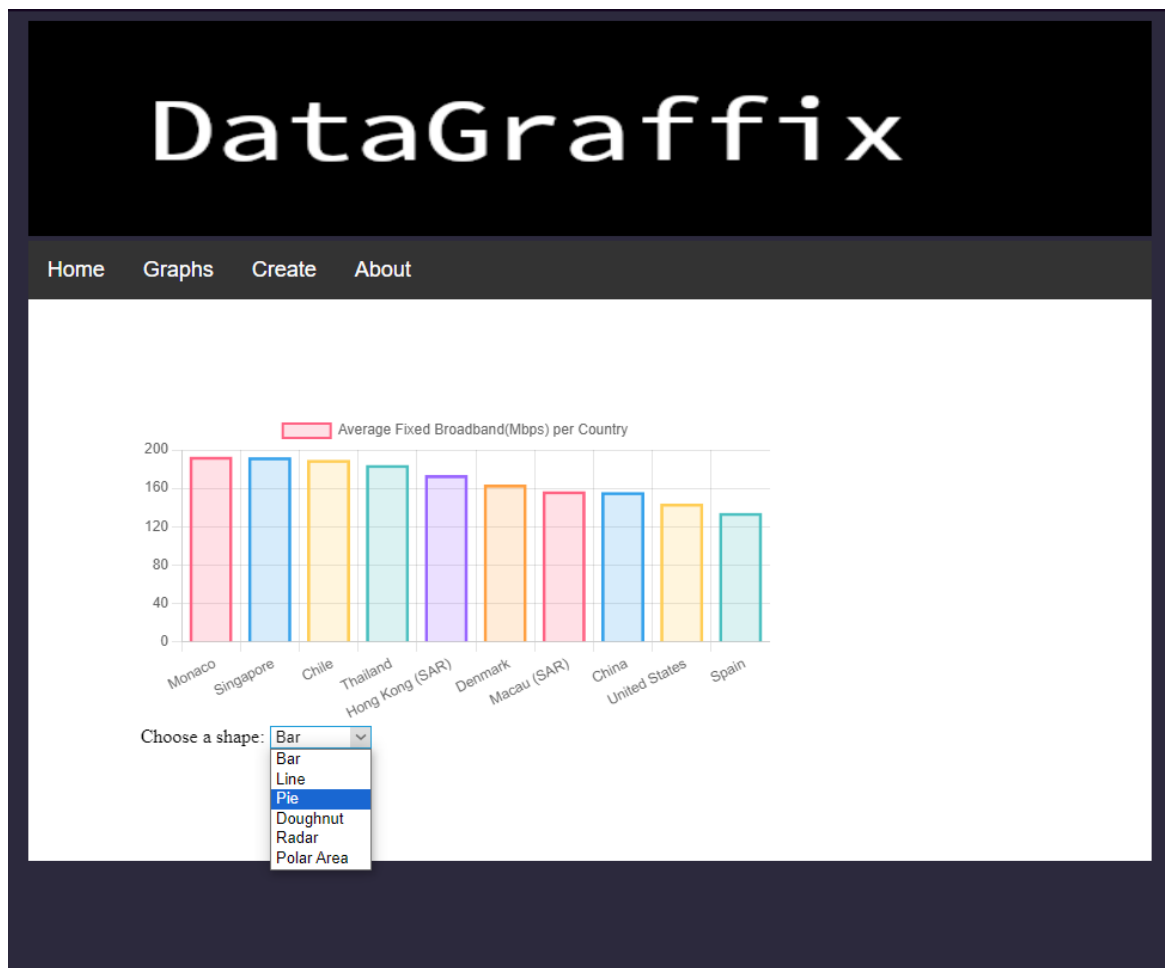
Στην γραμμή 94 δημιουργείται ένα αντικείμενο γραφήματος με όνομα `myChart` χρησιμοποιώντας την κλάση `Chart` της βιβλιοθήκης **Chart.js** όπως φαίνεται στην εικόνα 17. Το αντικείμενο γραφήματος αντιστοιχεί στο στοιχείο με `id="myChart"` του DOM, το οποίο είναι το στοιχείο με την ετικέτα `<canvas>`, και η αρχικοποίηση των δεδομένων και των ρυθμίσεων του αντικειμένου γίνονται με την βοήθεια του αντικειμένου `config`.

```
92 //Render Block
93 //Creates a chart Object
94 let myChart = new Chart(
95     document.getElementById("myChart"),
96     config
97 );
98
99
100 //Change Shape Function
101 function changeShape(chartShape){
102
103
104     myChart.destroy();
105     myChart = new Chart(
106         document.getElementById("myChart"),
107         {
108             type: chartShape.value,
109             data,
110             options: {
111
112                 scales: {
113                     y: {
114                         beginAtZero: true
115                     }
116                 }
117             }
118         }
119     );
120
121 }
```

Εικόνα 17 Αλλαγή τύπου γραφήματος μέσω της συνάρτησης `changeShape()`

Αυτός το κομμάτι κώδικα δημιουργεί ένα ραβδόγραμμα “bar” την πρώτη φορά που ο χρήστης θα φορτώσει την σελίδα στο πρόγραμμα περιήγησης του.

Στην συνέχεια ακολουθεί η συνάρτηση **changeShape()** η οποία είναι υπεύθυνη για την αλλαγή του τύπου του γραφήματος κάθε φορά που ο χρήστης αλλάζει επιλογή στο drop-down menu. Αρχικά η συνάρτηση καταστρέφει το παλιό αντικείμενο γραφήματος που ήδη υπάρχει με την μέθοδο **myChart.destroy()**. Ύστερα δημιουργείται ένα νέο αντικείμενο γραφήματος με όνομα myChart χρησιμοποιώντας επίσης την κλάση Chart της βιβλιοθήκης Chart.js. Στην συνέχεια γίνεται η αρχικοποίηση δεδομένων και ρυθμίσεων του αντικείμενου γραφήματος με τον ίδιο τρόπο όπως το αντικείμενο config με την μόνη διαφορά ότι πλέον ο τύπος του γραφήματος δεν είναι σταθερός αλλά καθορίζεται από την επιλογή του χρήστη όπως φαίνεται στην γραμμή 107. Αυτό έχει ως αποτέλεσμα κάθε φορά που αλλάζει ο χρήστης τον τύπο γραφήματος στο drop-down, αυτή η τιμή να αναθέτεται στο type μέσα στην συνάρτηση και στην συνέχεια να καλείται η συνάρτηση και να δημιουργεί ένα γράφημα με βάση την επιλογή του χρήστη.



Εικόνα 18 Επιλογή τύπου γραφήματος στην καρτέλα Graphs

3.2.6 To Script query.php

Το script **query.php** είναι υπεύθυνο για την λήψη των δεδομένων από την βάση που υπάρχει στο MariaDB. Κατά την εκτέλεση του script γίνεται ένα ερώτημα SQL στην βάση δεδομένων και στην συνέχεια τα δεδομένα ανακτώνται και αποθηκεύονται σε δύο πίνακες της PHP.

```
1  <?php
2  // Attempt select query execution
3  try{
4      $sql = "SELECT * FROM internetavg.intavgjanuary2022 LIMIT 10";
5
6      // Execute the query using PDO
7      $result = $pdo->query($sql);
8
9      // Check if any rows were returned by the query
10     if($result->rowCount() > 0) {
11         // Initialize arrays to store column values
12         $colCountry = array(); // Variable to store country values
13         $colIntAVG = array(); // Variable to store internet average values
14
15         // Loop through each row returned by the query
16         while($row = $result->fetch()) {
17             // Store country and internet average values in respective arrays
18             $colCountry[] = $row["Country"];
19             $colIntAVG[] = $row["Broadband Mbps"];
20         }
21
22         // Release the result set memory
23         unset($result);
24     } else {
25         // If no records were found, output a message
26         echo "No records matching your query were found.";
27     }
28 } catch(PDOException $e){
29     //error message
30     die("ERROR: Could not able to execute $sql. " . $e->getMessage());
31 }
32
33 // Close connection
34 unset($pdo);
35 ?>
```

Εικόνα 19 Ο κώδικας του query.php

Ο κώδικας είναι γραμμένος μέσα σε ένα try-catch block για την αντιμετώπιση σφαλμάτων που μπορεί να προκύψουν κατά την εκτέλεση του. Αν προκύψει κάποιο σφάλμα, η εξαίρεση PDOException πιάνεται και εμφανίζεται ένα μήνυμα λάθους.

Στην γραμμή 4 δημιουργείται μία μεταβλητή με όνομα **\$sql** και της ανατίθεται το string `"SELECT * FROM internetavg.intavgjanuary2022 LIMIT 10"`. Η μεταβλητή **\$sql**

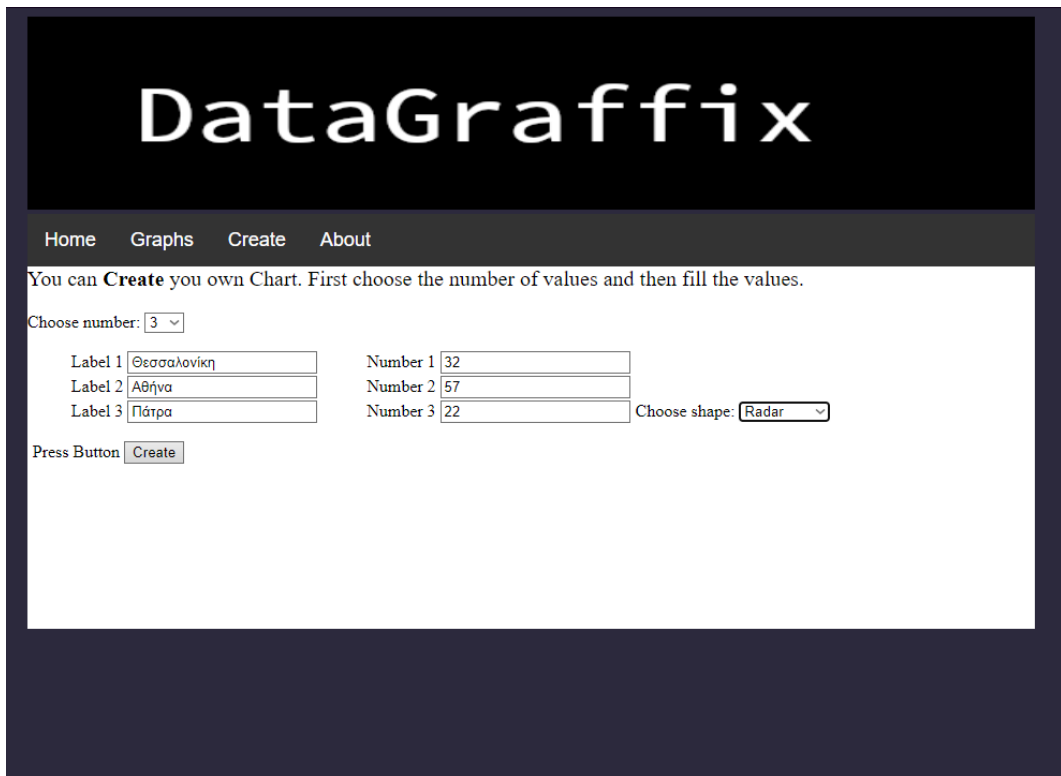
αντιπροσωπεύει το ερώτημα SQL που γίνεται στην βάση δεδομένων, το οποίο επιστρέφει όλα τα στοιχεία του συγκεκριμένου table από τις 10 πρώτες εγγραφές.

Στην συνέχεια εκτελείται το query μέσω της διεπαφής PDO (PHP Data Objects) χρησιμοποιώντας την μεταβλητή `$sql` και αποθηκεύονται τα αποτελέσματα στη μεταβλητή `$result`.

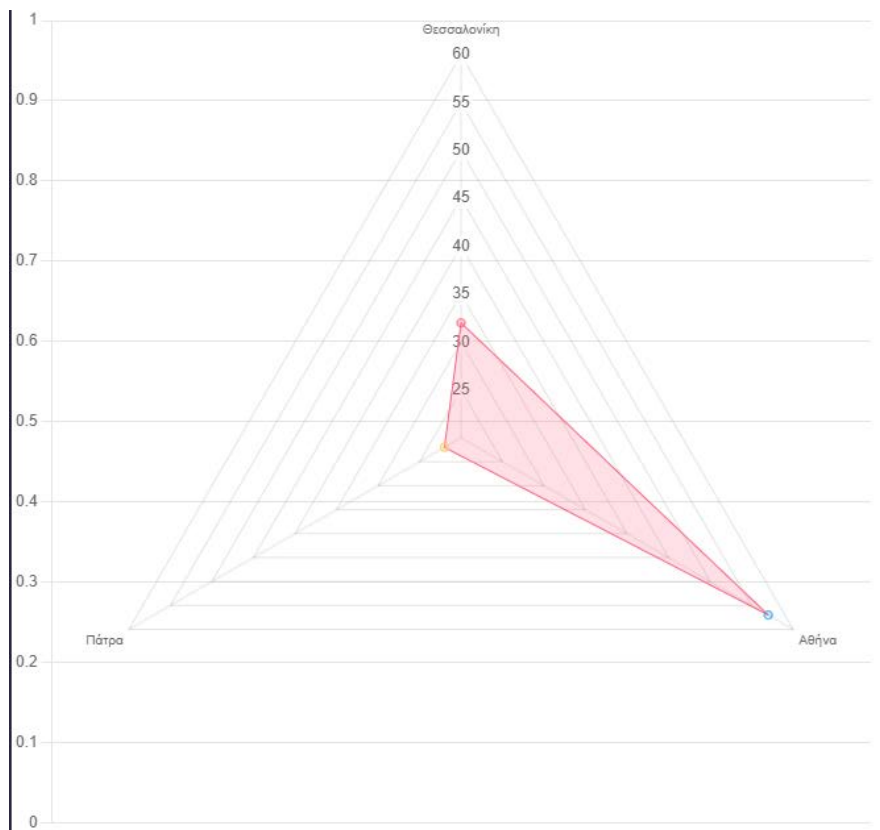
Η γραμμή 10 ελέγχει με την συνθήκη `if` αν υπάρχουν εγγραφές στο `$result` πριν συνεχίσει. Ύστερα δημιουργεί δύο πίνακες τον `$colCountry` και `$colIntAVG`. Έπειτα υπάρχει ένας βρόγχος `while` που εκτελείται μέχρις ότου δεν υπάρχουν άλλες εγγραφές στο αντικείμενο `$result`. Η μέθοδος `fetch()` ανακτεί την επόμενη εγγραφή που είναι αποθηκευμένη στο `$result` και στην συνέχεια αποθηκεύεται στο `$row`. Ύστερα αποθηκεύονται τα δεδομένα από την στήλη `Country` του `$row` στον πίνακα `$colCountry[]` και τα δεδομένα από την στήλη `Broadband Mbps` στον πίνακα `$colIntAVG[]` και στην γραμμή 23 ελευθερώνεται η μνήμη από το αντικείμενο `$result`. Εάν αποτύχει η συνθήκη `if` τότε τυπώνεται το αντίστοιχο μήνυμα στην γραμμή 26. Τέλος η γραμμή 34 αφαιρεί τη σύνδεση PDO για να κλείσει τη σύνδεση προς τη βάση δεδομένων.

3.2.7 Η Σελίδα `Create.php`

Στην καρτέλα `Create` της ιστοσελίδας, ο χρήστης έχει την δυνατότητα να εισάγει τα δικά του δεδομένα και παραμετροποιήσεις στην εφαρμογή και να δημιουργήσει ο ίδιος το γράφημα της αρεσκείας του. Ο χρήστης επιλέγει στην αρχή των αριθμό των εισόδων που θέλει να εισάγει στο σύστημα. Στην συνέχεια εμφανίζονται πεδία κειμένου και πεδία αριθμητικών τιμών τόσα όσα επέλεξε ο χρήστης. Εφόσον γεμίσει τα πεδία με τα δεδομένα που θέλει και επιλέξει τον τύπο του γραφήματος, με το κουμπί `Create` δημιουργεί το δικό του γράφημα. Τέλος με το κουμπί `Reset` επαναφέρει την σελίδα στην αρχική της κατάσταση.



Εικόνα 20 Εισαγωγή δεδομένων στα πεδία εισόδων στην καρτέλα Create



Εικόνα 21 Γράφημα Radar με βάση δεδομένα από το πληκτρολόγιο του χρήστη

Ο κύριος κώδικας για την καρτέλα Create βρίσκεται στο αρχείο **Create.php**. Μία ιδιαιτερότητα που έχει το συγκεκριμένο αρχείο είναι ότι στην επικεφαλίδα <head> εισάγεται ένα εξωτερικό αρχείο Javascript με όνομα **selectInputs.js**, το οποίο περιλαμβάνει μία συνάρτηση η οποία είναι υπεύθυνη για την εμφάνιση του κατάλληλου αριθμού πεδίων κειμένου και αριθμητικών πεδίων στην σελίδα με βάση την επιλογή του χρήστη.

```
26     <div class="chooseNumRec">
27         <label for="numbers">Choose number:</label>
28         <select name="numbers" id="selectNumber" onchange="selectInput(this)">
29             <option value="1">1</option>
30             <option value="2">2</option>
31             <option value="3">3</option>
32             <option value="4">4</option>
33             <option value="5">5</option>
34             <option value="6">6</option>
35             <option value="7">7</option>
36             <option value="8">8</option>
37             <option value="9">9</option>
38             <option value="10">10</option>
39         </select>
40     </div>
```

Εικόνα 22 Η επιλογή του αριθμού των εισόδων του χρήστη

Από την γραμμή 26 δημιουργείται ένα dropdown με αριθμούς από 1 έως 10. Όταν ο χρήστης επιλέξει έναν αριθμό, καλείται η συνάρτηση **selectInputs()** η οποία έχει ως αποτέλεσμα την εμφάνιση του κατάλληλου αριθμού πεδίων με βάση την επιλογή του χρήστη.

Στη συνέχεια δημιουργούνται δύο λίστες, κάθε μία αποτελούμενη από 10 στοιχεία. Η πρώτη λίστα χρησιμοποιείται για την εισαγωγή ετικετών από τον χρήστη, όπου κάθε στοιχείο της αντιστοιχεί σε ένα πεδίο κειμένου. Η δεύτερη λίστα χρησιμοποιείται για την εισαγωγή αριθμητικών δεδομένων από τον χρήστη, όπου κάθε στοιχείο της αντιστοιχεί σε ένα αριθμητικό πεδίο. Τα συγκεκριμένα πεδία δεν εμφανίζονται αρχικά κατά τη φόρτωση της σελίδας διότι υπάρχουν κανόνες CSS στο αρχείο **main.css** που τα αποκρύπτουν. Όταν ο χρήστης επιλέγει τον αριθμό εισόδων που επιθυμεί, καλείται η συνάρτηση **selectInputs()**. Αυτή η συνάρτηση τροποποιεί την CSS των εν λόγω πεδίων, καθιστώντας τα ορατά σύμφωνα με τις ανάγκες του χρήστη.

```

81  li{
82      padding: 1px;
83      display: none;
84  }
85  }

```

Εικόνα 23 Η απόκρυψη των αντικειμένων λίστας «li» στο αρχείο main.css

Παρακάτω βρίσκεται το τμήμα κώδικα που παρέχει στον χρήστη τη δυνατότητα επιλογής του τύπου γραφήματος με την χρήση ενός dropdown και τον έλεγχο της διαδικασίας δημιουργίας γραφήματος και επαναφοράς της ιστοσελίδας μέσω των κουμπιών Create και Reset. Το dropdown όπως και τα δύο κουμπιά είναι επίσης μη ορατά μέχρι ο χρήστης να επιλέξει τον αριθμό εισόδων όπως εξηγήθηκε προηγουμένως.

```

135  <div class="shapeInput">
136      <label for="shapes">Choose shape:</label>
137      <select name="shapes" id="selectShape" onchange="selectShape(this)">
138          <option value="bar">Bar</option>
139          <option value="line">Line</option>
140          <option value="pie">Pie</option>
141          <option value="doughnut">Doughnut</option>
142          <option value="radar">Radar</option>
143          <option value="polarArea">Polar Area</option>
144      </select>
145  </div>
146  <div class="buttons">
147      <label>Press Button</label>
148      <button class="createB" type="button" onclick="createButton()">Create</button>
149      <button class="resetB" type="button" onclick="resetButton()">Reset</button>
150  </div>

```

Εικόνα 24 Δημιουργία των κουμπιών Create και Reset

Στην γραμμή 137 δημιουργείται ένα dropdown, το οποίο περιλαμβάνει για επιλογές διάφορους τύπους γραφημάτων. Όταν ο χρήστης επιλέξει έναν από αυτούς, καλείται η συνάρτηση selectShape(), η οποία έχει ως σκοπό να παίρνει την τιμή από την επιλογή του χρήστη και να την αναθέτει στην μεταβλητή myChartShape. Αυτή η μεταβλητή συμβολίζει την επιλογή τύπου γραφήματος του χρήστη και χρησιμοποιείται για την αρχικοποίηση του αντικειμένου γραφήματος πριν την δημιουργία του.

Στην συνέχεια στην γραμμή 148 δημιουργείται το κουμπί Create. Όταν ο χρήστης πατάει αυτό το κουμπί καλείται η συνάρτηση createButton(). Η συγκεκριμένη συνάρτηση χρησιμοποιεί τα δεδομένα που έχει εισάγει ο χρήστης στην σελίδα και αρχικοποιεί το γράφημα με αυτά. Τέλος δημιουργεί το γράφημα με βάση αυτά τα δεδομένα, αποκρύπτει το κουμπί Create και εμφανίζει το κουμπί Reset.

```

159 <script>
160   let myCreateChart;
161   let myChartShape = 'bar';
162   let counter;
163
164   var style = document.createElement('style');
165   document.head.appendChild(style);
166
167   function createButton(){
168
169     counter = document.getElementById("selectNumber").value;
170     let labelArray = [];
171     let numberArray = [];
172     for(i=1;i<=counter;i++){
173       labelArray[i-1]=document.getElementById("label"+i).value;
174       numberArray[i-1]=document.getElementById("number"+i).value;
175     }
176
177
178     //Chart creation
179     let ctx = document.getElementById('createChart');
180     myCreateChart = new Chart(ctx, {
181       type: myChartShape ,
182       data: {
183         labels: labelArray,
184         datasets: [{
185           label: 'My Chart',
186           data: numberArray,
187           backgroundColor: [

```

Εικόνα 25 Η συνάρτηση createButton()

Στην εικόνα 25 φαίνεται ο κώδικας της συνάρτησης createButton(), ο οποίος αποτελείται από έναν βρόχο for και την δημιουργία ενός αντικειμένου γραφήματος. Πριν από αυτό δηλώνεται η μεταβλητή myCreateChart, που θα της ανατεθεί το αντικείμενο γραφήματος που θα δημιουργηθεί.

Επίσης η μεταβλητή myChartShape καθορίζει το σχήμα του γραφήματος, και έχει αρχική τιμή το ραβδόγραμμα “bar” μέχρι να αλλάξει η τιμή της από την συνάρτηση selectShape() όπως φάνηκε προηγουμένως με βάση την επιλογή του χρήστη.

Τέλος δηλώνεται η μεταβλητή counter η οποία συμβολίζει τον αριθμό των εισόδων που έχει επιλέξει ο χρήστης.

Στην γραμμή 169, ανατίθεται στην μεταβλητή counter ο αριθμός εισόδων που έχει επιλέξει ο χρήστης. Στην συνέχεια δημιουργούνται ο πίνακας labelArray[] που θα περιέχει τις ετικέτες που θα εισάγει ο χρήστης και ο πίνακας numberArray[] που θα περιέχει τις αριθμητικές μονάδες.

Ο βρόχος for χρησιμοποιείται για την ανάκτηση των δεδομένων που έχει εισάγει ο χρήστης στα πεδία κειμένου και στα αριθμητικά πεδία και τα αποθηκεύει στους

αντίστοιχους πίνακες που δημιουργήθηκαν παραπάνω. Η μεταβλητή counter είναι σημαντική διότι ο βρόχος πρέπει να κάνει τόσες επαναλήψεις όσος ο αριθμός των πεδίων που επέλεξε ο χρήστης.

Εφόσον πλέον υπάρχουν οι πίνακες με τα δεδομένα του χρήστη, στην γραμμή 179 ξεκινάει η δημιουργία του αντικείμενου γραφήματος, το οποίο αρχικοποιήθηκε με τύπο γραφήματος την τιμή της μεταβλητής myChartShape, ετικέτες(labels) τον πίνακα labelArray και αριθμητικά δεδομένα τον πίνακα numberArray. Το αντικείμενο γραφήματος δημιουργείται σε μία ορισμένη περιοχή κανβά του εγγράφου με id="createChart".

```
215 | | | | //Makes inputs disappear
216 | | | | style.textContent = `
217 | | | |   .resetB{
218 | | | |     display: inline-block;
219 | | | |   }
220 | | | |
221 | | | |   .createB, .chooseNumRec, .numberInput, .textInput, .shapeInput {
222 | | | |     display: none;
223 | | | |   }
224 | | | | `;
225 | | | |
```

Εικόνα 26 Δημιουργία CSS για την διαχείριση της εμφάνισης των κουμπιών.

Στην εικόνα 25 φαίνεται ότι δημιουργείται μία μεταβλητή style, η οποία θα χρησιμοποιηθεί για να εμφανίζει και να εξαφανίζει τα πεδία εισόδων καθώς επίσης και τα κουμπιά create button και reset button ανάλογα με την κατάσταση.

Το τελευταίο κομμάτι της συνάρτησης δημιουργεί κάποιους κανόνες CSS όπως φαίνεται στην εικόνα 26 και τους αναθέτει στην μεταβλητή style. Μετά την εκτέλεση της συνάρτησης createButton(), αυτοί οι κανόνες προστίθενται στην επικεφαλίδα του εγγράφου <head> και εφαρμόζονται. Αυτό που επιτυγχάνουν είναι να εξαφανίζουν το κουμπί Create και να εμφανίζουν το κουμπί Reset πλέον, δίνοντας την δυνατότητα στον χρήστη να διαγράψει το αντικείμενο γραφήματος και να επαναφέρει την σελίδα στην αρχική της μορφή.

Όπως φαίνεται στην εικόνα 24 πιο πάνω, δημιουργείται επίσης ένα κουμπί Reset που είναι αρχικά μη ορατό στον χρήστη μέχρι να δημιουργηθεί ένα αντικείμενο γραφήματος. Όταν ο χρήστης πατήσει στο κουμπί Reset εκτελείται η συνάρτηση `resetButton()` η οποία είναι υπεύθυνη για την διαγραφή του αντικείμενου γραφήματος και την επαναφορά της σελίδας στην αρχική της μορφή.

```
228 //Reset Button
229 function resetButton(){
230     myCreateChart.destroy();
231
232     //Resets inputs with blank space
233     for(i=1;i<=counter;i++){
234         document.getElementById("label"+i).value = "";
235         document.getElementById("number"+i).value = "";
236     }
237     //Resets Shape to 'Bar'
238     document.getElementById("selectShape").value = 'bar';
239
240     //Reset Button Disappear
241     style.textContent = `
242     .resetB {
243     |     display: none;
244     }
245
246
247     .chooseNumRec{
248     |     |     |     |     display: inline-block;
249     |     |     |     }
250     `;
```

Εικόνα 27 Οι λειτουργίες της συνάρτησης `resetButton()`

Το πρώτο πράγμα που κάνει η συνάρτηση **`resetButton()`** είναι να καταστρέφει το αντικείμενο γραφήματος όπως φαίνεται στην γραμμή 231.

Στην συνέχεια υπάρχει ένας βρόχος, παρόμοιος με αυτόν της συνάρτησης `createButton()`. Ο συγκεκριμένος βρόχος ανατρέχει τα πεδία εισόδων στα οποία έχει προσθέσει τιμές ο χρήστης και αλλάζει τις τιμές των πεδίων σε κενό.

Έπειτα ορίζεται μέσω του **`document.getElementById("selectShape").value = 'bar'`** η τιμή της μεταβλητής `selectShape` στην αρχική της τιμή που ήταν το "bar".

Τέλος δημιουργείται ένας κανόνας CSS που ανατίθεται στην μεταβλητή `styleR`. Ο συγκεκριμένος κανόνας εφαρμόζεται στο κουμπί Reset και το θέτει μη ορατό και εμφανίζει ξανά το dropdown επιλογής εισόδων από τον χρήστη μετά την εκτέλεση της συνάρτησης **`resetButton()`**.

3.2.8 To Script selectInputs.js

Ο κώδικας του αρχείου **selectInputs.js** παρέχει την συνάρτηση **selectInput(selectNum)**, η οποία καλείται όταν υπάρχει μια αλλαγή στο dropdown που επιλέγει τον αριθμό των εισόδων που επιθυμεί ο χρήστης στην καρτέλα Create της ιστοσελίδας. Η λειτουργία αυτή αλλάζει την εμφάνιση των πεδίων εισαγωγής ανάλογα με τον αριθμό που επιλέγει ο χρήστης και τα κάνει ορατά στην σελίδα με αποτέλεσμα ο χρήστης να έχει την δυνατότητα να εισάγει τα δεδομένα του σε αυτά.

Η συνάρτηση δίνει την δυνατότητα του ελέγχου της εμφάνισης των πεδίων, δημιουργώντας CSS κανόνες που επηρεάζουν την εμφάνιση των πεδίων καταλλήλως, ανάλογα με την επιλογή του χρήστη. Έτσι για παράδειγμα αν ο χρήστης επιλέξει αριθμό εισόδων 5, η συνάρτηση θα δημιουργήσει τον κατάλληλο κανόνα CSS για να κάνει ορατά τα 5 πρώτα πεδία κειμένου και τα 5 πρώτα αριθμητικά πεδία.

```
1 //Change the number of Inputs displaying
2 function selectInput(selectNum){
3     if(selectNum.value === "1"){
4         style.innerHTML = `
5             li{
6                 display: none;
7             }
8             .li1 {
9                 display: block;
10            }
11            .createB{
12                display: inline;
13            }
14
15            .shapeInput{
16                display: inline;
17            }
18            `;
19            document.head.appendChild(style);
20
21        }
22        else if(selectNum.value === "2"){
23            style.innerHTML = `
24                li{
25                    display: none;
26                }
27                .li1,.li2 {
28                    display: block;
29                }
30
31                .createB{
32                    display: inline;
33                }
34
35                .shapeInput{
36                    display: inline;
37                }
38                `;
39                document.head.appendChild(style);
40            }
41            else if(selectNum.value === "3"){
42                style.innerHTML = `
43                    li{
```

Εικόνα 28 Η συνάρτηση selectInput(selectNum)

Όπως φαίνεται στην εικόνα 28, η συνάρτηση στην πραγματικότητα αποτελείται από μία μεγάλη δομή ελέγχου if-else η οποία επεκτείνεται μέχρι την συνθήκη όπου η τιμή του dropdown(selectNum) να είναι το 10.

Στην αρχή του κώδικα δημιουργείται ένα στοιχείο <style> με όνομα style, το οποίο έχει την δυνατότητα να εισάγει CSS κανόνες μέσα στο έγγραφο της σελίδας.

Στην συνέχεια δημιουργείται η συνάρτηση selectInput(selectNum) η οποία δέχεται μία παράμετρο. Όταν η συνάρτηση καλείται μέσα στο αρχείο Create.php, στην θέση της παραμέτρου εισάγεται το αποτέλεσμα της επιλογής του χρήστη στο dropdown, το οποίο είναι υπεύθυνο για την επιλογή αριθμού πεδίων εισόδων.

Στην συνέχεια, μέσα στην συνάρτηση υπάρχει μία δομή if-else η οποία ελέγχει την τιμή της παραμέτρου που δέχεται η συνάρτηση και πράττει αναλόγως. Όταν καλείται η συνάρτηση μέσα στο Create.php, αυτό που κάνει είναι να ελέγχει ποιον αριθμό διάλεξε ο χρήστης στο dropdown και στην συνέχεια να εμφανίζει τον κατάλληλο αριθμό πεδίων. Αυτό πραγματοποιείται με τον κανόνα { **display:block;** } όπως φαίνεται στην γραμμή 9 η στην γραμμή 28. Επίσης στην γραμμή 11 ο κανόνας αυτός καθιστά το κουμπί Create ορατό και αντιστοίχως στην γραμμή 15 καθιστά το dropdown για την επιλογή του τύπου γραφήματος ορατό.

Όλοι αυτοί οι κανόνες γράφονται στο εσωτερικό του στοιχείου style με την εντολή **style.innerHTML=""**. Εφόσον το στοιχείο είναι τύπου <style>, αυτό που γίνεται στην πραγματικότητα είναι η δυναμική εισαγωγή κανόνων CSS στο έγγραφο με την χρήση της Javascript.

3.2.9 Η Σελίδα Upload.php

Το αρχείο Upload.php είναι υπεύθυνο για την δημιουργία της καρτέλας Upload στην ιστοσελίδα. Η καρτέλα Upload δίνει την δυνατότητα στον χρήστη να ανεβάσει ο ίδιος τα δικά του δεδομένα σε αρχείο Excel στην ιστοσελίδα της εφαρμογής με μορφή .xls , .xlsx ή .csv και στην συνέχεια να δημιουργήσει γραφήματα με βάση αυτά τα δεδομένα.

DataGraffix

Home Graphs Create Upload About

Upload your data to create a graph

No file chosen

Choose shape:

Press Button

Εικόνα 29 Η καρτέλα Upload

Το κουμπί **Choose File** επιτρέπει στον χρήστη να επιλέξει το αρχείο Excel που επιθυμεί από τον υπολογιστή του. Μόλις ο χρήστης πατήσει το κουμπί **Upload File** για να ανεβάσει το αρχείο στην ιστοσελίδα καλείται η συνάρτηση **uploadFile()**, τα δεδομένα αποθηκεύονται σε έναν πίνακα της Javascript με όνομα **excelData** κάνοντας χρήση της εξωτερικής βιβλιοθήκης **Sheet.js**.

```
//Uploads a file and saving it in an array
let excelData;
function uploadFile() {
const input = document.getElementById('fileInput');
const file = input.files[0];

if (file) {
const reader = new FileReader();

reader.onload = function(e) {
const data = new Uint8Array(e.target.result);
const workbook = XLSX.read(data, { type: 'array' });
const sheetName = workbook.SheetNames[0];
const worksheet = workbook.Sheets[sheetName];
excelData = XLSX.utils.sheet_to_json(worksheet, { header: 1 });

//console.log(excelData);
};

reader.readAsArrayBuffer(file);
} else {
console.log('No file selected.');
```

Εικόνα 30 Η συνάρτηση **uploadFile** και η χρήση της βιβλιοθήκης **Sheet.js** για την επεξεργασία του αρχείου Excel.

Εφόσον ο χρήστης έχει επιλέξει το είδος του γραφήματος που θέλει να κατασκευάσει και πατήσει στην συνέχεια το κουμπί **Create**, καλείται η συνάρτηση **createButton()**. Η συνάρτηση αυτή δημιουργεί αρχικά 2 πίνακες, τον **labelArray** ο οποίος θα πάρει τιμές που αντιπροσωπεύουν τις ετικέτες του γραφήματος και τον **numberArray** του οποίου οι τιμές αντιπροσωπεύουν τις αριθμητικές τιμές του γραφήματος. Στην συνέχεια αρχικοποιείται ο πίνακας **labelArray** με την πρώτη στήλη του πίνακα **excelData** που δημιουργήθηκε προηγουμένως, και ο **numberArray** με την δεύτερη του στήλη με μέγιστο αριθμό 20 τιμές σε κάθε πίνακα. Ύστερα χρησιμοποιούνται αυτοί οι 2 πίνακες για την αρχικοποίηση του αντικειμένου γραφήματος της **Chart.js** και μετά δημιουργείται το γράφημα.

```
function createButton(){

    let labelArray = [];
    let numberArray = [];

    for (let i = 0; i < excelData.length; i++) {

        if(i>19) { //max 20 entries, index no bigger than 19
            break
        }
        labelArray[i] = excelData[i][0]; // first column, index 0, labels
        numberArray[i] = excelData[i][1]; // second column, index 1, num values
    }

    //Chart creation
    let ctx = document.getElementById('createChart');
    myCreateChart = new Chart(ctx, {
        type: myChartShape ,
        data: {
            labels: labelArray,
            datasets: [{
                label: 'My Chart',
                data: numberArray,
                backgroundColor: [
                    'rgba(255, 99, 132, 0.2)',
```

Εικόνα 31 Δημιουργία και αρχικοποίηση των πινάκων **labelArray** και **numberArray** μέσα στην συνάρτηση **createButton**

Τέλος, μετά την δημιουργία του γραφήματος εμφανίζεται ένα **Reset** button στην θέση του **Create** button, το οποίο χρησιμοποιείται για να καταστρέψει το αντικείμενο γραφήματος που έχει δημιουργηθεί και να δώσει την δυνατότητα στο χρήστη για την δημιουργία ενός νέου αντικειμένου στην θέση του παλιού.

3.2.10 Η Σελίδα About.php

Σε αυτό το αρχείο εμπεριέχεται ο κώδικας της καρτέλας About, ο οποίος αποτελείται κυρίως από στοιχεία παραγράφων. Σε αυτό το μικρό ενημερωτικό τμήμα, παρουσιάζονται οι δυνατότητες που παρέχει η πλατφόρμα και οι δυνατότητες του χρήστη πάνω στην εφαρμογή.

3.2.11 Το Φύλλο Μορφοποίησης main.css

Τέλος το main.css, είναι ένα φύλλο μορφοποίησης που εμπεριέχει τους κανόνες CSS που εφαρμόζονται στην σελίδα της εφαρμογής. Οι κανόνες μορφοποιούν την εμφάνιση της σελίδας επεξεργαζόμενοι τα χρώματα που εμφανίζονται, τις γραμματοσειρές, τις θέσεις και τις αποστάσεις μεταξύ των στοιχείων της εφαρμογής κ.α.

3.3 Εκτέλεση της Εφαρμογής

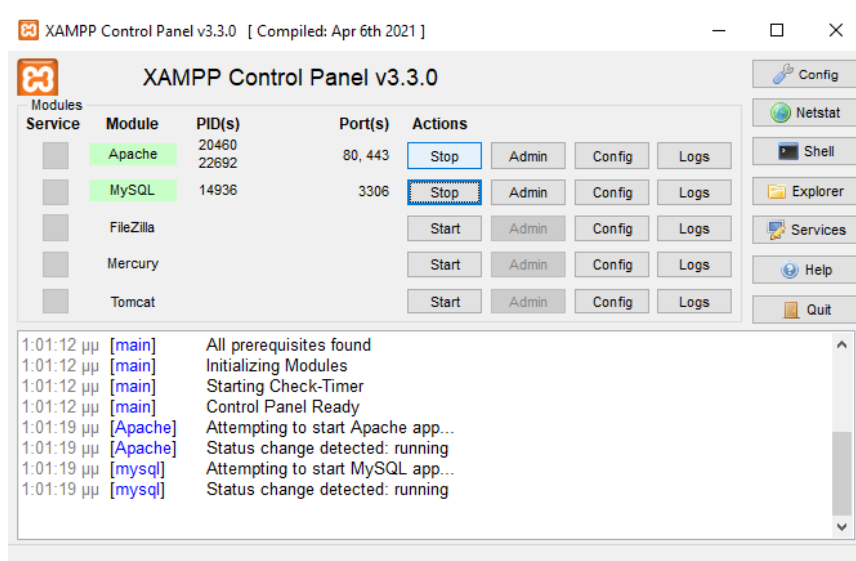
Σε αυτό το υποκεφάλαιο θα αναλυθούν τα βήματα που πρέπει να λάβει κάποιος υπόψιν έτσι ώστε να ξεκινήσει να εκτελείται μία διαδικτυακή εφαρμογή μέσω του XAAMP.

3.3.1 Η Τοποθεσία των Αρχείων

Αρχικά όπως ήδη αναφέρθηκε στο κεφάλαιο 3.2, είναι αναγκαίο όλα τα αρχεία που απαρτίζουν την εφαρμογή να βρίσκονται στον ίδιο φάκελο στην τοποθεσία `\xampp\htdocs\DataGraffix` όπως φαίνεται στην εικόνα 8. Αυτός ο φάκελος χρησιμοποιείται από τον Apache κάθε φορά που ο χρήστης ζητάει πρόσβαση στην ιστοσελίδα. Έτσι ο Apache ψάχνει τα ζητούμενα αρχεία μέσα στον φάκελο htdocs, και μετά την επεξεργασία των αρχείων τα στέλνει στο πρόγραμμα περιήγησης του χρήστη.

3.3.2 Control Panel του XAMPP

Το επόμενο σημαντικό κομμάτι για την εκκίνηση της εφαρμογής είναι το Control Panel του XAAMP. Το XAMPP Control Panel είναι ένα γραφικό περιβάλλον που παρέχεται με το XAMPP και χρησιμεύει για τον έλεγχο και τη διαχείριση βασικών συστατικών του XAMPP, όπως ο Apache HTTP Server και το MariaDB. Κάθε μία από αυτές τις λειτουργίες έχει τα δικά της κουμπιά εκκίνησης, διακοπής και επανεκκίνησης, δίνοντας στον χρήστη τον πλήρη έλεγχο αυτών των τεχνολογιών. Για την εκκίνηση της εφαρμογής είναι σημαντικό να ενεργοποιηθούν οι λειτουργίες του Apache και του MariaDB όπως φαίνεται στην εικόνα 32.



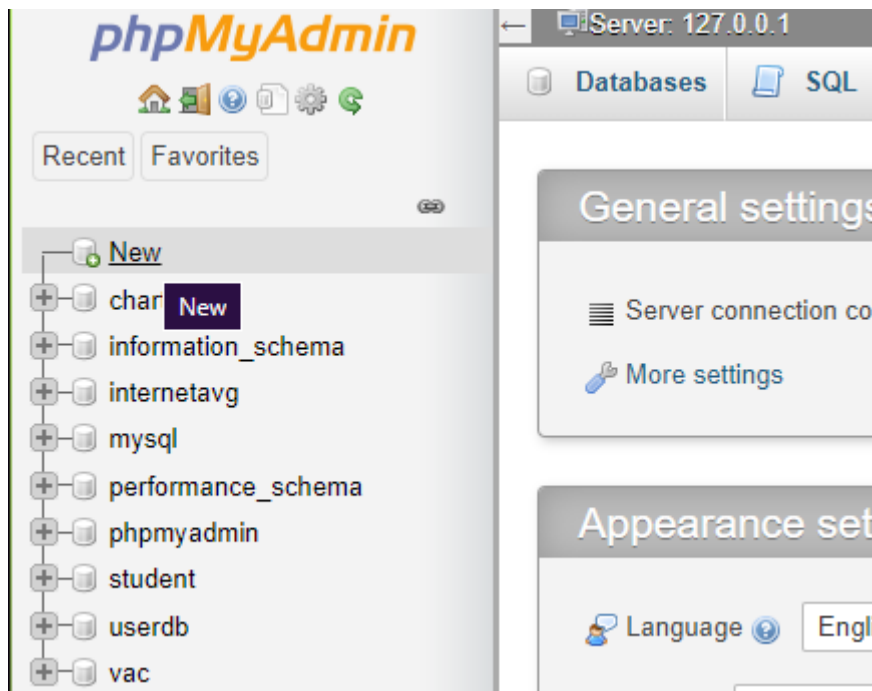
Εικόνα 32 Xampp Control Panel

3.3.3 phpMyAdmin

Εφόσον πλέον οι λειτουργίες του DBMS έχουν ενεργοποιηθεί, ένα σημαντικό κομμάτι για την εφαρμογή είναι η εισαγωγή των δεδομένων στην βάση. Αρχικά ο ενδιαφερόμενος πρέπει να γράψει στο πρόγραμμα περιήγησης του την τοποθεσία **/localhost/phpmyadmin** .

Το phpMyAdmin αποτελεί ένα δωρεάν εργαλείο γραμμένο σε PHP, σχεδιασμένο για τη διαχείριση βάσεων δεδομένων MySQL και MariaDB μέσω του διαδικτύου. Παρέχει ευρεία γκάμα λειτουργιών, επιτρέποντας στους χρήστες να διαχειρίζονται τις βάσεις δεδομένων τους μέσω του προγράμματος περιήγησης, ενώ παρέχει και τη δυνατότητα άμεσης εκτέλεσης εντολών SQL. [38]

Μέσα στο περιβάλλον του phpMyAdmin πρέπει να δημιουργηθεί μία βάση δεδομένων η οποία θα φιλοξενεί τα τραπέζια των δεδομένων τα οποία επιθυμεί κάποιος όπως φαίνεται στην εικόνα 33.



Εικόνα 33 Δημιουργία Βάσης Δεδομένων στο phpMyAdmin

Στην συνέχεια πρέπει να εισαχθούν τα δεδομένα σε αυτήν την βάση. Τα δεδομένα είναι αυτά που πάρθηκαν από την σελίδα Kaggle και βρίσκονται σε μορφή αρχείου Excel. Ο ενδιαφερόμενος αρχικά επιλέγει την βάση που δημιούργησε, και στην συνέχεια πρέπει να πατήσει το κουμπί import και να εισάγει στην βάση το αρχείο Excel με τα δεδομένα. Όταν το αρχείο query.php προσπαθεί να ανακτήσει δεδομένα χρησιμοποιεί το ερώτημα «**SELECT * FROM internetavg.intavgjanuary2022 LIMIT 10**» όπου το internetavg είναι το όνομα της βάσης και το intavgjanuary2022 είναι το όνομα του τραπέζιου που υπάρχουν τα δεδομένα.

3.3.4 Εκτέλεση της Εφαρμογής

Τέλος εφόσον έχουν τακτοποιηθεί τα παραπάνω ζητήματα, μπορεί πλέον να εκτελεστεί η εφαρμογή. Ο χρήστης πρέπει να γράψει την τοποθεσία /localhost/Datagraffix/ στο πρόγραμμα περιήγησης του. Αυτό θα έχει ως αποτέλεσμα να εκτελεστεί η αρχική σελίδα της εφαρμογής index.php που βρίσκεται τοπικά στον φάκελο htdocs.

Συμπεράσματα

Η εκπόνηση της διπλωματικής εργασίας στόχευσε στην δημιουργία μίας διαδικτυακής εφαρμογής με ιστοσελίδα, στην οποία λαμβάνει χώρα η οπτικοποίηση δεδομένων. Η οπτικοποίηση μπορεί να πραγματοποιηθεί με την χρήση βιβλιοθηκών οπτικοποίησης, οι οποίες παρέχουν έτοιμο κώδικα στον ενδιαφερόμενο για την κατασκευή των γραφημάτων

Η Chart.js είναι μία τέτοια βιβλιοθήκη οπτικοποίησης δεδομένων, η οποία χρησιμοποιήθηκε στην ανάπτυξη της εφαρμογής. Είναι πολύ εύκολη στην εκμάθηση και στην χρήση της. Παρέχει στον χρήστη την δυνατότητα κατασκευής ευπαρουσίαστων γραφικών και γραφημάτων με πολύ απλό τρόπο και δεν απαιτεί από τον χρήστη παρά βασικές γνώσεις πάνω στην HTML και την Javascript . Ωστόσο η βιβλιοθήκη υποστηρίζει έναν περιορισμένο αριθμό τύπων γραφημάτων.

Συμπερασματικά η δημιουργία μίας τέτοιας εφαρμογής απαιτεί την αρμονική συνεργασία τεχνολογιών τόσο στην πλευρά της ιστοσελίδας όσο και στην πλευρά του διακομιστή. Υπάρχουν πολλές διαθέσιμες βιβλιοθήκες οπτικοποίησης στο διαδίκτυο, οι οποίες διαφέρουν τόσο στην ευκολία χρήσης τους όσο και στον αριθμό και την ποικιλομορφία των ειδών γραφημάτων τους. Για την επιλογή της κατάλληλης βιβλιοθήκης πρέπει να ληφθεί υπόψιν το γνωσιολογικό υπόβαθρο και οι ανάγκες του ενδιαφερόμενου.

Μελλοντικά θα μπορούσε η εφαρμογή να προσαρμοστεί καταλλήλως έτσι ώστε να μετατραπεί η ιστοσελίδα σε ένα χρήσιμο εργαλείο για μαθησιακούς σκοπούς, το οποίο θα επιτρέπει στον επισκέπτη να δημιουργεί τα δικά του γραφήματα, να τα συγκρίνει με άλλα και τέλος να τα κατεβάζει στην προσωπική του συσκευή για περαιτέρω χρήση. Ύστερα θα μπορούσε να μεταφερθεί η ιστοσελίδα από τον τοπικό διακομιστή που βρίσκεται σε έναν online διακομιστή έτσι ώστε να είναι διαθέσιμη για όλους. Τέλος, η ιστοσελίδα θα μπορούσε να μετατραπεί σε έναν χώρο πληροφόρησης για την οπτικοποίηση δεδομένων όπου θα ενημερώνει τον επισκέπτη για τα οφέλη της οπτικοποίησης, για τις δυνατότητες του κάθε τύπου γραφήματος ξεχωριστά και τα μειονεκτήματά του.

Βιβλιογραφία

- 1] A. E. S. S. M. C. M. A. Matthew N. O. Sadiku, «DATA VISUALIZATION,» Δεκέμβριος 2016.
- 2] M. Friendly, «A Brief History of Data Visualization,» σε *Handbook of Data Visualization*, 2006.
- 3] T. S. Pamela Reynolds, «UC Davis DataLab,» 24 04 2023. [Ηλεκτρονικό]. Available: https://ucdavisdatalab.github.io/workshop_data_viz_principles/index.html. [Πρόσβαση 14 Οκτωβρίου 2023].
- 4] M. Friendly, «Milestones in the history of thematic cartography, statistica l graphics, and data visualization,» 2001.
- 5] E. B. Kate Brush, «techtarget,» [Ηλεκτρονικό]. Available: <https://www.techtarget.com/searchbusinessanalytics/definition/data-visualization>. [Πρόσβαση 13 Οκτωβρίου 2023].
- 6] «Chart.js,» [Ηλεκτρονικό]. Available: <https://www.chartjs.org/docs/latest/>. [Πρόσβαση 17 Οκτωβρίου 2023].
- 7] «Canvas API,» [Ηλεκτρονικό]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API. [Πρόσβαση 17 Οκτωβρίου 2023].
- 8] H. d. Rocha, Learn Chart.js: Create interactive visualizations for the Web with Chart.js 2, 2019.
- 9] H. d. Rocha, Learn D3.js: Create interactive data-driven visualizations for the web with the D3.js library.
- 10] «Plotly,» [Ηλεκτρονικό]. Available: <https://plotly.com/javascript/>. [Πρόσβαση 21 Οκτωβρίου 2023].
- 11] «Using Google Charts | Google For Developers,» [Ηλεκτρονικό]. Available: <https://developers.google.com/chart/interactive/docs>. [Πρόσβαση 22 Οκτωβρίου 2023].
- 12] P. Pedamkar, «Benefits of Data Visualization | Know Applications and Benefits,» [Ηλεκτρονικό]. Available: <https://www.educba.com/benefits-of-data-visualization/>. [Πρόσβαση 25 Οκτωβρίου 2023].
- A. S. Rawat, «10 Applications of Data visualizations | Analytics Steps,» [Ηλεκτρονικό].

- 13] Available: <https://www.analyticssteps.com/blogs/10-applications-data-visualization>.
- A. Sugandhi, «What is Data Visualization and Its Importance?,» [Ηλεκτρονικό].
- 14] Available: <https://www.knowledgehut.com/blog/business-intelligence-and-visualization/importance-of-data-visualization>. [Πρόσβαση 25 Οκτωβρίου 2023].
- «Which Industries & Professions Use Data Visualization?,» [Ηλεκτρονικό]. Available:
- 15] <https://www.nobledesktop.com/learn/data-visualization/industries-and-professions>. [Πρόσβαση 26 Οκτωβρίου 2023].
- M. West, HTML5 FOUNDATIONS, John Wiley & Sons, 2012.
- 16]
- «JavaScript | MDN,» [Ηλεκτρονικό]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [Πρόσβαση 1 Νοεμβρίου 2023].
- 17]
- M. SUCAN, «SVG or Canvas?,» 2010. [Ηλεκτρονικό]. Available:
- 18] <https://web.archive.org/web/20100623002104/http://dev.opera.com/articles/view/svg-or-canvas-choosing-between-the-two/#summary>. [Πρόσβαση 2 Νοεμβρίου 2023].
- E. S. K. G. Er. Saurabh Walia, *A Framework for Web Based Student Record Management System using PHP*, 2014.
- 19]
- D. D. Dvorski, *INSTALLING, CONFIGURING, AND DEVELOPING WITH XAMPP*, 2007.
- 20]
- «About the Apache HTTP Server Project,» [Ηλεκτρονικό]. Available:
- 21] https://httpd.apache.org/ABOUT_APACHE.html. [Πρόσβαση 5 Νοεμβρίου 2023].
- G. K. Roy T. Fielding, «COLLABORATIVE WORK,» *IEEE Internet Computing*, pp. 88-90,
- 22] 1997.
- «FAQ HTTPD,» [Ηλεκτρονικό]. Available:
- 23] <https://cwiki.apache.org/confluence/display/httpd/FAQ#FAQ-WhatisApache?>. [Πρόσβαση 5 Νοέμβρη 2023].
- «MariaDB in brief,» [Ηλεκτρονικό]. Available: <https://mariadb.org/en/>. [Πρόσβαση 6
- 24] Νοεμβρίου 2023].
- «About MariaDB Server,» [Ηλεκτρονικό]. Available: <https://mariadb.org/about/>.
- 25] [Πρόσβαση 6 Νοεμβρίου 2023].
- «PHP: Preface - Manual,» [Ηλεκτρονικό]. Available:
- 26] <https://www.php.net/manual/en/preface.php>. [Πρόσβαση 9 Νοεμβρίου 2023].
- E. B. S. C. D. R. L. Christian Paul W. Apiag, «A Review on PHP Programming Language».

27]

K. T. Rasmus Lerdorf, Programming PHP, O'Reilly Media, Inc, 2002.

28]

D. Book, «perl - The Perl 5 language interpreter - Perldoc Browser,» [Ηλεκτρονικό].

29] Available: <https://perldoc.perl.org/perl>. [Πρόσβαση 14 Νοεμβρίου 2023].

D. Book, «perlintro - a brief introduction and overview of Perl - Perldoc Browser,»

30] [Ηλεκτρονικό]. Available: <https://perldoc.perl.org/perlintro>. [Πρόσβαση 16 Νοεμβρίου 2023].

J. Lee, Beginning Perl, New York: Springer-Verlag, 2010.

31]

«Documentation for Visual Studio Code,» [Ηλεκτρονικό]. Available:

32] <https://code.visualstudio.com/docs>. [Πρόσβαση 30 Νοεμβρίου 2023].

K. Chris, «Visual Studio vs Visual Studio Code - What's The Difference Between These

33] IDE Code Editors?,» [Ηλεκτρονικό]. Available: <https://www.freecodecamp.org/news/visual-studio-vs-visual-studio-code/#whatisvisualstudiocode>. [Πρόσβαση Δεκεμβρίου 2 2023].

«Setting up Visual Studio Code,» [Ηλεκτρονικό]. Available:

34] <https://code.visualstudio.com/docs/setup/setup-overview>. [Πρόσβαση 3 Δεκεμβρίου 2023].

«Use Microsoft Live Share to collaborate with Visual Studio Code,» [Ηλεκτρονικό].

35] Available: <https://code.visualstudio.com/learn/collaboration/live-share>. [Πρόσβαση 3 Δεκεμβρίου 2023].

«Getting Started on Kaggle | Kaggle,» [Ηλεκτρονικό]. Available:

36] <https://www.kaggle.com/docs/datasets>. [Πρόσβαση 7 Δεκεμβρίου 2023].

J. Adegoke, «A Beginner's Guide to Kaggle for Data Science,» [Ηλεκτρονικό]. Available:

37] <https://www.makeuseof.com/beginners-guide-to-kaggle/>. [Πρόσβαση 7 Δεκεμβρίου 2023].

«phpMyAdmin,» [Ηλεκτρονικό]. Available: <https://www.phpmyadmin.net>. [Πρόσβαση

38] 16 Ιανουαρίου 2024].

«PHP: PDO::__construct - Manual,» [Ηλεκτρονικό]. Available:

39] <https://www.php.net/manual/en/pdo.construct.php>. [Πρόσβαση 20 Ιανουαρίου 2024].

Παράρτημα

Σύνδεση με την Βάση Δεδομένων

Ο κώδικας από το connection.php παρακάτω επιχειρεί μία PDO σύνδεση με την βάση δεδομένων.

```
<?php
<?php

// Definition of variables for the database connection
$username = "root";
$password = "";
$database = "internetavg";

// Attempt to create a PDO connection to the database
try {
    $pdo = new PDO("mysql:host=localhost;database=$database", $username,
$password);
    // Setting error handling mode for PDO
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e){
    // If there is an error during connection, print error message
    die("ERROR: Could not connect. " . $e->getMessage());
}

?>
```

Ανάκτηση των Δεδομένων

Ο κώδικας από το query.php επιχειρεί την ανάκτηση δεδομένων από την βάση και την αποθήκευση.

```
<?php
// Attempt select query execution
try{
    $sql = "SELECT * FROM internetavg.intavgjanuary2022 LIMIT 10";

    // Execute the query using PDO
    $result = $pdo->query($sql);

    // Check if any rows were returned by the query
    if($result->rowCount() > 0) {
        // Initialize arrays to store column values
        $colCountry = array(); // Variable to store country values
        $colIntAVG = array(); // Variable to store internet average values

        // Loop through each row returned by the query
        while($row = $result->fetch()) {
            // Store country and internet average values in respective arrays
            $colCountry[] = $row["Country"];
            $colIntAVG[] = $row["Broadband Mbps"];
        }

        // Release the result set memory
        unset($result);
    } else {
        // If no records were found, output a message
        echo "No records matching your query were found.";
    }
} catch(PDOException $e){
    //error message
    die("ERROR: Could not able to execute $sql. " . $e->getMessage());
}

// Close connection
unset($pdo);
?>
```

Ανάκτηση των Δεδομένων

Τμήμα κώδικα για την κατασκευή αντικειμένων γραφήματος

```
//Setup Block
const colCountry = <?php echo json_encode($colCountry); ?>; // Store the
'Country' data retrieved from PHP
const colIntAVG = <?php echo json_encode($colIntAVG); ?>; // Store the
'Broadband Mbps' data retrieved from PHP
let data = {
  labels: colCountry, // Define labels for the chart using the Country data
  datasets: [{
    label: 'Average Fixed Broadband(Mbps) per Country', //
Definition of label for the dataset
    data: colIntAVG, // Definition of the data for the chart using
the Broadband Mbps data
    backgroundColor: [
      'rgba(255, 99, 132, 0.2)',
      'rgba(54, 162, 235, 0.2)',
      'rgba(255, 206, 86, 0.2)',
      'rgba(75, 192, 192, 0.2)',
      'rgba(153, 102, 255, 0.2)',
      'rgba(255, 159, 64, 0.2)'
    ],
    borderColor: [
      'rgba(255, 99, 132, 1)',
      'rgba(54, 162, 235, 1)',
      'rgba(255, 206, 86, 1)',
      'rgba(75, 192, 192, 1)',
      'rgba(153, 102, 255, 1)',
      'rgba(255, 159, 64, 1)'
    ],
    borderWidth: 2
  }]
};

//Configuration Block
let config = {
  type: "bar", // Define the type of chart (bar chart)
  data, // Pass the data object to the chart
  options: {
    scales: {
      y: {
        beginAtZero: true
      }
    }
  }
}
```

```
}  
  
//Render Block  
//Creates a chart Object  
let myChart = new Chart(  
    document.getElementById("myChart"),  
    config  
);
```