ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Σχεδιασμός, ανάπτυξη και υλοποίηση δικτυακής πλατφόρμας για εξόρυξη δεδομένων από κείμενα ιατρικών βιβλιογραφικών βάσεων.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανδρέας Κουζαπάς

Επιβλέπων: Δημήτριος – Διονύσιος Κουτσούρης
Καθηγητής Ε.Μ.Π

Συνεπιβλέπων: Ουρανία Πετροπούλου ΕΔΙΠ Ε.Μ.Π

Αθήνα, Ιούλιος 2019

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Σχεδιασμός, ανάπτυξη και υλοποίηση δικτυακής πλατφόρμας για εξόρυξη δεδομένων από κείμενα ιατρικών βιβλιογραφικών βάσεων.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανδρέας Κουζαπάς

Επιβλέπων : Δημήτριος – Διονύσιος Κουτσούρης
Καθηγητής Ε.Μ.Π

Συνεπιβλέπων: Ουρανία Πετροπούλου ΕΔΙΠ Ε.Μ.Π

Εγκρίθηκε από τη τριμελή εξεταστική επιτροπή τον Ιούλιο του 2019.

.....	.....	.....
Δ.-Δ. Κουτσούρης	Γιώργος Ματσόπουλος	Παναγιώτης Τσανάκας
Καθηγητής Ε.Μ.Π	Αν. Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ

Αθήνα, Ιούλιος 2019

……

Ανδρέας Κουζαπάς

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Ο στόχος αυτής της διπλωματικής εργασίας είναι ο σχεδιασμός και ανάπτυξη μιας ηλεκτρονικής πλατφόρμας για εξόρυξη δεδομένων (data mining) από κείμενα ιατρικών βιβλιογραφικών βάσεων.

Στη σύγχρονη εποχή λόγω της ραγδαίας ανάπτυξης της τεχνολογίας τόσο σε υλικό όσο και σε λογισμικό μας δίνεται η ευχέρεια πρόσβασης σε μεγάλο όγκο δεδομένων. Αυτά τα δεδομένα όμως δεν είναι όλα χρήσιμα επομένως η χρήση τεχνικών μεσών για ανάκτηση χρήσιμων πληροφοριών από αυτά κρίνεται απαραίτητη. Εδώ έρχεται στο φως η τεχνική εξόρυξης δεδομένων που μας δίνει τη δυνατότητα άντλησης γνώσης μέσα από μεγάλους όγκους δεδομένων. Στη συγκεκριμένη διπλωματική εργασία θα ασχοληθούμε με εξόρυξη δεδομένων και πιο συγκεκριμένα λέξεων (text mining) με χρήση της γλώσσας R δια μέσω μιας ηλεκτρονικής δικτυακής πλατφόρμας ως προς την διευκόλυνση της.

Στα πλαίσια αυτής της εργασίας, διεξήχθη μια έρευνα σχετικά με τις τεχνολογίες διαδικτύου που είναι διαθέσιμες και κατά πόσο ποια είναι η καταλληλότερη για αυτήν την εργασία. Το αποτέλεσμα είναι η ανάπτυξη και σχεδίαση αυτής της ηλεκτρονικής πλατφόρμας η οποία δίνει στο χρήστη τη δυνατότητα άντλησης πληροφοριών από κείμενα ιατρικών βιβλιογραφικών βάσεων με σχετική ευκολία.

Λέξεις-Κλειδιά :

Εξόρυξη δεδομένων, Εξόρυξη κειμένων, Εξόρυξη γνώσης από κείμενα, Ανάκτηση πληροφοριών, Τεχνολογίες Διαδικτύου, Angular, React,Vuejs, Node.js, MongoDB, R, Διαδικτυακή Πλατφόρμα, Δεδομένα, Ιατρικές Βιβλιογραφίες

Abstract

The aim of this diploma thesis is the design and development of an online text mining platform aimed at online medical bibliographic databases.

Over the last years and due to the rapid technological development both in hardware and software areas, we have the flexibility to access a large volume of data. However, this data is not always useful, so the use of technical means to recover useful information is considered necessary. The technique and means of data mining come in hand, so that it enables us to uncover knowledge hidden in large volumes of data. This diploma thesis deals with the extraction of data and more specifically, text, using the R language via an online network platform.

In the context of the current diploma thesis, a state-of-the-art study was conducted on already available internet technologies in order to find the one best suited for our purposes. Its results were used for the development and design of the proposed online and user-friendly platform, which gives the user the ability to extract information from online medical bibliographic databases.

Keywords:

Data Mining, Text-Mining, Text Knowledge Mining , Information Retrieval, Network Technologies, Angular, React, Vue, Node.js, MongoDB, R, Dashboard, Data, Medical bibliographies

Ευχαριστίες

Η παρούσα διπλωματική εργασία, με την οποία ολοκληρώνεται η ακαδημαϊκή μου πορεία στη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου, δεν θα μπορούσε να διεκπεραιωθεί χωρίς τη βοήθεια διάφορων ανθρώπων με τους οποίους συνεργάστηκα και θα ήθελα να ευχαριστήσω από καρδιάς.

Καταρχήν, οφείλω ένα μεγάλο ευχαριστώ στον κ. Δημήτριο Κουτσούρη, Καθηγητή Ε.Μ.Π., για την εμπιστοσύνη που μου έδειξε και για την ευκαιρία που μου προσέφερε να εκπονήσω αυτή τη διπλωματική εργασία στο Εργαστήριο Βιοιατρικής Τεχνολογίας της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Επιπλέον ιδιαίτερα θα ήθελα να ευχαριστήσω την κα. Ουρανία Πετροπούλου, ΕΔΙΠ Ε.Μ.Π. και τον κ. Παναγιώτη Κατρακάζα, υποψήφιο Διδάκτορα Ε.Μ.Π. για την βοήθεια, την υπομονή τους και τον πολύτιμο χρόνο που διέθεσαν καθοδηγώντας με σε όλα τα στάδια της εργασίας.

Τέλος, με εξίσου μεγάλη θέρμη θέλω να αναφερθώ και να ευχαριστήσω την οικογένεια μου, η οποία με στήριξε όλα αυτά τα χρόνια σε όλες τις δύσκολες στιγμές, καθώς και τους φίλους και τους συμφοιτητές μου, οι οποίοι στάθηκαν δίπλα μου σε όλη τη διάρκεια της ακαδημαϊκής μου πορείας, ο καθένας με τον δικό του ξεχωριστό τρόπο.

Ανδρέας Κουζαπάς

Αθήνα, Ιούλιος 2019

Πρόλογος

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελεί ο σχεδιασμός, η ανάπτυξη και η υλοποίηση μιας δικτυακής πλατφόρμας εξόρυξης δεδομένων από ιατρικές βιβλιογραφικές βάσεις.

Η δομή της εργασίας συνοψίζεται στις εξής ενότητες:

Στο πρώτο κεφάλαιο γίνεται μια θεωρητική αναφορά για την εξόρυξη δεδομένων και εξόρυξη δεδομένων από κείμενα όπως επίσης και μια αναφορά στις υπάρχουσες τεχνολογίες διαδικτύου και στη σχετική μελέτη που έγινε ως προς την επιλογή των καταλληλότερων για αυτό το σκοπό.

Στο δεύτερο κεφάλαιο γίνεται μια πιο εκτενής παρουσίαση της εξόρυξης δεδομένων από κείμενα και συγκεκριμένα με τη χρήση της γλώσσας R ως προς την υλοποίηση της.

Στο τρίτο κεφάλαιο γίνεται εκτενής ανάλυση της διαδικασίας σχεδιασμού και υλοποίησης της πλατφόρμας, όπως επίσης και αναφορά στις τεχνολογίες που έχουν χρησιμοποιηθεί για την ανάπτυξη της.

Στο τέταρτο κεφάλαιο παρουσιάζεται το αποτέλεσμα της εργασίας και γίνεται λεπτομερής παρουσίαση της πλατφόρμας που έχει υλοποιηθεί, καθώς και των επιμέρους κομματιών που την απαρτίζουν με τη χρήση εικόνων και επεξηγήσεων.

Τέλος, στο πέμπτο κεφάλαιο γίνεται μια σύνοψη του ολικού ιστότοπου που έχει κατασκευαστεί, παρουσιάζονται δυσκολίες που αντιμετωπίστηκαν, περιορισμοί που λήφθηκαν υπόψη και μελλοντικές επεκτάσεις που μπορούν να πραγματοποιηθούν.

Πίνακας περιεχομένων

Πίνακας Εικόνων

Κεφάλαιο 1: Θεωρητικό Πλαίσιο

1.1 Εισαγωγή

Αναμφισβήτητα στον 21 αιώνα ο όγκος των δεδομένων που είναι διαθέσιμος στους χρήστες είναι τεράστιος, και αυξάνεται καθημερινά με εκθετικό ρυθμό. Αυτός ο όγκος αποθηκευμένων πληροφοριών οφείλεται τόσο στην ανάπτυξη της τεχνολογίας σε θέματα λογισμικού, υλικού, μείωσης κόστους κατασκευής αλλά και στην ένταξη της πληροφορικής στην καθημερινή ζωή. Αφού πλέον η πληροφορική βρίσκεται σε κάθε τομέα της σύγχρονης κοινωνίας καθιστά τους χρήστες ως πρωταγωνιστικούς παράγοντες παραγωγής πληροφοριών[1].

Πλέον η κοινωνία βρίσκεται στην εποχή των μεγάλων δεδομένων (Big Data). Δηλαδή δεδομένα τόσο μεγάλα σε όγκο αλλά και πολύπλοκά σε γνώση τα οποία είναι αδιανόητο να μπορέσει ο ανθρώπινος νους να επεξεργαστεί με τις παραδοσιακές τεχνικές. Για παράδειγμα μέχρι το 2000 ένας συνηθισμένος υπολογιστής στο σπίτι μπορούσε να έχει περίπου 10 Giga Bytes (GB) εσωτερικής μνήμης. Σήμερα μόνο το Facebook[1] εισάγει καθημερινά 500 Terra Bytes (TB) νέα δεδομένα, ένα Αεροπλάνο Boeing 737 παράγει 240 TB δεδομένα για μια πτήση κατά μήκος των ΗΠΑ και γενικά αισθητήρες που βρίσκονται σε αντικείμενα καθημερινής χρήσης παράγουν συνεχώς νέα δεδομένα και πληροφορίες που αποθηκεύονται σε τεράστιες βάσεις δεδομένων[2]. Σε αυτό βοήθησε και η πρόοδος στις τεχνολογίες αποθήκευσης δεδομένων αφού πλέον διατίθενται βάσεις δεδομένων ειδικές για Big Data οι οποίες προσφέρουν γρήγορη και αποδοτική συλλογή και αποθήκευση πληροφοριών, δοσοληψία μεταξύ βάσεων και ανάλυση αυτών.

Στον τομέα της βιοιατρικής και επιστήμης υγείας ο όγκος πληροφοριών που λαμβάνονται καθημερινά είναι πολύ μεγάλος. Όμως τόσες πολλές πληροφορίες και δεδομένα δεν συνεπάγονται πάντα καθαρή γνώση. Η μαζική συλλογή και αποθήκευση τόσο μεγάλων ποσοτήτων δεδομένων συχνά οδηγεί στην μη σωστή αξιοποίηση τους αφού καταλήγουν σε τεράστιες αποθήκες δεδομένων που χάνουν την αξία τους και μετατρέπονται απλά σε αποθήκες χωρίς ουσία και γνώση.

[1] https://www.facebook.com

Αυτή η διαθεσιμότητα τεράστιου όγκου πληροφορίας έχει ωθήσει την βιομηχανία πληροφόρησης στην αναζήτηση τρόπων ανάλυσης και μετατροπής όλων αυτών των δεδομένων σε χρήσιμη πληροφορία και γνώση. Η ανάλυση τόσο μεγάλου όγκου δεδομένων θα ήταν αδύνατη χωρίς συγκεκριμένες τεχνικές. Επομένως κρίνεται αναγκαία η ένταξη της τεχνικής εξόρυξης δεδομένων.

1.2 Ορισμός Εξόρυξης δεδομένων (Data Mining)

Η εξόρυξη δεδομένων αποτελεί μια λύση στην ανάγκη επεξεργασίας τεράστιων όγκων αποθηκευμένων δεδομένων και εξαγωγή χρήσιμης πληροφορίας-γνώσης από αυτά. Από το όνομα φαίνεται ο παραλληλισμός με την εξόρυξη πολύτιμων μετάλλων όπως για παράδειγμά χρυσό από μεγάλους όγκους χώματος. Σε αυτή την περίπτωση χρυσός είναι η γνώση ενώ ο τεράστιος όγκος αχρείαστων δεδομένων το χώμα. Για την εξόρυξη πολύτιμης γνώσης χρειάζονται μεθοδολογίες από πολλούς επιστημονικούς κλάδους όπως για παράδειγμα μηχανική μάθηση, βάσεις δεδομένων, στατιστική, αναγνώριση προτύπων και άλλοι.

Ο όρος εξόρυξη δεδομένων πιο σωστά μπορεί να οριστεί και ως η ανακάλυψη γνώσης σε βάσεις δεδομένων (Knowledge Discovery in Databases – KDD) για τη συνολική διαδικασία ανακάλυψης προτύπων μέσα από μεγάλα και περίπλοκα σύνολα δεδομένων. Αυτή η διαδικασία ανακάλυψης γνώσης αποτελείται από μια αλληλουχία αυτοτελών τμημάτων επεξεργασίας τα οποία οδηγούν στη γνώση με τη καθαυτό εξόρυξη γνώσης να αποτελεί ένα από τα στάδια της. Κάθε στάδιο αυτής της διαδικασίας είναι σημαντικό ως προς την εξαγωγή της επιθυμητής πληροφορίας και δεν γίνεται απαραίτητα μόνο μια φορά. Δηλαδή η διαδικασία είναι αμφίδρομη και μπορεί να γίνει με πολλές επαναλήψεις επιμέρους διαδικασιών. Τα στάδια της ανακάλυψης γνώσης από βάσεις δεδομένων και οι επιμέρους εργασίες τους απεικονίζονται στην **Εικόνα 1** και είναι τα ακόλουθα.

1. **Συλλογή, Ολοκλήρωση και Καθαρισμός των Δεδομένων :**
   Αρχικά πρέπει να γίνει συλλογή των πηγαίων δεδομένων τα οποία μπορεί να βρίσκονται σε διάφορες πηγές όπως βάσεις δεδομένων. Αυτά τα δεδομένα όμως μπορεί να έχουν προβλήματα όπως εσφαλμένες τιμές, αντιφάσεις και έτσι πρέπει να καθαριστούν και στη συνέχεια να ομογενοποιηθούν.

2. **Επιλογή Δεδομένων και Μετασχηματισμός τους:**

   Οι μέθοδοι της εξόρυξης δεδομένων είναι ισχυρώς καθοδηγούμενες από τα δεδομένα έτσι η επιλογή των κατάλληλων δεδομένων για τη δημιουργία του συνόλου δεδομένων στο οποίο θα γίνει η εξόρυξη είναι κομβικής σημασίας. Αυτό επιτυγχάνεται με δειγματοληψία και επιλογή χαρακτηριστικών από τον αναλυτή που δεν θα απασχολήσουν περεταίρω αυτή την εργασία. Στη συνέχεια γίνεται ο μετασχηματισμός στα δεδομένα για να προσαρμοστούν σε απαιτήσεις των μεθόδων ανάλυσης. Το τελικό αποτέλεσμα αυτού του σταδίου είναι ένα σύνολο δεδομένων που θα χρησιμοποιηθεί για την εξαγωγή προτύπων.

3. **Εξόρυξη Δεδομένων:**

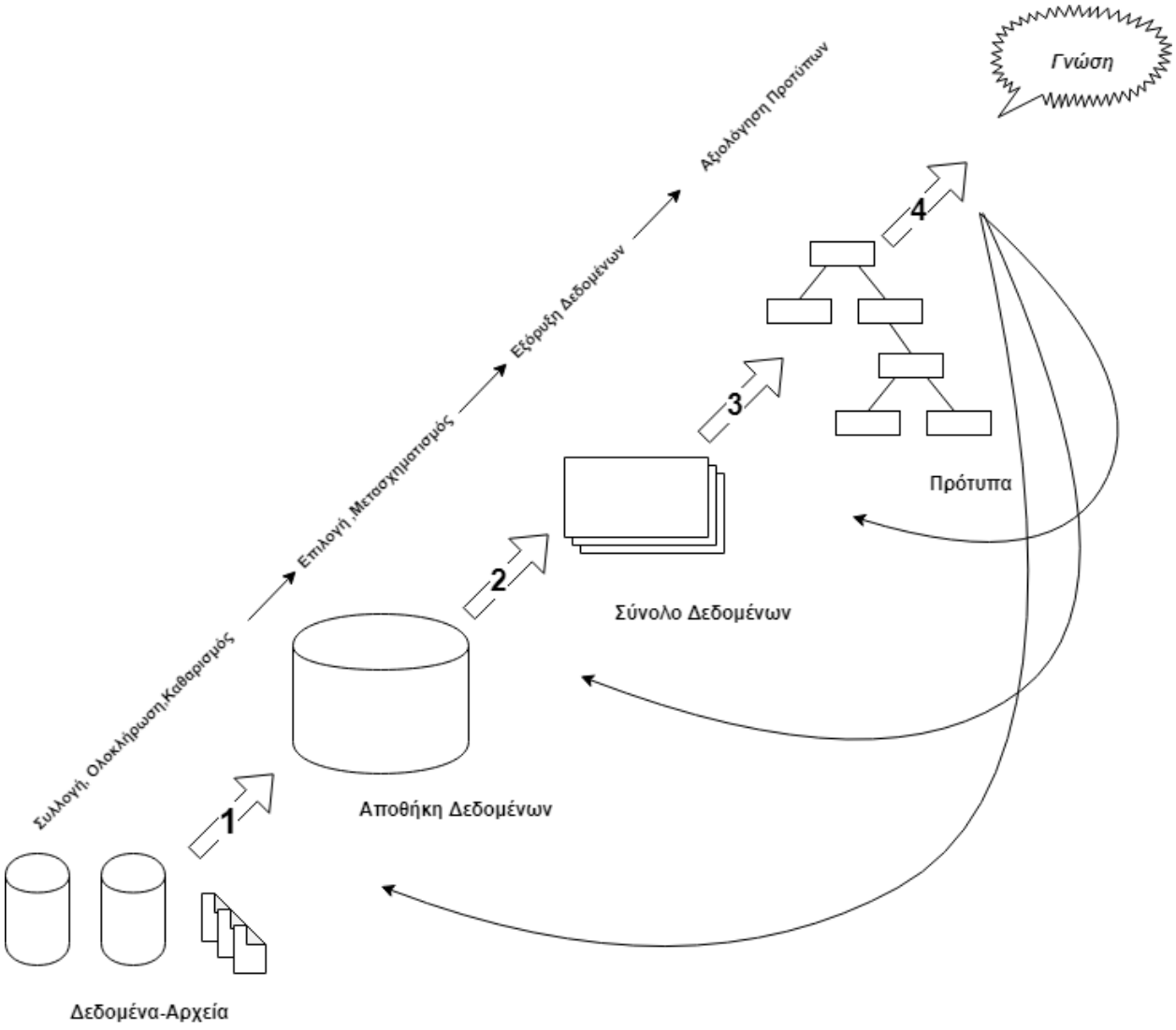
   Στο στάδιο αυτό γίνεται η καθεαυτό εξόρυξη δεδομένων ώστε να εξαχθούν πρότυπα δεδομένων. Δύο κύρια είδη ανάλυσης που μπορούν να εφαρμοστούν είναι: η περιγραφική ανάλυση (descriptive analytics) η οποία στοχεύει στην κατάδειξη ομαδοποιήσεων και ιδιοτήτων των δεδομένων και η προγνωστική ανάλυση (predictive analytics) που αποσκοπεί στην διατύπωση προβλέψεων για το μέλλον με τη χρήση μοντέλων. Υπάρχουν διάφορα είδη εργασιών εξόρυξης δεδομένων με πιο σημαντικά την κατηγοριοποίηση, ανάλυση συστάδων και ανάλυση κανόνων.

4. **Αξιολόγηση Προτύπων :**

   Τα πρότυπα που πρόεκυψαν από το προηγούμενο στάδιο αξιολογούνται. Αν δεν είναι ικανοποιητικά γίνεται επανάληψη προηγούμενων σταδίων με πιθανώς τροποποιημένα δεδομένα ή χρήση διαφορετικών μεθόδων εξόρυξης.

5. **Ανακάλυψη Γνώσης**

   Για την αναπαράσταση της γνώσης μπορούν να εφαρμοστούν διάφορες τεχνικές οπτικοποίησης και απεικόνισης.



Εικόνα 1: Στάδια διαδικασίας Ανακάλυψης Γνώσης

Όπως φαίνεται και από το σχήμα τα στάδια 1 και 2 με τις επιμέρους εργασίες τους είναι η προ-επεξεργασία των δεδομένων που θα χρησιμοποιηθούν για την εξόρυξη. Η εξόρυξη δεδομένων μπορεί να αλληλεπιδρά με τον χρήστη ή με μια βάση γνώσης και είναι ένα μόνο στάδιο στην όλη διαδικασία. Είναι όμως το πιο σημαντικό, διότι αποκαλύπτει κρυμμένα πρότυπα για αξιολόγηση. Στην εργασία αυτή θα υλοποιηθεί μια διεπαφή χρήστη προς την εξόρυξη δεδομένων από ιατρικές βιβλιογραφικές πηγές η οποία θα επιτρέπει στο χρήστη να αλληλεπιδρά με το σύστημα θέτοντας το δικό του θέμα για εξόρυξη δεδομένων παρέχοντας πληροφορίες και παράγοντες που θα εστιάσουν την αναζήτηση.

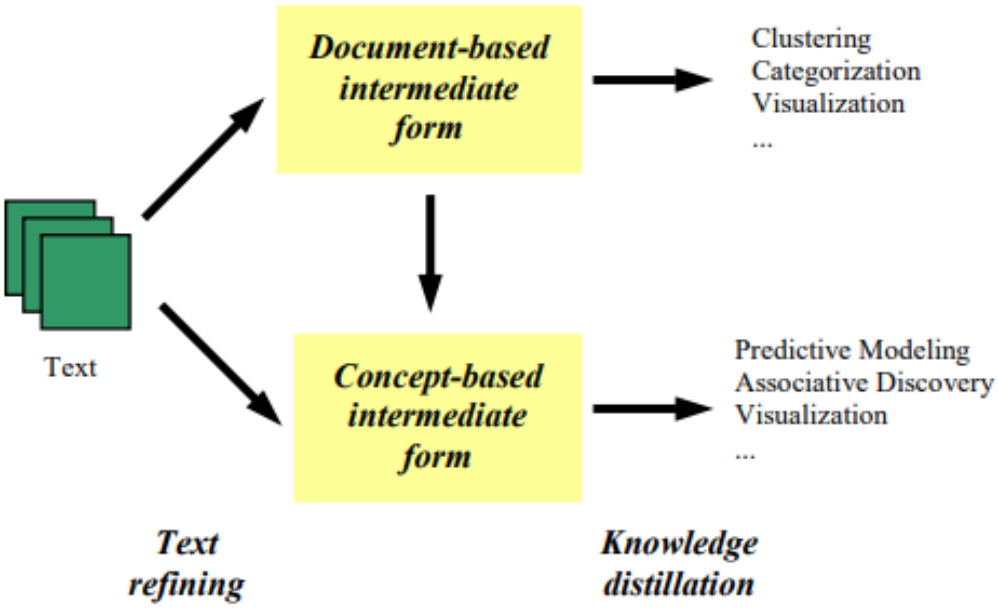
Επιπλέον μια διεπαφή χρήστη, επιτρέπει στον χρήστη να έχει πρόσβαση σε βάσεις δεδομένων ή αποθήκες δεδομένων, να αξιολογήσει εξορυγμένα πρότυπα και να οπτικοποιήσει τα πρότυπα σε διαφορετικές μορφές πράγμα που στην παρούσα εργασία τίθεται ως μελλοντική επέκταση.

1.3 Εξόρυξη δεδομένων από κείμενα (Text Mining)

Το μεγαλύτερο ποσοστό διαθέσιμων πληροφοριών στη σύγχρονη εποχή βρίσκεται αποθηκευμένο σε βάσεις δεδομένων αποκλειστικά από κείμενα. Σε αυτές υπάρχουν έγγραφα από εφημερίδες, ερευνητικές δημοσιεύσεις, βιβλία, ιστοσελίδες και άλλες διαφορετικές πηγές. Πλέον όμως με την εξέλιξη της τεχνολογίας και την ένταξη της πληροφορικής στην καθημερινή ζωή σχεδόν κάθε είδους έγγραφο υπάρχει και σε ηλεκτρονική μορφή. «Έτσι και ο ίδιος ο Παγκόσμιος Ιστός, μπορεί να θεωρηθεί σαν μια τεράστια, αλληλοσυνδεόμενη, δυναμική βάση κειμένου»[3].

Οι βάσεις κειμένων συχνά έχουν ένα ημι-δομημένο περιεχόμενο για την ευκολότερη αξιοποίηση τους. Δηλαδή ένα κείμενο μπορεί να έχει ορισμένα δομημένα πεδία όπως για παράδειγμα τίτλο, συγγραφέα, ημερομηνία αλλά και άλλες αδόμητες μορφές κειμένου όπως για παράδειγμα η περίληψη και τα περιεχόμενα του. Λόγω του ότι τα κείμενα δεν μπορούν να είναι αυστηρώς δομημένα, έχουν γίνει πολλές μελέτες για την μοντελοποίηση τους και έχουν αναπτυχθεί τεχνικές ανάκτησης δεδομένων όπως μέθοδοι τοποθέτησης δεικτών σε κείμενα για ευκολότερο χειρισμό τους. Χωρίς τη γνώση του περιεχομένου σε ένα έγγραφο είναι δύσκολο να σχηματιστούν αποδοτικά ερωτήματα για ανάλυση και εξαγωγή χρήσιμης πληροφορίας από αυτό, και έτσι κρίνεται αναγκαία η χρήση εργαλείων σύγκρισης κειμένων. Εδώ γίνεται χρήση της εξόρυξης δεδομένων από κείμενα αφού οι χρήστες με διάφορα εργαλεία συγκρίνουν, βαθμολογούν τη σχετικότητα και σημαντικότητα των κειμένων, βρίσκουν πρότυπα και ομοιότητες ανάμεσα σε αυτά. Αυτός ο λόγος καθιστά την εξόρυξη δεδομένων από κείμενα απαραίτητο κομμάτι στην αναζήτηση γνώσης με εξόρυξη δεδομένων. Στην παρούσα διπλωματική εργασία θα αναλυθεί περεταίρω η εξόρυξη δεδομένων από κείμενα μέσω της γλώσσας R σε επόμενη ενότητα.

Η εξόρυξη κειμένων μπορεί να παρουσιαστεί ως αποτέλεσμα δύο φάσεων: την μετατροπή κειμένων ελεύθερης μορφής σε πιο δομημένη μορφή (text refining) και την εξόρυξη γνώσης ή μοτίβων που οδηγούν στη γνώση από αυτή τη μορφή. Η ενδιάμεση μορφή μετά την πρώτη φάση μπορεί να είναι ημι-δομημένη. Επιπλέον η ενδιάμεση φάση μπορεί να είναι βασισμένη σε έγγραφο (document-based), δηλαδή κάθε οντότητα να αντιπροσωπεύει ένα έγγραφο, ή μια βάση δεδομένων (concept-based) στην οποία κάθε οντότητα αντιπροσωπεύει ένα αντικείμενο ή έννοια σε ένα συγκεκριμένο τομέα. Παραδείγματα εργασιών document-based εξόρυξης δεδομένων από κείμενα αποτελούν η συσσωμάτωση (clustering), η οπτικοποίηση (visualization) και η κατηγοριοποίηση (categorization). Ενώ παραδείγματα Concept-Based εξόρυξης δεδομένων από κείμενα είναι η προγνωστική μοντελοποίηση (predictive modelling) και η συσχετιστική ανακάλυψη (associative discovery). Μια ενδιάμεση document-based φάση μπορεί να μετατραπεί σε concept-based φάση επαναπροσδιορίζοντας ή εξάγοντας σχετικές πληροφορίες αναλόγως με το πεδίο ενδιαφέροντος[4].



Εικόνα 2: Οι δύο φάσεις εξόρυξης γνώσης κειμένου με παραδείγματα επί μέρους εργασιών τους.[2]

[2] http://www.ntu.edu.sg/home/asahtan/papers/tm_pakdd99.pdf

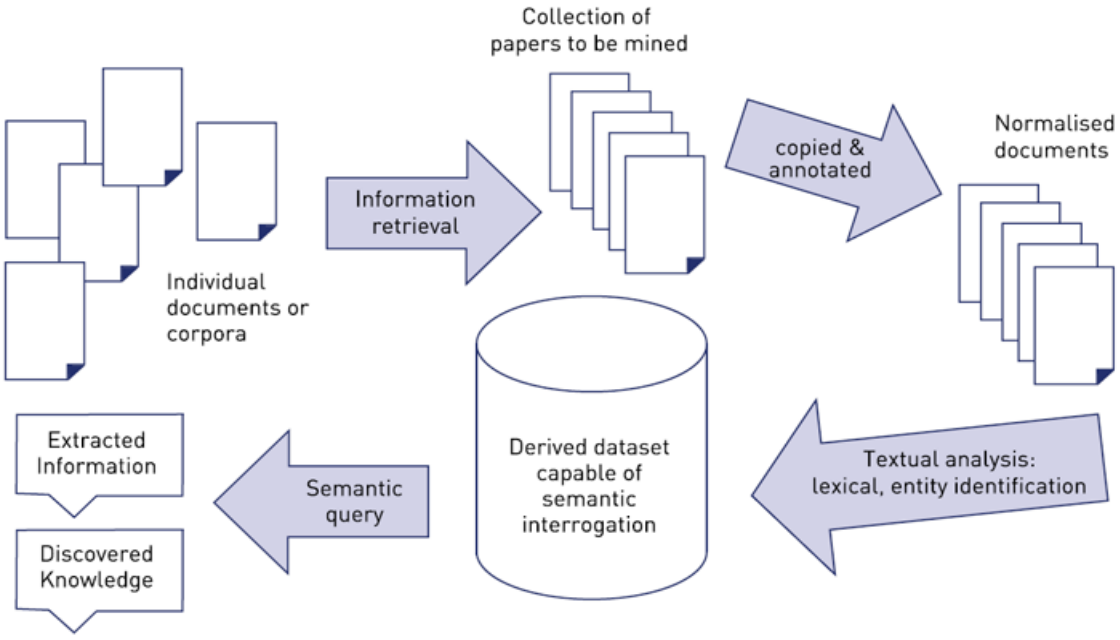
1.3.1 Ανάκτηση Πληροφοριών από κείμενα

Λόγω της αφθονίας γραπτών πληροφοριών, η ανάκτηση πληροφοριών με ακρίβεια και σχετικότητα είναι πολύ σημαντική. Καλούνται να εντοπιστούν σχετικά έγγραφα σε μια συλλογή εγγράφων βασισμένα στο ερώτημα ενός χρήστη. Συνήθως ο χρήστης καλείται να εισάγει κάποιες λέξεις – κλειδιά που περιγράφουν μια ανάγκη για συγκεκριμένες πληροφορίες ή ένα σχετικό κείμενο. Υπάρχουν διάφορες μέθοδοι ανάκτησης εγγράφων οι οποίες όμως μπορούν να χωριστούν σε δύο γενικές κατηγορίες. Ανάλογα με τον τρόπο αντιμετώπισης του προβλήματος ανάκτησης μπορούν να χωριστούν σε πρόβλημα επιλογής εγγράφων ή σε πρόβλημα ταξινόμησης εγγράφων.

Στις μεθόδους επιλογής εγγράφων, η ανάκτηση πληροφοριών ακολουθεί ένα αυστηρό κανόνα για την επιλογή των σχετικών εγγράφων με βάση το ερώτημα που έχει τεθεί. Δηλαδή τα έγγραφα περιορίζονται μόνο σε αυτά που καλύπτουν εξ ολοκλήρου το ερώτημα. Ένα παράδειγμα μεθόδου επιλογής εγγράφων είναι το μοντέλο ανάκτησης Boolean, κατά το οποίο τα επιλεγμένα έγγραφα για ανάκτηση αντιπροσωπεύονται από ένα σύνολο λέξεων-κλειδιά τα οποία ο χρήστης καλείται να συνοδεύσει από μια έκφραση Boolean. Δηλαδή καλείται να επιλέξει αν θέλει να ανακτήσει τα έγραφα τα οποία παρέχουν όλες τις λέξεις κλειδιά μέσα, λογικό AND ή αν θέλει να περιέχονται οποιεσδήποτε από αυτές, λογικό OR . Παραδείγματα είναι “Hearing Aid AND Diabetes”, ”GHABP OR MOCA“. Με τη χρήση λογικού AND η ανάκτηση γίνεται με μεγαλύτερη ακρίβεια και μεγαλύτερη ανάκληση. Ο όρος ακρίβεια μπορεί να οριστεί ως το ποσοστό των σχετικών εγγράφων που έχουν ανακτηθεί ως προς όλα τα ανακτημένα έγγραφα. Ενώ με τον όρο ανάκληση ορίζεται το ποσοστό των σχετικών εγγράφων που ανακτήθηκαν ως προς όλα τα σχετικά έγγραφα που έπρεπε να ανακτηθούν. Στην περίπτωση της μεθόδου επιλογής εγγράφων με το μοντέλο ανάκτησης Boolean υπάρχει μια σχετική δυσκολία ως προς την αποτίμηση της περιγραφής με ακρίβεια αφού όταν ο χρήστης δεν έχει επαρκή γνώση της συλλογής εγγράφων που διατίθεται δεν μπορεί να εκφράσει σωστά ερωτήματα.

Αυτή η δυσκολία αντιμετωπίζεται σε ένα βαθμό με τη χρήση μεθόδων ταξινόμησης εγγράφων. Σε αυτή την μέθοδο το ερώτημα που έχει τεθεί από το χρήστη χρησιμοποιείται ώστε να ταξινομήσει όλα τα έγγραφα με σειρά σχετικότητας. Για το λόγο αυτό η συγκεκριμένη μέθοδος

χρησιμοποιείται περισσότερο στα σύγχρονα συστήματα ανάκτησης δεδομένων. Υπάρχουν αρκετές διαφορετικές μέθοδοι ταξινόμησης με τις πλείστες να στηρίζονται στην θεωρία στατιστικής, πιθανοτήτων και άλλων μαθηματικών μεθοδολογιών. Σκοπός της μεθόδου ταξινόμησης είναι να ταιριάξει όσο το καλύτερο δυνατό το ερώτημα που τέθηκε από το χρήστη με τη χρήση λέξεων-κλειδιών και να ταξινομηθούν τα αποτελέσματα ανάλογα με την σχετικότητα σε αυτό το ερώτημα. Ο βαθμός σχετικότητας ενός εγγράφου μπορεί να υπολογιστεί βάση της συχνότητας λέξεων που υπάρχουν σε αυτό και κατά πόσο αυτές οι λέξεις είναι σχετικές με το ερώτημα που έχει ανατεθεί. Για παράδειγμα σε αναζήτηση με λέξη-κλειδί την ακοή, εκτός από τη βαρύτητα που φέρει ο αριθμός χρήσης αυτής της λέξης, εξίσου βαρύτητα έχει και η λέξη αυτί, κώφωση, ήχος, αίσθηση και άλλες λέξεις παρόμοιας κατηγορίας. Κάθε μία έχοντας διαφορετικό βαθμό βαρύτητας. Εντούτοις, είναι εγγενώς δύσκολο να υπάρχει ένα ακριβές μέτρο του βαθμού σχετικότητας ανάμεσα σε ένα σύνολο λέξεων-κλειδιών.



Εικόνα 3: Κύκλος εξόρυξης δεδομένων από κείμενα[3]

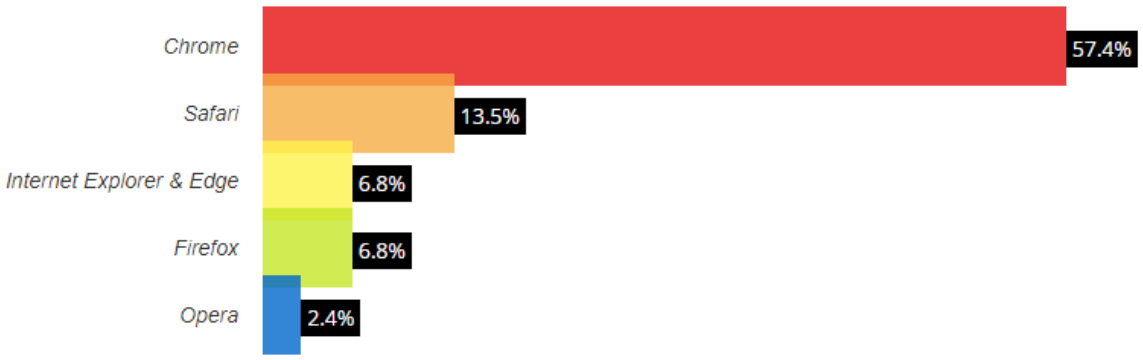
[3] https://libguides.cam.ac.uk/tdm/definitions

1.4 Τεχνολογίες Διαδικτύου

1.4.1 Φυλλομετρητής Ιστού (Web Browser)

Φυλλομετρητής Ιστού ή αλλιώς Web Browser ορίζεται ως ένα λογισμικό που επιτρέπει στο χρήστη να προβάλλει και να αντιδρά με διάφορες ιστοσελίδες. Δηλαδή επιτρέπει στο χρήστη να προβάλλει κείμενα, βίντεο, εικόνες και άλλες πληροφορίες από σελίδες που είναι αναρτημένες είτε στον παγκόσμιο ιστό είτε σε τοπικό δίκτυο. Οι Web Browsers έχουν τη δυνατότητα να παρουσιάζουν σελίδες που είναι κατασκευασμένες χρησιμοποιώντας τη γλώσσα Hypertext Mark-up Language (HTML) και Extensible Mark-up Language (XML) μεταφράζοντας τις σε περιεχόμενο που είναι εφικτό να διαβαστεί από τους χρήστες[4].

Υπάρχει μια ποικιλία διαθέσιμών browsers τόσο για διαφορετικά λειτουργικά συστήματα όσο και για τις διαφορετικές επεκτάσεις που παρέχουν ο καθένας. Πιο συχνά χρησιμοποιημένοι browsers είναι ο Internet Explorer, Firefox, Google Chrome, Safari και Opera. Στην Εικόνα 4 φαίνεται το διάγραμμα των πιο δημοφιλών Web Browsers από έρευνα που πραγματοποιήθηκε το Μάιο του 2019[5].



Εικόνα 4: Διάγραμμα δημοτικότητας Web Browser για το Μάιο 2019[4]

[4] https://www.w3counter.com/globalstats.php

1.4.2 HTML (Hypertext Markup Language)

Η HTML είναι μια γλώσσα που επιτρέπει στο χρήστη να κατασκευάσει ιστοσελίδες στον παγκόσμιο ιστό. Τα αρχικά HTML σημαίνουν Hypertext Mark-up Language. Ο όρος Hypertext ή αλλιώς υπερκείμενο είναι ένα σύνολο από πληροφορίες μέσα στο οποίο μπορεί να γίνει μετάβαση με μη γραμμικό τρόπο με τη χρήση συνδέσμων. Δηλαδή δίνεται η δυνατότητα πλοήγησης από μια σελίδα σε μια άλλη και αντίστροφα. Mark-up language είναι μια γλώσσα η οποία χρησιμοποιείται για μορφοποίηση και διαμόρφωση διάταξης εγγράφων κειμένου. Δηλαδή μπορεί να μετατρέπει απλό κείμενο σε δυναμικό και διαδραστικό περιεχόμενο (εικόνες, πίνακες, συνδέσμους). Αυτό γίνεται με τη χρήση ετικετών (tags) που προσθέτονται γύρω από τις λέξεις ή προτάσεις που θέλουν να παρουσιαστούν με κάποιο συγκεκριμένο τύπο μορφοποίησης. Η HTML διαθέτει ένα πεπερασμένο αριθμό ετικετών που μπορούν να χρησιμοποιηθούν, με τον αριθμό αυτό να αυξάνεται με το χρόνο. Στην εικόνα 5 φαίνεται ένα παράδειγμα απλής HTML με χρήση ετικετών.

```
<!DOCTYPE>
<html>
<head>
<title>Web page title</title>
</head>
<body>
<h1>Write Your First Heading</h1>
<p>Write Your First Paragraph.</p>
</body>
</html>
```

Εικόνα 5: Παράδειγμα δομής απλής HTML σελίδας με χρήση ετικετών

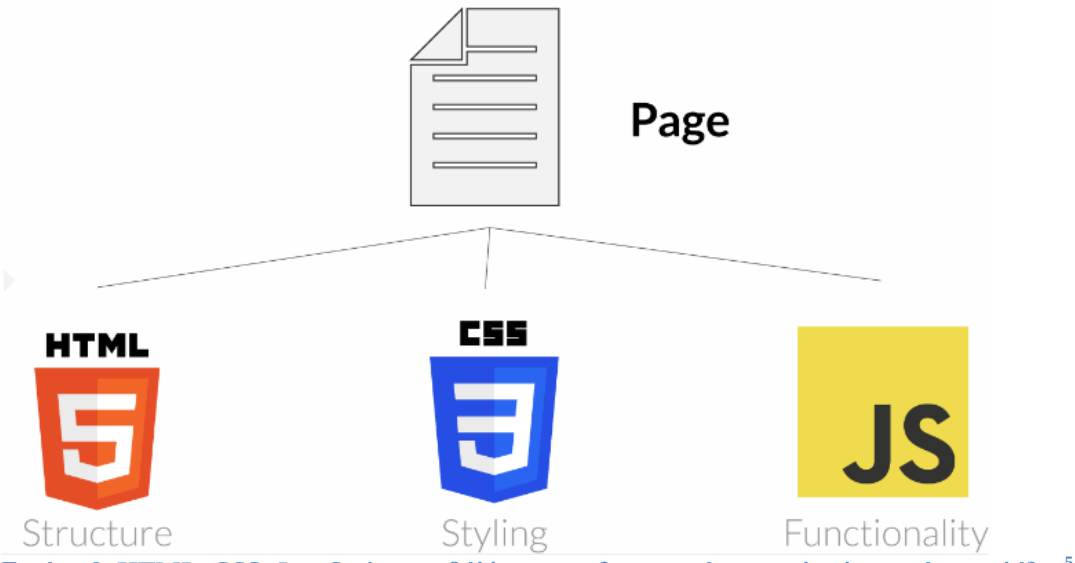
Η HTML δεν είναι γλώσσα προγραμματισμού έτσι δεν έχει τη δυνατότητα κατασκευής σελίδων με δυναμικό περιεχόμενο. Η δημιουργία ιστοσελίδων αποκλειστικά με HTML κρίνεται ανεπαρκής για τα σημερινά δεδομένα του δικτυακού προγραμματισμού. Οι σύγχρονες ιστοσελίδες αποτελούνται πλέον από ένα συνδυασμό δομής (structure), ύφους (style) και διαδραστικότητας (interactivity). Υπεύθυνες για τα τρία αυτά διαφορετικά στοιχεία είναι οι

τεχνολογίες HTML, CSS και JavaScript, αντίστοιχα, οι οποίες από κοινού συνεισφέρουν στη διανομή πλούσιου περιεχομένου σελίδων ιστού. Η HTML μέσω κατάλληλων tag μπορεί να συμπεριλάβει πλέον CSS και JavaScript αρχεία[6].

1.4.3 JavaScript

Η JavaScript δημιουργήθηκε από την Netscape Communications Corporation. Υποστηρίζει αντικειμενοστρεφές και συναρτησιακό στυλ προγραμματισμού. Η JavaScript δεν θα πρέπει να συγχέεται με τη Java, που είναι διαφορετική γλώσσα προγραμματισμού και έχει διαφορετικές εφαρμογές. Με την εισαγωγή της JavaScript οι ιστοσελίδες μπορούν να είναι πιο δυναμικές αφού γίνεται προσθήκη κουμπιών, εκτέλεση υπολογισμών και αλγορίθμων, προσθήκη συναρτήσεων για συγκεκριμένα γεγονότα (event handlers) όπως για παράδειγμα onClick, onHover, onSubmit events που δίνουν στην σελίδα πιο δυναμικό και αντιδραστικό χαρακτήρα. Επίσης δίνουν την ικανότητα επικοινωνίας με τον εξυπηρετητή και γενικότερα όλες τις δυνατότητες που προσφέρει μια πλήρης (κατά Turing) γλώσσα προγραμματισμού.

Σε πολλές περιπτώσεις ο κώδικας JavaScript μεταφράζεται και εκτελείται από τούς browser μέσω των JavaScript μηχανών που υπάρχουν στον πυρήνα τους. Έτσι η εκτέλεση δεν βασίζεται στην ταχύτητα του δικτύου άλλα αποκλειστικά στο τοπικό λογισμικό και υλικό.



Εικόνα 6: HTML, CSS, JavaScript συμβάλλουν στη δημιουργία μιας ολοκληρωμένης σελίδας[5]

[5] https://blog.codeanalogies.com/2018/05/09/the-relationship-between-html-css-and-javascript-explained/

Υπάρχει βέβαια και υλοποίηση JavaScript στο διακομιστή (server side) π.χ. με την πλατφόρμα Node.js, ενός μοντέλου προγραμματισμού βασισμένου στα συμβάντα (events). Στην παρούσα εργασία έχει χρησιμοποιηθεί η πλατφόρμα Node.js για τη δημιουργία του διακομιστή της πλατφόρμας στην οποία θα γίνει πιο εκτενής αναφορά στη συνέχεια.

Η τάση για δημιουργία πιο δυναμικών ιστοσελίδων με το συνδυασμό HTML, CSS, JavaScript, έχει οδηγήσει στην ανάπτυξη ειδικών βιβλιοθηκών (frameworks) τα οποία βοηθούν το έργο του προγραμματιστή.

1.4.4 Rest API (Representational State Transfer)

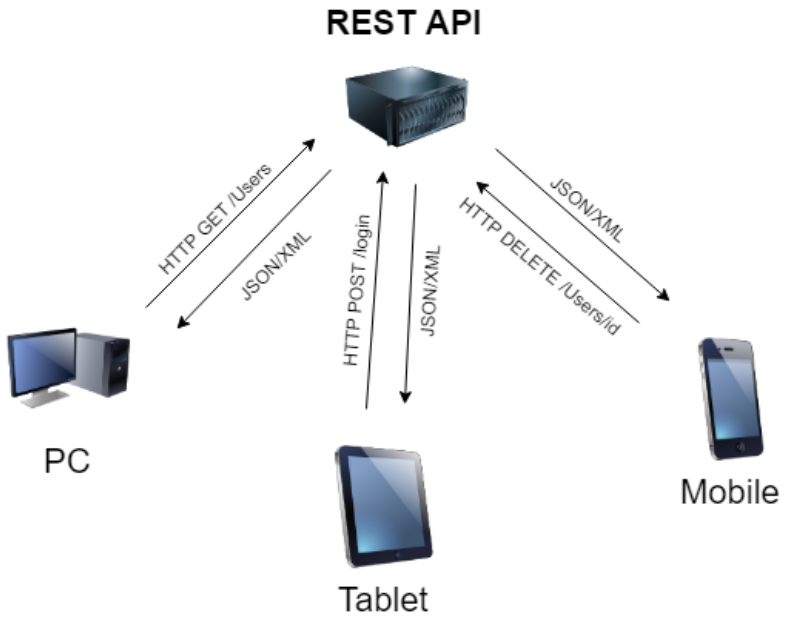
Το REST (Representational State Transfer) αποτελεί αρχιτεκτονική σχεδίασης δικτυακών υπηρεσιών. Πρωτοεμφανίστηκε το 2000 από τον Roy Fielding στην ακαδημαϊκή του διατριβή με τίτλο «Architectural Styles and the Design of Network-based Software Architectures»[7]. Συστήματα σχεδιασμένα σε REST αρχιτεκτονική χαρακτηρίζονται από καλή επίδοση, αξιοπιστία και δυνατότητα κλιμακωσιμότητας. Τα RESTful συστήματα επικοινωνούν δια μέσω του πρωτοκόλλου HTTP και έχουν τη δυνατότητα να δεχτούν αιτήματα από client ανεξαρτήτως γλώσσας υλοποίησης (**Εικόνα 7**). Βασικές μέθοδοι RESTful API είναι οι:

- **GET**: Συνήθως χρησιμοποιείται για την ανάκτηση πόρων (δεδομένων) από τον εξυπηρετητή.
    - ***Παράδειγμα*** : Ανάκτηση λίστας χρηστών από τον διαχειριστή.
- **POST**: Συνήθως χρησιμοποιείται για την δημιουργία/αποστολή πόρων (δεδομένων) στον εξυπηρετητή.
    - ***Παράδειγμα*** : Προσθήκη νέου χρήστη.
- **PUT**: Χρησιμοποιείται για την αλλαγή της κατάστασης ενός πόρου ή την ενημέρωσή του στον εξυπηρετητή.
    - ***Παράδειγμα*** : Αλλαγή προσωπικών στοιχείων χρήστη.
- **DELETE**: Χρησιμοποιείται για την διαγραφή ή απομάκρυνση ενός πόρου στον εξυπηρετητή.
    - ***Παράδειγμα*** : Διαγραφή Χρήστη.

Επειδή γίνεται χρήση συγκεκριμένων HTTP μεθόδων για την ανάκτηση ή αποστολή δεδομένων από/προς τον εξυπηρετητή, το είδος του αιτήματος καθορίζει την ενέργεια που θέλουμε να εφαρμόσουμε στο αντικείμενο αυτό και το περιεχόμενο του αιτήματος περιέχει διάφορες εξειδικεύσεις της ενέργειας αυτής. Έτσι με τη χρήση REST μεθόδων αρχικά δίνεται η διεύθυνση του πόρου που θα ασκηθεί η ενέργεια (endpoint) μαζί με το είδος μεθόδου και στη συνέχεια το περιεχόμενο σε μορφή JSON ή XML.

Η υπηρεσία REST χρησιμοποίει HTTP κωδικούς κατάστασης για την ευκολότερη κατανόηση της κατάστασης του, αφού υπάρχει ένας παγκόσμιος διακανονισμός ως προς την επεξήγηση τους. Για παράδειγμα ο κωδικός κατάστασης 200 αντιστοιχεί στην επιτυχία ενέργειας, 404 στο ότι ο πόρος δεν βρέθηκε και το 500 για εσωτερικό σφάλμα του εξυπηρετητή. Αξιοσημείωτοι είναι και οι κωδικοί 401, 403 ο οποίοι αντιστοιχούν στο ότι ο χρήστης δεν έχει τη δικαιοδοσία να εκτελέσει μια ενέργεια, ή στην απαγορευμένη ενέργεια αντίστοιχα.

Επιπλέον, είναι μια Stateless υπηρεσία δηλαδή δεν έχει κατάσταση. Ο όρος Stateless δηλώνει πως οποιαδήποτε κλήση σε μια υπηρεσία REST πρέπει να είναι ανεξάρτητη από άλλη προγενέστερη της. Ο Server δεν απασχολείται με προηγούμενες κλήσεις και ούτε γνωρίζει τι έχει προηγηθεί παρά μόνο επεξεργάζεται την κάθε κλήση ξεχωριστά και ανεξάρτητα.



Εικόνα 7: Επικοινωνία Client – REST API με HTTP ανεξαρτήτως πλατφόρμας/γλώσσας

1.5 Έρευνα επιλογής εργαλείου ανάπτυξης πλατφόρμας (Framework)

Όπως αναφέρθηκε σε προηγούμενη παράγραφο η ανάγκη χρησιμοποίησης JavaScript σε συνδυασμό με HTML και CSS για την δημιουργία δυναμικών και διαδραστικών ιστοσελίδων οδήγησε στην ανάπτυξη ειδικών βιβλιοθηκών (frameworks) τα οποία βοηθούν το έργο του προγραμματιστή. Εν έτη 2019 υπάρχουν τρία κύρια JavaScript Frameworks που υπερισχύουν στην αγορά τα οποία θα παρουσιαστούν στη συνέχεια.

1.5.1 Angular

Αναπτύχθηκε από την Google και κυκλοφόρησε για πρώτη φορά το 2010. Πρόκειται για ένα JavaScript framework βασισμένο σε Typescript. Τελευταία σταθερή έκδοση είναι το Angular 8 που κυκλοφόρησε τον Μάιο του 2019. Σημαντικότεροι υποστηρικτές της Angular είναι Google, YouTube, UPS, Microsoft[8].

Πλεονεκτήματα :

- Άψογος συνδυασμός TypeScript με JavaScript.
- Angular-language-service η οποία επιτρέπει τον αυτοματισμό και την διαδραστικότητα στα components με τη χρήση εξωτερικών HTML αρχείων.
- Πολλές νέες επεκτάσεις και npm βιβλιοθήκες από το CLI, και συνεχής ανάπτυξη Web Components βασισμένες στην Angular.
- Πλήρης λεπτομερής οδηγός υποστήριξης που επιτρέπει στο χρήστη τη λήψη όλων των απαραίτητων πληροφοριών που χρειάζονται για υλοποίηση μιας εφαρμογής.
- Επιτρέπει τη σύνδεση δεδομένων μονής κατεύθυνσης (one-way data binding) η οποία ελαχιστοποιεί τα πιθανά λάθη.
- MVVM (Model-View-ViewModel) που επιτρέπει στους προγραμματιστές να δουλεύουν ξεχωριστά στην ίδια ενότητα εφαρμογών χρησιμοποιώντας το ίδιο σύνολο δεδομένων.
- Ειδική δομή και αρχιτεκτονική που επιτρέπει την μεγάλη κλιμάκωση έργων.

Μειονεκτήματα :

- Περιέχει πολλά δομικά στοιχεία (injectables, components, pipes, modules) που καθιστούν την εκμάθηση της πιο πολύπλοκή από τις άλλες οι οποίες αποτελούνται από ένα ολικό component.
- Σχετικά βραδύτερη απόδοση και αρχικός χρόνος φόρτωσης.
- Συνεχείς ενημερώσεις οι οποίες συχνά εισάγουν μεγάλες αλλαγές που οδηγούν στην ανάγκη για αλλαγή, συνεχή εκμάθηση και συντήρηση των εφαρμογών από τους προγραμματιστές. Για παράδειγμα η AngularJS έχει διαφορετικές εντολές από την Angular 2+ έστω και αν θεωρείται αναβάθμιση της.

1.5.2 ReactJS

Η React εμφανίστηκε ως ένα έργο ανοικτού κώδικα από το Facebook το 2013, και μπορεί να θεωρηθεί ως μια JavaScript βιβλιοθήκη. Τελευταία σταθερή έκδοση είναι το React 16.8 που κυκλοφόρησε τον Φεβρουάριο του 2019. Χρησιμοποιείται κυρίως από Facebook, Instagram, Uber[9].

Πλεονεκτήματα :

- Εύκολη στην εκμάθηση λόγω απλότητας με τη χρήση JSX (τροποποιημένη εκδοχή HTML) σύνταξης με πλήρη λεπτομερή οδηγό υποστήριξης.
- Οι προγραμματιστές δαπανούν περισσότερο χρόνο γράφοντας JavaScript κώδικα και λιγότερο χρόνο για πιο συγκεκριμένο κώδικα react.
- Εξαιρετικά γρήγορη, με εφαρμογή Virtual - DOM και με διάφορες βελτιστοποιήσεις rendering.
- Μεγάλη υποστήριξη για rendering από πλευράς διακομιστή, καθιστώντας το ένα ισχυρό πλαίσιο για εφαρμογές με επίκεντρο το περιεχόμενο (content-focused application).
- One-way Data-binding επομένως λιγότερα σφάλματα.
- Εφαρμόζονται έννοιες συναρτησιακού προγραμματισμού δημιουργώντας έτσι αρκετό επαναχρησιμοποιούμενο κώδικα που κάνει τις δοκιμές πιο εύκολες.
- Η εναλλαγή μεταξύ των εκδόσεων είναι γενικά πολύ εύκολη, με το Facebook να παρέχει "codemods" για να αυτοματοποιήσει ένα μεγάλο μέρος της διαδικασίας.

- Οι δεξιότητες που αποκτήθηκαν μαθαίνοντας React μπορούν να εφαρμοστούν σχετικά άμεσα στην ανάπτυξη React Native εφαρμογών (για smartphones/tablets).

Μειονεκτήματα :

- Η React δεν έχει συγκεκριμένο τρόπο ανάπτυξης και αφήνει την επιλογή στους προγραμματιστές. Αυτό μπορεί να οδηγήσει συχνά σε λάθος διαδικασίες ανάπτυξης με πολλά λάθη (bugs). Για αυτό σε μεγάλες εργασίες πρέπει να υπάρχει ένας ισχυρός διακανονισμός σχετικά με την υλοποίηση που θα ακολουθηθεί ώστε να προληφθεί η ασυνέπεια.

- Υπάρχουν διαφορετικοί τρόποι μορφοποίησης των σελίδων, είτε με απλό CSS είτε με CSS-in-JS (emotion και styled components) επομένως οι προγραμματιστές διχάζονται στο θέμα ποιο τρόπο να επιλέξουν.

- Η React σιγά σιγά απομακρύνεται από τις κλάσεις και τον αντικειμενοστρεφή προγραμματισμό (object oriented programming).

- Μερικές φορές η χρήση JSX μπορεί να προκαλέσει σύγχυση στους προγραμματιστές όταν γίνει ανάμιξη μορφοποίησης με τη λογική.

1.5.3 Vue

Η Vue υλοποιήθηκε από τον Evan You πρώην υπάλληλο της Google to 2014, και τα τελευταία χρόνια έχει γνωρίσει μεγάλη δημοσιότητα, έστω και αν δεν έχει πίσω της υποστήριξη από μεγάλες εταιρείες όπως τις προηγούμενες (Angular-Google,React-Facebook). Τελευταία σταθερή έκδοση είναι το Vuejs 2.6 που κυκλοφόρησε τον Φεβρουάριο του 2019. Χρησιμοποιείται κυρίως από Xiaomi, WizzAir, EuroNews, Gitlab[10].

Πλεονεκτήματα:

- Εξουσιοδοτημένη HTML. Αυτό δείχνει την ομοιότητα στα χαρακτηριστικά με την Angular καθώς επιτρέπει την βελτιστοποίηση χειρισμού των html block με χρήση διαφορετικών εργαλείων.
- Λεπτομερής οδηγός υποστήριξης που μπορεί να σώσει πολύ χρόνο στον προγραμματιστή για να αναπτύξει μια εφαρμογή χρησιμοποιώντας μόνο τις βασικές γνώσεις HTML και JavaScript.
- Προσαρμογή λόγω της ομοιότητας με την Angular και React. Όσον αφορά το σχεδιασμό και την αρχιτεκτονική της είναι πιο εύκολη η εναλλαγή προς αυτήν.
- Μεγάλη κλιμακωσιμότητα. Μπορεί να βοηθήσει στην ανάπτυξη αρκετά μεγάλων επαναχρησιμοποιήσιμων προτύπων που μπορούν να κατασκευαστούν εύκολα και γρήγορα σύμφωνα με την απλή δομή της.
- Πολύ μικρό μέγεθος. Η Vue μπορεί να έχει μέγεθος μέχρι και 20 Kilo-Bytes (KB) διατηρώντας την ταχύτητα και την ευελιξία της που επιτρέπει την επίτευξη πολύ καλύτερων επιδόσεων σε σύγκριση με άλλα πλαίσια (frameworks).

Μειονεκτήματα **:**

- Πολύ λίγες πηγές και υποστήριξη. Σε σχέση με το μερίδιο αγοράς που αντιστοιχεί σε React και Angular η Vue δεν έχει τόσο μεγάλο ποσοστό, γεγονός που έχει σαν αποτέλεσμα να μην υπάρχει υποστήριξη από άλλους προγραμματιστές σε ιστοσελίδες ανταλλαγής γνώσεων για διάφορες απορίες (stackoverflow, github).
- Δεν υπάρχει ακόμα εμπειρία με πιθανές λύσεις γεγονός που οδηγεί σε προβλήματα κατά την ενσωμάτωση της σε τεράστια έργα, αφού κάθε προγραμματιστής ακολουθεί κάποιον δικό του τρόπο σύνταξης και οργάνωσης λόγω της μεγάλης ελαστικότητας που προσφέρει η Vue.

1.5.4 Στατιστικά δημοτικότητας και επαγγελματικής αποκατάστασης

Εκτός από τις διαφορές στην δομή, ανάπτυξη και υλοποίηση του κάθε framework σημαντικός παράγοντας στην επιλογή του καταλληλότερου για αυτή την εργασία είναι η δημοτικότητα, η προοπτική επαγγελματικής αποκατάστασης και η μελλοντική ανάπτυξη τους.

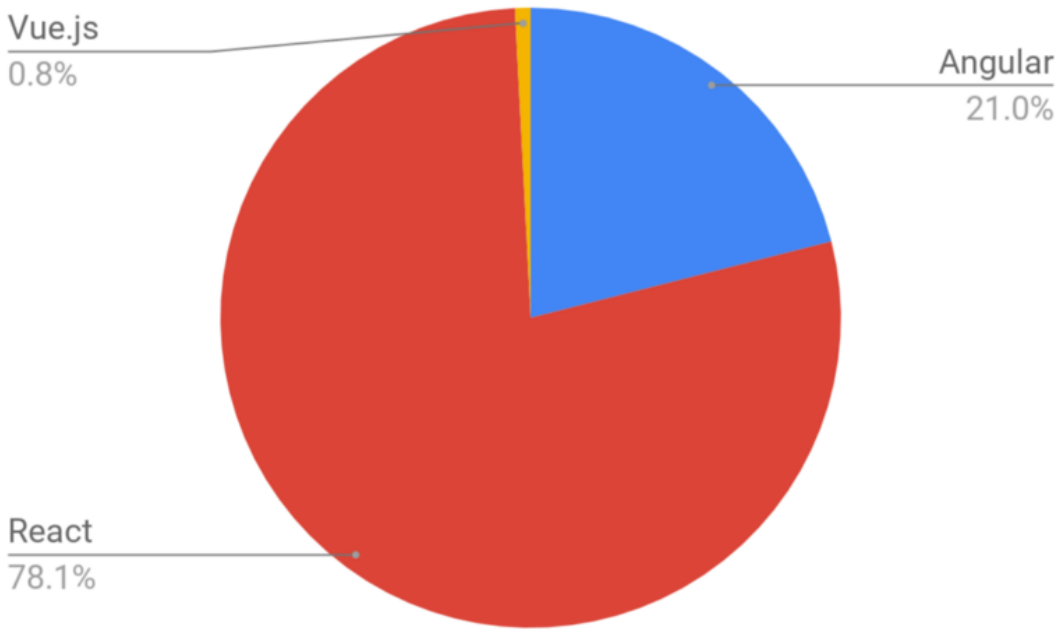
Στον **Πίνακα 1** φαίνονται τα στατιστικά για το κάθε Framework ξεχωριστά με τα στοιχεία να έχουν ληφθεί από τα κεντρικά Github repositories τους. Όπως φαίνεται από τον πίνακα η React και η View έχουν μεγάλους αριθμούς από watchers που παρακολουθούν αυτά τα frameworks αλλά και μεγάλο αριθμό starts. Βλέποντας όμως τους συνεισφέροντες (contributors) καταλαβαίνουμε το λόγο που η View ακόμα είναι στα αρχικά της στάδια αφού ο αριθμός τους είναι ο πιο μικρός. Η Angular ως η πιο παλιά από τις τρεις έχει έναν ικανοποιητικό αριθμό contributors που την καθιστούν σταθερή και έμπιστη για την χρησιμοποίηση της, όπως επίσης και τα περισσότερα commits, όμως το ενδιαφέρον για τις επεκτάσεις και το μέλλον της ως framework είναι μειωμένο.

	Angular[6]	**React**[7]	**Vue**[8]
Watchers	3,294	6,632	5,916
Stars	49,955	130,838	140,998
Forks	13,193	24,099	20,278
Contributors	948	1,294	272
Commits	14,448	11,020	3,018

Πίνακας 1: Στατιστικά Github για κάθε framework

[6] https://github.com/angular/angular
[7] https://github.com/facebook/react
[8] https://github.com/vuejs/vue

Στην **εικόνα 8** φαίνεται το ποσοστό ανοικτών θέσεων εργασίας που είναι διαθέσιμες ανά framework σε έρευνα που έγινε στο indeed.com πάνω σε 60000 ελεύθερες θέσεις[11]. Φαίνεται ξεκάθαρα πως η γνώση μιας εκ των δύο, Angular ή React, δίνει ένα τεράστιο ποσοστό πιθανότητας εύρεσης εργασίας με την React να κυριαρχεί. Η Vue έστω κι αν έχει μεγάλο κοινό ως προς την ανάπτυξη της ακόμη δεν έχει φτάσει σε ικανοποιητικό επίπεδο.



Εικόνα 8: Ποσοστό θέσεων εργασίας στα συγκεκριμένα frameworks[9]

Λαμβάνοντας υπόψη όλα τα πιο πάνω κατέληξα στην επιλογή της **React** ως το framework στο οποίο θα αναπτυχτεί πάνω η εργασία. Πολύ σημαντικό ρόλο στην επιλογή αυτή συνέβαλε το γεγονός ότι η React έχει αναπτυχθεί από την Facebook, η οποία στην εποχή που διανύουμε είναι μια από τις πιο ισχυρές, κερδοφόρες και σταθερές εταιρείες στην παγκόσμια κοινότητα. Η δημοτικότητα της και ο τεράστιος όγκος υλικού για βοήθεια και υποστήριξη που υπάρχει στο διαδίκτυο όπως επίσης και η προοπτική μελλοντικής ανάπτυξης και επαγγελματικής αποκατάστασης ήταν καθοριστικές για αυτή την επιλογή μου. Επίσης επιλέχθηκε η React για λόγους απλότητας και ευκολίας εκμάθησης αφού υπάρχει μεγάλη ομοιότητα με την JavaScript δίνοντας όμως μεγαλύτερη απόδοση από την Angular.

[9] https://medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d

Κεφάλαιο 2 : Εξόρυξη δεδομένων από κείμενα με χρήση R

2.1 Η γλωσσά R

Η R[10] είναι μια γλώσσα προγραμματισμού κυρίως διαδεδομένη για εργασίες με στατιστικούς υπολογισμούς, παραγωγή οπτικών και γραφικών απεικονίσεων και για επεξεργασία και ανάλυση δεδομένων κατά την εξόρυξη δεδομένων. Εκτός από γλώσσα προγραμματισμού μπορεί να θεωρηθεί και ως ένα περιβάλλον λογισμικού. Έχει βασιστεί στη γλώσσα προγραμματισμού S η οποία αναπτύχθηκε από τον John Chambers στα εργαστήρια της Bell. Η R δημιουργήθηκε από τους Ross Ihaka και Robert Gentleman, στο πανεπιστήμιο Auckland στη Νέα Ζηλανδία στις αρχές της δεκαετίας του 1990. Η R έγινε ιδιαίτερα δημοφιλής λόγω της ευκολίας στην εκμάθηση της, της συμβατότητας της με όλα τα κυρίως λειτουργικά συστήματα (Linux, Mac OS, Windows), την εξαιρετική απόδοση της σε επαναληπτικές διεργασίες και τη δωρεάν διαθεσιμότητα της. Επιπλέον η R παρέχει ένα μεγάλο αριθμό έτοιμων πακέτων που προσφέρονται στον χρήστη με πολύ αναλυτικά εγχειρίδια χρήσης. Ο αριθμός των πακέτων αυτών αυξάνεται με τη πάροδο του χρόνου αφού δίνεται η ευχέρεια στους ερευνητές να προτείνουν, να δημιουργήσουν και να ανεβάσουν καινούριες μεθοδολογίες σε μορφή πακέτων υποστηριζόμενων από την R.

Πλεονεκτήματα R σε σχέση με άλλα εμπορικά πακέτα:

- Διανέμεται δωρεάν και είναι ανοικτού κώδικα. Άλλα εμπορικά πακέτα παρόμοια με R έχουν μεγάλο κόστος που τα καθιστούν προσιτά μόνο από πανεπιστήμια, ερευνητικά κέντρα ή εταιρείες και όχι από απλούς χρήστες ή ερευνητές. Η χρήση αδείας ανοικτού κώδικα σημαίνει ότι ο καθένας μπορεί να συνεισφέρει και να τροποποιήσει το δικό του κώδικα. Με αυτό τον τρόπο μπορούν όλοι οι χρήστες να συνεισφέρουν ουσιαστικά στην ανάπτυξη της R.
- Ένα πρόγραμμα σε R είναι ελαφρύ και δεν χρειάζεται εξεζητημένο τεχνολογικό υλικό για να τρέξει. Δεν έχει πολλές απαιτήσεις σε μνήμη και αυτό δίνει τη δυνατότητα σε μεγαλύτερο ποσοστό χρηστών να δουλέψουν και να τρέξουν προγράμματα χωρίς να χρειάζονται υπολογιστές με μεγαλύτερη ισχύ.

[10] https://www.r-project.org/

- Η R είναι μια πλήρης και δομημένη γλώσσα προγραμματισμού, που προσφέρει στο χρήστη τεράστιες δυνατότητες επεξεργασίας και επίλυσης ποικίλων προβλημάτων με κάθε μορφή δεδομένων. Επίσης συνεργάζεται με άλλες γλώσσες προγραμματισμού, αφού μπορεί κανείς να καλέσει και να τρέξει μέσα από την R εντολές που έχουν γραφτεί σε C++ ή/και FORTRAN[12].
- Είναι ιδανική για αλληλεπίδραση με βάσεις δεδομένων αφού παρέχει αποτελεσματικό τρόπο ανάκτησης και αποθήκευσης δεδομένων ανεξαρτήτως μορφής.
- Η R είναι ευρέως χρησιμοποιούμενη για τη δυνατότητα παραγωγής από αυτήν πολλών ειδών διαγραμμάτων και γραφημάτων. Υπάρχει μεγάλη ποικιλία έτοιμων διαγραμμάτων σε πάρα πολλές μορφές που παρέχουν τη δυνατότητα στο χρήστη να φτιάξει πολύπλοκα διαγράμματα απλά με χρήση βιβλιοθηκών (ggplot). Τα διαγράμματα μπορούν να προσαρμοστούν ανάλογα με τις ανάγκες παρουσίασης που υπάρχουν, έτσι ο χρήστης παρουσιάζει τα αποτελέσματα του με πιο οπτικά μέσα χωρίς ιδιαίτερη προσθήκη άσκοπων γραμμών κώδικα.
- Η R παρέχει εξαιρετική απόδοση της σε επαναληπτικές διεργασίες και επιτρέπει παράλληλους υπολογισμούς που σημαίνει ότι ο χρόνος υπολογισμών μπορεί να περιοριστεί εντυπωσιακά.

Σε αντίθεση με άλλες γλώσσες προγραμματισμού η R δεν παρέχει μεταβλητές. Στη θέση των μεταβλητών υπάρχουν τα αντικείμενα τα οποία ανήκουν σε μια κλάση που μπορεί να θεωρηθεί ως ο «τύπος» τους. Δηλαδή τα πάντα στην R θεωρούνται ως αντικείμενα (objects). Κατά την ανάπτυξη κώδικα σε R η κλάση κάθε αντικειμένου τίθεται αυτόματα μετά από τον καθορισμό τιμής σε αυτό. Δεν απαιτείται δηλαδή δήλωση κλάσεων για κάθε αντικείμενο.

Υπάρχουν πέντε βασικές κλάσεις αντικειμένων:

- χαρακτήρας (character)
- αριθμητικός – πραγματικοί αριθμοί (numeric)
- ακέραιος (integer)
- σύνθετος (complex)
- λογικός (logical – True/False)

Επιπλέον η R παρέχει και τη δυνατότητα χρήσης δομών δεδομένων ως κλάσεις αντικειμένων. Βασικότερη δομή δεδομένων είναι το διάνυσμα (vector) που όμως μπορεί να αποτελείται μόνο από αντικείμενα ίδιου τύπου. Παρόμοιας λογικής είναι και οι δομές δεδομένων πινάκων μίας ή πολλών διαστάσεων. Για δομές δεδομένων που περιέχουν αντικείμενα διαφορετικών τύπων γίνεται χρήση των πλαισίων δεδομένων (data.frames) και λιστών (lists). Οι δομές δεδομένων περιέχουν αντικείμενα των κλάσεων που αναπτύχθηκαν πιο πάνω με τη μόνη διαφορά πως οι λίστες μπορούν να περιέχουν και δομές δεδομένων[13].

Η γλώσσα R παρέχει στο χρήστη βοήθεια δια μέσω της κονσόλας της. Αν δηλαδή κάποιος χρήστης έχει απορία σχετικά με κάποιο αντικείμενο ή πακέτο που χρησιμοποιεί στο πρόγραμμα του, μπορεί να ζητήσει πληροφορίες για αυτό δια μέσου της γραμμής εντολών της κονσόλας, πληκτρολογώντας ένα ερωτηματικό(?) μπροστά από το όνομα αυτού. Με αυτό τον τρόπο η R επιστρέφει στο χρήστη το εγχειρίδιο για αυτόν τον όρο που έχει θέσει από την επίσημη σελίδα. Επιπλέον όπως έχει αναφερθεί προηγουμένως, σε κάθε πρόγραμμα R γίνεται χρήση πολλών έτοιμων πακέτων που υπάρχουν στη βιβλιοθήκη της. Για την εγκατάσταση πακέτων ο χρήστης πρέπει να καλέσει τη συνάρτηση install.packages() εισάγοντας στο εσωτερικό των παρενθέσεων το όνομα του πακέτου αυτού. Το πακέτο τότε αποθηκεύεται στην βιβλιοθήκη της R που είναι εγκατεστημένη στον υπολογιστή και ο χρήστης στη συνέχεια απλά πρέπει να τη φορτώσει στο περιβάλλον που εργάζεται κάνοντας χρήση της συνάρτησης library() με το όνομα του.

Στη συνέχεια θα αναφερθούμε στην διαδικασία εξόρυξης δεδομένων που ακολουθήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας και ακολούθως θα γίνει επεξήγηση μερικών βασικών πακέτων R που έχουν χρησιμοποιηθεί.

2.2 Διαδικασία εξόρυξης δεδομένων δια μέσω R.

Σε αυτή την ενότητα θα παρουσιαστεί η μεθοδολογία η οποία ακολουθήθηκε για την εξόρυξη δεδομένων από τη βάση δεδομένων για βιβλιογραφικά κείμενα (PubMed). Αρχικά θα παρουσιαστούν τα βήματα/διαδικασίες που ακολουθήθηκαν για προετοιμασία των αποτελεσμάτων και στη συνέχεια θα γίνει μια μικρή επεξήγηση τους[14].

1) **Λήψη των σχετικών περιλήψεων κειμένων από PubMed**

   Γίνεται λήψη των σχετικών κειμένων που βασίζονται στα στοιχεία που έχει εισάγει ο χρήστης (δεδομένα, παράγοντες, χρονολογίες από και μέχρι).

2) **Δημιουργία Συλλογής (corpus) στοιχείων**

   Βασισμένη στα στοιχεία που έχει εισάγει ο χρήστης δημιουργείται μια συλλογή στην οποία θα γίνει η ανάλυση και η εξόρυξη δεδομένων.

3) **Αποθήκευση Συλλογής**

   Η συλλογή αποθηκεύεται με τη μορφή αρχείου .csv.

4) **Ενημέρωση λίστας Stop Word**

   Ενημερώνεται η λίστα των Stop Word με βάση τις επιλογές του χρήστη.

5) **Εκτέλεση εξόρυξης λέξεων στη συλλογή**

   Γίνεται η εκτέλεση της διαδικασίας εξόρυξης λέξεων στη συλλογή που έχει ανακτηθεί και κατασκευαστεί βάσει των στοιχείων που δόθηκαν στο προηγούμενο βήμα.

6) **Αριθμός άρθρων ανά χρονολογία.**

   Επιστρέφεται ο αριθμός των άρθρων που έχουν ανακτηθεί ανά χρονολογικό έτος.

7) **Περιλήψεις ανακτημένων κειμένων.**

   Επιστρέφεται για κάθε κείμενο που έχει ανακτηθεί η περίληψη (abstract) του.

8) **PubMed ID κειμένων.**

   Κάθε κείμενο που έχει ανακτηθεί συνοδεύεται από τον χαρακτηριστικό αριθμό που έχει στη βάση δεδομένων PubMed (PMID).

9) **Συχνότητα εμφάνισης n λέξεων.**

   Επιστρέφεται η συχνότητα εμφάνισης n λέξεων, όπου n ακέραιος αριθμός που έχει επιλεχτεί από τον προγραμματιστή.

10) **Εμφάνιση συχνότητας λέξεων ανά χρονολογικό έτος.**

    Επιστρέφεται η συχνότητα εμφάνισης των λέξεων για κάθε έτος.

11) **Οπτικοποίηση εξόρυξης λέξεων με χρήση διαγραμμάτων.**

Τα αποτελέσματα παρουσιάζονται σε μορφές διαγραμμάτων για την καλύτερη αξιοποίηση τους από τους χρήστες. Εμφανίζονται με διαγράμματα τύπου Barplot, WordCloud και Hierarchical Clustering.

12) **Αποθήκευση αποτελεσμάτων σε μορφή .pdf και .json**

Αποθήκευση αποτελεσμάτων για την περαιτέρω ανάλυση και χρησιμοποίηση τους.

2.2.1 Λήψη σχετικών περιλήψεων κειμένων από PubMed

Κατά την εξερεύνηση μεγάλων βάσεων δεδομένων (όπως η PubMed), οι αναλυτές αντιμετωπίζουν το πρόβλημα της επιλογής σχετικών εγγράφων για ποιοτική έρευνα και περαιτέρω ποσοτική ανάλυση. Σε αυτή την περίπτωση η συλλογή (corpus) από την οποία καλείται η διαδικασία να ανακτήσει κείμενα ανάλογα με τις λέξεις-κλειδιά που εισήγαγε ο χρήστης είναι δυναμικής φύσης και αποτελείται από αρκετές εκατοντάδες χιλιάδες αντικείμενα. Το μεγαλύτερο ποσοστό από αυτά μπορεί να θεωρηθεί άσχετό με το ερευνητικό ερώτημα που τίθεται. Για το λόγο αυτό ακολουθείται μια προσέγγιση αναζήτησης κειμένων δια μέσου των περιλήψεων τους. Δηλαδή γίνεται ανάκτηση κειμένων των οποίων η περίληψη είναι σχετική με το ερώτημα που έχει τεθεί από το χρήστη.

Ένα ερώτημα αναζήτησης που είναι πολύ γενικό θα επιστρέψει πολλές περιλήψεις που μπορεί να θεωρηθούν άχρηστες, ενώ ένα πολύ συγκεκριμένο ερώτημα μπορεί να είναι πολύ περιοριστικό για μια εκτεταμένη αναζήτηση. Για αυτό το λόγο γίνεται χρήση μοντέλων ανάκτησης δεδομένων Boolean ή ακόμα περιορισμός της αναζήτησης σε συγκεκριμένα πεδία όπως για παράδειγμα το όνομα του συγγραφέα ή τον τίτλο του άρθρου.

Ένας από τους κύριους στόχους κάθε διαδικασίας ανάκτησης δεδομένων είναι η μείωση του αρχικού μεγάλου συνόλου δεδομένων αναζήτησης σε μικρότερο το οποίο είναι και πιο εύχρηστο αφού τα δεδομένα δυνητικά θα είναι και πιο σχετικά μεταξύ τους. Σε αυτή τη περίπτωση γίνεται χρήση μοντέλων που μπορούν να θεωρηθούν ως λεξικά με βάση τα συμφραζόμενα (contextualized dictionaries) τα οποία ο χρήστης μπορεί προαιρετικά να χρησιμοποιήσει όπως παρουσιάζεται στο Κεφάλαιο 4.3.1 με χρήση παραδείγματος. Με αυτό τον τρόπο ο χρήστης καθοδηγείται ως προς την εισαγωγή ερωτημάτων ώστε να συγκεκριμενοποιείται το πεδίο αναζήτησης για ανάκτηση κειμένων.

2.2.2 Δημιουργία Συλλογής (Corpus)

Η εξόρυξη δεδομένων από μεγάλες συλλογές εγγράφων είναι μια πολύπλοκη διαδικασία. Η ύπαρξη μιας δομής δεδομένων για το κείμενο διευκολύνει την περαιτέρω ανάλυση των εγγράφων. Ο μετασχηματισμός εγγράφων υπολογίζεται στο συνολικό υπολογιστικό χρόνο μιας ανάλυσης κειμένου αφού συχνά απαιτείται η μετατροπή όλων των λέξεων σε αριθμητικά δεδομένα για ποσοτικοποίηση της αξιολόγησης, της στατιστικής ανάλυσης ή της μοντελοποίησης τους. Συνήθως, για ένα τέτοιο μετασχηματισμό τα έγγραφα πρέπει να χωριστούν σε ενιαίες λεξικές μονάδες, οι οποίες στη συνέχεια προσμετρούνται.

Ο πιο συνηθισμένος τρόπος αναπαράστασης των εγγράφων είναι ως απλές λέξεις χωρίς γραμματική και σειρά (bag of words) και ο υπολογισμός του αριθμού της εμφάνισης της κάθε λέξης ή φράσης. Αυτή η αναπαράσταση οδηγεί σε μια αναπαράσταση διανυσμάτων η οποία μπορεί να αναλυθεί με αλγόριθμους μείωσης διαστάσεων, μηχανική μάθηση(machine learning) και στατιστική.

Στην συγκεκριμένη εργασία έχει γίνει χρήση αλγορίθμου topic modeling ο οποίος είναι ένας από τους πιο δημοφιλείς πιθανολογικούς αλγορίθμους ομαδοποίησης. Έγινε χρήση του για την δημιουργία της συλλογής (corpus) για τα ανακτημένα έγγραφα κειμένου. Τα θεματολογικά μοντέλα (topic models) βασίζονται στην ιδέα πως τα κείμενα μπορούν να θεωρηθούν ως μια μίξη από θεματολογίες, οι οποίες ορίζονται ως κατανομές πιθανότητας πάνω στις λέξεις ενός λεξιλογίου. Σε ένα μοντέλο θέματος (topic model) μπορούν να εντοπιστούν συνήθως θέματα με ανεπιθύμητο περιεχόμενο. Σε αντίθεση με άλλες μεθόδους εξαγωγής λέξεων-κλειδιών οι οποίες παραμελούν την αλληλεξάρτηση των όρων, η προσέγγιση μοντέλου θέματος επιτρέπει να αποκλείονται τέτοιες ανεπιθύμητες σημασιολογικές συστάδες[15].

2.2.3 Stop Words και προετοιμασία εξόρυξης δεδομένων από κείμενα

Ανάλογα με την εφαρμογή, η ανάλυση και μέτρηση των δεδομένων μπορεί να γίνεται σε ολόκληρα έγγραφα, παραγράφους ή και απλές προτάσεις για τον περιορισμό του περιεχομένου και την πιο σχετική συγκέντρωση πληροφοριών. Μετά τον ορισμό του πεδίου που θα αναλυθεί από τη μονάδα ανάλυσης και τον αντίστοιχο διαχωρισμό δεδομένων πρέπει να γίνει εντοπισμός των λεξικών μονάδων, οι οποίες πλέον είναι γνωστές ως tokens. Η διαδικασία αυτή ονομάζεται “tokenization” και κατά τη χρήση της γίνεται χωρισμός όλων των διακριτών μορφών λέξεων

μιας συλλογής (corpus). Αυτές οι διακριτές μορφές λέξεων θεωρούνται ως ξεχωριστοί τύποι και η μέτρηση τους από κάθε μονάδα ανάλυσης κωδικοποιείται και αποθηκεύεται σε διαφορετική συλλογή γνωστή ως Document-Term Matrix (DTM). Ο τρόπος με τον οποίο γίνεται διαχωρισμός ενός κειμένου σε tokens επηρεάζει την ανάλυση που ακολουθεί αφού ορίζονται οι ατομικοί τύποι που αντιστοιχούν στη σημασιολογία του κειμένου που έχει αναλυθεί. Τα tokens μπορεί να είναι μεμονωμένοι όροι, σημεία στίξης ή και ομάδες λέξεων που χρησιμοποιούνται αναλόγως της σημασιολογίας τους. Υπάρχει μια ποικιλία διαδικασιών για την προ-επεξεργασία κειμένων δεδομένων πριν την εισαγωγή τους στα DTM. Μετά την προ-επεξεργασία και την εισαγωγή τους στα DTM συχνά γίνεται περεταίρω μαθηματική ανάλυση σε αυτά με σκοπό την προετοιμασία τους για την επακόλουθη εξόρυξη δεδομένων.

Τα κυριότερα βήματα προ-επεξεργασίας που λαμβάνουν μέρος στην εργασία αυτή είναι:

- **Tokenization:** Ο διαχωρισμός του κειμένου σε tokens μπορεί να επιτευχθεί σε πολλές γλώσσες απλά με την αφαίρεση των χαρακτήρων λευκού διαστήματος (white spaces) μεταξύ των λέξεων. Ωστόσο, αυτή η προσέγγιση χάνει τον διαχωρισμό των σημείων στίξης από μεμονωμένους όρους ή δεν καλύπτει την αναγνώριση μονάδων πολλαπλών λέξεων (multi-words).
- **Cleaning:** Σε αρκετές περιπτώσεις χρήσης δεν είναι όλοι οι αναγνωρισμένοι τύποι λεξικών μονάδων χρήσιμοι, αφού δεν συμβάλλουν στην επιθυμητή ανάκτηση πληροφορίας ως προς τη σημασιολογία που χρειάζεται να αναλυθεί. Για παράδειγμα, λέξεις όπως τα άρθρα ή οι αντωνυμίες συχνά δεν καλύπτουν τις σχετικές πτυχές της έννοιας της σημασιολογίας. Το ίδιο μπορεί να ισχύει για σημεία στίξης ή αριθμούς στο κείμενο. Εάν είναι χρήσιμο, τέτοιοι τύποι λεξικών μονάδων μπορούν να παραληφθούν για να μειώσουν την ποσότητα των δεδομένων ώστε η ανάλυση να επικεντρωθεί στις πιο σημαντικές γλωσσικές πτυχές. Για αυτό το λόγο δίνεται δυνατότητα στο χρήστη να εισάγει δικές του λέξεις (stop words) οι οποίες δεν χρειάζονται περαιτέρω ανάλυση αφού δεν συμβάλλουν στην επιθυμητή σημασιολογική έννοια που χρειάζεται να ανακτηθεί. Αυτές οι λέξεις αφαιρούνται κατά τη διαδικασία αυτή από τα δεδομένα που θα αναλυθούν.
- **Unification:** Οι λεξικές μονάδες εμφανίζονται με διαφορετικούς τρόπους ορθογραφίας και συντακτικής μορφής. Παραλλαγές του ιδίου ουσιαστικού μπορεί να εμφανιστούν σε

ενικό, πληθυντικό και ρήματα μπορεί να υπάρξουν σε διαφορετικές μορφές. Σε αυτή τη διαδικασία, αυτές οι μορφές ενοποιούνται σε μια ενιαία βασική μορφή, για να αντιμετωπιστούν περιστατικά διακυμάνσεων πανομοιότυπων δεδομένων. Πιο συνηθισμένες διαδικασίες ενοποίησης είναι η μετατροπή όλων των χαρακτήρων σε πεζά (lower-case), το Stemming που συνήθως σε μια διαδικασία απομακρύνει τα άκρα των λέξεων για την επίτευξη του στόχου αυτού και συχνά περιλαμβάνει την αφαίρεση των προθεμάτων και τη διαδικασία Lemmatization που αποσκοπεί στη σωστή χρήση λεξιλογίου και μορφολογικής ανάλυσης των λέξεων, με σκοπό την αποκοπή καταλήξεων ώστε να επιστραφεί η βασική θεματολογία στην οποία στηρίζεται η κάθε λέξη[16].

Αυτές οι διαδικασίες προ-επεξεργασίας κειμένων διαμορφώνουν ένα σύνολο από λεκτικούς τύπους που θα ληφθούν υπόψη στη δημιουργία ενός DTM με τον εντοπισμό, μετασχηματισμό και φιλτράρισμα λέξεων, χωρίς να αλλοιώνεται το γνωστικό περιεχόμενο του κειμένου. Δεν υπάρχει ιδανική ή σωστή πορεία προ-επεξεργασίας δεδομένων και αποτελεί επιλογή του αναλυτή.

2.2.4 Λεξικομετρική Ανάλυση κειμένων

Ανάλυση συχνότητας (Frequency analysis): Σε αυτή την ανάλυση οι συγκεκριμένοι όροι ή οι έννοιες που εμφανίζονται στα έγγραφα, μετρώνται και οι μετρήσεις χωρίζονται ανά έτη όπου και γίνεται σύγκριση τους. Η παρατήρηση των συχνοτήτων εμφάνισης των όρων κατά τη διάρκεια των ετών, μπορεί να αποκαλύψει τις κορυφές και τις διακυμάνσεις της χρήσης τους και αναλόγως να εξαχθούν συμπεράσματα σχετικά με τη θεματολογία τους.

Εξαγωγή πληροφοριών (Information Extraction-IE): Αυτή η ανάλυση αποσκοπεί στον εντοπισμό ονομάτων, όρων ή εννοιών σε ένα έγγραφο. Συνήθως, πραγματοποιείται με πιθανολογική ταξινόμηση που καθορίζει την πιο πιθανή κατηγορία για οποιοδήποτε όρο σε μια πρόταση. Είναι χρήσιμο να προσδιορίσει τους όρους ή τις έννοιες που σχετίζονται με οποιαδήποτε άλλη πληροφορία που προσδιορίζεται σε ένα κείμενο.

Co-occurrence analysis: Η κοινή εμφάνιση όρων σε ένα κείμενο παρατηρείται και αξιολογείται από μια στατιστική ανάλυση. Δηλαδή, για κάθε τύπο λέξης εμφανίζει μια λίστα με άλλες λέξεις που συνυπάρχουν με αυτήν πιο συχνά. Για παράδειγμα αυτές οι λέξεις παρουσιάζονται

γειτονικές σε προτάσεις ή απλά συνδυάζονται αρκετά συχνά σε ολοκληρωμένες προτάσεις. Αυτό μπορεί να αποκαλύψει σημασιολογικά πεδία από αλληλοσχετιζόμενους όρους. Η περαιτέρω σύγκριση και η ταξινόμηση τέτοιων σημασιολογικών πεδίων μπορεί να αποκαλύψει πραγματικά συναφείς όρους οι οποίοι σχετίζονται σημασιολογικά. Στα πλαίσια της πτυχιακής εργασίας έχει χρησιμοποιηθεί μια προσέγγιση ιεραρχικής ομαδοποίησης με χρήση συστάδων (Hierarchical Clustering).

2.3 Πακέτα R

Όπως αναφέρθηκε στην προηγούμενη ενότητα η R παρέχει πάρα πολλά εξωτερικά πακέτα υποστήριξης και επέκτασης για την ευκολότερη χρήση της. Για την υλοποίηση εξόρυξης δεδομένων από κείμενα ιατρικών βιβλιογραφικών βάσεων με χρήση της γλώσσας R χρησιμοποιήθηκαν τα εξής πακέτα :

RISmed[11]: Το πακέτο RISmed χρησιμοποιείται για τη λήψη κειμένων από τη βάση δεδομένων του εθνικού κέντρου Βιοτεχνολογικών πληροφοριών των Ηνωμένων πολιτειών της Αμερικής (United States of America National Center for Biotechnology Information -NCBI). Παρέχει αρκετά εργαλεία για την ανάκτηση βιβλιογραφικών δεδομένων από τις NCBI βάσεις δεδομένων συμπεριλαμβανόμενης και της PubMed.

Qdap[12]**(Quantitative Discourse Analysis Package):** Είναι ένα πακέτο R που χρησιμοποιείται για ποιοτική ανάλυση δεδομένων προσφέροντας ποσοτική ανάλυση λόγου, ανάλυση εγγράφων και επεξεργασία φυσικής γλώσσας.

Παρέχει:

- Εργαλεία για προ-επεξεργασία δεδομένων από έγγραφα
- Μετρητές συχνοτήτων λέξεων, προτάσεων και άλλων τύπων δεδομένων
- Συνάθροιση χρησιμοποιώντας μεταβλητές ομαδοποίησης
- Ανάκτηση λέξεων και οπτικοποίηση

[11] https://cran.r-*project*.org/web/packages/RISmed/RISmed.pdf
[12] https://cran.r-project.org/web/packages/qdap/qdap.pdf

- Στατιστική ανάλυση

SnowballC[13]: Το πακέτο SnowballC παρέχει στην R μια διασύνδεση με την βιβλιοθήκη libstemmer της γλώσσας C η οποία υλοποιεί τον αλγόριθμο Porter's word Stemming για την απλοποίηση και ομαδοποίηση των λέξεων με βάση τη κοινή τους ρίζα για καλύτερη σύγκριση μεταξύ των λεξιλογίων.

tm[14]: Είναι μια βιβλιοθήκη (framework) για εφαρμογές εξόρυξης δεδομένων.

• **WordCloud**[15]**:** Το πακέτο WordCloud παρέχει στο χρήστη την ικανότητα δημιουργίας διαγράμματος συννεφόλεξου για την καλύτερη οπτικοποίηση των αποτελεσμάτων.

• **Jsonlite**[16]**:** Είναι ένα πακέτο που παρέχει ένα γρήγορο κατασκευαστή και αναλυτή αντικειμένων JSON βελτιστοποιημένο για στατιστικά δεδομένα και τον παγκόσμιο ιστό. Το πακέτο προσφέρει ευέλικτα, ισχυρά εργαλεία υψηλής απόδοσης αλληλεπίδρασης με τα JSON αντικείμενα στην R και είναι ιδιαίτερα ισχυρό για την κατασκευή αγωγών (pipes) και την αλληλεπίδραση με ένα web API.

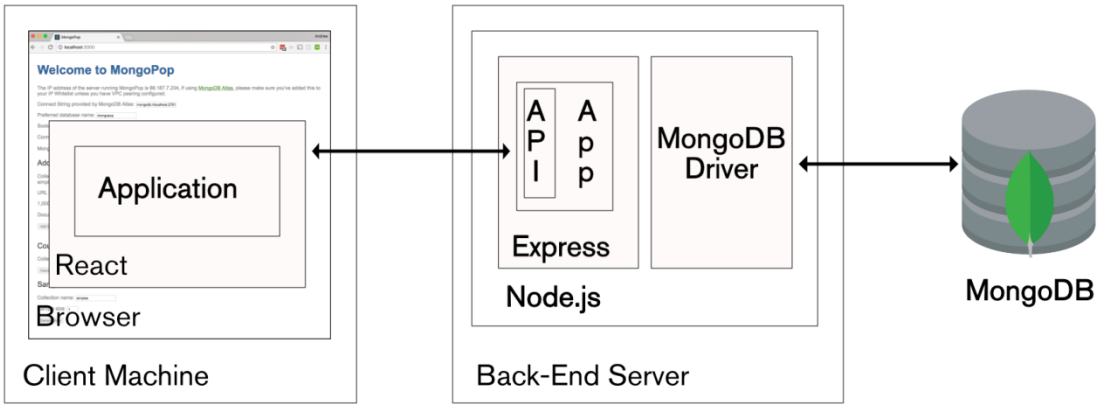
• **Dendextend**[17]: Το πακέτο Dendextend προσφέρει μια συλλογή από συναρτήσεις επεκτάσεων για τα δενδρογράμματα στην R.

[13] https://cran.r-project.org/web/packages/SnowballC/index.html
[14] https://cran.r-project.org/web/packages/tm/index.html
[15] https://cran.r-project.org/web/packages/wordcloud/index.html
[16] https://cran.r-project.org/web/packages/jsonlite/index.html
[17] https://cran.r-project.org/web/packages/dendextend/

Κεφάλαιο 3 : Απαιτούμενα και σχεδιασμός

3.1 Αρχιτεκτονική και υπηρεσίες

Η αρχιτεκτονική που ακολουθήθηκε για την ανάπτυξη της πλατφόρμας στηρίζεται στο μοντέλο Διακομιστή-Πελάτη (Server-Client), με την επικοινωνία μεταξύ των δύο να επιτυγχάνεται δια μέσω HTTP αιτημάτων. Έχει γίνει χρήση αρχιτεκτονικής που συνδυάζει MongoDB, ExpressJS(Node.js), ReactJS (Full MERN Stack). Για την υλοποίηση της σελίδας πελάτη (Client) έγινε χρήση της ReactJS η οποία είναι ένα JavaScript framework για ανάπτυξη δυναμικών και διαδραστικών σελίδων για web browsers. Ενώ για την υλοποίηση του διακομιστή (Server) έγινε χρήση της Node.js για δημιουργία ενός RESTful API, η οποία είναι βασισμένη στην JavaScript και προσφέρει πάρα πολλές βιβλιοθήκες για την ευκολότερη υλοποίηση της πλατφόρμας. Αρχική ιδέα ήταν να υπάρχουν 2 είδη χρηστών, διαχειριστής (admin) και απλός χρήστης (user), με τον πρώτο να έχει τη δυνατότητα εγγραφής καινούριων χρηστών. Για την ύπαρξη χρηστών και την σωστή οργάνωση των στοιχείων τους γίνεται χρήση της βάσης δεδομένων MongoDB. Για την εξουσιοδότηση των χρηστών γίνεται χρήση JSON Tokens. Στη συνέχεια θα γίνει μια εκτενέστερη αναφορά σε κάθε εργαλείο που έχει χρησιμοποιηθεί.



Εικόνα 9: Αρχιτεκτονική Υλοποίησης Πλατφόρμας (Full MERN Stack)[18]

[18] https://www.mongodb.com/blog/post/the-modern-application-stack-part-5-using-reactjs-es6-and-jsx-to-build-a-ui-the-rise-of-mern

3.1.1 JSON Objects

Το JSON (JavaScript Object Notation) είναι μια μορφή δεδομένων που συχνά χρησιμοποιείται για την ανταλλαγή δεδομένων λόγω του μικρού μεγέθους της. Είναι εύκολο να διαβαστεί και να συνταχθεί από το χρήστη όπως επίσης και να αναλυθεί, επεξεργαστεί από τις μηχανές. Το JSON είναι μια μορφή κειμένου πλήρως ανεξάρτητη από τη γλώσσα προγραμματισμού η οποία χρησιμοποιείται.

Το JSON μπορεί να εμφανιστεί σε δύο δομές:

- Μια συλλογή ζευγών ονόματος/τιμής. Σε διάφορες γλώσσες, αυτό πραγματοποιείται ως αντικείμενο, εγγραφή, struct.
- Μια ταξινομημένη λίστα τιμών. Στις περισσότερες γλώσσες, αυτό πραγματοποιείται ως ένας πίνακας, μια λίστα.

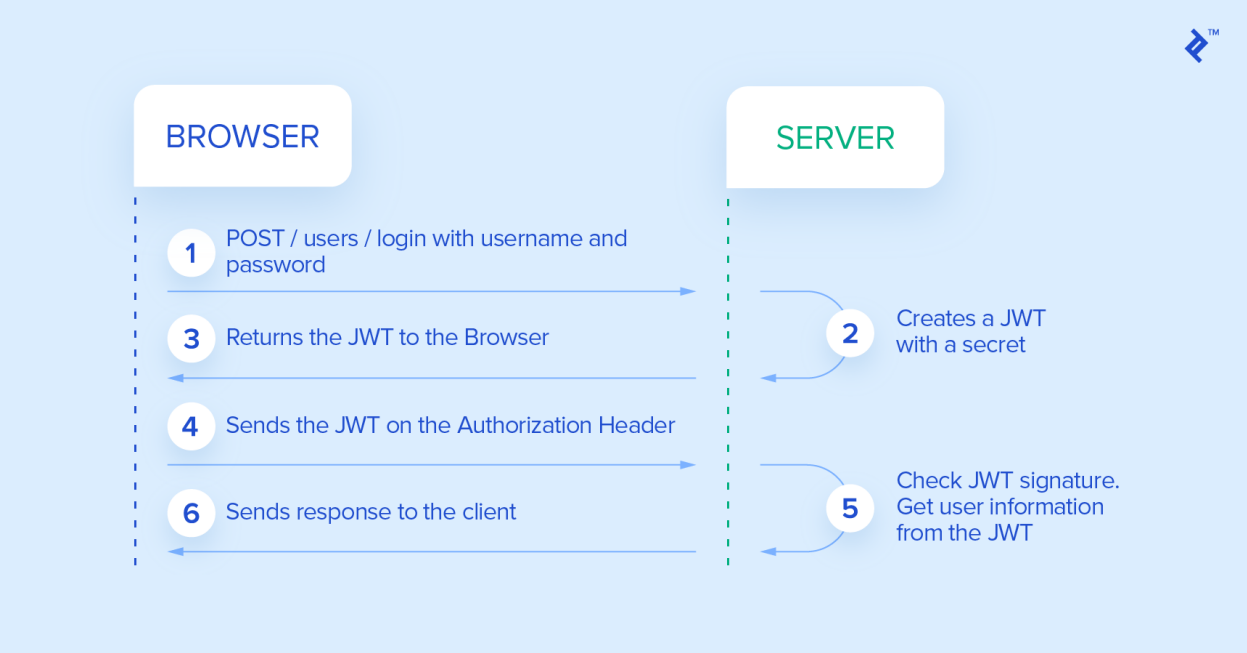
Παράδειγμα JSON Object :


```
{
    "_id" : "5cd2f32e2090b736d0a2a031",
    "admin" : true,
    "username" : "John27",
    "email" : "User2@gmail.com",
    "password" : "$2a$08$OE8ZaGhFx2SOICjwYpFJouayIaFJudVee0dg5ktJcpyYof/UUDWre",
    "firstname" : "John",
    "lastname" : "Black",
    "date_of_birth" : "1994-11-01",
    "gender" : "M"
}
```


3.1.2 JSON Web Token

JSON Web Token (JWT) είναι μια πληροφορία σε ασφαλή και συμπυκνωμένη μορφή η οποία πρόκειται να μεταφερθεί μεταξύ πελάτη και εξυπηρετητή (Client–Server). Η πληροφορία αυτή είναι τύπου JSON που προκύπτει μετά από JSON Web Signature (JWS) μετατρέποντας την σε πιστοποιημένη, προστατευμένη με Message Authentication Code (MAC) και/ή κρυπτογραφημένη μορφή. Στην Εικόνα 10 φαίνεται ένα παράδειγμα χρήσης JWT για έλεγχο και διατήρηση σύνδεσής χρήστη στην πλατφόρμα. Κατά την σύνδεση του χρήστη δημιουργείται ένα

JWT το οποίο επιστρέφεται στον Browser. Αυτός με τη σειρά του αποθηκεύει το JWT είτε ως cookie είτε στο local storage ώστε να μπορεί να γίνει χρήση τους στις επόμενες HTTP αιτήσεις προσθέτοντας το στο Authentication Header για τον έλεγχο και επαλήθευση του χρήστη .



Εικόνα 10: Σχήμα λειτουργίας JWT Authentication[19]

[19] https://www.toptal.com/java/rest-security-with-jwt-spring-security-and-java

3.2 Εργαλεία Ανάπτυξης πλατφόρμας

3.2.1 Node.js

Το node.js είναι μία πλατφόρμα ανάπτυξης λογισμικού σε περιβάλλον JavaScript που χρησιμοποιείται κυρίως για ανάπτυξη server-side κώδικα. Είναι βασισμένη στη μηχανή JavaScript V8 του Chrome γεγονός που εγγυάται την αποδοτικότητα της. Το node.js κάνει χρήση μόνο μιας διαδικασίας (single-process) χωρίς να δημιουργεί καινούρια νήματα σε κάθε αίτηση (single-threaded). Αυτό μπορεί να φαίνεται σαν σπατάλη όλων των άλλων πόρων ενός συστήματος, εφόσον πλέον όλοι οι υπολογιστές έχουν περισσότερους από έναν επεξεργαστές. Αυτό δεν ισχύει όμως γιατί με τη διαφορετική προσέγγιση που ακολουθεί το node.js είναι πολύ πιο αποδοτικό. Ουσιαστικά ακολουθεί ασύγχρονη επικοινωνία εισόδου/εξόδου (I/O) με αποτέλεσμα να μην γίνεται αποκλεισμός όλου του κώδικα όταν αναμένεται απάντηση από ένα γεγονός. Αυτό επιτυγχάνεται με τη χρήση callback functions. Όταν ο επεξεργαστής περιμένει ένα γεγονός (πχ διάβασμα αρχείου) ορίζεται μια τέτοια συνάρτηση που θα εκτελεστεί όταν ολοκληρωθεί αυτό το γεγονός, έτσι ο επεξεργαστής παραμένει ελεύθερος χωρίς να παγώνει. Όταν ολοκληρωθεί αυτό το γεγονός, καλείται η callback function η οποία θα κρατήσει τον επεξεργαστή κατειλημμένο μέχρι να ολοκληρωθεί η διαδικασία. Έτσι επιτυγχάνεται η ελαχιστοποίηση της αναμονής και η μικρή απαίτηση σε μνήμη, επιτρέποντας στο node.js να διαχειρίζεται χιλιάδες ταυτόχρονες συνδέσεις με ένα μόνο επεξεργαστή. Επιπλέον υπάρχει και τρόπος συγχρονισμού των γεγονότων που χρησιμοποιούνται σε λιγότερες περιπτώσεις.

Μεγάλο πλεονέκτημα του Node.js είναι ότι είναι υλοποιημένο σε γλώσσα JavaScript πράγμα που βοηθάει τους front-end developers, αφού δεν χρειάζεται η γνώση περισσότερων γλωσσών για την δημιουργία μιας ολοκληρωμένης εφαρμογής. Επίσης παρέχει μια τεράστια κοινότητα ανοικτού λογισμικού που το υποστηρίζει, παρέχοντας πάνω από 500 000 πακέτα ανοικτού κώδικα. Κάθε προγραμματιστής μπορεί ελεύθερα και εύκολα να έχει πρόσβαση σε αυτά τα πακέτα με τη χρήση του διαχειριστή πακέτων NPM.

Σημαντικότερα πακέτα που χρησιμοποιήθηκαν :

- **Express**: Ιδανικό για ανάπτυξη εφαρμογών ιστού και APIs, καθώς υλοποιεί πάρα πολλές βοηθητικές μεθόδους HTTP και middleware.
- **Mongoose**: Απαραίτητο για την σύνδεση Node.js με τη βάση δεδομένων MongoDB. Παρέχει μοντελοποίηση (model), σχηματισμό (schema) και μετατροπή δεδομένων (Casting) σε ιδανικό τύπο για χρήση της MongoDB.
- **Bcrypt**: Γίνεται χρήση για κατακερματισμό (hashing) κωδικών που προσφέρει μεγαλύτερη ασφάλεια αφού δεν μεταφέρνεται αυτούσιος ο κωδικός του χρήστη μεταξύ client-server κατά την ανταλλαγή αιτημάτων.
- **Jsonwebtoken**: Χρησιμοποιήθηκε για την επαλήθευση στοιχείων των χρηστών για τη διατήρηση της σύνδεσης τους στην πλατφόρμα.

Μεγάλο πλεονέκτημα του node.js είναι και η ευκολία επικοινωνίας με object databases όπως η MongoDB. Κατά την ανάκτηση ή προσθήκη δεδομένων προς τη βάση τα δεδομένα παρέχονται σε μορφή JSON και έτσι δεν χρειάζεται καμιά άλλη μετατροπή τύπου δεδομένων αφού το Node.js μπορεί να επεξεργαστεί με ευκολία τέτοιου τύπου δεδομένα.

3.2.2 MongoDB

Η MongoDB αποτελεί ένα δωρεάν, ανοιχτού κώδικα σύστημα διαχείρισης βάσεων δεδομένων. Ανήκει στις NoSQL βάσεις δεδομένων, δηλαδή δεν ακολουθεί την κλασική δομή βάσεων που βασίζονται σε πίνακες και σχεσιακές σχέσεις (relationships) μεταξύ τους, ούτε στη γλώσσα SQL για την διαχείριση των στοιχείων με ερωτήματα(queries), όψεις(views) και άλλες ενέργειες. Τα SQL ερωτήματα δεν είναι κατάλληλα για δεδομένα σε μορφή αντικειμένων λόγω του ότι η ανάκτηση και η αποθήκευση τους απαιτεί πολλά και σύνθετα queries σε βαθμό δύσκολο και περίπλοκο για τον προγραμματιστή. Επίσης για μεγάλες ποσότητες δεδομένων η χρήση queries δεν είναι τόσο αποδοτική και κρίνεται σχεδόν απαγορευτική για Big Data προβλήματα. Επιπλέον μια κεντρική διαφορά από τις σχεσιακές βάσεις δεδομένων είναι η έλλειψη ρητής δομής δεδομένων. Οι NoSQL βάσεις συμπεραίνουν την δομή από τα αποθηκευμένα δεδομένα ανάλογα με το πιο μοντέλο έχει χρησιμοποιηθεί.

Η αποθήκευση των δεδομένων γίνεται στη μορφή BSON (binary JSON). Αυτό διευκολύνει πάρα πολύ τη χρήση της με το Node.js αφού γίνεται χρήση JSON δεδομένων και δεν απαιτείται καθόλου μετατροπή σε μορφή ειδική για αποθήκευση ή ανάκτηση από τη βάση. Στην εφαρμογή που αναπτύχθηκε στα πλαίσια της διπλωματικής εργασίας επιλέχτηκε η MongoDB κυρίως λόγω δημοτικότητας και της άψογης δυνατότητας συνδυασμού της με το Node.js. Υπάρχουν πολλές βιβλιοθήκες ανεπτυγμένες σε περιβάλλον Node.js για χρήση MongoDB με πολλές ισχυρές δυνατότητες. Στην εφαρμογή μας επιλέχτηκε η Mongoose. Η βάση για την πλατφόρμα που έχει δημιουργηθεί αρχικά χρησιμοποιείται μόνο για την αποθήκευση των χρηστών.

Παραδείγματα χρήσης MongoDB δια μέσω Node.js

- Εγκατάσταση βιβλιοθήκης mongoose : "*npm install mongoose -save*"
- Εισαγωγή βιβλιοθήκης για χρήση με Node.js : "*var mongoose = require("mongoose")"*
- Σύνδεση με Βάση Δεδομένων :

```
mongoose.connect(
  "mongodb://127.0.0.1:27017/EvotDashDB",
  { useNewUrlParser: true },
  function(err) {
    if (err) {
      console.log("Not connected to the database: " + err);
    } else {
      console.log("Succesfully connected to MongoDB ");
    }
  }
);
mongoose.set("useCreateIndex", true);
```

- Δημιουργία Schema για Πίνακα δεδομένων και εξαγωγή μοντέλου :

```
var Schema = mongoose.Schema;
var UserSchema = new Schema({
  username: { type: String, required: true, unique: true },
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true },
  firstname: { type: String, required: true },
  lastname: { type: String, required: true },
  date_of_birth: { type: Date, required: true },
  gender: { type: String, required: true },
  admin: { type: Boolean, default: false },
  token: { type: String }
});
module.exports = mongoose.model("User", UserSchema);
```

- Παράδειγμα Επικοινωνίας με βάση
    - Επιστροφή όλων των user : "*let users = Users.find({}) ;*"
    - Προσθήκη User : *" let users = Users.create(userObject)"*

Η αμφίδρομη επικοινωνία μεταξύ server και βάσης δεδομένων γίνεται πολύ εύκολα με τη χρήση JSON Objects όπως φαίνεται από τα παραδείγματα. Η βάση λαμβάνει JSON Object και απαντά επίσης με τον ίδιο τρόπο. Η χρήση της βιβλιοθήκης mongoose καθιστά την επικοινωνία σε ζήτημα μιας γραμμής κώδικα πράγμα που δείχνει ξεκάθαρα τη διαφορά με τις SQL βάσεις δεδομένων και το λόγο επιλογής της με το Node.js.

3.2.3 React

Η React.js είναι ένα JavaScript Framework/Library το οποίο χρησιμοποιείται για την ανάπτυξη διαδραστικών web εφαρμογών. Υπάρχει και η έκδοση React Native η οποία σχετίζεται με την ανάπτυξη εφαρμογών για έξυπνες συσκευές (smartphones, tablets). Στην συγκεκριμένη εργασία χρησιμοποιήθηκε η React.js. Δημιουργήθηκε από τον Jordan Walke, μηχανικό λογισμικού του Facebook. Ήταν επηρεασμένος από το XHP, ένα στοιχείο HTML για την PHP. Χρησιμοποιήθηκε για πρώτη φορά στη ροή ενημερώσεων (newsfeed) του Facebook το 2011 και αργότερα στο Instagram το 2012. Έγινε κώδικας ανοιχτού λογισμικού (open-sourced) στο JSConf US το Μάιο του 2013[17].

Λόγω της δημοτικότητας και της ανάπτυξης της τα τελευταία χρόνια υπάρχουν διαθέσιμες πάρα πολλές βιβλιοθήκες ανοικτού κώδικα που διευκολύνουν τη δουλεία του προγραμματιστή. Για παράδειγμα η χρήση βιβλιοθηκών βοηθάει στην διαχείριση καταστάσεων (state), τη δρομολόγηση (routing) και την αλληλεπίδραση με API (http calls).

Σημαντικό χαρακτηριστικό της ReactJS είναι η χρήση ενός εικονικού Μοντέλου Αντικειμένου Εγγράφου (Virtual DOM). Αυτό γίνεται δημιουργώντας μια δομή δεδομένων εντός της μνήμης και στη συνέχεια υπολογίζονται οι προκύπτουσες διαφορές για αποτελεσματική ενημέρωση του DOM που εμφανίζεται στο πρόγραμμα περιήγησης. Αυτό επιτρέπει στον προγραμματιστή να γράφει κώδικα σαν να αποδίδεται ολόκληρη η σελίδα σε κάθε αλλαγή, ενώ οι ρουτίνες της React αποδίδουν μόνο τα υποστοιχεία του DOM που πραγματικά αλλάζουν.

Η ReactJS λειτουργεί με χρήση δηλωτικής λογικής μέσω αντικειμένων (components). Κάθε component αποτελεί ένα μικρό αυτοτελές κομμάτι της διεπαφής χρήστη υλοποιώντας κάποιες συγκεκριμένες λειτουργίες. Μπορεί να συμπεριλαμβάνεται σε άλλο component ή να συνδυάζεται με άλλα component για τη δημιουργία μιας ολοκληρωμένης σελίδας. Μεταξύ component μπορεί να δοθεί και ο χαρακτηρισμός parent- child αφού αποτελούν διαφορετικά στοιχεία συνδυασμένα για το τελικό αποτέλεσμα. Στη συνέχεια θα παρουσιαστούν τα κυρίως χαρακτηριστικά που παρουσιάζει αυτή η βιβλιοθήκη.

JavaScript XML (JSX):

Η JSX είναι μια επέκταση της JavaScript που χρησιμοποιείται για την περιγραφή των components που θα αποτελέσουν την εφαρμογή. Μοιάζει πολύ με την HTML και αυτό μπορεί μερικές φορές να συγχύζει. Η χρησιμοποίηση JSX για τη σύνταξη της React συνιστάται αλλά δεν είναι υποχρεωτική, μπορεί να γραφτεί και σαν απλή JavaScript. Στο παράδειγμα[20] πιο κάτω φαίνεται η δήλωση μίας μεταβλητής element η οποία δεν είναι ούτε string άλλα ούτε και HTML. Είναι ένα απλό JSX element.

```
const element = <h1>Hello, world!</h1>;
```

Στο επόμενο παράδειγμα φαίνεται ο συνδυασμός JavaScript μέσα στην JSX. Σε ένα JSX element μπορεί να προστεθεί οποιαδήποτε JavaScript έκφραση φτάνει αυτή να είναι μέσα σε αγκύλες ({curly braces}). Αυτό περιλαμβάνει αριθμητικές εντολές, συναρτήσεις, αντικείμενα[18].

```
const name = 'Josh Perez';
const element = <h1>Hello, {name}</h1>;
```

[20] https://reactjs.org/docs/introducing-jsx.html

```
function formatName(user) {
  return user.firstName + ' ' + user.lastName;
}

const user = {
  firstName: 'Harper',
  lastName: 'Perez'
};

const element = (
  <h1>
    Hello, {formatName(user)}!
  </h1>
);
```

Props

Κατά τη δήλωση ενός component δίνεται η δυνατότητα προσθήκης παραμέτρων σε αυτό. Αυτές οι παράμετροι είναι σαν read-only τιμές, οι οποίες είναι διαθέσιμες εντός του component και μπορούν να χρησιμοποιηθούν για τη συμπεριφορά του ή/και για την όψη του.

Η τιμή κάθε prop δεν μπορεί να αλλαχτεί μέσα στο component, αλλά ίδιο component μπορεί να δηλωθεί πολλές φορές με διαφορετικές τιμές props. Κυρίως χρησιμοποιείται για την μετάδοση δεδομένων από ένα parent-component σε ένα child-component στο οποίο θα ορίζει με κάποιο τρόπο τη συμπεριφορά του ή την όψη του. Αυτό ονομάζεται αλλιώς ως μονόδρομη πρόσδεση δεδομένων (one-way data binding).

Για παράδειγμα δίνεται ένα κομμάτι κώδικας από την εφαρμογή που έχει αναπτυχθεί που δείχνει ξεκάθαρα τη δήλωση props και θα εξηγηθεί η διαφορά στην έξοδο του component ανάλογα με αυτά. Στο παράδειγμα έχουμε ως component το YearSelect ένα component πού φτιάχτηκε για επιλογή χρονολογίας. Δέχεται σαν props τιμές για disabled, name, label, selectYear, minYear. Κάθε ένα από αυτά καθορίζουν την εμφάνιση και τη λειτουργικότητα του. Το minYear καθορίζει την μικρότερη χρονολογία που θα ξεκινούν οι επιλογές. Έτσι όπως φαίνεται στο ένα component η χρονολογία θα ξεκινά από το 1940, ενώ στο άλλο από την επιλογή που έχει γίνει στο πρώτο. Μέσα στα components η πρόσβαση στα props γίνεται εύκολα με χρήση απλά του props.minYear.

```
<YearSelect
  disabled={this.state.searching}
  name="startDate"
  label="From"
  selectYear={e => this.handleDateChange(e)}
  minYear={1940}
/>
```

```
<YearSelect
  disabled={this.state.searching}
  name="endDate"
  label="To"
  selectYear={e => this.handleDateChange(e)}
  minYear={this.state.startDate}
/>
```

State

Κάθε component έχει τη δική του κατάσταση (state) που συμβολίζεται ως ένα αντικείμενο. Αυτό το αντικείμενο καλείται state και ο προγραμματιστής το αρχικοποιεί εντός constructor. Τα state μοιάζουν με τα props μόνο που οι τιμές τους δηλώνονται αποκλειστικά μέσα στο component και μπορούν να αλλάξουν τιμές μέσω μίας μεθόδου που ονομάζεται setState(). Χρησιμοποιούνται για τον ίδιο σκοπό με τα props δηλαδή για την προσαρμογή του component.

Παράδειγμα αρχικοποίησης State :

```
this.state = {
  username: "",
  password: ""
};
```

Παράδειγμα αλλαγής τιμής :

```
handleChange(event) {
  const { value } = event.target;
  this.setState({
    username: value
  });
}
```

Στα ποιο πάνω κομμάτια κώδικα φαίνεται η χρήση του state αφού για παράδειγμα σε μια φόρμα σύνδεσης αρχικά η τιμή username, password θα είναι κενή περιμένοντας από το χρήστη να εισάγει τα στοιχεία του. Η συνάρτηση handleChange καλείται για την αλλαγή του state κατά την εισαγωγή τιμής.

Lifecycle Methods

Η React χρησιμοποιεί πολλές μεθόδους κύκλου ζωής (Lifecycle Methods) που διευκολύνουν τη δουλεία του προγραμματιστή αφού είναι πολύ χρήσιμες σε αρκετές περιπτώσεις. Στην εφαρμογή που αναπτύχτηκε χρησιμοποιήθηκαν μόνο η componentDidMount και render.

- **ComponentDidMount**: Η μέθοδος αυτή καλείται αυτόματα μόλις το component έχει δημιουργηθεί δηλαδή έχει συνδεθεί με ένα κόμβο στο DOM (Did Mount on the DOM). Αυτή η μέθοδος συχνά χρησιμοποιείται για φόρτωση δεδομένων από μια εξωτερική πηγή μέσω API. Καλείται μόνο μια φορά στον κύκλο ζωής ενός component ακριβώς μετά το πρώτο render.

- **Render**: Είναι η πιο σημαντική μέθοδος και είναι απαραίτητη σε κάθε component. Καλείται σε κάθε αλλαγή κατάστασης (state) ενός component. Αυτό έχει ως αποτέλεσμα την ενημέρωση της οθόνης του χρήστη στις πιο πρόσφατες αλλαγές και τη διατήρηση της ακεραιότητας δεδομένων προς αυτόν. Στην οθόνη δεν παρουσιάζεται σαν ανανέωση όλης της σελίδας. Αντιθέτως ο χρήστης βλέπει την αλλαγή αυτόματα και δυναμικά.

Κεφάλαιο 4 : Υλοποίηση – Αποτελέσματα

Σε αυτό το κεφάλαιο θα παρουσιαστεί ολόκληρη η εφαρμογή που έχει υλοποιηθεί με τα επιμέρους κομμάτια που την απαρτίζουν εξηγώντας παράλληλα τη λειτουργία τους. Σκοπός της ανάπτυξης αυτής της πλατφόρμας είναι η χρήση της με ευκολία σε φιλικό περιβάλλον προς το χρήστη. Για την χρήση αυτής της πλατφόρμας είναι αναγκαίο κάποιος χρήστης να είναι εγγεγραμμένος. Στα αρχικά στάδια ανάπτυξης αυτής της εργασίας έχουμε επιλέξει να μην μπορεί να εγγραφεί όποιος θέλει στην εφαρμογή αλλά να μπορεί μόνο να εγγραφεί δια μέσω ενός διαχειριστή. Δηλαδή μόνο ο διαχειριστής μπορεί να προσθέσει χρήστες για την πλατφόρμα. Οι χρήστες χωρίζονται σε 2 είδη, απλούς χρήστες και διαχειριστές.

Η πλατφόρμα που αναπτύχτηκε έχει πρόσβαση στα Ιατρικά βιβλιογραφικά κείμενα της PubMed. Η PubMed δίνει πρόσβαση στο API της για την ανάκτηση των κειμένων δωρεάν και επιλέχτηκε για αυτό το λόγο. Μετά την ανάκτηση των κειμένων βασισμένα στον τομέα αναζήτησης του χρήστη, γίνεται Text Mining από την πλευρά του εξυπηρετητή και τα αποτελέσματα παρουσιάζονται στην οθόνη του χρήστη για την περαιτέρω ανάλυση τους.

Η πρόσβαση στην πλατφόρμα από τον τοπικό υπολογιστή που βρίσκεται γίνεται με τη χρήση του URL http://localhost:3000/. Σε αρχικά στάδια ο ιστότοπος δεν έχει δημοσιευτεί στον παγκόσμιο ιστό με δικό του Domain Name, γεγονός που θα γίνει σε μελλοντική επέκταση. Επίσης χρήση του ιστότοπου μπορεί να γίνει από οποιονδήποτε υπολογιστή βρίσκεται στο ίδιο δίκτυο με χρήση του IP address του τοπικού υπολογιστή. Η ολοκληρωμένη εφαρμογή αποτελείται από τις εξής σελίδες και το URL τους:

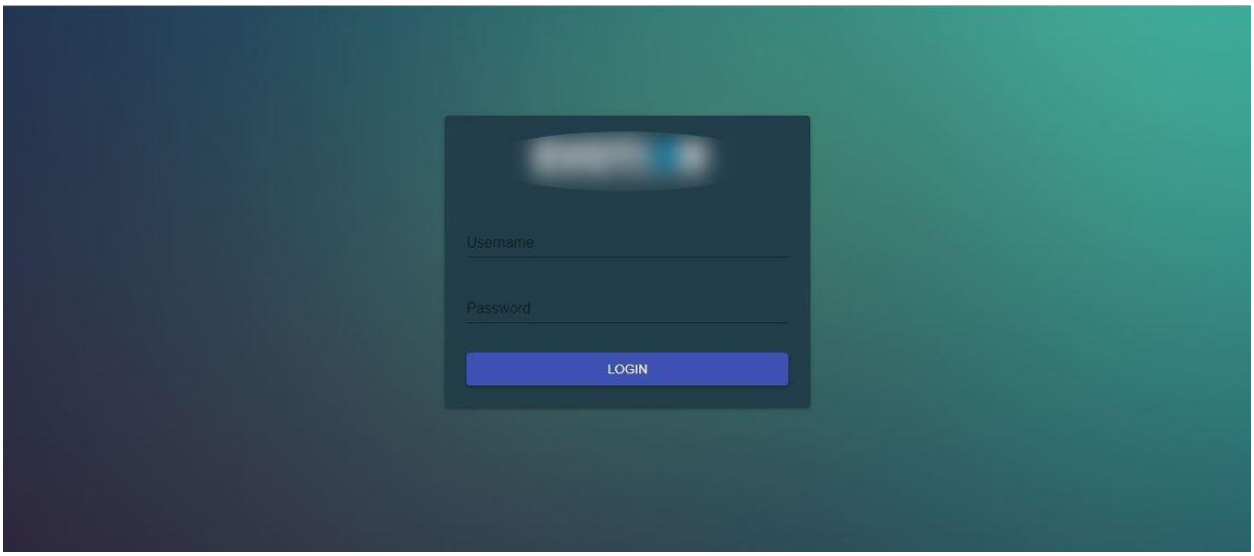
- Login Page : URL = http://localhost:3000/Login ή http://localhost:3000/
- Dashboard : URL = http://localhost:3000/Dashboard
- Reports : URL = http://localhost:3000/Reports
- Admin Page : URL = http://localhost:3000/AdminPage
- Account Page : URL = http://localhost:3000/Account

Ο χρήστης δεν υποχρεούται να γνωρίζει τη δρομολόγηση του ιστότοπου παρά μόνο το αρχικό URL της σελίδας. Η δρομολόγηση παρουσιάζεται εδώ απλά για λόγους επεξήγησης.

Στη συνέχεια θα γίνει εκτενής παρουσίαση της κάθε σελίδας με εισαγωγή στιγμιότυπων οθόνης, κατά τη χρήση της πλατφόρμας και επεξήγηση των επί μέρους component που την απαρτίζουν.

4.1 Login Page

Κατά τη μετάβαση στο URL της σελίδας στον Web Browser αν ο χρήστης δεν είναι ήδη συνδεδεμένος στην πλατφόρμα τότε θα μεταβεί στην αρχική σελίδα σύνδεσης ανεξαρτήτως από το URL που θα ακολουθήσει. Δηλαδή αν ο χρήστης προσπαθήσει να συνδεθεί απευθείας στο Dashboard και δεν είναι συνδεδεμένος τότε θα μεταφερθεί στη σελίδα σύνδεσης. Όπως έχουμε παρουσιάσει σε προηγούμενη ενότητα η διατήρηση σύνδεσης γίνεται με την βοήθεια JWT (JSON Web Token). Στην εικόνα 11 φαίνεται η αρχική οθόνη σύνδεσης.



Εικόνα 11: Αρχική σελίδα σύνδεσης (Login)

Σε αυτή την οθόνη ο χρήστης καλείται να εισάγει το Username και Password που αντιστοιχούν στον λογαριασμό που διαθέτει. Η σελίδα είναι κατάλληλα διαμορφωμένη ώστε να ανταποκρίνεται σε οποιοδήποτε λάθος προκύψει κατά την σύνδεση του χρήστη παρέχοντας του το κατάλληλο μήνυμα.

Πιθανά λάθη χρήστη :

- Κενό Username: “Please Provide Username”
- Κενό Password: “Please Provide Password”

- Λάθος Username: “User Not Found, Please Try Again”
- Λάθος Κωδικός: “Wrong Password , Please Try Again”

Στη συνέχεια όταν ο χρήστης συμπληρώσει τα στοιχεία πατώντας το κουμπί Login ή πατώντας το πλήκτρο Enter και επαληθευτούν τα στοιχεία του μεταφέρεται στη σελίδα Dashboard.

Η επαλήθευση των στοιχείων του γίνεται με τη χρήση του endpoint /login από το RESTful API. Η React στέλνει HTTP POST request στο REST API. Αν τα στοιχεία είναι σωστά δημιουργείται από τον εξυπηρετητή ένα JWT το οποίο αποθηκεύεται στον browser για τη διατήρηση της σύνδεσης του, και στη συνέχεια ο χρήστης μεταφέρεται στη σελίδα Dashboard.

4.2 Navigation Bar

Μέτα τη σύνδεση του χρήστη κάθε σελίδα της εφαρμογής που αναπτύχτηκε συνοδεύεται από την μπάρα πλοήγησης (Navigation Bar). Στην Εικόνα 12 φαίνεται το Navigation Bar και στη συνέχεια θα γίνει ανάλυση των επιμέρους κομματιών που το απαρτίζουν.

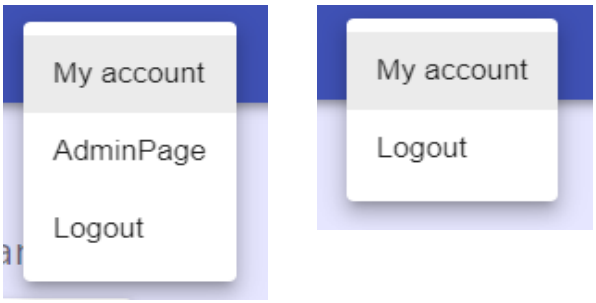
Εικόνα 12: Navigation Bar

Στο αριστερό κομμάτι του Navigation Bar φαίνεται το Logo της εφαρμογής το οποίο αν πατηθεί μεταφέρει το χρήστη στην κεντρική σελίδα. Στη δεξιά πλευρά φαίνονται τα αρχικά του χρήστη (Όνομα- Επίθετο) καθώς επίσης και το κατάλληλο Avatar σχετικά με το αν ο χρήστης είναι admin ή απλός χρήστης.

Εικόνα 13: Avatar Admin και User αντίστοιχα

Αν ο χρήστης πατήσει πάνω στο Avatar του Navigation Bar εμφανίζεται ένα μενού με επιλογές. Στο μενού δίνονται οι επιλογές My Account και Logout. Αν ο χρήστης είναι διαχειριστής παρουσιάζεται μια επιπλέον επιλογή για τη σελίδα διαχείρισης AdminPage. Κάθε επιλογή

μεταφέρει το χρήστη στην ανάλογη σελίδα. Με το Logout ο χρήστης αποσυνδέεται από την εφαρμογή διαγράφοντας το JWT από την μνήμη του Browser.



Εικόνα 14: Μενού επιλογών Navigation Bar.

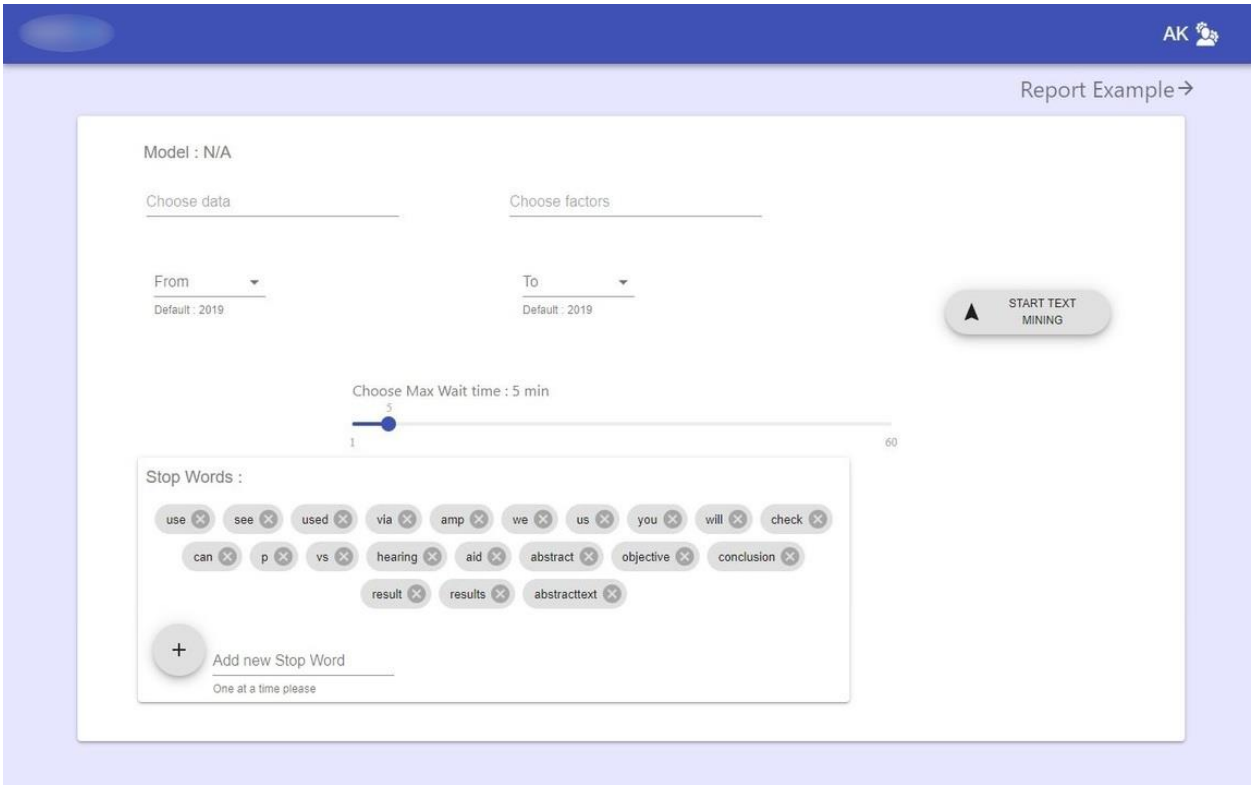
4.3 Dashboard

Αυτή είναι η βασική σελίδα της εφαρμογής . Σε αυτή τη σελίδα ο χρήστης καλείται να εισάγει τα δεδομένα και τις παραμέτρους που ενδιαφέρεται ώστε να προχωρήσει στην εξόρυξη δεδομένων. Συγκεκριμένα δίνεται στο χρήστη η δυνατότητα εισαγωγής δεδομένων (Data) και παραγόντων (Factors) ως προς το θέμα αλλά και το διάστημα χρονολογιών έκδοσης που τον ενδιαφέρουν για την εξόρυξη δεδομένων από τις βιβλιογραφικές πηγές. Έχουμε επιλέξει ως μέθοδο το μοντέλο ανάκτησης Boolean με χρήση λογικού AND για την πιο συγκεκριμένη ανάκτηση δεδομένων.

Η εξόρυξη δεδομένων από κείμενα που θα ακολουθήσει ασχολείται με τις πιο συχνά χρησιμοποιημένες λέξεις στα κείμενα που θα ανακτηθούν, βάσει του πεδίου ενδιαφέροντος του χρήστη. Για αυτό το λόγο δίνεται η δυνατότητα στο χρήστη να μπορεί να επιλέξει λέξεις οι οποίες δεν τον ενδιαφέρουν από τα άρθρα που θα ανακτηθούν. Αυτές οι λέξεις λέγονται Stop Words και δεν θα ληφθούν υπόψη στην παρουσίαση των αποτελεσμάτων.

Λόγω έλλειψης τεχνολογικού υλικού για την υποστήριξη τέτοιων πολύπλοκων εργασιών ή/και της μη πλήρους αποδοτικότητας του αλγορίθμου για την εξόρυξη δεδομένων από τεράστιες βάσεις κειμένων, τίθεται και το θέμα του χρόνου που ο χρήστης είναι διαθέσιμος να αφιερώσει για αυτό το σκοπό. Γι’ αυτό το λόγο ο χρήστης έχει τη δυνατότητα να επιλέξει πόσο χρόνο θα δώσει στη μηχανή για εξαγωγή αποτελεσμάτων. Όταν ο χρόνος αυτός τελειώσει η εξόρυξη δεδομένων σταματά αυτόματα. Επίσης ο χρήστης μπορεί να διακόψει την εξόρυξη δεδομένων οποιαδήποτε στιγμή θελήσει. Στη συνέχεια θα παρουσιαστεί η ολική σελίδα Dashboard και

ακολούθως θα γίνει επεξήγηση των επιμέρους κομματιών που την απαρτίζουν. Στην εικόνα 15 φαίνεται η ολική σελίδα Dashboard.



Εικόνα 15: Σελίδα Dashboard

4.3.1 Εισαγωγή Δεδομένων και Παραγόντων (Data-Factors)

Ο χρήστης καλείται να εισάγει δεδομένα και παράγοντες για να ορίσει το θέμα πάνω στο οποίο θα γίνει η ανάκτηση πηγών από την βάση δεδομένων. Κρίθηκε χρήσιμο να υπάρχει ένα είδος αυτοσυμπλήρωσης ή πρότασης λέξεων/όρων ιατρικού περιεχομένου χωρίς όμως αυτό να εμποδίζει τον χρήστη από το να εισάγει οποιουσδήποτε άλλους όρους επιθυμεί. Αρχικά έγινε χρήση ενός component από την βιβλιοθήκη material-ui[21] το οποίο ονομάζεται AutoComplete. Το AutoComplete είναι ένα κλασσικό component εισαγωγής κειμένου (text-box) το οποίο έχει σαν επέκταση τη δυνατότητα εμπλουτισμού του από προτεινόμενες λέξεις/όρους.

[21] https://material-ui.com/

Οι προτεινόμενες λέξεις και όροι που δίνονται στα components περιέχονται σε ένα εξωτερικό αρχείο χωρισμένες σε μοντέλα με τη μορφή αντικειμένου. Κάθε αντικείμενο περιέχει ως στοιχεία το όνομα του μοντέλου, ένα πίνακα από δεδομένα, και ένα πίνακα από παράγοντες που αντιστοιχούν σε αυτά τα δεδομένα. Παράδειγμα ενός μοντέλου φαίνεται στην Εικόνα 16.

```
{
  modelName: 'PHPDM4',
  data: ['MOCA', 'HUI3', 'Hearing aid usage'],
  factors: [
    'Diabetes',
    'Smoking',
    'Vascular disease',
    'Dementia',
    'Occupation',
    'Education',
    'Age',
    'Reading Span',
    'Verbal reaction time',
    'Mood monitoring',
    'Reverse digit Recall',
    'HADS',
    'Social Engagement',
  ],
},
```

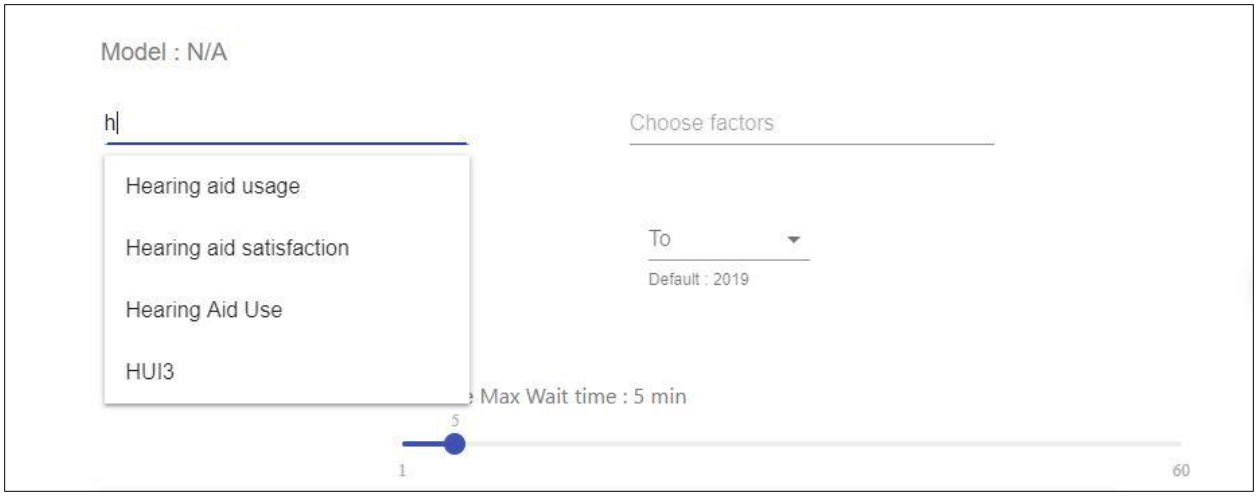
Εικόνα 16: Παράδειγμα μοντέλου για φιλτράρισμα προτεινόμενων λέξεων

Στην εφαρμογή μας υπάρχουν δύο Component εισαγωγής κειμένου. Ένα για δεδομένα (data) και ένα για τους παράγοντες (factors). Αρχικά το Component για εισαγωγή factors είναι απενεργοποιημένο, δηλαδή δεν μπορεί ο χρήστης να εισάγει factors πριν την εισαγωγή data. Όταν ο χρήστης ξεκινήσει να πληκτρολογεί στο πεδίο εισαγωγής κειμένου για data αυτόματα κάτω από το πεδίο παρουσιάζονται προτεινόμενες/συμπληρωμένες πιθανές λέξεις/όροι σε μορφή λίστας. Ο χρήστης μπορεί να επιλέξει από τη λίστα ή να εισάγει ότι άλλο θέλει. Η εισαγωγή τιμών και στα δύο πεδία είναι υποχρεωτική και έτσι αν ο χρήστης δεν συμπληρώσει κάποιο από τα δύο δεν μπορεί να συνεχίσει στην εξόρυξη δεδομένων και του παρουσιάζεται ένα μήνυμα λάθους.

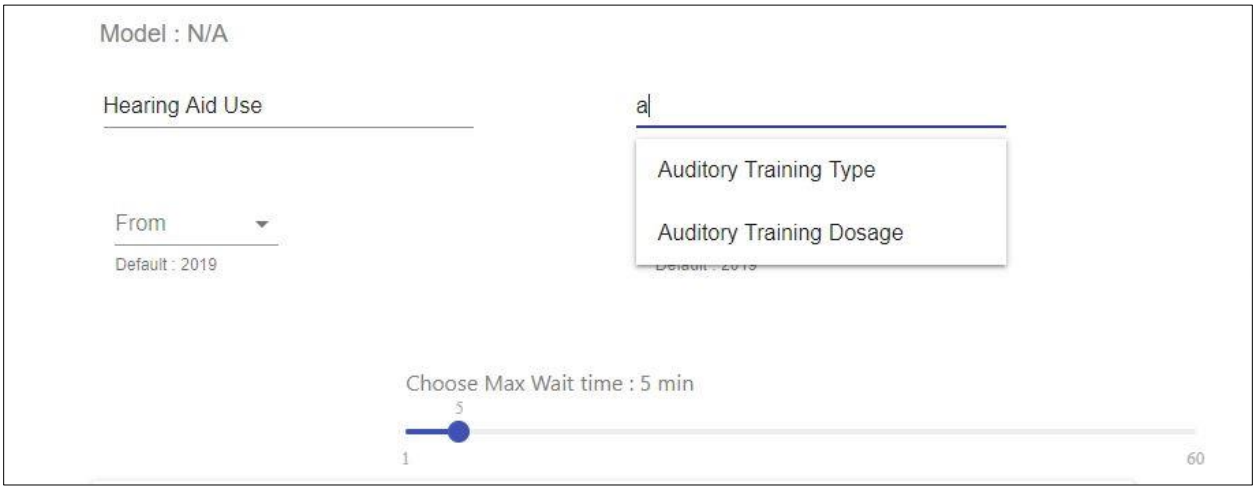
Με τη συμπλήρωση του πεδίου για data το πεδίο για τα factors έχει ενεργοποιηθεί. Οι προτεινόμενες επιλογές κατά την πληκτρολόγηση από τον χρήστη στο πεδίο αυτό είναι φιλτραρισμένες ούτως ώστε να αντιστοιχούν σε ότι έχει εισάγει για data. Αυτό βοηθάει το χρήστη στο να έχει μια καλύτερη σχέση μεταξύ data και factors. Αν ο χρήστης δεν έχει επιλέξει

κάποιο όρο από τις προτεινόμενες επιλογές στην επιλογή data δεν θα υπάρχουν αντίστοιχες προτάσεις για factors μιας και δεν θα αντιστοιχούν σε κάποιο καθορισμένο μοντέλο. Αν όμως ο χρήστης έχει επιλέξει και στις δυο περιπτώσεις λέξεις/όρους από τις προτεινόμενες, θα μπορεί να δει στο πάνω μέρος της οθόνης του σε ποιο μοντέλο αντιστοιχούν. Στην **εικόνα 17** φαίνεται η πρόταση όρων κατά την πληκτρολόγηση από τον χρήστη στο πεδίο data. Ο χρήστης προαιρετικά μπορεί να επιλέξει έναν όρο από αυτή τη λίστα.

Στην **εικόνα 18** φαίνεται η πρόταση όρων κατά την πληκτρολόγηση από τον χρήστη στο πεδίο factors. Στο παράδειγμα ο χρήστης έχει ήδη επιλέξει έναν όρο που αντιστοιχεί σε κάποιο μοντέλο επομένως υπάρχουν προτεινόμενοι όροι για factors. Στην **εικόνα 19** φαίνεται η εμφάνιση του μοντέλου μετά την επιλογή όρων.



Εικόνα 17: Πρόταση όρων κατά την πληκτρολόγηση από το χρήστη(data)



Εικόνα 18: Πρόταση φιλτραρισμένων όρων ανάλογα με επιλεγμένο data(factors)

Εικόνα 19: Παρουσίαση Μοντέλου που αντιστοιχούν data και factors

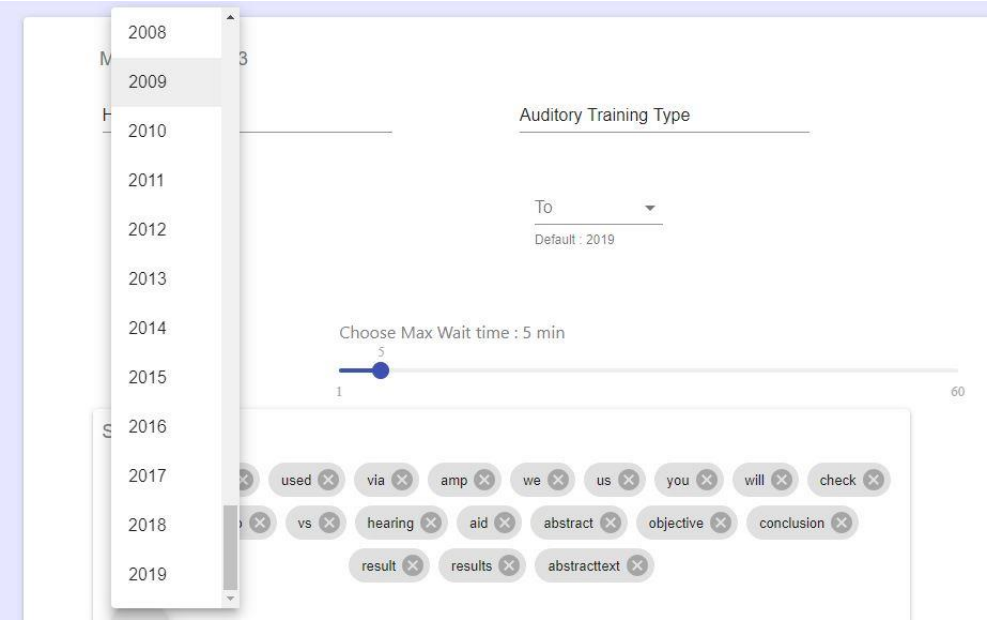
Η χρήση μοντέλων και προτάσεων από το σύστημα είναι προαιρετική και γίνεται με σκοπό να δώσει κάποιο συγκεκριμένο χαρακτήρα στην εφαρμογή. Σε αυτή την περίπτωση τα μοντέλα που αναπτύχθηκαν έχουν σαν θεματολογία τους την ακοή. Μοντέλα με διαφορετικό περιεχόμενο θεωρούνται ως μελλοντικές επεκτάσεις.

4.3.2 Επιλογή Χρονολογικού διαστήματος ενδιαφέροντος

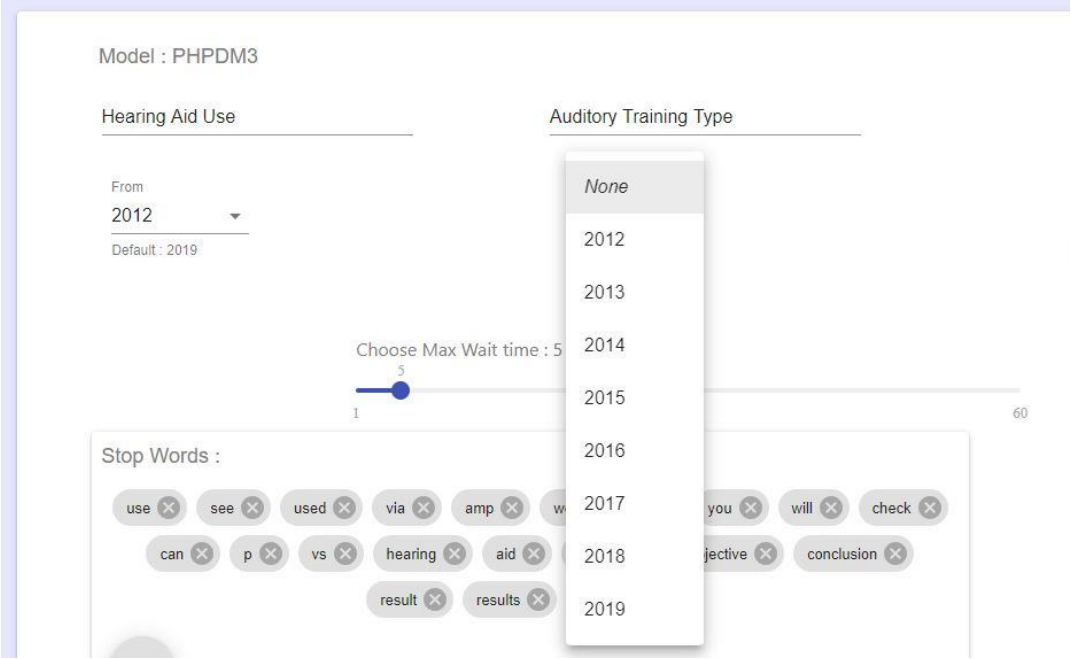
Εκτός από δεδομένα και παράγοντες που ενδιαφέρουν το χρήστη ως προς την ανάκτηση δεδομένων για την εξόρυξη που θα ακολουθήσει, ο χρήστης έχει τη δυνατότητα να επιλέξει και το χρονολογικό διάστημα που τον ενδιαφέρει. Για παράδειγμα μπορεί να έχει προτίμηση να γίνει ανάκτηση δεδομένων μόνο για κείμενα που έχουν δημοσιευτεί από το 2010 μέχρι σήμερα, ή να έχει ένα ιδιαίτερο ενδιαφέρον για κείμενα της δεκαετίας του 1990.

Έχει γίνει χρήση του Component Select από την βιβλιοθήκη material-ui. Το Component έχει επεκταθεί με επιλογές από χρονολογίες από το 1940 μέχρι την χρονολογία που διανύουμε.

Αν ο χρήστης δεν επιλέξει κάποιες συγκεκριμένες χρονολογίες τότε είναι προκαθορισμένη η χρόνια την οποία διανύουμε. Επίσης για την αποφυγή οποιουδήποτε λάθους σχετικά με τις χρονολογίες (από - μέχρι) έχει γίνει προγραμματισμός ώστε η χρονολογία που θα επιλεχθεί στο πεδίο “μέχρι” να είναι πάντα μεγαλύτερη από την χρονολογία που θα επιλεχθεί στο πεδίο “από”. Στην εικόνα 20 και 21 φαίνονται στιγμιότυπα οθόνης για αυτές τις επιλογές.



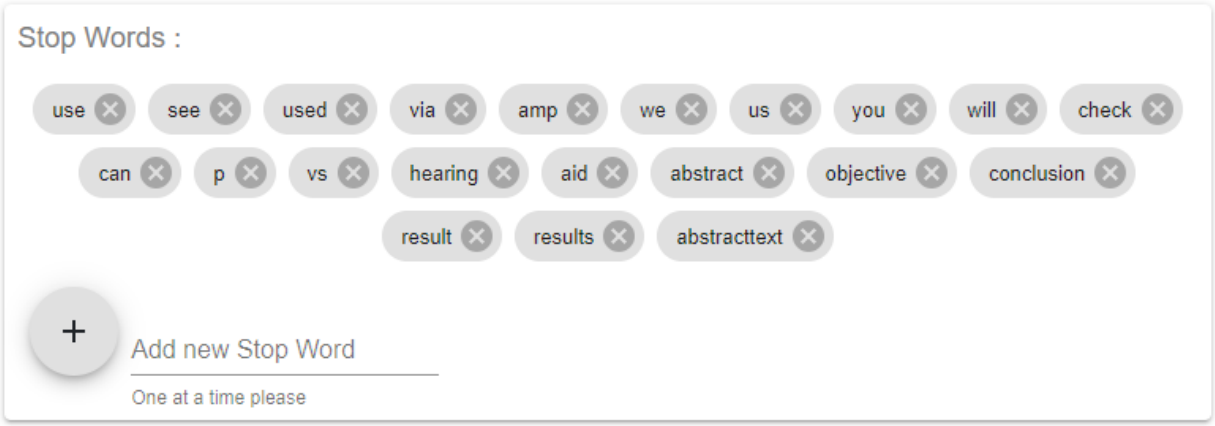
Εικόνα 20: Παράδειγμα επιλογής χρονολογίας "Από"



Εικόνα 21: Η επιλογή χρονολογίας "Μέχρι" είναι φιλτραρισμένη από την προηγούμενη επιλογή.

4.3.3 Επιλογή Stop Words

Στην οθόνη του χρήστη σε ειδικό πλαίσιο παρουσιάζονται μερικές λέξεις στη μορφή που φαίνεται στην Εικόνα 22. Αυτές οι λέξεις καλούνται Stop Words και είναι λέξεις οι οποίες δεν υπάρχει λόγος να αναλυθούν και να ληφθούν υπόψη κατά την εξόρυξη δεδομένων από κείμενα. Κατά την εξόρυξη δεδομένων πραγματοποιείται μια ανάλυση με τις πιο χρησιμοποιημένες λέξεις και γίνεται ομαδοποίηση τους σε συστάδες για περαιτέρω μελέτη τους. Έχουν επιλεχτεί μερικές προκαθορισμένες λέξεις που κατά τη κρίση μου δεν ενδιαφέρουν κάποιο χρήστη για περαιτέρω μελέτη τους.



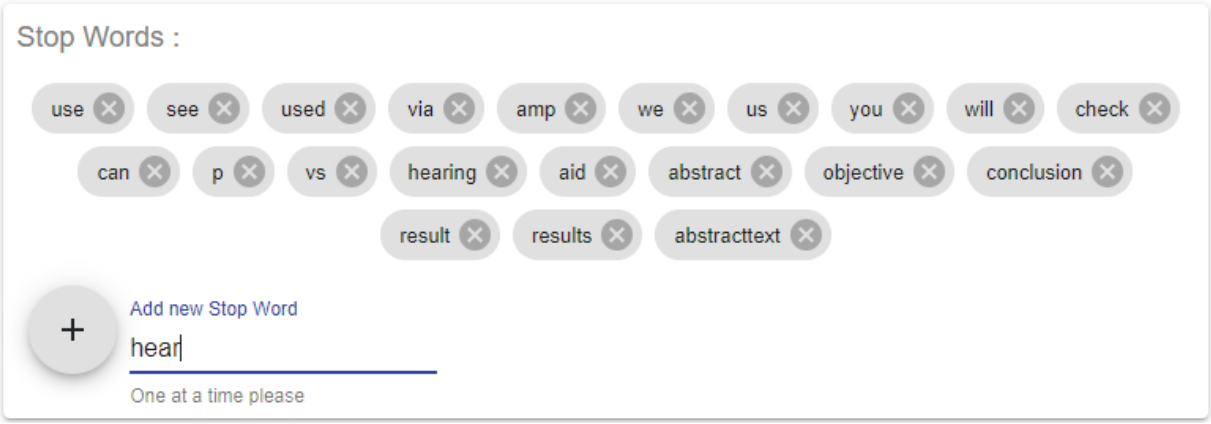
Εικόνα 22: Πλαίσιο επιλογής Stop Word.

Έχει γίνει χρήση του Component Chip από τη βιβλιοθήκη material-ui για την κάθε λέξη συνοδευόμενη από το κουμπί διαγράφης. Στην οθόνη του χρήστη παρουσιάζεται μια λίστα από Chips. Ο χρήστης μπορεί να αφαιρέσει όποια λέξη από τις υπάρχουσες επιθυμεί απλά πατώντας στο κουμπί "X" που βρίσκεται δίπλα στη λέξη που επιθυμεί να διαγράψει. Πατώντας πάνω στο κουμπί "X" η λέξη διαγράφεται απευθείας από τη λίστα και εξαφανίζεται από το πλαίσιο.

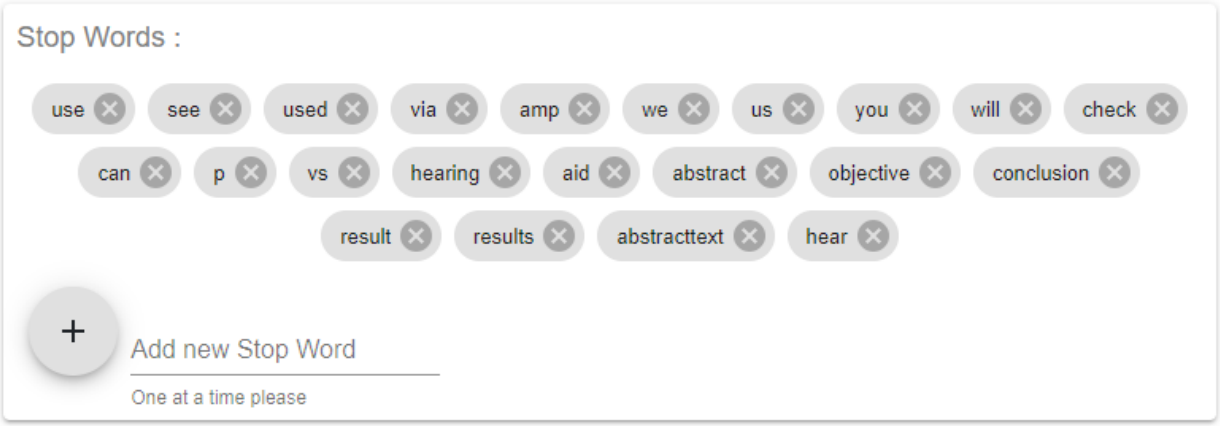
Εικόνα 23: Διαγραφή Stop Word

Εικόνα 24: Η λέξη Aid δεν υπάρχει πλέον στη λίστα Stop Words.

Επιπλέον, ο χρήστης μπορεί να προσθέσει όσες καινούριες λέξεις θέλει στη λίστα απλά εισάγοντας τις στο πλαίσιο κειμένου (text-box), μια κάθε φορά, που βρίσκεται στο κάτω μέρος του πλαισίου και στη συνέχεια πατώντας στο κουμπί "+". Η νέα λίστα φαίνεται δυναμικά στο πλαίσιο αμέσως μετά το πάτημα του κουμπιού "+", ώστε ο χρήστης να γνωρίζει πως έχει γίνει σωστά η προσθήκη της και να μπορεί να την διαγράψει αμέσως αν έχει γίνει τυχόν ορθογραφικό λάθος.



Εικόνα 25: Ο χρήστης πληκτρολογεί νέα λέξη για εισαγωγή στη λίστα Stop Words.



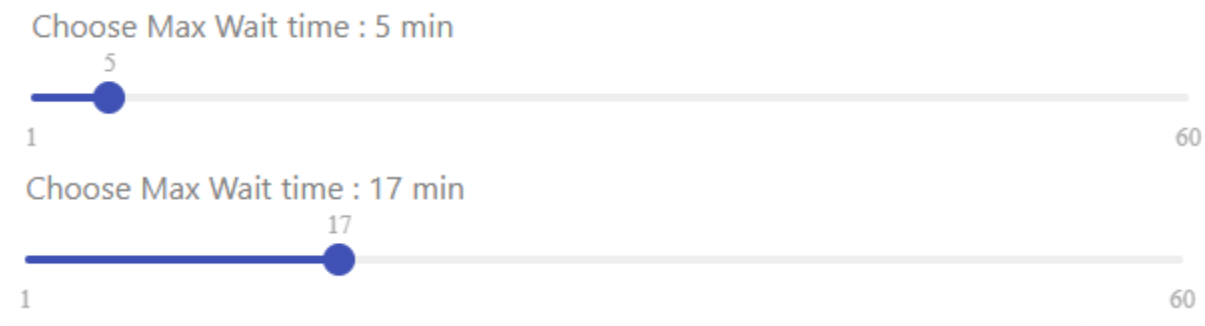
Εικόνα 26: Το νέο Stop Word έχει εμφανιστεί στην οθόνη.

4.3.4 Ορισμός μέγιστου χρόνου αναμονής

Όπως έχει αναφερθεί στην εισαγωγή της σελίδας Dashboard ο χρήστης έχει τη δυνατότητα να ορίσει στην πλατφόρμα το χρόνο που επιθυμεί να διαθέσει για την διαδικασία εξόρυξης δεδομένων. Αυτός ο χρόνος δεν ορίζει ότι η ολική διαδικασία θα διαρκέσει ακριβώς το συγκεκριμένο χρονικό διάστημα, απλά ορίζει ένα μέγιστο όριο αναμονής. Αν δηλαδή η διαδικασία ολοκληρωθεί θα παρουσιαστούν τα αποτελέσματα άμεσα.

Ο ορισμός χρόνου αναμονής γίνεται με της χρήση μιας μπάρας με τιμές από 1 λεπτό έως 60 λεπτά. Σε αυτή την περίπτωση έγινε χρήση του Component InputRange από την εξωτερική βιβλιοθήκη react-input-range η εγκατάσταση του οποίου έγινε με χρήση του διαχειριστή πακέτων npm.

Έχουν προκαθοριστεί ως αρχική τιμή τα 5 λεπτά ενδεικτικά με τον χρήστη να μπορεί να επιλέξει αν θέλει κάποια άλλη τιμή. Κατά την εκκίνηση της διαδικασίας εξόρυξης δεδομένων η μπάρα εξαφανίζεται και παρουσιάζεται ένας μετρητής αντίστροφης μέτρησης έτσι ώστε ο χρήστης να έχει αντίληψη του χρόνου που έχει περάσει.



Εικόνα 27: Παράδειγμα ορισμού μέγιστου χρόνου αναμονής

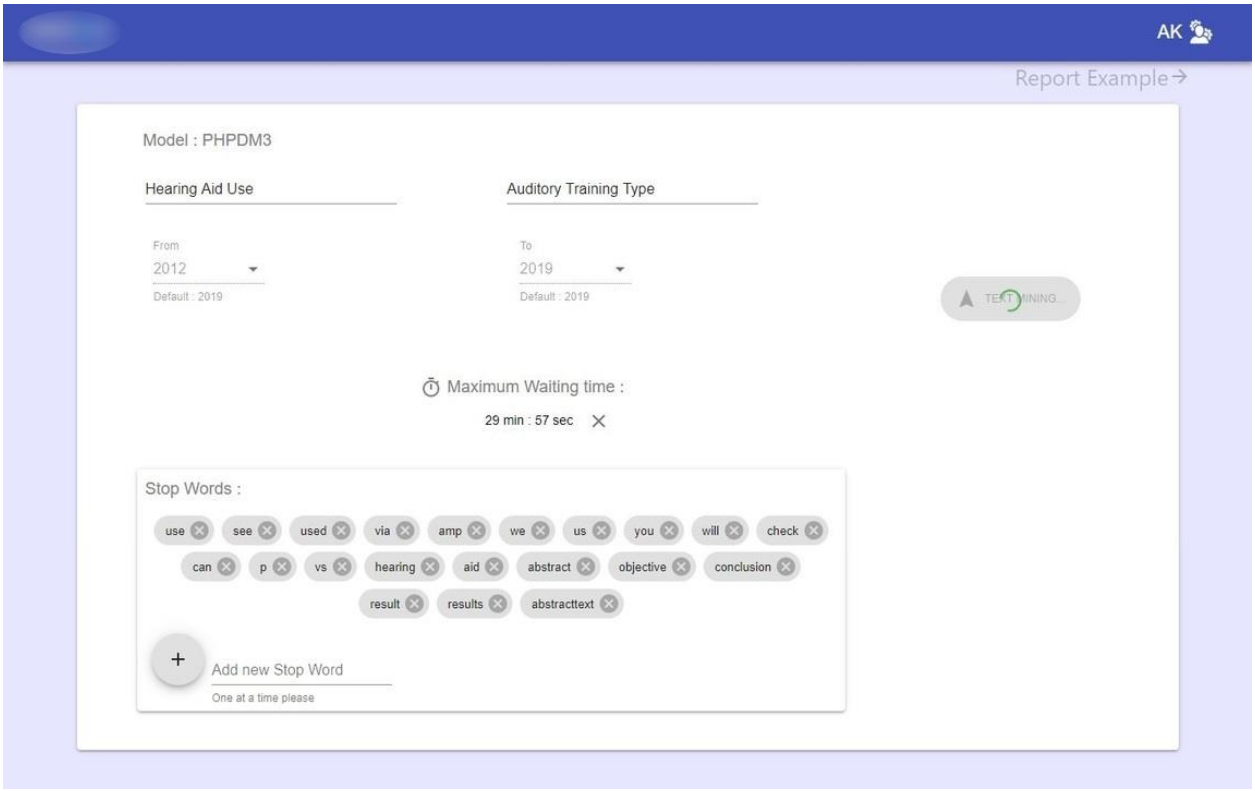
4.3.5 Εκκίνηση Εξόρυξης Δεδομένων από κείμενα

Μετά την συμπλήρωση όλων των παραγόντων, ο χρήστης για να ξεκινήσει τη διαδικασία εξόρυξης δεδομένων από κείμενα πρέπει να πατήσει στο κουμπί "Start Text Mining" που βρίσκεται στα δεξιά της σελίδας. Αν ο χρήστης δεν έχει εισάγει Δεδομένα ή Παράγοντες στα πεδία Data και Factors αντίστοιχα, η πλατφόρμα δεν του επιτρέπει να ξεκινήσει την εξόρυξη δεδομένων και παρουσιάζει ανάλογο μήνυμα.

Εικόνα 28: Data και Factors δεν μπορούν να είναι κενά

Αν ο χρήστης δεν επιλέξει διαφορετικές χρονολογίες ενδιαφέροντος τότε θα ανακτηθούν μόνο κείμενα που έχουν εκδοθεί την ίδια χρόνια. Η προσθήκη και η διαγραφή Stop Words είναι προαιρετική και υπόκεινται στην εμπειρία και τις ανάγκες του χρήστη.

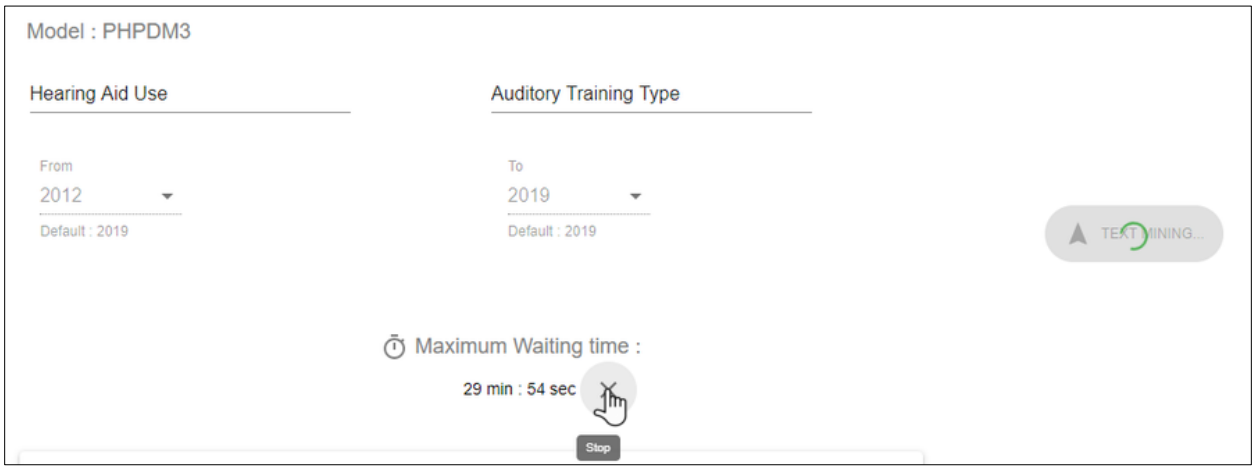
Στην **Εικόνα 29** φαίνεται η οθόνη που βλέπει ο χρήστης κατά την διαδικασία εξόρυξης δεδομένων. Σε αυτό το παράδειγμα ο χρήστης έχει ζητήσει ανάκτηση κειμένων για “Hearing Aid Use” AND “Auditory Training Type” με χρονολογίες έκδοσης από το 2012 μέχρι το 2019 με μέγιστο χρόνο αναμονής 30 λεπτά.



Εικόνα 29: Εκκίνηση Διαδικασίας Εξόρυξης Δεδομένων από κείμενα

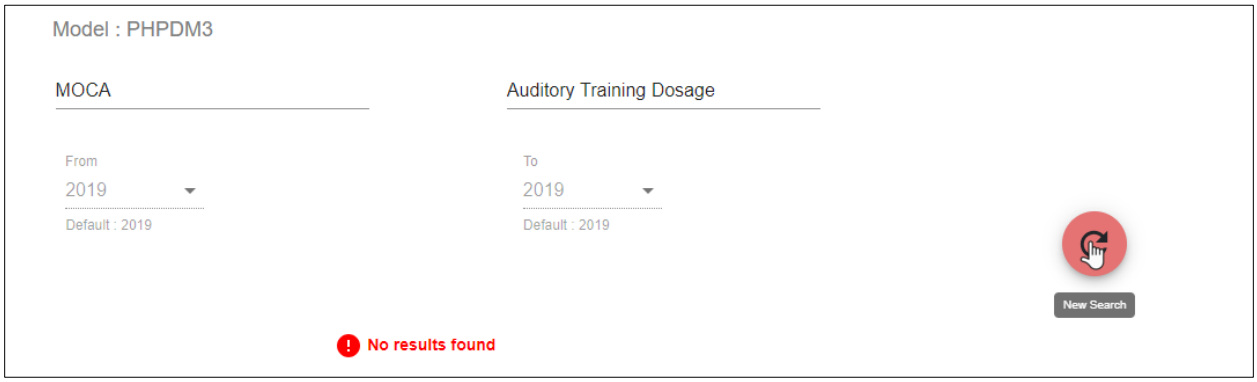
Όταν ο χρήστης πατήσει το κουμπί Start Text Mining για να ξεκινήσει η διαδικασία αλλάζει η κατάσταση και το κείμενο στο εσωτερικό του κουμπιού σε disabled και “Text Mining” αντίστοιχα. Στο εσωτερικό του κουμπιού έχει τοποθετηθεί ένα spinner για να γνωρίζει ο χρήστης ότι η διεργασία πραγματοποιείται. Όλα τα components έχουν αλλάξει κατάσταση σε disabled ώστε να μην δέχονται αλλαγές μετά την εκκίνηση της διαδικασίας. Στο κέντρο της οθόνης στη θέση της μπάρας εισαγωγής μέγιστου χρόνου αναμονής υπάρχει ένας αντίστροφος μετρητής για τον υπολειπόμενο χρόνο μέγιστης αναμονής και δίπλα από το μετρητή υπάρχει το πλήκτρο

ακύρωσης της διαδικασίας. Ο χρήστης έχει τη δυνατότητα ακύρωσης της διαδικασίας οποιαδήποτε στιγμή.



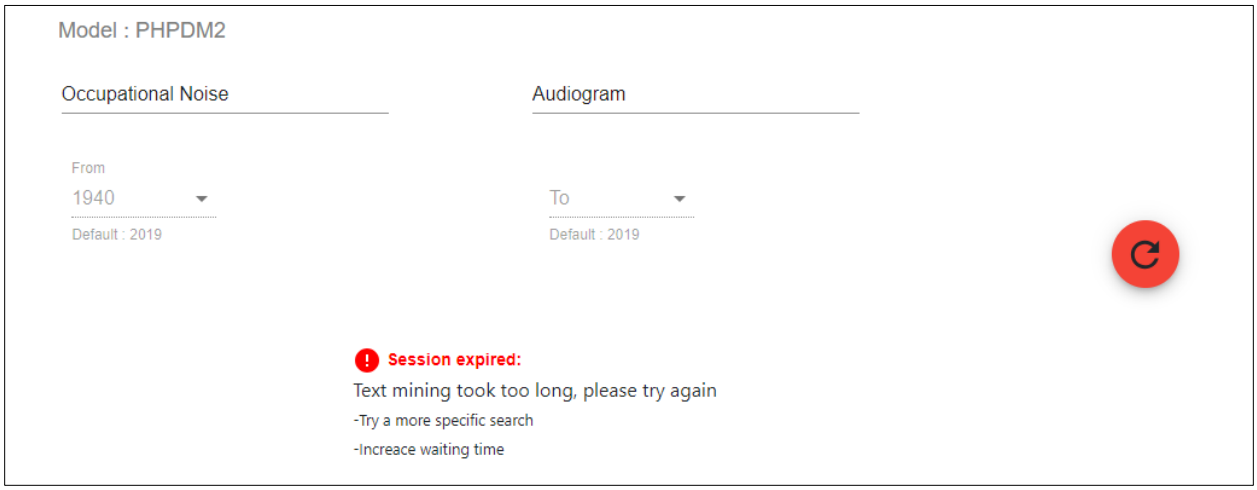
Εικόνα 30: Δυνατότητα ακύρωσης Διαδικασίας οποιαδήποτε στιγμή.

Υπάρχει περίπτωση η ανάκτηση κειμένων να μην επιστρέψει κανένα αποτέλεσμα είτε λόγω έλλειψης πληροφορίας για το θέμα που έχει αναζητηθεί, είτε λόγω λανθασμένων ή μη συνδυαζόμενων όρων ανάκτησης. Όταν δεν υπάρχουν αποτελέσματα ο χρήστης πρέπει να ειδοποιείται με το κατάλληλο μήνυμα και να του δίνεται η δυνατότητα για νέα αναζήτηση. Στην **Εικόνα 31** φαίνεται η οθόνη του χρήστη όταν δεν βρεθούν αποτελέσματα. Ο χρήστης για να επανεισάγει καινούρια δεδομένα ή να τα τροποποιήσει πρέπει να πατήσει πάνω στο κόκκινο κουμπί Refresh που εμφανίζεται στην δεξιά πλευρά της σελίδας.



Εικόνα 31: Η ανάκτηση δεδομένων δεν βρήκε αποτελέσματα, νέα αναζήτηση.

Αν η διαδικασία εξόρυξης δεδομένων διαρκέσει πολλή ώρα με αποτέλεσμα ο μέγιστος χρόνος αναμονής που έθεσε ο χρήστης εξαντληθεί, τότε η διαδικασία θα διακοπεί αυτόματα και στην οθόνη του χρήστη θα παρουσιαστεί το κατάλληλο μήνυμα.



Εικόνα 32: Ο μέγιστος χρόνος αναμονής έχει εξαντληθεί.

Όταν η διαδικασία ολοκληρωθεί με επιτυχία ο χρήστης μεταφέρεται αυτόματα στη σελίδα Reports η οποία θα αναπτυχθεί στην επόμενη παράγραφο.

Ο χρήστης έχει τη δυνατότητα προβολής της σελίδας Report σε οποιαδήποτε στιγμή πατώντας απλά στο εικονιζόμενο πάνω δεξιά βέλος το οποίο αναγράφει Report Example.

Εικόνα 33: Προβολή σελίδας Αναφοράς

4.4 Report Page

Σε αυτή τη σελίδα παρουσιάζονται αναλυτικά τα αποτελέσματα της εξόρυξης δεδομένων βάσει των επιλογών του χρήστη. Ο χρήστης μεταφέρεται αυτόματα από τη σελίδα Dashboard στη σελίδα Report μετά το πέρας της διαδικασίας εξόρυξης δεδομένων. Αν ο χρήστης επιλέξει να δει ένα παράδειγμα αποτελεσμάτων χωρίς να εισάγει ο ίδιος πληροφορίες για εξόρυξη δεδομένων τότε σε αυτή τη σελίδα θα εμφανιστεί ένα παράδειγμα αναφοράς από κάποια προηγούμενη εξόρυξη δεδομένων που πραγματοποιήθηκε.

Στη σελίδα παρουσιάζονται οι παράμετροι που λήφθηκαν υπόψη για την ανάκτηση κείμενων από τα οποία στη συνέχεια έγινε η εξόρυξη δεδομένων. Οι παράμετροι αυτοί είναι τα δεδομένα (Data) και οι παράγοντες (factors), χρονολογία από, χρονολογία μέχρι, συνοδευμένα με τον αριθμό των κείμενων που έχουν ανακτηθεί και αναλυθεί.

Στη συνέχεια παρουσιάζονται τα αποτελέσματα από την εξόρυξη λέξεων από τα κείμενα. Η εξόρυξη λέξεων από τα κείμενα επιστρέφει σε μορφή εικόνων τρία διαφορετικά είδη ανάλυσης τους. Το πρώτο είδος ανάλυσης παρουσιάζεται στη μορφή γραφικής παράστασης και αποτελείται από τις 50 πιο χρησιμοποιημένες λέξεις. Η ανάλυση αυτή καλείται Word Frequency. Το δεύτερο είδος ανάλυσης αποτελείται από μία γραφική παράσταση ομαδοποίησης των λέξεων σε συστάδες. Η ανάλυση αυτή καλείται Clustering. Τελευταίο είδος ανάλυσης αποτελεσμάτων είναι μια μορφή συννεφόλεξου η οποία καλείται WordCloud. Ο χρήστης έχει τη δυνατότητα προβολής και αποθήκευσης των αποτελεσμάτων σε μορφή PDF για την καλύτερη αξιοποίηση τους.

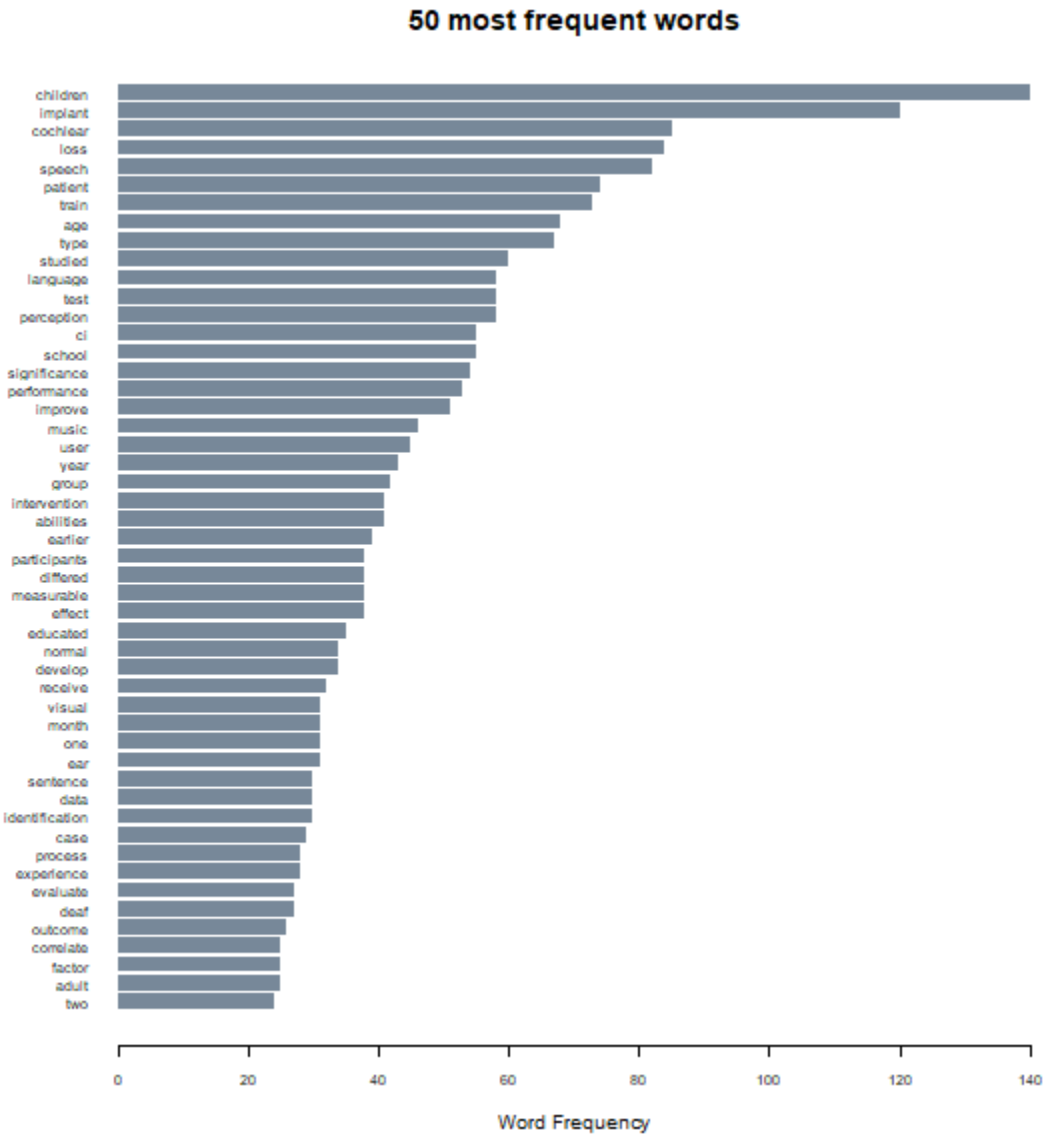
Στο τέλος της σελίδας παρουσιάζονται όλα τα κείμενα τα οποία ανακτήθηκαν και έχει γίνει εξόρυξη δεδομένων από αυτά σε μορφή πίνακα. Ο χρήστης μπορεί να έχει πρόσβαση στα άρθρα αυτά για την περαιτέρω μελέτη τους.

Ακολουθεί στιγμιότυπο οθόνης από την αναφορά για την εξόρυξη δεδομένων σε πεδίο αναζήτησης **“**Hearing Aid Use” AND “Auditory Training Type” από το έτος 2000 μέχρι το έτος 2019. Για το πεδίο ενδιαφέροντος που έχει εισάγει ο χρήστης η μέθοδος ανάκτησης κειμένων έχει επιστρέψει 47 κείμενα στα οποία έχει γίνει εξόρυξη λέξεων.

Εικόνα 34: Αναφορά εξόρυξης Δεδομένων.

Ο χρήστης πατώντας στο πεδίο “OPEN IN NEW TAB” έχει τη δυνατότητα να ανοίξει την κάθε εικόνα σε νέα σελίδα ώστε να μπορέσει να δει αναλυτικά τα αποτελέσματα. Στη συνέχεια θα παρουσιαστούν τα αποτελέσματα της κάθε ανάλυσης σε καλύτερη ποιότητα.

4.4.1 Word Frequency

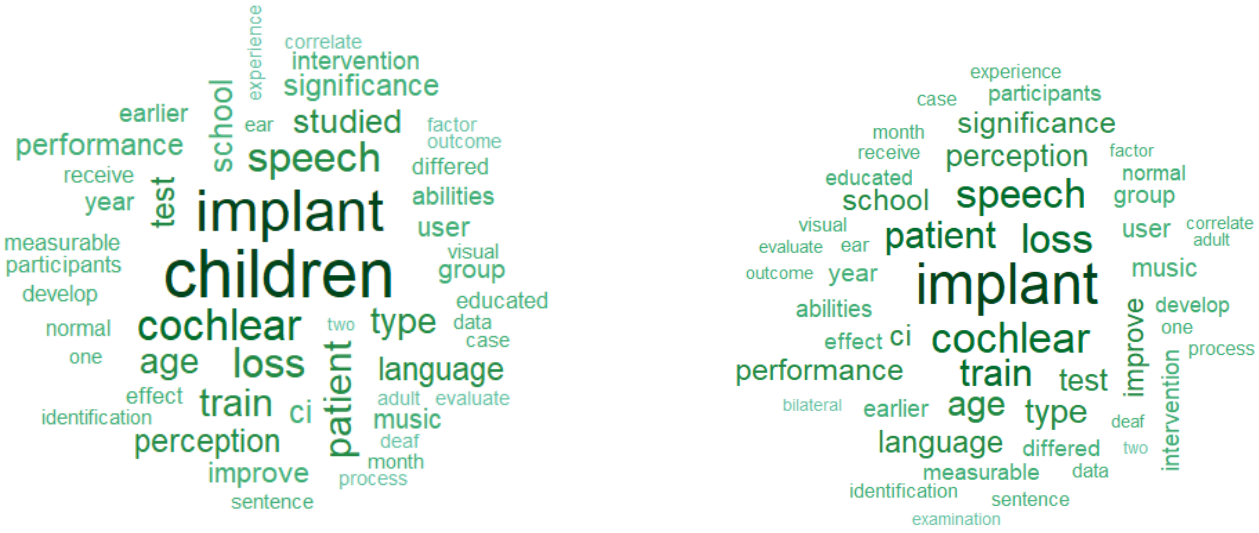


Εικόνα 35: Γραφική Παράσταση 50 πιο συχνά χρησιμοποιημένων λέξεων.

Από την γραφική παράσταση συμπεραίνεται ότι στα κείμενα που ανακτήθηκαν σχετικά με **“**Hearing Aid Use” AND “Auditory Training Type” από το έτος 2000 μέχρι το έτος 2019 πιο συχνά χρησιμοποιημένη λέξη είναι η λέξη «children» και δεύτερη η λέξη «implant». Στην

περίπτωση αυτή ο χρήστης μπορεί να αναγνωρίσει την σημασία των Stop Word λέξεων αφού για παράδειγμα αν η λέξη «children» δεν τον απασχολεί ιδιαίτερα, μπορεί να επαναλάβει την διαδικασία εξόρυξης δεδομένων με τα ίδια χαρακτηριστικά προσθέτοντας όμως λέξεις που δεν τον ενδιαφέρουν στη λίστα Stop Words. Για την ευκολότερη κατανόηση της σημασίας των Stop Words στην επόμενη ενότητα φαίνεται παράδειγμα με χρήση των λέξεων «children» και «studied» στη λίστα των Stop Words.

4.4.2 WordCloud

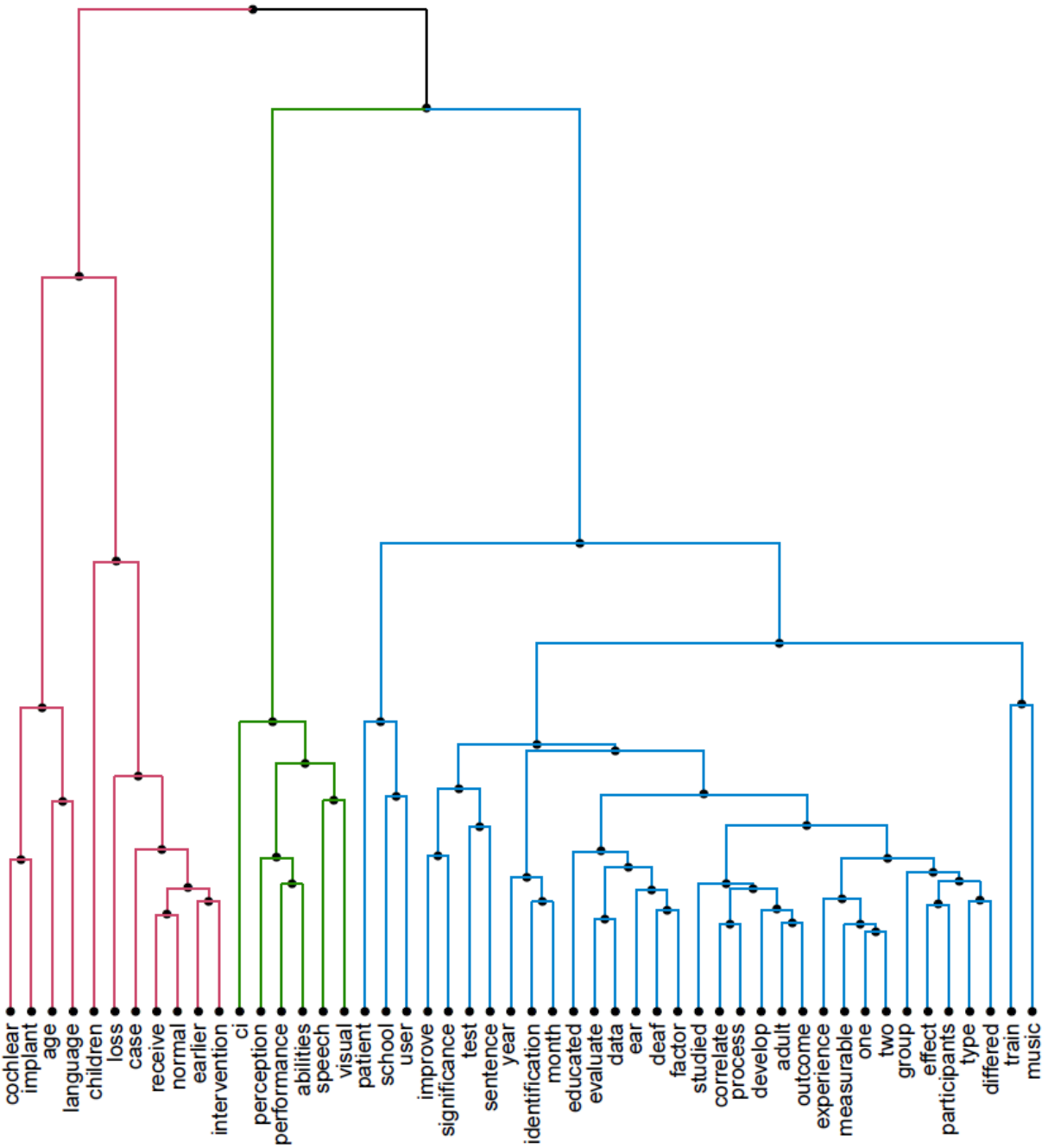


Εικόνα 36: Σύννεφο Λέξεων (WordCloud) με τις πιο συχνά χρησιμοποιημένες λέξεις

Στο αριστερά WordCloud φαίνεται το αποτέλεσμα από την αρχική εξόρυξη δεδομένων χωρίς Stop Words, ενώ στο δεξιά έχουν προστεθεί οι λέξεις «children» και «studied» στη λίστα Stop Words με αποτέλεσμα να μην εμφανίζονται πλέον.

Τα σύννεφα λέξεων δίνουν μια οπτική εικόνα των λέξεων που χρησιμοποιούνται πιο συχνά στα κείμενα που έχουν ανακτηθεί. Η πληροφορία που λαμβάνεται από αυτήν την εικόνα είναι η ίδια με την πληροφορία που λαμβάνεται από την γραφική παράσταση της προηγούμενης ενότητας με πιο οπτικό και φιλικό τρόπο προς το χρήστη. Η χρήση τέτοιων διαγραμμάτων είναι ευρέως διαδεδομένη σε πάρα πολλές σελίδες ή παρουσιάσεις διαφόρων θεμάτων. Με αυτό τον τρόπο μπορούμε με μια ματιά να δούμε την κύρια ιδέα για κάποιο θέμα σε μια σελίδα ή άρθρο, ή στην περίπτωση μας για μία ομάδα από κείμενα σχετικά με ένα πεδίο ενδιαφέροντος.

4.4.3 Clustering



Εικόνα 37: Hieratical Clustering στις 50 πιο χρησιμοποιημένες λέξεις

Η προσέγγιση ιεραρχικής ομαδοποίησης (hieratical clustering) δημιουργεί μια ένθετη ακολουθία έτσι ώστε να επιτευχθεί μια ιεραρχική δομή. Οι στρατηγικές της ιεραρχικής ομαδοποίησης εμπίπτουν σε δύο κατηγορίες: συσσωρευτική (agglomerative) και διαιρετική (divisive)[19]. Στη συγκεκριμένη περίπτωση γίνεται χρήση της agglomerative κατηγορίας, η οποία είναι μια προσέγγιση από τη βάση προς την κορυφή στην οποία κάθε αντικείμενο ξεκινά στο δικό του σύμπλεγμα και δημιουργούνται ζευγάρια συστάδων τα οποία συγχωνεύονται ανεβαίνοντας επίπεδο μέχρι να επιτευχθεί κάποια κατάσταση τερματισμού. Για τους σκοπούς της διπλωματικής εργασίας έχει επιλεχθεί τερματικός σταθμός ο ορισμός τριών τελικών συστάδων. Δηλαδή όταν η συγχώνευση φτάσει στην ομαδοποίηση όλων των λέξεων σε τρεις συνολικά συστάδες η διαδικασία τερματίζει.

4.4.4 Πίνακας κειμένων που ανακτήθηκαν

Στο παράδειγμα εξόρυξης δεδομένων που αναλύουμε σε αυτό το κεφάλαιο ανακτήθηκαν 47 κείμενα. Ο χρήστης μπορεί να έχει πρόσβαση σε αυτά τα κείμενα από τον πίνακα που υπάρχει στο κάτω μέρος της σελίδας. Ο πίνακας αυτός αποτελείται από τον αριθμό του κειμένου (PMID), τον τίτλο του και ένα κομμάτι από την περίληψη του. Στον πίνακα αυτό έχει χρησιμοποιηθεί σελιδοποίηση για την πιο ευχάριστη και φιλική παρουσίαση των κειμένων προς το χρήστη.

PMID	Title	Abstract
11201322	Speech perception with steeply sloping hearing loss: effects of frequency transposition.	Six adults with a very steeply sloping high-frequency hearing loss listened to monosyllabic words in several conditions. In the first condition, their... SHOW MORE
11211441	First auditory brainstem implant in the Czech Republic.	In the Czech Republic, the first implantation of a stimulation electrode into the brainstem was performed on 11 January 1999 in the Department of ORL,... SHOW MORE
11405147	[Status of hearing aid use by children in schools for the hearing impaired and deaf].	To evaluate and possibly improve the hearing aid fittings of children attending the Westphalian School for the Hearing Impaired or the Westphalian Sch... SHOW MORE
11462274	[Presbyacusis--hearing loss in old age].	Presbyacusis is a very common type of hearing loss, often having profound effects on the quality of life in old age. Since the number of elderly perso... SHOW MORE
12122028	Hearing after congenital deafness: central auditory plasticity and sensory deprivation.	The congenitally deaf cat suffers from a degeneration of the inner ear. The organ of Corti bears no hair cells, yet the auditory afferents are preserv... SHOW MORE

Rows per page: 5 1-5 of 47

Εικόνα 38: Πίνακας κειμένων που έχουν ανακτηθεί και αναλυθεί από τη διαδικασία.

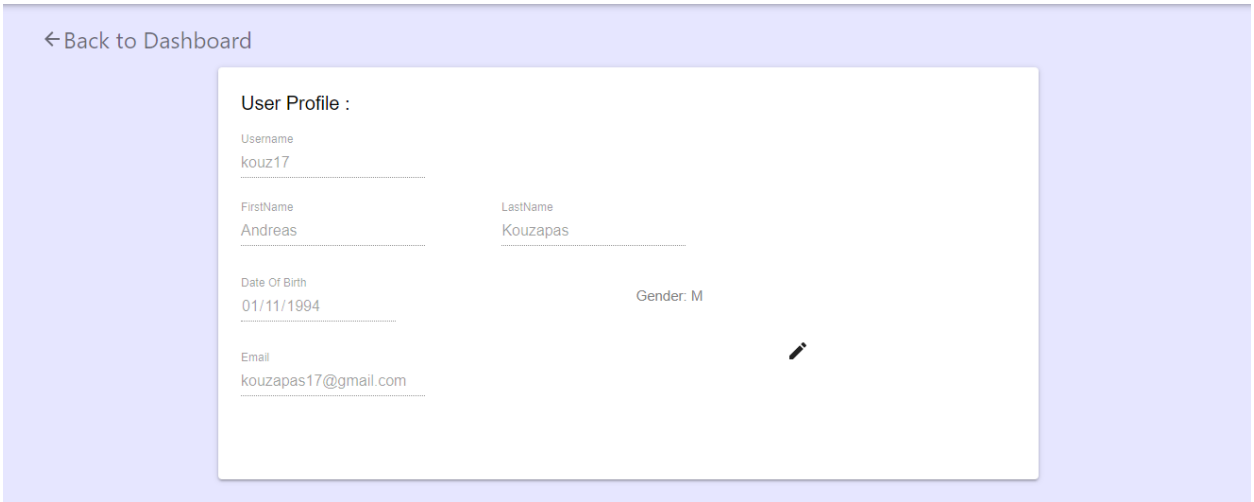
Επιπλέον δίνεται η δυνατότητα στο χρήστη να αλλάξει τον αριθμό των κειμένων που παρουσιάζονται σε κάθε σελίδα από 5 σε 10 και 25 στοιχεία.

Αν ο χρήστης επιθυμεί να έχει πρόσβαση σε όλη την περίληψη του κειμένου που τον ενδιαφέρει μπορεί να πατήσει στην επιλογή “SHOW MORE” που βρίσκεται στο πεδίο αυτό. Με αυτό τον τρόπο θα μπορεί να διαβάσει από ένα αναδυόμενο παράθυρο ολοκληρωμένη την περίληψη του συγκεκριμένου κειμένου όπως φαίνεται στην Εικόνα 39.

Εικόνα 39: Αναδυόμενο παράθυρο περίληψης κειμένου

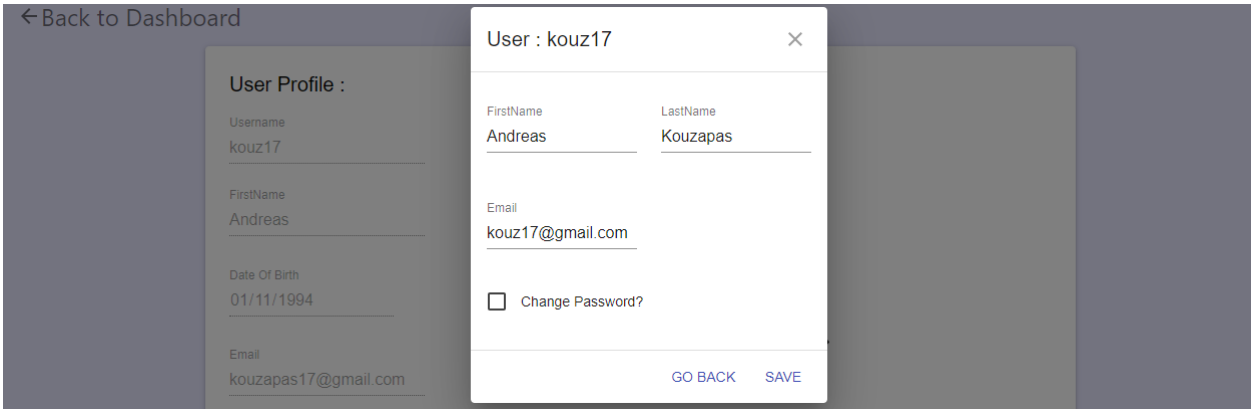
4.5 My Account

Κάθε χρήστης έχει πρόσβαση στην σελίδα του λογαριασμού του από το Navigation Bar μενού. Στη σελίδα My Account κάθε χρήστης μπορεί να προβάλει τα στοιχεία του λογαριασμού του.



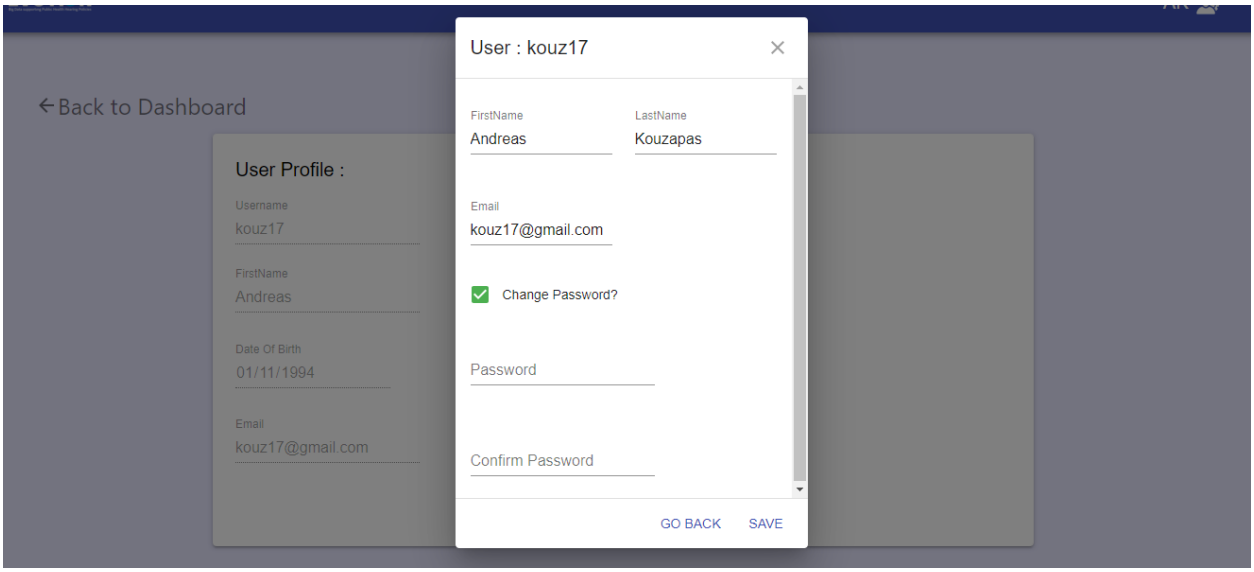
Εικόνα 40: Σελίδα λογαριασμού χρήστη.

Ο χρήστης δεν έχει δικαίωμα αλλαγής username για λόγους ασφαλείας και μοναδικότητας. Επίσης έχουμε υποθέσει ότι δεν είναι αναγκαίο ο χρήστης να έχει τη δυνατότητα αλλαγής ημερομηνίας γέννησης και φύλου. Για επεξεργασία των προσωπικών του στοιχείων ο χρήστης επιλέγει το κουμπί επεξεργασίας (Edit logo) που φαίνεται στην **Εικόνα 40** ώστε να εμφανιστεί η φόρμα αλλαγής των στοιχείων που χρειάζεται. Η φόρμα παρουσιάζεται στη μορφή αναδυόμενου παράθυρου όπως φαίνεται στην **Εικόνα 41**. Ο χρήστης μπορεί να αλλάξει όνομα, επίθετο και email. Επίσης δίνεται η δυνατότητα αλλαγής κωδικού, με την κατάλληλη επαλήθευση.

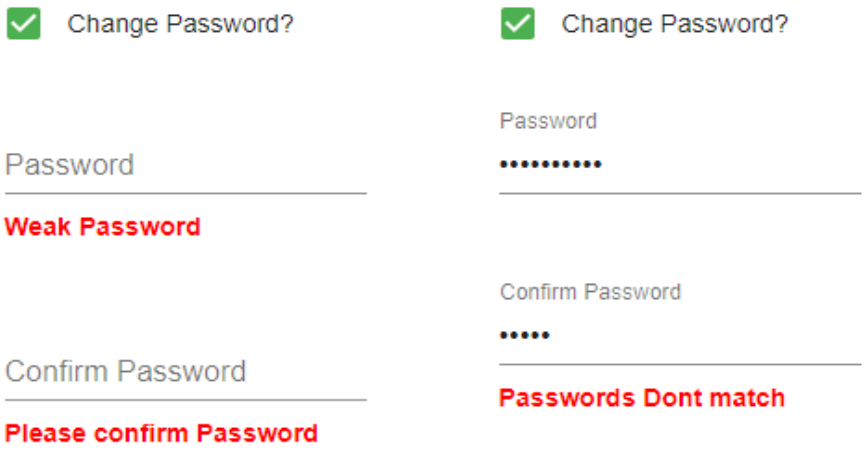


Εικόνα 41: Φόρμα αλλαγής στοιχείων χρήστη

Αν ο χρήστης θέλει να αλλάξει τον κωδικό του επιλέγει το κουτί “Change Password?” ώστε να εμφανιστούν τα κατάλληλα πεδία αλλαγής κωδικού. Για λόγους ασφαλείας ο χρήστης καλείται να επαληθεύσει τον κωδικό του. Ο νέος κωδικός πρέπει να αποτελείται από τουλάχιστον 8 χαρακτήρες για να είναι έγκυρος. Σε περίπτωση σφάλματος ο χρήστης βλέπει το κατάλληλο μήνυμα όπως φαίνεται στην **Εικόνα 43**.



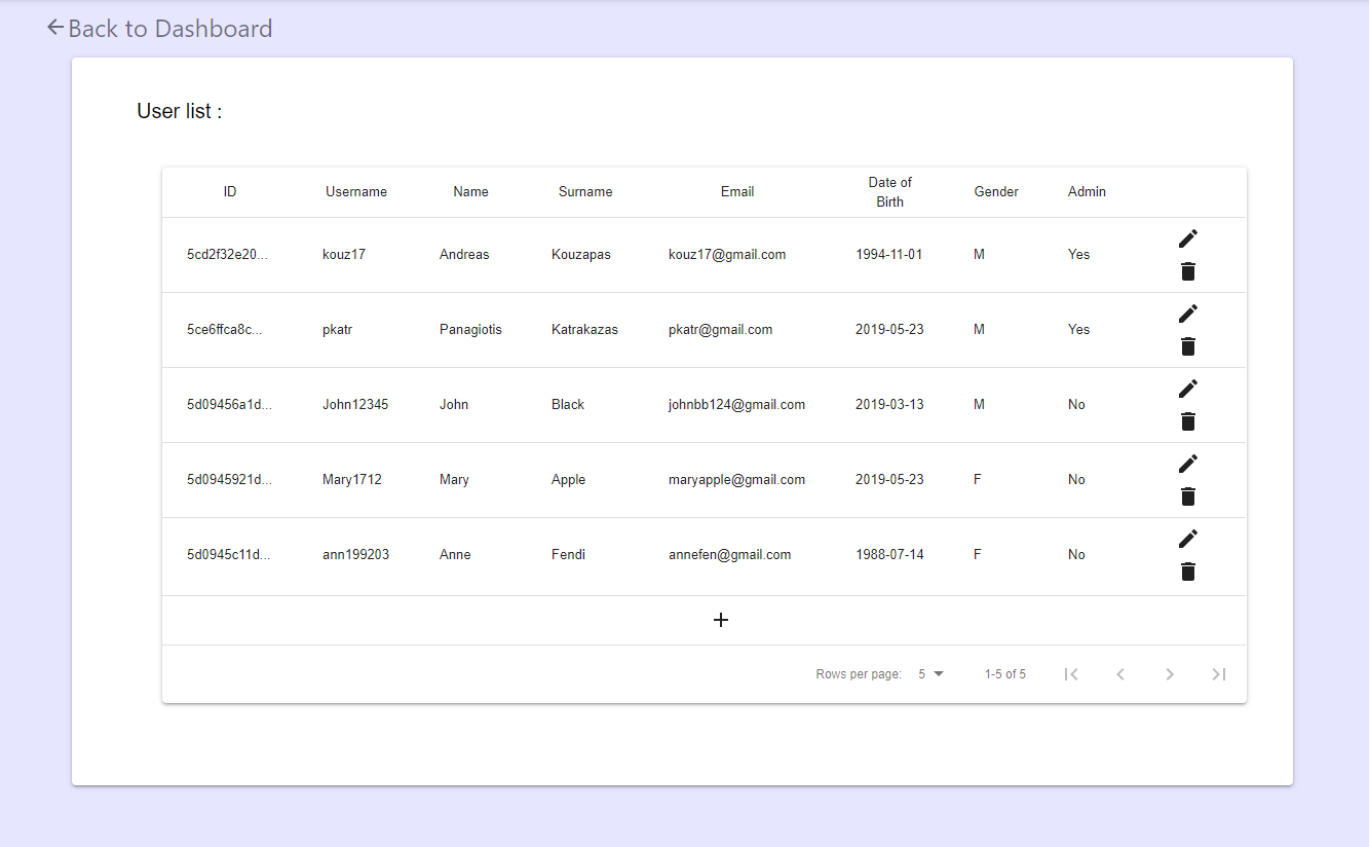
Εικόνα 42: Φόρμα αλλαγής κωδικού



Εικόνα 43: Έλεγχος και επαλήθευση κωδικού

4.6 Admin Page

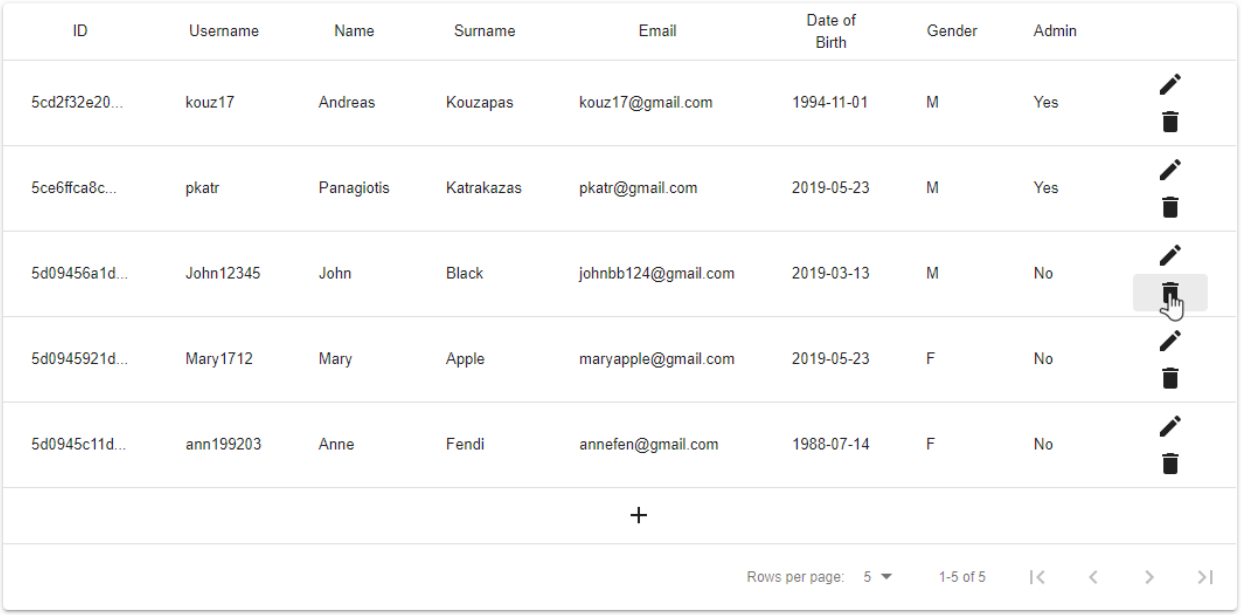
Η σελίδα αυτή είναι προσβάσιμη μόνο από τους διαχειριστές της πλατφόρμας. Κατά την ανάπτυξη αυτής της εφαρμογής ως αρχική ιδέα ήταν το κοινό στο οποίο είναι διαθέσιμη να είναι περιορισμένο. Δηλαδή δεν δίνεται η δυνατότητα σε κάποιο χρήστη να εγγραφεί με δική του πρωτοβουλία. Η εγγραφή νέων χρηστών γίνεται μόνο από τους διαχειριστές της πλατφόρμας, οι οποίοι έχουν επίσης το δικαίωμα επεξεργασίας στοιχείων για μετατροπή ενός απλού χρήστη σε διαχειριστή και η διαγραφή ενός χρήστη.



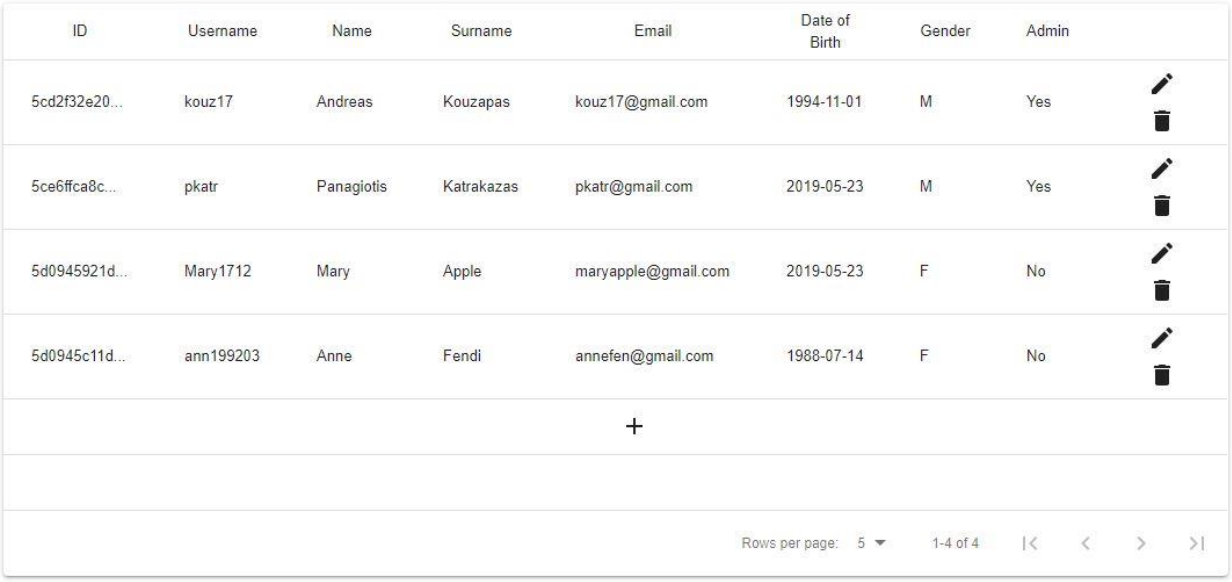
Εικόνα 44: Σελίδα Διαχειριστή (Admin Page)

Η σελίδα αποτελείται από ένα πίνακα ο οποίος περιέχει ως στοιχεία τον μοναδικό αριθμό του χρήστη που δίνεται αυτόματα από τη βάση δεδομένων(MongoDB id), username, όνομα, επίθετο, email, ημερομηνία γέννησης, φύλο και είδος χρήστη.

Για να διαγραφεί ένας χρήστης αρκεί απλά ο διαχειριστής να πατήσει το αντίστοιχο κουμπί δίπλα από το χρήστη που θέλει να διαγράψει, με τον πίνακα να ανανεώνεται αυτόματα με τη νέα λίστα.

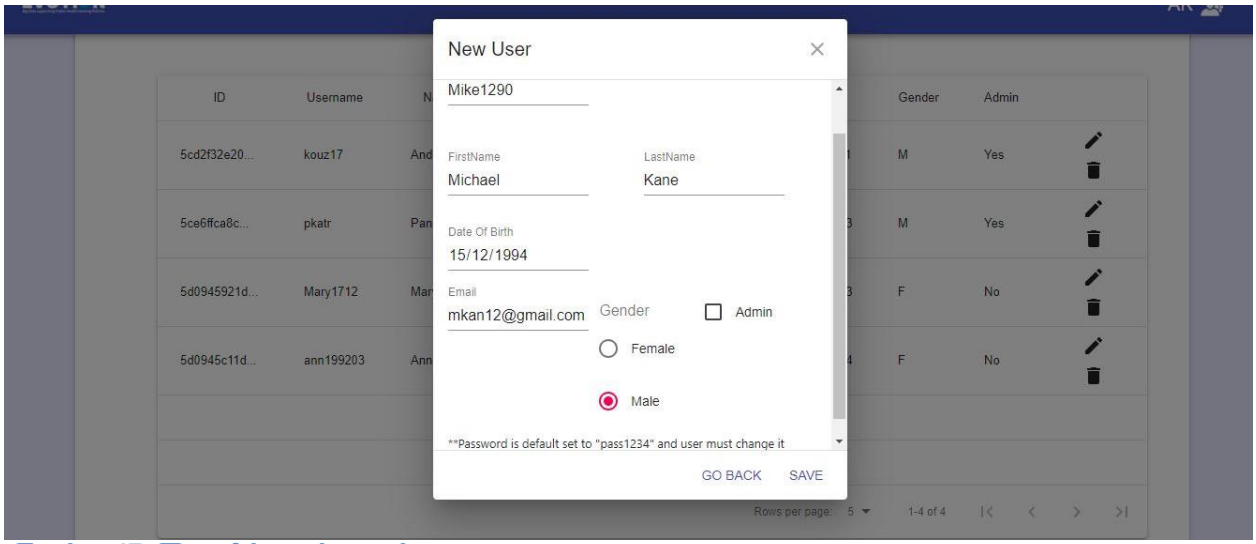


Εικόνα 45: Διαγραφή χρήστη.



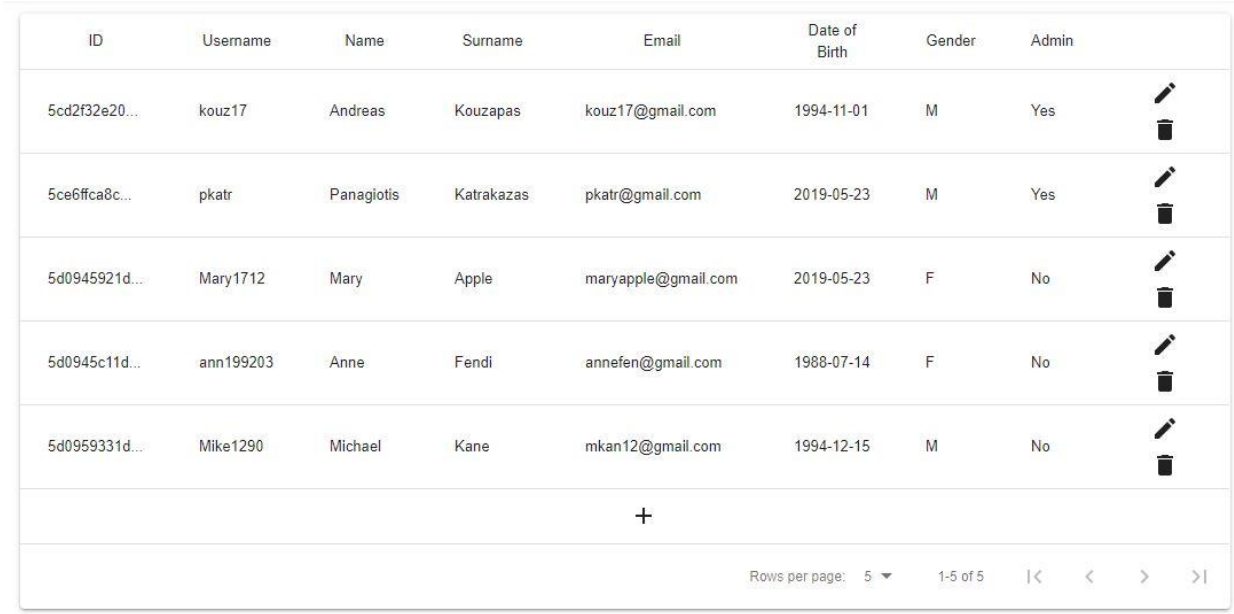
Εικόνα 46: Ο χρήστης αφαιρέθηκε από τη λίστα.

Ο διαχειριστής μπορεί να προσθέσει νέο χρήστη πατώντας στο κουμπί “+” που βρίσκεται στο τέλος του πίνακα. Η φόρμα εγγραφής νέου χρήστη εμφανίζεται σε αναδυόμενο παράθυρο και φαίνεται στην **Εικόνα 47**.



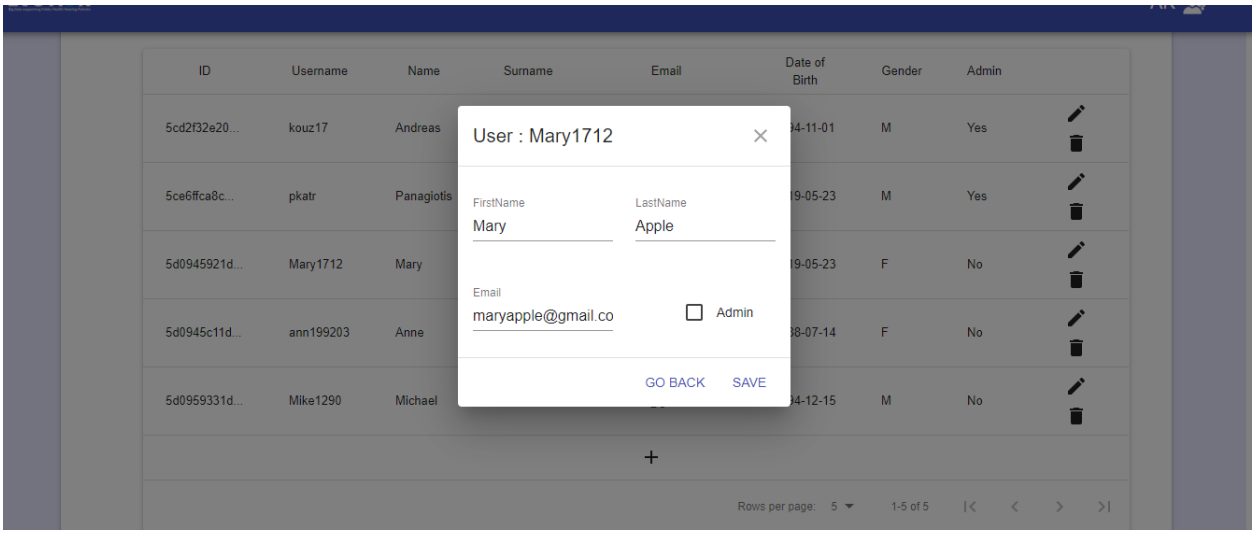
Εικόνα 47: Προσθήκη νέου χρήστη.

Για την εγγραφή νέου χρήστη στην πλατφόρμα ο διαχειριστής καλείται να εισάγει μετά από συνεννόηση με τον νέο χρήστη, τα προσωπικά του στοιχεία. Η φόρμα εγγραφής αποτελείται από πεδία εισαγωγής username, όνομα, επίθετο, ημερομηνία γέννησης, email, φύλο και αν πρόκειται για απλό χρήστη ή διαχειριστή. Όπως φαίνεται στο τέλος της φόρμας για λόγους ασφαλείας αρχικά ο κωδικός είναι προκαθορισμένος σε “pass1234” και ο νέος χρήστης καλείται να τον αλλάξει στον κωδικό αρεσκείας του. Με την συμπλήρωση των στοιχείων του νέου χρήστη και την επιλογή “SAVE” γίνεται ανανέωση του πίνακα των χρηστών με τη νέα λίστα.



Εικόνα 48: Ο νέος χρήστης έχει προστεθεί στη λίστα.

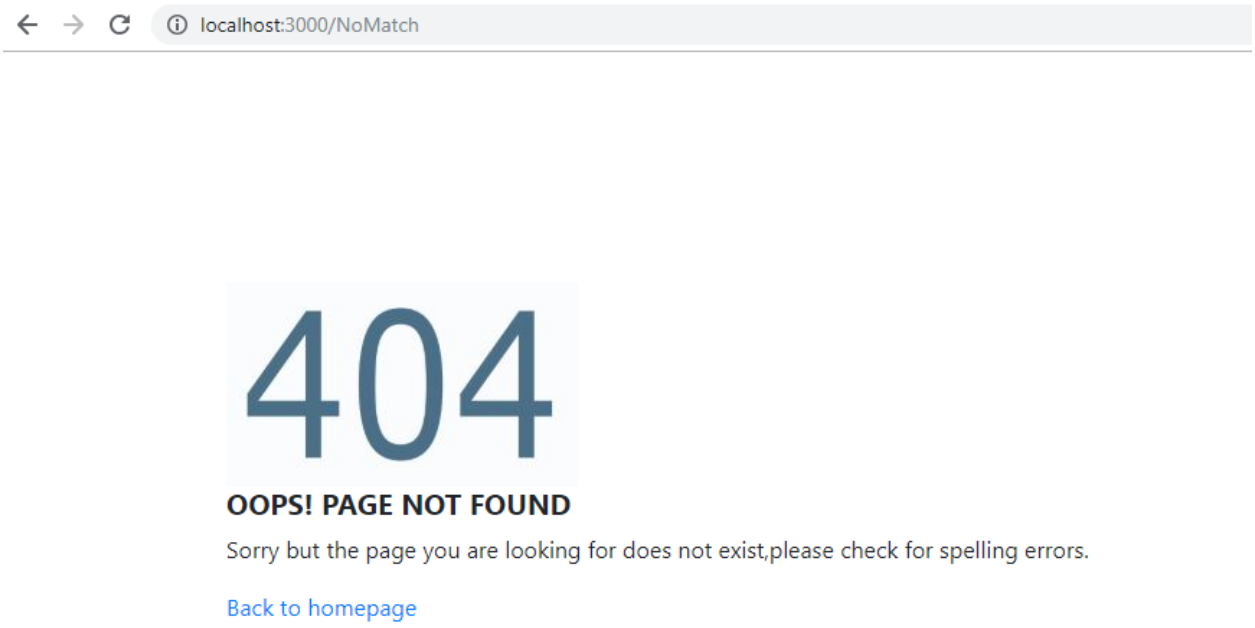
Επιπλέον ο διαχειριστής έχει το δικαίωμα αναβάθμισης ενός απλού χρήστη σε διαχειριστή ή το αντίστροφο. Αυτό γίνεται πατώντας το αντίστοιχο κουμπί δίπλα από τον χρήστη που επιθυμεί να επεξεργαστεί και στη συνέχεια αλλάζοντας όποιο στοιχείο επιθυμεί από το αναδυόμενο παράθυρο που εμφανίζεται στην οθόνη του.



Εικόνα 49: Επεξεργασία στοιχείων χρήστη

4.7 Page Not Found

Αυτή η σελίδα είναι φτιαγμένη για λόγους τυχών σφαλμάτων κατά την πλοήγηση στην εφαρμογή. Αν κάποιος χρήστης πληκτρολογήσει λανθασμένο URL, ή κατά την χρήση της πλατφόρμας παρουσιαστεί κάποιο εσωτερικό σφάλμα τότε γίνεται η παραπομπή σε αυτή τη σελίδα προς πληροφόρηση του χρήστη.



Εικόνα 50: Σελίδα λανθασμένης πλοήγησης ή σφάλματος

Ο χρήστης πατώντας στον σύνδεσμο “Back to homepage” μεταφέρεται στη σελίδα Dashboard αν η σύνδεση του είναι ακόμη ενεργή. Αν έχει λήξει η σύνδεση του ή έχει προκύψει εσωτερικό σφάλμα τότε ο χρήστης θα μεταφερθεί στην αρχική σελίδα σύνδεσης.

Κεφάλαιο 5: Σύνοψη, Περιορισμοί & Μελλοντικές Επεκτάσεις

Στο κεφάλαιο αυτό ακολουθεί μια σύντομη και ολοκληρωμένη περιγραφή της πλατφόρμας που αναπτύχτηκε. Στη συνέχεια θα αναφερθούμε σε κάποιους περιορισμούς που αντικρίσαμε κατά την εκπόνηση της διπλωματικής εργασίας και τέλος θα δώσουμε συγκεκριμένες προτάσεις για μελλοντικά σχέδια και βελτιώσεις.

5.1 Σύνοψη

Στα πλαίσια της διπλωματικής εργασίας στόχος ήταν η ανάπτυξη μιας δικτυακής πλατφόρμας για εξόρυξη δεδομένων από κείμενα Ιατρικών Βιβλιογραφικών Βάσεων. Βασικός σκοπός ήταν η δημιουργία ενός γραφικού περιβάλλοντος (GUI-Graphical User Interface) για την ευκολία ανάκτησης και παρουσίασης αποτελεσμάτων μετά από εξόρυξη δεδομένων από κείμενα. Η εργασία υλοποιήθηκε με συνδυασμό βάσης δεδομένων (MongoDB) για την διατήρηση χρηστών, Node.js για υποστήριξη, RESTful API για σύνδεση με βάση και εκτέλεση εξόρυξης δεδομένων δια μέσω της γλώσσας R και ReactJS για ανάπτυξη του γραφικού περιβάλλοντος εισαγωγής και προβολής δεδομένων από το χρήστη. Ο χρήστης μετά την σύνδεση του στην πλατφόρμα έχει τη δυνατότητα εισαγωγής δεδομένων για ανάκτηση και εξόρυξη δεδομένων από κείμενα σχετικά με τον τομέα ενδιαφέροντος του. Τα αποτελέσματα παρουσιάζονται σε διαφορετική σελίδα και δίνουν τη δυνατότητα στο χρήστη να τα μελετήσει ή να τα χρησιμοποιήσει για άλλους σκοπούς. Για παράδειγμα μπορεί να χρησιμοποιήσει την πλατφόρμα για εξαγωγή ενός WordCloud για χρήση σε προσωπική του ιστοσελίδα ή κάποιο άρθρο.

5.2 Περιορισμοί

Κατά την εκπόνηση της παρούσας διπλωματικής εργασίας προέκυψαν αρκετές δυσκολίες και περιορισμοί. Αρχικά το θέμα εξόρυξης δεδομένων από κείμενα, είναι ένα θέμα που απασχολεί την κοινωνία την σύγχρονη εποχή με αποτέλεσμα να υπάρχουν πολλές εκδοχές και μεθοδολογίες οι οποίες εξελίσσονται καθημερινά. Αυτό έχει ως αποτέλεσμα την ανάπτυξη νέων μεθοδολογιών ή βελτίωση υπάρχουσων σε θέματα απόδοσης. Η μεθοδολογία εξόρυξης δεδομένων από κείμενα που χρησιμοποιήθηκε δια μέσω της γλώσσας R παρέχει πολλές δυνατότητες προς το χρήστη, όμως η έλλειψη τεχνολογικού υλικού υποστήριξης τις περιορίζει.

Δηλαδή ο χρόνος ανάκτησης κειμένων και ακολούθως η εξόρυξη δεδομένων από αυτά δεν είναι ικανοποιητικός για μεγάλο όγκο ανακτημένων κειμένων.

Επίσης δυσκολία στην ανάπτυξη της πλατφόρμας παρουσιάστηκε στην επιλογή των κατάλληλων διαδικτυακών τεχνολογιών υποστήριξης και υλοποίησης της. Υπάρχει μια πληθώρα επιλογών από μεθοδολογίες ανάπτυξης διαδικτυακών εφαρμογών και η επιλογή μιας συγκεκριμένης για την υλοποίηση της εργασίας χρειάστηκε μελέτη και σύγκριση μεταξύ τους. Λόγο προηγούμενης εμπειρίας προτιμηθήκαν επιλογές που ασχολούνταν κυρίως με JavaScript. Αυτή η προτίμηση οδήγησε αρχικά στην επιλογή της node.js για το back-end και στη συνέχεια, μετά από τη μελέτη που παρουσιάστηκε στο κεφάλαιο 1, στη React.

Η μεγαλύτερη δυσκολία που παρουσιάστηκε κατά την υλοποίηση της εφαρμογής ήταν η χρήση κώδικα γλώσσας R δια μέσω της node.js. Πριν να επιλεχθεί η node.js για την υλοποίηση της εργασίας έγινε μια μελέτη σχετικά με το αν υπήρχε τρόπος συνδυασμού της με την γλώσσα R. Υπάρχει μια ειδική βιβλιοθήκη ονομαζόμενη «react-script» δια μέσω της οποίας μπορούσε να γίνει αυτή η συνένωση των δύο γλωσσών. Στη δική μας περίπτωση όμως λόγω του ότι το πρόγραμμα που είχε γραφτεί σε R επέστρεφε αρχεία και δεχόταν παραμέτρους από τη γραμμή εντολών για τη λειτουργία της δεν ήταν αρκετή. Για το λόγο αυτό έγινε χρήση της γλώσσας R εξολοκλήρου δια μέσω της γραμμής εντολών (command line). Η node.js παρέχει τη δυνατότητα στον προγραμματιστή να τρέξει εντολές στο command line με τη χρήση της επέκτασης «exec»[22]. Για λόγους απόδοσης και βελτιστοποίησης της ποιότητας επικοινωνίας με τον χρήστη χρειάστηκε η μετάβαση από τον ασύγχρονο χαρακτήρα της node.js σε συγχρονισμένο.

Επιπλέον λόγω του ότι η εξόρυξη δεδομένων από κείμενα είναι μία πολύπλοκη διαδικασία και χρειάζεται αρκετό χρόνο για τη διεκπεραίωση της, κρίθηκε αναγκαία η δυνατότητα στο χρήστη να μπορεί να την ακυρώσει ανά πάσα στιγμή. Αυτό δεν ήταν δυνατό με τη χρήση της node.js βασιζόμενης σε single-process και single-threaded αρχιτεκτονική. Μετά από μελέτη σχετικά με την ικανότητα δημιουργίας νέου νήματος (thread) στην node.js για την αντιμετώπιση αυτού του θέματος, κατέληξα στη χρήση της επέκτασης «cluster»[23] για τη δημιουργία νέου νήματος για την ανεξάρτητη λειτουργία της διαδικασίας εξόρυξης δεδομένων. Αυτή η παραλλαγή της

[22] https://nodejs.org/api/child_process.html#child_process_child_process_exec_command_options_callback
[23] https://nodejs.org/api/cluster.html#cluster_cluster_ismaster

λειτουργίας της node.js εκτός από τη δυνατότητα που δίνει στο χρήστη για τερματισμό της εξόρυξης δεδομένων ασύγχρονα, βελτιώνει παράλληλα και την απόδοση του προγράμματος αφού η εξόρυξη δεδομένων δεν διατηρεί τον εξυπηρετητή κατειλημμένο αλλά αντιθέτως οι δύο διεργασίες λειτουργούν ανεξάρτητα.

5.3 Μελλοντικές Επεκτάσεις

Στη συνέχεια θα παρουσιαστούν πιθανές βελτιώσεις και επεκτάσεις που μπορούν να υλοποιηθούν σε μεταγενέστερο στάδιο.

Αρχικά λόγω του ότι η πλατφόρμα είναι προσβάσιμη μόνο στο εσωτερικό του δικτύου που ανήκει ο υπολογιστής ο οποίος φιλοξενεί τον εξυπηρετητή και τη βάση δεδομένων, μια πρόταση επέκτασης είναι η δημοσίευση της μέσω μιας δημόσιας διεύθυνσης. Επίσης κατά τη δημοσίευση της πλατφόρμας σε μεγαλύτερο κοινό καλό θα ήταν να δίνεται η δυνατότητα εγγραφής στους ενδιαφερόμενους χρήστες μέσω μιας φόρμας και στη συνέχεια η αποδοχή αιτήματος εγγραφής από τους διαχειριστές αντί να γίνεται εξολοκλήρου η εγγραφή χρηστών από αυτούς. Αυτό όχι μόνο θα διευκολύνει τους διαχειριστές αλλά θα διευρύνει και το κοινό που μπορεί να χρησιμοποιήσει την πλατφόρμα.

Επιπλέον μια μελλοντική επέκταση είναι η δυνατότητα επιλογής από τους χρήστες του μοντέλου ανάκτησης εγγράφων. Στην συγκεκριμένη περίπτωση γίνεται χρήση του μοντέλου επιλογής εγγράφων Boolean με χρήση του λογικού AND, επομένως μελλοντική επέκταση αποτελεί η δυνατότητα χρήσης μοντέλου επιλογής εγγράφων Boolean με χρήση του λογικού OR ή χρήση μοντέλου ταξινόμησης εγγράφων. Για να γίνει όμως εφικτή αυτή η επέκταση κρίνεται αναγκαία η ανάπτυξη του τεχνολογικού υλικού υποστήριξης και η βελτιστοποίηση της απόδοσης του αλγορίθμου εξόρυξης δεδομένων αφού ο αριθμός των ανακτημένων εγγράφων κάθε φορά θα είναι μεγαλύτερος.

Επιπρόσθετα όπως αναφέρθηκε σε προηγούμενο κεφάλαιο σχετικά με την ανάπτυξη μοντέλων δεδομένων για πρόταση όρων και λέξεων για τη διευκόλυνση των χρηστών κατά την εισαγωγή όρων ανάκτησης εγγράφων, μια μελλοντική ανάπτυξη είναι η προσθήκη επιπλέων μοντέλων τόσο στον τομέα ενδιαφέροντος που έχει ήδη αναπτυχθεί, όσο και σε διαφορετικούς τομείς.

Αυτό μπορεί να γίνει ευκολότερα με τη χρήση της MongoDB και τη δημιουργία εγγράφων ειδικά για αυτό το σκοπό.

Ακόμη, για διεύρυνση του διαθέσιμου υλικού για ανάκτηση δεδομένων από τους χρήστες, η χρήση περισσότερων από μίας αποθήκης βιοιατρικής βιβλιογραφίας θα ήταν προτιμότερη. Στην περίπτωση της εφαρμογής που έχει αναπτυχτεί γίνεται χρήση μόνο μιας γνωσιακής βάσης βιοιατρικών δεδομένων (PubMed) η οποία επιτρέπει τη χρήση του API της από εξωτερικούς παράγοντες.

Τέλος μία μελλοντική επέκταση μπορεί να είναι η διατήρηση ενός ιστορικού αναζητήσεως για κάθε χρήστη ώστε να δίνεται η δυνατότητα στον ίδιο να ανατρέξει σε παλαιότερες αναφορές εξόρυξης δεδομένων που είχε πραγματοποιήσει. Αυτό θα διευκόλυνε τους χρήστες, εφόσον δεν θα χρειάζεται κάθε φορά να περιμένουν μέχρι να γίνει ανάκτηση και εξόρυξη δεδομένων για όρους τους οποίους είχαν χρησιμοποιήσει στο παρελθόν. Επίσης θα μπορούσε να υπάρξει και ένας τρόπος αξιολόγησης των εξορυγμένων προτύπων που είχαν παρουσιαστεί στο χρήστη ο οποίος θα λειτουργούσε ως ανατροφοδότηση (feedback) προς τους διαχειριστές της πλατφόρμας.

Βιβλιογραφία

[1] E. Kyrkos and E. Κύρκος, “Εξόρυξη Γνώσης από Δεδομένα,” Jan. 2016.

[2] “Big Data Explained | MongoDB.” [Online]. Available: https://www.mongodb.com/big-data-explained. [Accessed: 30-May-2019].

[3] Π. Σ. Κατρακάζας, “Διερεύνηση Ιατρικών Υποθέσεων Για Τον Καρκίνο Του Τραχήλου Της Μήτρας Με Χρήση Προηγμένων Τεχνικών Εξόρυξης Γνώσης Σε Κείμενα Διαδικτυακών Βάσεων Βιοιατρικής Βιβλιογραφίας,” 2014.

[4] “What is a Web Browser? - Definition from Techopedia.” [Online]. Available: https://www.techopedia.com/definition/288/web-browser. [Accessed: 08-Jun-2019].

[5] “Best Web Browser 2019: Chrome, Edge, Firefox & More Compared - Tech Advisor.” [Online]. Available: https://www.techadvisor.co.uk/test-centre/software/best-web-browsers-3635255/. [Accessed: 08-Jun-2019].

[6] K. Cheilas, A. Vakaloudis, A. Politis, K. Χειλάς, A. Βακαλούδης, and A. Πολίτης, “HTML και CSS,” Jan. 2016.

[7] R. T. Fielding, “Representational State Transfer (REST),” University of California, 2000.

[8] “Angular.” [Online]. Available: https://angular.io/. [Accessed: 11-Jun-2019].

[9] “React – A JavaScript library for building user interfaces.” [Online]. Available: https://reactjs.org/. [Accessed: 11-Jun-2019].

[10] “Vue.js.” [Online]. Available: https://vuejs.org/. [Accessed: 11-Jun-2019].

[11] “React vs Angular vs Vue.js—What to choose in 2019? (updated).” [Online]. Available: https://medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d. [Accessed: 11-Jun-2019].

[12] I. Ntzoufras, D. Karlis, I. Ντζούφρας, and Δ. Καρλής, “Εισαγωγή στον προγραμματισμό και στη στατιστική ανάλυση με R,” Feb. 2016.

[13] V. Verykios, V. Kagklis, E. Stavropoulos, B. Βερύκιος, B. Καγκλής, and H. Σταυρόπουλος, “Η επιστήμη των δεδομένων μέσα από τη γλώσσα R,” Feb. 2016.

[14] J. Caudwell, “LibGuides: Text & Data Mining: What is TDM?”

[15] M. Steyvers and T. Griffiths, “Probabilistic Topic Models.”

[16] “Stemming and lemmatization.” [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html. [Accessed: 26-Jun-2019].

[17] Δ. Χαρκιολάκη, “Συγκριτική μελέτη μεταξύ της javascript βιβλιοθήκης reactjs και του πλαισίου ανάπτυξης angular,” 2019.

[18] “Introducing JSX – React.” [Online]. Available: https://reactjs.org/docs/introducing-jsx.html. [Accessed: 14-Jun-2019].

[19] N. Garg and R. K. Gupta, “Clustering Techniques for Text Mining: A Review,” no. 5, p. 2016, 2016.