



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Εξατομικευμένη εύρεση γευμάτων
με Μηχανική Μάθηση και Επεξεργασία Φυσικής Γλώσσας**

ΜΠΟΥΣΙΟΣ ΑΡΗΣ

A.M. 18390061

Εισηγητής: Χρήστος Τρούσσας, Επ. Καθηγητής

Αθήνα, Απρίλιος 2023

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Εξατομικευμένη εύρεση γευμάτων με Μηχανική Μάθηση και Επεξεργασία
Φυσικής Γλώσσας**

ΜΠΟΥΣΙΟΣ ΑΡΗΣ

A.M. 18390061

Εισηγητής:

Χρήστος Τρούσσας, Επ. Καθηγητής

Εξεταστική Επιτροπή:

**Χρήστος Τρούσσας, Επ. Καθηγητής
Ακριβή Κρούσκα, Μέλος ΕΔΙΠ
Παναγιώτα Τσελέντη, Μέλος ΕΔΙΠ**

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Άρης Μπούσιος του Βασίλειου, με αριθμό μητρώου 18390061 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Βεβαιώνω ότι είμαι συγγραφέας αυτής της Διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

«Βεβαιώνω ότι είμαι συγγραφέας της παρούσας διπλωματικής εργασίας και ότι έχω αναφέρει ή παραπέμψει σε αυτή, ρητά και συγκεκριμένα, όλες τις πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, προτάσεων ή λέξεων, είτε αυτές μεταφέρονται επακριβώς (στο πρωτότυπο ή μεταφρασμένες) είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για την συγκεκριμένη διπλωματική εργασία»

Ο/Η Δηλών/ούσα

Άρης Μπούσιος

Α. Μπούσιος

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό των εφαρμογών στη μηχανική μάθηση. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής Χρήστος Τρούσσας, τον οποίο θα ήθελα να ευχαριστήσω. Ακόμα θα ήθελα να ευχαριστήσω την οικογένειά μου για τη συμπαράσταση κατά τη διάρκεια των σπουδών μου.

ΠΕΡΙΛΗΨΗ

Η διπλωματική εργασία αυτή πραγματεύεται την κατασκευή μιας Fullstack διαδικτυακής εφαρμογής που στοχεύει στην παροχή ενός εξατομικευμένου ευρετή γευμάτων, αξιοποιώντας την δύναμη της μηχανικής μάθησης. Στο πλαίσιο της σύγχρονης κοινωνίας, όπου η τεχνολογία και οι διαδικτυακές εφαρμογές καταλαμβάνουν ένα ολοένα και πιο ουσιαστικό μέρος της καθημερινότητάς μας, το έργο επικεντρώνεται στην εκμετάλλευση αυτών των τάσεων για την ανάπτυξη μιας καινοτόμας εφαρμογής. Η εφαρμογή αυτή συνδυάζει σύγχρονες τεχνολογίες Frontend και Backend για την παροχή μιας ολοκληρωμένης εμπειρίας στον χρήστη. Περιλαμβάνει σελίδες σύνδεσης και εγγραφής για τη διαχείριση των χρηστών, καθώς και μια κύρια σελίδα για την αναζήτηση συνταγών μέσω μιας φιλικής προς τον χρήστη διεπαφής. Ένα από τα κύρια χαρακτηριστικά αυτής της εφαρμογής είναι η ενσωμάτωση τεχνικών κατανόησης φυσικής γλώσσας (NLP) μέσω της χρήσης του προ-εκπαιδευμένου μοντέλου t5-recipe-generator της Hugging face. Αυτό το μοντέλο, που έχει εκπαιδευτεί σε πάνω από 2 εκατομμύρια συνταγές, επιτρέπει την προσαρμογή της αναζήτησης συνταγών βάσει των αναγκών και των προτιμήσεων του χρήστη. Συμπέρασμα της διπλωματικής είναι η ευκολία της χρήσης μηχανικής μάθησης σε διαδικτυακές εφαρμογές για την ενίσχυση των λειτουργιών του συστήματος για μια καλύτερη εμπειρία ως προς τον χρήστη.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Μηχανική μάθηση, Διαδικτυακές εφαρμογές

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μοντέλα μηχανικής μάθησης, NLP, React.js, Fullstack, ML

ABSTRACT

This thesis deals with the construction of a Fullstack web application that aims to provide a personalized meal finder, utilizing the power of machine learning in the context of modern society, where technology and online applications occupy an increasingly essential part of our daily lives, the project focuses on exploiting these trends to develop an innovative application. This application combines modern Frontend and Backend technologies to provide a complete user experience. It includes login and registration pages to manage users, as well as a main page to search for recipes through a user-friendly interface. One of the main features of this

app is the integration of natural language understanding (NLP) techniques through the use of Hugging face's pre-trained t5-recipe-generator model. This model, which has been trained on over 2 million recipes, allows the recipe search to be customized based on the user's needs and preferences. The conclusion of the thesis is the ease of using machine learning in web applications to enhance system functions for a better user experience.

SCIENTIFIC AREA: Machine learning, online applications

KEYWORDS: Machine learning models, NLP, React.js, Fullstack, ML

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	8
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ.....	13
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ	14
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ.....	14
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ.....	15
ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ.....	16
ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	18
2.1 ΠΑΓΚΟΣΜΙΟΣ ΙΣΤΟΣ	18
2.2 ΔΙΑΔΙΚΤΥΑΚΕΣ ΕΦΑΡΜΟΓΕΣ	18
2.3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ CLIENT-SERVER ΤΩΝ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ	20
2.4 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΗΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ CLIENT SERVER	22
2.5 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ	23
2.6 ΕΚΠΑΙΔΕΥΣΗ ΜΟΝΤΕΛΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ	23
2.7 ΠΡΟ-ΕΚΠΑΙΔΕΥΜΕΝΑ ΜΟΝΤΕΛΑ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ	24
2.8 ΕΠΕΞΕΡΓΑΣΙΑ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ (NLP)	25
2.8 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ	27
ΚΕΦΑΛΑΙΟ 3: ΜΕΘΟΔΟΛΟΓΙΑ	30
3.1 ΣΤΟΧΟΣ ΚΑΙ ΤΑ ΥΠΟΠΡΟΒΛΗΜΑΤΑ.....	30
3.2 ΠΡΩΙΜΑ ΣΤΑΔΙΑ.....	30
3.3 ΕΜΠΟΔΙΑ ΚΑΤΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ	31
ΚΕΦΑΛΑΙΟ 4: ΤΟ ΣΥΣΤΗΜΑ ΜΟΥ.....	32
4.1 ΔΟΜΗ ΑΡΧΕΙΩΝ ΕΦΑΡΜΟΓΗΣ.....	32
4.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΦΑΡΜΟΓΗΣ.....	33
4.2.1 Η ΒΙΒΛΙΟΘΗΚΗ REACT	33
4.2.2 Η ΒΙΒΛΙΟΘΗΚΗ TYPESCRIPT.....	34
4.2.3 Η ΒΙΒΛΙΟΘΗΚΗ MATERIALUI	34
4.2.4 Η ΒΙΒΛΙΟΘΗΚΗ STYLED COMPONENTS	34
4.2.5 Η ΒΙΒΛΙΟΘΗΚΗ REACT ROUTER DOM	34
4.2.6 Η ΒΙΒΛΙΟΘΗΚΗ MONGOOSE	35
4.2.7 Η ΒΙΒΛΙΟΘΗΚΗ AXIOS.....	35
4.2.8 Η ΒΙΒΛΙΟΘΗΚΗ TRANSFORMERS.....	35
4.2.9 Η ΒΙΒΛΙΟΘΗΚΗ AUTOTOKENIZER	36
4.2.10 Η ΒΙΒΛΙΟΘΗΚΗ FLAXAUTOMODELFORSEQ2SEQLM.....	37
4.2.11 Η ΒΙΒΛΙΟΘΗΚΗ PYTORCH	37
4.3 ΕΜΠΡΟΣΘΙΟ ΑΚΡΟ (FRONTEND)	38

4.3.1 ΣΕΛΙΔΑ ΕΓΓΡΑΦΗΣ	38
4.3.2 ΣΕΛΙΔΑ ΣΥΝΔΕΣΗΣ	39
4.3.3 ΚΥΡΙΑ ΣΕΛΙΔΑ.....	41
4.4 ΠΙΣΩ ΑΚΡΟ (BACKEND).....	44
4.4.1 ΓΕΝΙΚΗ ΔΟΜΗ ΤΟΥ BACKEND	44
4.4.2 Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ MONGODB	47
4.4.3 ΤΟ ΠΕΡΙΒΑΛΛΟΝ CONDA.....	49
4.5 Η ΔΟΜΗ ΚΑΙ ΤΟ ΜΟΝΤΕΛΟ T5-RECIPE-GENERATOR.....	49
4.5.1 Η ΔΟΜΗ ΤΟΥ ΑΡΧΕΙΟΥ	49
4.5.2 ΠΑΡΑΜΕΤΡΟΙ ΜΟΝΤΕΛΟΥ	50
4.5.3 ΜΟΝΤΕΛΟ T5	51
4.5.4 ΜΕΤΑΦΟΡΑ ΜΑΘΗΣΗΣ	54
4.5.5 T5-RECIPE-GENERATOR.....	54
ΚΕΦΑΛΑΙΟ 5: ΑΞΙΟΛΟΓΗΣΗ	57
5.1 ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	57
5.2 ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΧΡΗΣΗΣ ΑΠΟ ΧΡΗΣΤΕΣ	63
5.2.1 ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ.....	63
5.2.2 ΕΥΧΡΗΣΤΙΑ	64
5.2.3 ΑΠΟΔΟΣΗ.....	65
5.2.4 ΑΙΣΘΗΤΙΚΗ	66
ΚΕΦΑΛΑΙΟ 6: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ.....	68
6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ	68
6.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΞΕΛΙΞΕΙΣ	68
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	70

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 4.1: Η σελίδα εγγραφής χρηστών.....	38
Εικόνα 4.2: Εμφάνιση μηνύματος ειδοποίησης προς τον χρήστη.....	39
Εικόνα 4.3: Η σελίδα σύνδεσης χρηστών.....	40
Εικόνα 4.4: Προσπάθεια λανθασμένης εισόδου.....	40
Εικόνα 4.5: Κύρια σελίδα της εφαρμογής.....	42
Εικόνα 4.6: Κύρια σελίδα της εφαρμογής με εμφάνιση συνταγής.....	43
Εικόνα 4.7: Σελίδα διαχειριστή Atlas της βάσης MongoDB.....	47
Εικόνα 4.8: Οι συλλογές της βάσης.....	48

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 2.1: Αρχιτεκτονική client-server (χρήση του προγράμματος Dia).....	21
Σχήμα 4.1: Δομή αρχείων εφαρμογής.....	33
Σχήμα 4.2: Διάγραμμα εφαρμογής Frontend κομματιού.....	44

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 4.1: Εκδόσεις βιβλιοθηκών εφαρμογής.....	37
Πίνακας 4.2: Μετρικές αξιολόγησης απόδοσης μοντέλου από [29].....	55

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

HTTP Hypertext Transfer Protocol

TCP Transmission Control Protocol

NLP Natural Language Processing

SPA Single Page Application

NPM Node Packet Manager

UI User Interface

CSS Cascading Style Sheets

Js JavaScript

Ts TypeScript

Pip pip installs packages

RNN Recurrent Neural Networks

LTSM Long Short-Term Memory

URL Uniform Resource Locator

JSON JavaScript Object Notation

BSON Binary JSON

CRUD Create, Read, Update, Delete

COSIM Cosine Similarity

WER Word Error Rate

BLEU Bilingual Evaluation Understudy

GLEU Google-BLEU

METEOR Metric for Evaluation of Translation with Explicit Ordering

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια, έχει αυξηθεί το ενδιαφέρον για τις εξατομικευμένες προτάσεις σε διάφορους τομείς στις διαδικτυακές εφαρμογές καθώς οι χρήστες απολαμβάνουν μια προσαρμοσμένη εμπειρία με βάση τα ενδιαφέροντά τους και διαφέρει από χρήστη σε χρήστη. Ένας τέτοιος τομέας είναι η διατροφή καθώς οι άνθρωποι γίνονται όλο και πιο επιδέξιοι στον αντίκτυπο της διατροφής στη συνολική τους υγεία.

Για την ανάγκη αυτή έχουν αναπτυχθεί εξατομικευμένα συστήματα συστάσεων με σκοπό την παροχή ατομικών διατροφικών συμβουλών και προτάσεων βασισμένων στις ανάγκες και τις προτιμήσεις του κάθε χρήστη. Ένα σύστημα συστάσεων [1] αποτελεί ένα εργαλείο που αναλαμβάνει το φιλτράρισμα και την προβολή πληροφοριών με βάση τα ενδιαφέροντα και τις προτιμήσεις του χρήστη, περιορίζοντας τον όγκο των δεδομένων που παρουσιάζονται. Βασίζεται στην ιδέα ότι οι χρήστες θα έχουν ενδιαφέρον για ένα συγκεκριμένο περιεχόμενο και θα θέλουν να λάβουν συστάσεις που ανταποκρίνονται στα ενδιαφέροντά τους. Είναι ιδιαίτερα χρήσιμο όταν υπάρχει υπερφόρτωση πληροφοριών και αναζήτησης. Αντί να χρειάζεται η χειροκίνητη αναζήτηση και το φιλτράρισμα μεγάλο όγκο πληροφοριών, το σύστημα μπορεί να προσφέρει μια εξατομικευμένη και προσαρμοσμένη εμπειρία.

Ένα από τα προβλήματα στη δημιουργία εξατομικευμένων συστάσεων είναι η τεράστια ποσότητα δεδομένων που πρέπει να επεξεργαστεί και να αναλυθεί για τη δημιουργία ακριβών και χρήσιμων προτάσεων. Για να αντιμετωπιστεί αυτό το πρόβλημα, έγινε η μεταστροφή στις τεχνικές μηχανικής μάθησης που μπορούν να εκπαιδευτούν και να μάθουν από μεγάλα σύνολα δεδομένων για να αναγνωρίσουν μοτίβα και να κάνουν προβλέψεις. Αυτά τα ονομάζουμε μοντέλα μηχανικής μάθησης (trained ml model).

Τα προ-εκπαιδευμένα μοντέλα μηχανικής μάθησης (pre-trained ml model), δηλαδή τα μοντέλα που έχουν ήδη εκπαιδευτεί σε κάποιο σύνολο δεδομένων με συγκεκριμένο αντικείμενο, για παράδειγμα οι συνταγές φαγητών, αντιπροσωπεύουν μια σημαντική εξέλιξη στον τομέα της τεχνητής νοημοσύνης και της μηχανικής μάθησης. Από την πρώτη στιγμή της γέννησης της μηχανικής ο στόχος ήταν να επιτευχθεί ένα σύστημα που θα μπορούσε να μάθει από τα δεδομένα και να εκπαιδευτεί στον τρόπο με τον οποίο λειτουργεί ο κόσμος. Οι αλγόριθμοι μηχανικής μάθησης βασίζονται στην ιδέα ότι ένας υπολογιστής μπορεί να μάθει να αναγνωρίζει πρότυπα στα δεδομένα και να κάνει προβλέψεις σε βάση αυτών των προτύπων.

Τα μοντέλα μηχανικής μάθησης μπορούν να εκπαιδευτούν σε μεγάλα σύνολα δεδομένων και να χρησιμοποιούνται για την επίλυση προβλημάτων που δεν έχουν σχέση με τα δεδομένα στα οποία εκπαιδεύτηκαν. Αυτό σημαίνει ότι ένα προ-εκπαιδευμένο μοντέλο μπορεί να παράγει ακριβείς προβλέψεις για δεδομένα που δεν είχαν χρησιμοποιηθεί κατά την εκπαίδευση του. Αυτό είναι επίτευγμα της τεχνολογίας της μεταφοράς μάθησης, που μας δίνει τη δυνατότητα να χρησιμοποιήσουμε τις γνώσεις που έχουν αποκτηθεί από την εκπαίδευση σε ένα πρόβλημα για να βελτιώσουμε την απόδοση σε άλλο πρόβλημα.

Η εργασία αυτή παρουσιάζει μια προσέγγιση της χρήσης τέτοιων μοντέλων και τεχνικών για τη δημιουργία ενός εξατομικευμένου εργαλείου σε μορφή σελίδας για την εύρεση γευμάτων με χρήση προ-εκπαιδευμένων μοντέλων μηχανικής μάθησης. Η σελίδα αυτή δέχεται από τον χρήστη τις προτιμήσεις του ως προς τα υλικά καθώς και ένα ή περισσότερα ιδιαίτερα χαρακτηριστικά - φίλτρα πχ gluten free. Τα δεδομένα αυτά χρησιμοποιούνται στο προ εκπαιδευμένο μοντέλο το οποίο θα παράξει κάποιο γεύμα παρέχοντας τα υλικά κατασκευής του καθώς και τα βήματα εκτέλεσης της συνταγής. Η εφαρμογή θεωρείται προσωποποιημένη καθώς διατηρεί τις προτιμήσεις αλλά και το ιστορικό του εκάστοτε χρήστη αποθηκεύοντας τα σε βάση δεδομένων και ανακτώνται σε κάθε σύνδεσή του με το σύστημα.

ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

2.1 ΠΑΓΚΟΣΜΙΟΣ ΙΣΤΟΣ

Ο παγκόσμιος ιστός ή διαδίκτυο (World Wide Web - WWW) είναι μια τεχνολογία που εφευρέθηκε τη δεκαετία του 1980 από τον Βρετανό επιστήμονα Tim Berners-Lee [2]. Η ιδέα του Berners-Lee ήταν να δημιουργήσει ένα σύστημα που θα επέτρεπε στους χρήστες να ανταλλάσσουν πληροφορίες και ιδέες μέσω ενός δικτύου υπολογιστών που θα ήταν παγκόσμια προσβάσιμο.

Η πρώτη ιστοσελίδα του κόσμου δημιουργήθηκε το 1991 [2], και από τότε το WWW έχει αλλάξει ριζικά τον τρόπο που επικοινωνούμε, εργαζόμαστε, και αγοράζουμε προϊόντα. Σήμερα, το WWW έχει γίνει ένα αναπόσπαστο κομμάτι της καθημερινότητάς μας, και σχεδόν κάθε επιχείρηση ή οργανισμός έχει μια ιστοσελίδα.

Σήμερα, οι ιστοσελίδες έχουν γίνει ανεπαίσθητο κομμάτι της καθημερινότητάς μας, καθώς χρησιμοποιούμε το διαδίκτυο για να βρούμε πληροφορίες, να επικοινωνήσουμε με τους φίλους και την οικογένειά μας, να αγοράσουμε προϊόντα και υπηρεσίες, και ακόμη και να εργαστούμε.

Στον σημερινό κόσμο, ο αριθμός τους έχει αυξηθεί σε εκατοντάδες εκατομμύρια. Περιλαμβάνουν όλα τα είδη των πληροφοριών, από τα μικρά προσωπικά ιστολόγια και τα ηλεκτρονικά καταστήματα έως τις μεγάλες εταιρικές ιστοσελίδες και τα διαδικτυακά κοινωνικά δίκτυα.

Εκτός από την αύξηση του αριθμού των ιστοσελίδων, οι ιστοσελίδες έχουν εξελιχθεί σημαντικά από το 1991. Αρχικά, οι ιστοσελίδες ήταν απλά στατικά κείμενα με μερικές εικόνες. Σήμερα, οι ιστοσελίδες είναι δυναμικές, πλούσιες σε περιεχόμενο και πλούσιες σε δυνατότητες, που επιτρέπουν στους χρήστες να αλληλεπιδρούν μαζί τους.

2.2 ΔΙΑΔΙΚΤΥΑΚΕΣ ΕΦΑΡΜΟΓΕΣ

Οι διαδικτυακές εφαρμογές (web applications) [3] είναι μια εξέλιξη του WWW, που επιτρέπει στους χρήστες να χρησιμοποιούν λογισμικό μέσω ενός προγράμματος περιήγησης στον ιστό. Αυτό σημαίνει ότι οι χρήστες μπορούν να αποκτήσουν πρόσβαση σε λογισμικό και δεδομένα από οπουδήποτε στον κόσμο, χωρίς να χρειάζεται να το εγκαταστήσουν στον δικό τους υπολογιστή.

Τα web applications έχουν καταλάβει μια όλο και μεγαλύτερη θέση στην καθημερινότητα μας, καθώς επιτρέπουν στους χρήστες να εκτελούν διάφορες λειτουργίες και εργασίες απευθείας από τον προγράμματος περιήγησης τους. Μερικά από τα πιο γνωστά web applications περιλαμβάνουν τα social media, τις υπηρεσίες email, τις υπηρεσίες αποθήκευσης στον νέφος, τις υπηρεσίες βίντεο και μουσικής streaming, και πολλά άλλα.

Η διαθεσιμότητα των web applications σε πολλές συσκευές, όπως κινητά τηλέφωνα και tablet, έχει δώσει τη δυνατότητα στους χρήστες να έχουν πρόσβαση στα δεδομένα τους και να εκτελούν τις

λειτουργίες τους οπουδήποτε και αν βρίσκονται. Επιπλέον, η ανάπτυξη των web applications έχει δώσει τη δυνατότητα στους προγραμματιστές να δημιουργούν λογισμικό που επιτρέπει τη συλλογή και την ανάλυση δεδομένων, καθώς και τη διεξαγωγή διάφορων επιχειρηματικών δραστηριοτήτων, όπως η πώληση προϊόντων και υπηρεσιών Έχουν καταφέρει να κυριαρχήσουν στη σύγχρονη κοινωνία, ανοίγοντας νέες δυνατότητες στους χρήστες και στις επιχειρήσεις. Αυτό οδήγησε σε μια αύξηση της ζήτησης για προγραμματιστές που ειδικεύονται στην ανάπτυξη web applications και για εταιρείες που προσφέρουν web hosting και υπηρεσίες ανάπτυξης λογισμικού.

Μια από τις πιο δημοφιλείς διαδικτυακές εφαρμογές είναι οι ιστοσελίδες. Οι ιστοσελίδες είναι συλλογές αρχείων που παρέχονται από έναν διακομιστή στους πελάτες τους μέσω του Διαδικτύου και περιλαμβάνουν κυρίως κείμενο, εικόνες, ήχο και βίντεο. Οι ιστοσελίδες μπορούν να είναι στατικές, δηλαδή να παρέχουν στους χρήστες σταθερό περιεχόμενο, ή δυναμικές, δηλαδή να παρέχουν στους χρήστες διαφορετικό περιεχόμενο ανάλογα με τις απαιτήσεις τους και τις πληροφορίες που δίνουν στην ιστοσελίδα.

Άλλα παραδείγματα διαδικτυακών εφαρμογών περιλαμβάνουν τα συστήματα διαχείρισης περιεχομένου, τα κοινωνικά δίκτυα, τις εφαρμογές e-mail, τις εφαρμογές τραπεζικών συναλλαγών, τα συστήματα κράτησης ξενοδοχείων και πτήσεων, και τα ηλεκτρονικά καταστήματα. Αυτές οι εφαρμογές μπορούν να παρέχουν στους χρήστες πολλαπλές λειτουργίες και να ανταποκρίνονται σε διαφορετικές ανάγκες και απαιτήσεις.

Η εξατομικευμένη εύρεση γευμάτων βασίζεται σε ένα web application στο οποίο ο χρήστης παρέχει τα δεδομένα στο εμπρόσθιο άκρο της εφαρμογής (front end) και έπειτα επεξεργάζονται και αποθηκεύονται από το πίσω άκρο (back end) και σε βάση δεδομένων αντίστοιχα.

Συνήθως χρησιμοποιούνται γενικά για επικοινωνία με άλλα συστήματα, όπως Βάσεις Δεδομένων, Υπηρεσίες Πληροφορικής και APIs (Διεπαφές Προγραμματισμού Εφαρμογών). Η τελευταία αποτελεί σημαντική τεχνολογία που επιτρέπει στις διαφορετικές εφαρμογές να επικοινωνούν μεταξύ τους και να ανταλλάσσουν δεδομένα. Για παράδειγμα, μπορεί να χρησιμοποιηθεί ένα API για να συνδέσει μια εφαρμογή κοινωνικής δικτύωσης με μια εφαρμογή καταστήματος, ώστε οι χρήστες να μπορούν να αγοράζουν από το κατάστημα μέσα από την εφαρμογή κοινωνικής δικτύωσης.

Προκειμένου να αναπτυχθεί μια εφαρμογή απαιτείται καλή κατανόηση της αρχιτεκτονικής των εφαρμογών και της ασφάλειας τους, καθώς μπορεί να εκτίθενται σε κινδύνους όπως επιθέσεις διακομιστή, κλοπή δεδομένων, και άλλα.

Γενικά, η ανάπτυξη αυτών των εφαρμογών είναι ένα πολύπλοκο θέμα και απαιτεί πολλές γνώσεις στον τομέα της πληροφορικής και της ανάπτυξης λογισμικού. Ωστόσο, με τη βοήθεια της σωστής εκπαίδευσης και της κατάλληλης τεκμηρίωσης, μπορεί να γίνει δυνατόν να αναπτυχθούν υψηλής ποιότητας διαδικτυακές εφαρμογές που μπορούν να παρέχουν ευέλικτες και προσαρμοστικές λύσεις στους χρήστες τους. Οι διαδικτυακές εφαρμογές μπορούν να παρέχουν πολλές διαφορετικές υπηρεσίες, όπως αποθήκευση και κοινή χρήση αρχείων, διαχείριση περιεχομένου, επικοινωνία και συνεργασία, ηλεκτρονικό εμπόριο, και πολλά άλλα.

Η ανάπτυξη συνήθως χρειάζεται ένα ομαδικό περιβάλλον ανάπτυξης λογισμικού, όπου οι προγραμματιστές, οι σχεδιαστές και οι διαχειριστές βάσεων δεδομένων συνεργάζονται μαζί για να δημιουργήσουν μια ολοκληρωμένη εφαρμογή. Η συνεργασία μεταξύ των μελών της ομάδας ανάπτυξης είναι σημαντική για να επιτευχθεί μια επιτυχημένη ανάπτυξη και κυκλοφορία της εφαρμογής.

2.3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ CLIENT-SERVER ΤΩΝ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ

Ένα δημοφιλές μοντέλο αρχιτεκτονικής είναι η αρχιτεκτονική πελάτη-εξυπηρετητή (client-server) [4] στην οποία οι εφαρμογές διαμορφώνονται ως δυο ξεχωριστές διεργασίες, η client και η server, που αλληλεπιδρούν μεταξύ τους μέσω μιας δικτυακής σύνδεσης. Συγκεκριμένα, ο client αναλαμβάνει την εμφάνιση των δεδομένων στον χρήστη μέσω του προγράμματος περιήγησης (browser), ενώ ο server είναι υπεύθυνη για τη διαχείριση και την παροχή των δεδομένων που απαιτούνται από την client.

ο client είναι υπεύθυνος για το να ζητάει υπηρεσίες ή πόρους από τον server. Οι clients είναι συνήθως συσκευές τελικών χρηστών, όπως προσωπικοί υπολογιστές, smartphones ή άλλες συσκευές με δυνατότητα πρόσβασης στο web. Είναι υπεύθυνοι για την παρουσίαση της διεπαφής χρήστη, την επεξεργασία της εισόδου του χρήστη και την ανταλλαγή πληροφοριών με τον server. Ο client στέλνει αιτήματα στον server και λαμβάνει τις κατάλληλες απαντήσεις, οι οποίες μπορεί να περιλαμβάνουν δεδομένα, ενημερώσεις κατάστασης ή μηνύματα σφάλματος. Ο client επεξεργάζεται και εμφανίζει την απάντηση του server στο χρήστη.

Από την άλλη πλευρά, ο server είναι η οντότητα που είναι υπεύθυνη για την παροχή των απαιτούμενων υπηρεσιών και πόρων στον client. Επεξεργάζεται τα εισερχόμενα αιτήματα του client, πραγματοποιεί απαραίτητους υπολογισμούς, ανακτά ή ενημερώνει δεδομένα στη βάση δεδομένων και επιστρέφει τα αποτελέσματα στον client. Οι servers είναι συνήθως ισχυροί υπολογιστές ή συστήματα σε συστάδες, σχεδιασμένα για να χειρίζονται πολλαπλές ταυτόχρονες συνδέσεις και να εκτελούν εργασίες γρήγορα και αποδοτικά.

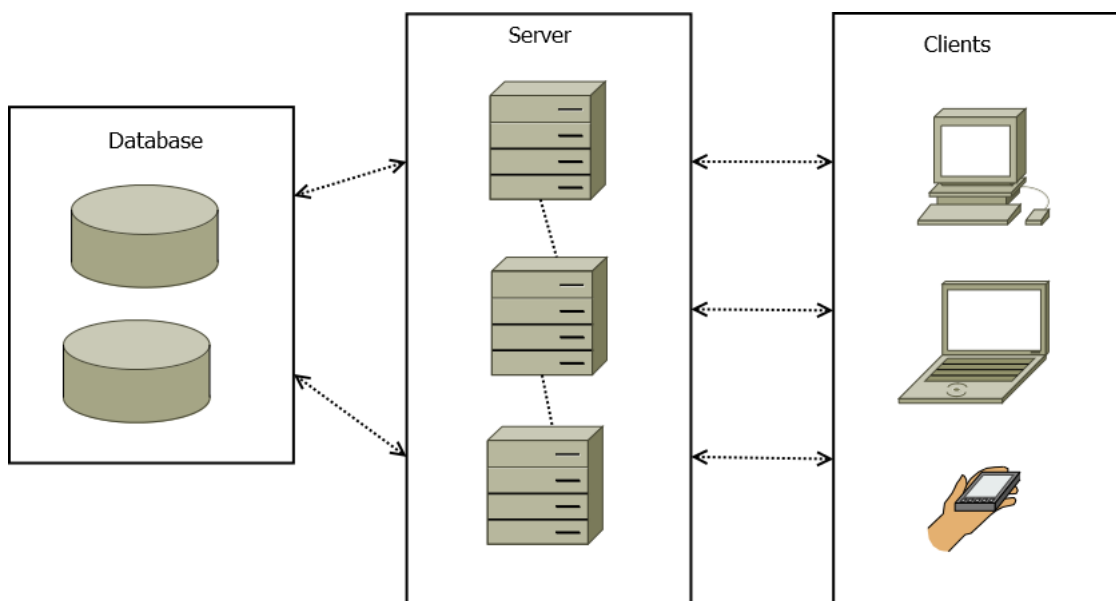
Ο ρόλος της βάσης δεδομένων στην αρχιτεκτονική client-server είναι να αποθηκεύει, να διαχειρίζεται και να παρέχει πρόσβαση στα δεδομένα που χρειάζεται η εφαρμογή. Η βάση δεδομένων συνήθως φιλοξενείται σε έναν server ή σε μια συστάδα server, ξεχωριστά από τον εξυπηρετητή εφαρμογών. Αυτή η διαχωριστικότητα επιτρέπει καλύτερη επεκτασιμότητα, ασφάλεια και απόδοση. Σε αυτό το μοντέλο, ο client δεν αλληλοεπιδρά απευθείας με τη βάση δεδομένων. Αντ' αυτού, ο client στέλνει αιτήματα στον server, ο οποίος ανακτά ή ενημερώνει τα σχετικά δεδομένα στη βάση δεδομένων και στέλνει το αποτέλεσμα πίσω στον client.

Για να διασφαλιστεί μια άψογη και αποδοτική διαδικασία επικοινωνίας μεταξύ clients και servers, η αρχιτεκτονική client-server στηρίζεται συχνά σε πρωτόκολλα που έχουν τυποποιηθεί, όπως το HTTP για web-based εφαρμογές ή το SQL για επικοινωνία βάσεων δεδομένων. Αυτά τα

πρωτόκολλα παρέχουν μια κοινή γλώσσα για την ανταλλαγή πληροφοριών μεταξύ clients και servers, εξασφαλίζοντας τη συνεργασία μεταξύ διαφορετικών συστημάτων.

Το πρωτόκολλο HTTP (Hypertext Transfer Protocol) είναι ένα ευρέως χρησιμοποιούμενο πρωτόκολλο επιπέδου εφαρμογής για τη μετάδοση δεδομένων μεταξύ clients και servers σε ένα δίκτυο, ιδιαίτερα στον Παγκόσμιο Ιστό. Το HTTP βασίζεται σε ένα μοντέλο αιτήματος-απόκρισης, όπου ο client στέλνει ένα αίτημα για έναν πόρο, όπως μια ιστοσελίδα ή ένα αρχείο, σε έναν server. Ο server επεξεργάζεται το αίτημα και στέλνει μια κατάλληλη απόκριση πίσω στον client, συνήθως περιλαμβάνοντας τον απαιτούμενο πόρο ή ένα μήνυμα σφάλματος. Το πρωτόκολλο καθορίζει ένα πρότυπο σύνολο μεθόδων (π.χ. GET, POST, PUT, DELETE) και κωδικούς κατάστασης (π.χ. 200 OK, 404 Not Found) για να διευκολύνει τη σαφή επικοινωνία και τη συνεργασία μεταξύ διαφορετικών συστημάτων. Το HTTP λειτουργεί επάνω στο πρωτόκολλο ελέγχου μετάδοσης (TCP), εξασφαλίζοντας αξιόπιστη και ταξινομημένη μετάδοση δεδομένων. Η ευρεία υιοθέτηση του HTTP οφείλεται κυρίως στην απλότητα, ευκολία χρήσης και συμβατότητά του με διάφορες τεχνολογίες ιστού, καθιστώντας το τη ραχοκοκαλιά της σύγχρονης επικοινωνίας στον Παγκόσμιο Ιστό.

Η ανάπτυξη τόσο του client όσο και του server απαιτεί πολλές τεχνολογίες, όπως HTML, CSS, JavaScript, για το frontend, ενώ για το backend οι προγραμματιστικές γλώσσες όπως η PHP, η Python, Ruby ή javascript ή πλατφόρμες ανάπτυξης όπως το Node.js, το Ruby on Rails και το Django που βοηθούν στην ανάπτυξη διαδικτυακών εφαρμογών με πιο αποδοτικό και συνεπή τρόπο. Σε κάποιες από τις παραπάνω θα γίνει αναφορά στις επόμενες ενότητες καθώς χρησιμοποιήθηκαν εκτενώς για την υλοποίηση της εφαρμογής.



Σχήμα 2.1: Αρχιτεκτονική client-server (χρήση του προγράμματος Dia)

2.4 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΗΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ CLIENT SERVER

Η αρχιτεκτονική client-server προσφέρει αρκετά πλεονεκτήματα και μειονεκτήματα, τα οποία είναι ουσιώδη να λαμβάνονται υπόψη κατά τον σχεδιασμό και την υλοποίηση λογισμικών συστημάτων. Ένα από τα βασικά πλεονεκτήματα αυτής της αρχιτεκτονικής είναι η αποτελεσματική κατανομή εργασιών και πόρων μεταξύ clients και servers, δυνατότητα που επιτρέπει καλύτερη διαχείριση και αξιοποίηση πόρων. Με τη μεταφορά περίπλοκων εργασιών επεξεργασίας στον server, τα client devices μπορούν να είναι λιγότερο ισχυρά και να επικεντρωθούν στην παροχή μιας ομαλής εμπειρίας χρήστη. Αυτός ο διαχωρισμός αρμοδιοτήτων οδηγεί επίσης σε βελτιωμένη συντηρησιμότητα, καθώς ενημερώσεις ή διορθώσεις μπορούν να εφαρμοστούν στον server χωρίς να απαιτούν τροποποιήσεις στην πλευρά του client.

Η επεκτασιμότητα είναι ένα ακόμη σημαντικό πλεονέκτημα του μοντέλου client-server. Καθώς ο αριθμός των clients αυξάνεται, μπορούν να διατίθενται επιπλέον πόροι του server για να ανατιμήσουν το αυξανόμενο φορτίο εργασίας, εξασφαλίζοντας σταθερά επίπεδα απόδοσης. Επιπλέον, η κεντρική φύση του server επιτρέπει καλύτερο έλεγχο της ασφάλειας δεδομένων και εφαρμογών, καθώς η πρόσβαση σε ευαίσθητες πληροφορίες μπορεί να ρυθμιστεί αυστηρά.

Ωστόσο, η αρχιτεκτονική client-server έχει και μειονεκτήματα. Ένα σημαντικό μειονέκτημα είναι η δυνατότητα ενός μοναδικού σημείου αποτυχίας, καθώς ο server παίζει κρίσιμο ρόλο στην παροχή υπηρεσιών και πόρων στους clients. Εάν ο server πέσει ή αντιμετωπίσει προβλήματα απόδοσης, μπορεί να επηρεάσει όλους τους συνδεδεμένους clients, προκαλώντας εκτεταμένη διακοπή υπηρεσιών. Η εφαρμογή μηχανισμών αντιστάθμισης και επαναφοράς λειτουργίας μπορεί να αμβλυνθεί αυτό το πρόβλημα, αλλά αυτό προσθέτει πολυπλοκότητα και κόστος στο σύστημα.

Ένα άλλο μειονέκτημα είναι η αυξημένη εξάρτηση από τη συνδεσιμότητα και το εύρος ζώνης του δικτύου. Οι clients πρέπει να διατηρούν μια σταθερή σύνδεση με τον server για να έχουν πρόσβαση σε πόρους και υπηρεσίες, κάτι που μπορεί να είναι προβληματικό σε περιοχές με κακή υποδομή δικτύου ή κατά τη διάρκεια περιόδων υψηλής κυκλοφορίας δεδομένων στο δίκτυο. Επιπλέον, όσο αυξάνεται ο αριθμός των clients, τόσο αυξάνεται και η ζήτηση για πόρους του server και το εύρος ζώνης του δικτύου, κάτι που μπορεί να οδηγήσει σε σημεία συμφόρησης και πιο αργούς χρόνους απόκρισης.

Τέλος, η αρχιτεκτονική client-server μπορεί να προκαλέσει ανησυχίες όσον αφορά την ιδιωτικότητα και την κυριαρχία των δεδομένων. Καθώς οι clients αποθηκεύουν και επεξεργάζονται τα δεδομένα τους σε απομακρυσμένους servers, γίνονται εξαρτημένοι από παρόχους τρίτων για τη διατήρηση της ακεραιότητας, της ασφάλειας και της ιδιωτικότητας των πληροφοριών τους. Αυτή η εξάρτηση μπορεί να καθιστά δύσκολο τον εξασφαλισμό της προστασίας των δεδομένων, ιδίως όταν ασχολούμαστε με ευαίσθητες ή ρυθμιζόμενες πληροφορίες.

Η επιλογή αυτής της αρχιτεκτονικής πρέπει να βασίζεται στις ανάγκες και τις απαιτήσεις του συγκεκριμένου συστήματος, λαμβάνοντας υπόψη τους περιορισμούς και τις δυνατότητες του δικτύου, των συσκευών και των εφαρμογών.

2.5 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Η μηχανική μάθηση [5][6] είναι ένας τομέας της τεχνητής νοημοσύνης που εξετάζει πώς οι υπολογιστές μπορούν να μάθουν από τα δεδομένα χωρίς να χρειάζεται να τους προγραμματίσουμε εξειδικευμένα για κάθε εργασία. Οι αλγόριθμοι μηχανικής μάθησης μπορούν να αναπτυχθούν για να κατανοήσουν τα πρότυπα και τις σχέσεις μεταξύ των δεδομένων και να κάνουν προβλέψεις ή να λαμβάνουν αποφάσεις. Η ιστορία της μηχανικής μάθησης ξεκινά από τη δεκαετία του 1940, όταν ο Alan Turing εισήγαγε την ιδέα των μηχανών που μπορούν να "μάθουν" από την εμπειρία. Κατά τη διάρκεια των επόμενων δεκαετιών, ερευνητές ανέπτυξαν τις πρώτες μεθόδους μηχανικής μάθησης, όπως τα νευρωνικά δίκτυα και τους αλγόριθμους αναγνώρισης προτύπων. Τα τελευταία χρόνια, η αύξηση της υπολογιστικής ισχύος και η διαθεσιμότητα των μεγάλων συνόλων δεδομένων οδήγησαν σε σημαντικές προόδους στον τομέα, όπως η βαθιά μάθηση και οι αλγόριθμοι ενίσχυσης μάθησης.

Η μηχανική μάθηση μπορεί να χωριστεί σε τρεις βασικές κατηγορίες, την **εποπτευόμενη**, **μη εποπτευόμενη** και **ενισχυτική μάθηση** [7]. Στην Εποπτευόμενη Μάθηση ο αλγόριθμος εκπαιδεύεται χρησιμοποιώντας ετικετοποιημένα δεδομένα, δηλαδή δεδομένα με τις σωστές απαντήσεις. Ο στόχος είναι να δημιουργήσει ένα μοντέλο που μπορεί να κάνει προβλέψεις για νέα δεδομένα με βάση την εκπαίδευση που έλαβε. Στη μη εποπτευόμενη μάθηση, ο αλγόριθμος λαμβάνει δεδομένα χωρίς ετικέτες και προσπαθεί να βρει μια δομή ή σχέσεις μέσα στα δεδομένα. Κοινές τεχνικές περιλαμβάνουν την ομαδοποίηση και τη μείωση της διαστατικότητας. Στην ενισχυτική μάθηση, ο αλγόριθμος μαθαίνει πώς να λάβει αποφάσεις μέσω της αλληλεπίδρασης με το περιβάλλον του. Το μοντέλο προσπαθεί να βελτιστοποιήσει μια συνάρτηση αξίας, λαμβάνοντας υπόψη τις επιβραβεύσεις και τις ποινές που λαμβάνει κατά τη διάρκεια της εκπαίδευσής του. Το αποτέλεσμα είναι ένας πράκτορας που μπορεί να λάβει αποφάσεις με βάση το τρέχον περιβάλλον και τις προηγούμενες εμπειρίες του.

2.6 ΕΚΠΑΙΔΕΥΣΗ ΜΟΝΤΕΛΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Η εκπαίδευση μοντέλων μηχανικής μάθησης είναι η διαδικασία κατά την οποία ένας αλγόριθμος μαθαίνει να προβλέπει, ταξινομεί ή να εξάγει πληροφορίες από τα δεδομένα. Αυτή η διαδικασία περιλαμβάνει μια σειρά από βήματα. Αρχικά την συλλογή δεδομένων. Η εκπαίδευση ενός μοντέλου απαιτεί μια μεγάλη ποσότητα δεδομένων. Τα δεδομένα πρέπει να είναι αντιπροσωπευτικά του προβλήματος που προσπαθούμε να λύσουμε και να περιλαμβάνουν όλες τις πιθανές περιπτώσεις.

Ακολουθεί μια προ επεξεργασία δεδομένων όπου προτού τα δεδομένα χρησιμοποιηθούν για την εκπαίδευση ενός μοντέλου, πρέπει να καθαριστούν, να τεθούν σε κατάλληλη μορφή και να επιλεγούν οι κατάλληλες χαρακτηριστικές μεταβλητές.

Ανάλογα με το πρόβλημα που προσπαθούμε να λύσουμε, πρέπει να επιλέξουμε τον κατάλληλο αλγόριθμο μηχανικής μάθησης. Κάποιοι αλγόριθμοι είναι κατάλληλοι για προβλήματα **παλινδρόμησης**, ενώ άλλοι είναι καλύτεροι για **ταξινόμηση ή συσταδοποίηση**.

Έπειτα η Εκπαίδευση του μοντέλου, όπου κατά τη διάρκεια της εκπαίδευσης, το μοντέλο προσαρμόζει τις παραμέτρους του, ώστε να βελτιώσει την απόδοσή του στα δεδομένα εκπαίδευσης. Αυτό συνήθως γίνεται μέσω μιας διαδικασίας που ονομάζεται βελτιστοποίηση, όπου ο αλγόριθμος προσπαθεί να ελαχιστοποιήσει μια συνάρτηση κόστους που αντιστοιχεί στο σφάλμα του μοντέλου.

Θα πρέπει να αξιολογήσουμε τη γενίκευση του μοντέλου, χρησιμοποιώντας ένα σύνολο δεδομένων επικύρωσης που δεν έχει χρησιμοποιηθεί κατά την εκπαίδευση. Με βάση την απόδοση του μοντέλου σε αυτό το σύνολο, μπορούμε να προσαρμόσουμε τις παραμέτρους του αλγορίθμου, την αρχιτεκτονική του μοντέλου ή τις τεχνικές προ επεξεργασίας δεδομένων.

Στη συνέχεια όταν το μοντέλο έχει προσαρμοστεί και βελτιστοποιηθεί χρησιμοποιούμε ένα τελικό σύνολο δεδομένων, γνωστό ως σύνολο δεδομένων ελέγχου, για να αξιολογήσουμε την απόδοση του μοντέλου σε νέα, αδιάθετα δεδομένα. Η απόδοση σε αυτό το σύνολο δίνει μια εκτίμηση της ικανότητας του μοντέλου να γενικεύει σε νέες καταστάσεις.

Ένα σημαντικό ζήτημα στη μηχανική μάθηση είναι η υπερεκπαίδευση, που συμβαίνει όταν το μοντέλο προσαρμόζεται υπερβολικά στα δεδομένα εκπαίδευσης και χάνει την ικανότητα να γενικεύει σε νέα δεδομένα. Για να αντιμετωπιστεί αυτό το πρόβλημα, μπορούμε να χρησιμοποιήσουμε τεχνικές όπως η πρόωρη διακοπή, η κανονικοποίηση και η επιλογή χαρακτηριστικών.

Στη μηχανική μάθηση, είναι σημαντικό να εκτιμούμε όχι μόνο την απόδοση του μοντέλου αλλά και την αβεβαιότητα των προβλέψεών του. Ορισμένες προσεγγίσεις για την εκτίμηση της αβεβαιότητας περιλαμβάνουν τη χρήση στατιστικών μεθόδων, ή την εφαρμογή προσεγγίσεων με βάση τη θεωρία της Μπεϊζιανής στατιστικής (Bayesian statistics). Η εκτίμηση της αβεβαιότητας μπορεί να βοηθήσει στη λήψη αποφάσεων και την αξιολόγηση των κινδύνων που συνδέονται με τη χρήση των προβλέψεων του μοντέλου στην πράξη.

Τέλος μετά την ανάπτυξη και την εφαρμογή ενός μοντέλου μηχανικής μάθησης, συνεχίζουμε να παρακολουθούμε την απόδοσή του στον χρόνο και να ενημερώνουμε το μοντέλο όταν χρειάζεται. Η συλλογή και η επεξεργασία νέων δεδομένων, η επανεκπαίδευση του μοντέλου και η προσαρμογή των παραμέτρων του αλγορίθμου μπορεί να είναι απαραίτητες για να διατηρηθεί η απόδοση του μοντέλου στα επιθυμητά επίπεδα.

2.7 ΠΡΟ-ΕΚΠΑΙΔΕΥΜΕΝΑ ΜΟΝΤΕΛΑ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Τα προ εκπαιδευμένα μοντέλα (**pretrained models**) μηχανικής μάθησης είναι μοντέλα που έχουν ήδη εκπαιδευτεί σε ένα μεγάλο σύνολο δεδομένων και έχουν μάθει γενικά χαρακτηριστικά που μπορούν να μεταφερθούν σε νέα, σχετικά προβλήματα. Η χρήση προ εκπαιδευμένων μοντέλων μπορεί να εξοικονομήσει πολύ χρόνο και υπολογιστικούς πόρους, καθώς επιτρέπει στους ερευνητές και τους προγραμματιστές να εκμεταλλευτούν τη γνώση που έχει ήδη αποκτηθεί από το μοντέλο.

Τα προ εκπαιδευμένα μοντέλα μπορούν να χρησιμοποιηθούν σε διάφορους τομείς της μηχανικής μάθησης, όπως η επεξεργασία φυσικής γλώσσας (NLP), η ανάλυση εικόνας και η αναγνώριση προτύπων. Μερικά δημοφιλή προ εκπαιδευμένα μοντέλα περιλαμβάνουν τα BERT, GPT [8] και άλλα.

Για να χρησιμοποιηθεί ένα προ εκπαιδευμένο μοντέλο, θα πρέπει να προσαρμοστεί στο συγκεκριμένο πρόβλημα που καλείται κάποιος να λύσει. Αυτό συχνά περιλαμβάνει την προσθήκη και την εκπαίδευση επιθετικών στρωμάτων στο μοντέλο, μια διαδικασία που ονομάζεται μεταφορά μάθησης (transfer learning)[8]. Στη μεταφορά μάθησης, τα βασικά στρώματα του προ εκπαιδευμένου μοντέλου παραμένουν σταθερά, ενώ τα νέα στρώματα προσαρμόζονται στα δεδομένα του συγκεκριμένου προβλήματος.

Η χρήση προ εκπαιδευμένων μοντέλων προσφέρει ορισμένα πλεονεκτήματα. Πρώτον, μειώνει τον χρόνο και τους πόρους που απαιτούνται για την εκπαίδευση ενός μοντέλου από το μηδέν. Δεύτερον, τα προ εκπαιδευμένα μοντέλα έχουν συχνά καλύτερη απόδοση σε σύγκριση με τα μοντέλα που εκπαιδεύονται αποκλειστικά σε ένα μικρό σύνολο δεδομένων, καθώς έχουν μάθει γενικά χαρακτηριστικά από μεγάλα σύνολα δεδομένων. Τρίτον, η μεταφορά μάθησης μπορεί να βελτιώσει τη γενίκευση του μοντέλου, καθώς το προ εκπαιδευμένο μοντέλο έχει ήδη εκπαιδευτεί για να αναγνωρίζει διάφορες πτυχές των δεδομένων.

Ωστόσο, υπάρχουν και ορισμένες προκλήσεις κατά τη χρήση προ εκπαιδευμένων μοντέλων. Αρχικά, τα μοντέλα αυτά μπορεί να είναι πολύπλοκα και να απαιτούν μεγάλη υπολογιστική ισχύ, κάτι που μπορεί να δημιουργήσει προβλήματα στην εφαρμογή και την απόδοση του μοντέλου σε περιορισμένους υπολογιστικούς πόρους. Έπειτα, τα προ εκπαιδευμένα μοντέλα μπορεί να περιλαμβάνουν προκαταλήψεις ή άγνωστα ζητήματα που προέρχονται από τα δεδομένα που χρησιμοποιήθηκαν για την προ εκπαίδευσή τους, και αυτά τα ζητήματα μπορεί να μεταφερθούν στην εφαρμογή του μοντέλου σε νέα προβλήματα.

2.8 ΕΠΕΞΕΡΓΑΣΙΑ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ (NLP)

Όπως αναφέρθηκε, ένας τομέας που μπορεί να χρησιμοποιηθεί η μηχανική μάθηση είναι η Επεξεργασία Φυσικής Γλώσσας (NLP). Η NLP είναι ένα υπό πεδίο της τεχνητής νοημοσύνης και της γλωσσολογίας που επικεντρώνεται στο να επιτρέψει στους υπολογιστές να κατανοήσουν, ερμηνεύσουν και να δημιουργήσουν ανθρώπινη γλώσσα με τρόπο που έχει ταυτόχρονα νόημα και χρησιμότητα. Συνδυάζει υπολογιστικές μεθόδους με γλωσσολογικές γνώσεις για να διευκολύνει την αποτελεσματική επικοινωνία μεταξύ ανθρώπων και μηχανών.

Οι ρίζες της NLP μπορούν να ανιχνευθούν στα 1950s, με την ανάπτυξη των πρώτων υπολογιστικών μοντέλων για την επεξεργασία γλώσσας. Ένα από τα πρώτα πειράματα ήταν το Τεστ Turing, που προτάθηκε από τον Βρετανό μαθηματικό και επιστήμονα των υπολογιστών Alan Turing το 1950 [10]. Το τεστ στόχευε στην αξιολόγηση της ικανότητας ενός μηχανήματος να εκδηλώσει έξυπνη συμπεριφορά που δεν μπορεί να διαφοροποιηθεί από αυτήν του ανθρώπου.

Κατά το 1969, το πεδίο της NLP απέκτησε ώθηση με την ανάπτυξη των πρώτων συστημάτων βασισμένων σε κανόνες, όπως το General Problem Solver (GPS) [12], τα οποία σχεδιάστηκαν για να εκτελούν εργασίες όπως η επίλυση προβλημάτων και η κατανόηση εντολών φυσικής γλώσσας σε ένα περιορισμένο πλαίσιο. Τη δεκαετία που ακολούθησε είδε την εμφάνιση στατιστικών μεθόδων στην NLP, οι οποίες βασίζονταν στην ανάλυση μεγάλων ποσοτήτων δεδομένων για να κάνουν προβλέψεις σχετικά με τη γλώσσα.

Η εισαγωγή του World Wide Web κατά την δεκαετία του 1990 οδήγησε σε μία εκθετική αύξηση της διαθεσιμότητας ψηφιακών κειμένων ως δεδομένων, ενισχύοντας τον αναπτυξιακό ρυθμό των τεχνικών της NLP. Στα πιο πρόσφατα χρόνια, η βαθιά μάθηση και τα νευρωνικά δίκτυα έχουν επαναστατήσει στο πεδίο της NLP, οδηγώντας στην ανάπτυξη προηγμένων μοντέλων γλώσσας όπως τα GPT και BERT [8]. Αυτά τα μοντέλα μπορούν να εκτελούν μια ευρεία γκάμα εργασιών, όπως η ταξινόμηση κειμένου, η ανάλυση απόχρωσης, η μηχανική μετάφραση και η απάντηση σε ερωτήσεις, με υψηλά επίπεδα ακρίβειας.

Η Επεξεργασία Φυσικής Γλώσσας (NLP) [32][33] έχει μια ευρεία γκάμα εφαρμογών, επιτρέποντας στις μηχανές να κατανοούν, να αναλύουν και να παράγουν ανθρώπινη γλώσσα για διάφορα έργα. Μερικές από τις κύριες δυνατότητες και χρήσεις της NLP περιλαμβάνουν:

1. Ταξινόμηση Κειμένου: Η NLP μπορεί να κατατάξει το κείμενο σε διαφορετικές κατηγορίες ή ομάδες με βάση προκαθορισμένα κριτήρια, όπως το θέμα ή το συναίσθημα. Οι εφαρμογές της ταξινόμησης κειμένου περιλαμβάνουν το φιλτράρισμα ανεπιθύμητων, την κατηγοριοποίηση ειδήσεων και την οργάνωση εγγράφων.
2. Ανάλυση Συναισθήματος: Η NLP μπορεί να προσδιορίσει το συναίσθημα ή τη συγκίνηση που εκφράζεται σε ένα κείμενο, όπως θετικό, αρνητικό ή ουδέτερο. Αυτή η τεχνική χρησιμοποιείται συχνά για τη μέτρηση της δημόσιας γνώμης, την παρακολούθηση της φήμης των εταιρειών και την ανάλυση των τάσεων των κοινωνικών μέσων.
3. Μηχανική Μετάφραση: Η NLP μπορεί να μεταφράσει αυτόματα κείμενο από τη μια γλώσσα σε μια άλλη, βοηθώντας στη διευκόλυνση της παγκόσμιας επικοινωνίας. Παραδείγματα συστημάτων μηχανικής μετάφρασης περιλαμβάνουν το Google Translate και το DeepL.
4. Σύνοψη Κειμένου: Η NLP μπορεί να δημιουργήσει συνοπτικές περιλήψεις μεγαλύτερων κειμένων, διατηρώντας τις ουσιαστικές πληροφορίες και το πλαίσιο. Οι τεχνικές σύνοψης κειμένου χρησιμοποιούνται στη συγκέντρωση ειδήσεων, τη σύνοψη ερευνητικών εργασιών και την επιμέλεια περιεχομένου.
5. Απάντηση Ερωτήσεων: Η NLP μπορεί να κατανοήσει και να ανταποκριθεί σε ερωτήσεις που τίθενται σε φυσική γλώσσα, παρέχοντας σχετικές απαντήσεις βάσει των διαθέσιμων πληροφοριών. Αυτή η δυνατότητα χρησιμοποιείται συνήθως σε εικονικούς βοηθούς, chatbots και συστήματα υποστήριξης πελατών.
6. Παραγωγή Κειμένου: Η NLP μπορεί να δημιουργήσει νέο, έγκυρο κείμενο με βάση την εισαγωγή ή το πλαίσιο. Προηγμένα μοντέλα γλώσσας, όπως το GPT, έχουν επιδείξει

εντυπωσιακές δυνατότητες στη δημιουργία συνεκτικού και πλαισιακά σχετικού κειμένου για διάφορους σκοπούς, όπως δημιουργική γραφή, παραγωγή περιεχομένου και εξατομικευμένα μηνύματα.

7. Αναγνώριση Ομιλίας: Η NLP μπορεί να μετατρέψει την προφορική γλώσσα σε γραπτό κείμενο, επιτρέποντας εφαρμογές και υπηρεσίες που ελέγχονται με φωνητικές εντολές, όπως εικονικοί βοηθοί (π.χ., Siri, Alexa και Google Assistant), υπηρεσίες μεταγραφής και συστήματα φωνητικών εντολών.
8. Έλεγχος Ορθογραφίας και Διόρθωση Γραμματικής: Η NLP μπορεί να εντοπίσει και να διορθώσει σφάλματα ορθογραφίας και γραμματικής στο κείμενο, βελτιώνοντας την ποιότητα της γραπτής επικοινωνίας και υποστηρίζοντας εφαρμογές μάθησης γλώσσας.

Η ένταξη της επεξεργασίας φυσικής γλώσσας (NLP) με τη μηχανική μάθηση έχει οδηγήσει σε σημαντικές προόδους στην κατανόηση και ερμηνεία της ανθρώπινης γλώσσας. Συνδυάζοντας τη δύναμη της NLP και των αλγορίθμων μηχανικής μάθησης, μπορούμε να επεξεργαστούμε, να αναλύσουμε και να μάθουμε από τεράστια ποσά κειμενικών δεδομένων, επιτρέποντάς μας να δημιουργήσουμε έξυπνα συστήματα που μπορούν να ταξινομήσουν με ακρίβεια κείμενα, να παράγουν ανθρώπινες απαντήσεις και ακόμη και να προβλέψουν γλωσσικά μοτίβα. Η χρήση εποπτευόμενων, μη εποπτευόμενων και ημι-εποπτευόμενων τεχνικών μάθησης μας επιτρέπει να προσαρμόζουμε τα μοντέλα NLP ανάλογα με τα διαθέσιμα δεδομένα και τις ειδικές απαιτήσεις της εφαρμογής, οδηγώντας σε βελτιωμένες επιδόσεις και ευελιξία. Από την ανάλυση συναισθήματος και την αυτοματοποιημένη υποστήριξη πελατών μέχρι τη μηχανική μετάφραση και τη σύνοψη, η συνεργασία μεταξύ της NLP και της μηχανικής μάθησης επαναστατεί τον τρόπο που αλληλοεπιδρούμε.

2.8 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

Για το κομμάτι της βιβλιογραφικής ανασκόπησης αξιοποιήθηκαν επιστημονικά άρθρα, διαδικτυακές σελίδες αλλά και κυρίως εργασίες μέσω του αποθετηρίου «Πολυνόη». Από το υλικό αυτό πάρθηκαν ιδέες για την υλοποίηση της εφαρμογής τόσο στο κομμάτι του Frontend, Backend όσο και για το μοντέλο μηχανικής μάθησης.

Μια από τις εργασίες που μελετήθηκε ήταν μια διπλωματική εργασία με τίτλο «FeedMeRight-Recipe Recommendation System» [12] η οποία πραγματευόταν τους διάφορους τρόπους και τεχνικές για την σχεδίαση και υλοποίηση συστημάτων συστάσεων καθώς και με την χρήση μηχανικής μάθησης για την επίτευξη αυτών. Πειράματα που έγιναν στα πλαίσια της εργασίας σε μεγάλους όγκους δεδομένων χρησιμοποιώντας τεχνικές όπως βασισμένη σε περιεχόμενο (content based), συνεργατικό φίλτράρισμα (collaborative filtering) και υβριδικό (hybrid). Κατά την χρήση της content based προσέγγισης, καλύτερα αποτελέσματα υπήρχαν όταν γινόταν χρήση περισσότερων ορισμάτων, δηλαδή

υλικών για την παραγωγή υλικών. Στην δεύτερη προσέγγιση, η οποία βασίστηκε κυρίως σε βαθμολογίες έδειξε βελτιώσεις στην ακρίβεια αλλά και στην ακρίβεια ανάκλησης. Την καλύτερη απόδοση είχε η Τρίτη προσέγγιση καθώς συνδύαζε τις δύο προηγούμενες τεχνικές.

Το content based filtering [13] βασίζεται στην ανάλυση των χαρακτηριστικών ή των ιδιοτήτων των αντικειμένων για να κάνει συστάσεις. Προτείνει αντικείμενα που είναι παρόμοια με αυτά που ένας χρήστης έχει ήδη αρέσει ή αλληλοεπιδράσει με το παρελθόν. Το σύστημα εξετάζει τα χαρακτηριστικά των αντικειμένων, όπως είδος, λέξεις-κλειδιά, μεταδεδομένα ή περιγραφές κειμένου και τα συνδυάζει με τις προτιμήσεις των χρηστών. Το collaborative filtering επικεντρώνεται στις προτιμήσεις και τη συμπεριφορά πολλών χρηστών για να κάνει συστάσεις. Αναγνωρίζει τις ομοιότητες μεταξύ χρηστών ή αντικειμένων με βάση τις προηγούμενες αλληλεπιδράσεις ή αξιολογήσεις τους. Ενώ το hybrid συνδυάζει τα πλεονεκτήματα των προσεγγίσεων της βασισμένης σε περιεχόμενο και της συνεργατικής φιλτράρισης. Χρησιμοποιεί τόσο τα χαρακτηριστικά των αντικειμένων όσο και τη συμπεριφορά των χρηστών για να δημιουργήσει συστάσεις.

Το άρθρο με τίτλο "A Recommendation System for Online Courses" [14] εστιάζει στη δημιουργία ενός συστήματος προτάσεων για διαδικτυακά μαθήματα. Ο στόχος είναι να παρέχεται στους χρήστες προτάσεις μαθημάτων που ταιριάζουν με το προφίλ τους και είναι παρόμοια με αυτά που προτίθενται άλλοι χρήστες. Οι συγγραφείς χρησιμοποίησαν συνδυαστικά τις τεχνικές του content-based, του collaborative filtering και του hybrid, με σκοπό να προσφέρουν πιο ακριβείς προτάσεις και να διασφαλίσουν την επιτυχή χρήση του συστήματος από τους χρήστες. Στο άρθρο γίνεται αναφορά σε τεχνικές σχετικά με τα συστήματα προτάσεων, τους αλγορίθμους και τη λειτουργία ενός συστήματος για τη σωστή προτεινόμενη μάθηση. Τα συμπεράσματα της έρευνας δείχνουν ότι οι αλγόριθμοι προτάσεων είναι σχετικά απλοί και εύκολοι στην εφαρμογή τους σε οποιαδήποτε γλώσσα προγραμματισμού. Ωστόσο, αναφέρεται και ένα μειονέκτημα, ότι είναι δυσκολότερο να προταθούν σωστά μαθήματα σε έναν χρήστη που εγγράφεται για πρώτη φορά λόγω έλλειψης πληροφοριών. Το άρθρο αποδεικνύει τη σημασία των αλγορίθμων προτάσεων στην εξατομίκευση της εκπαίδευσης και στη βελτίωση της εμπειρίας των χρηστών μέσω της προσφοράς προσαρμοσμένων μαθημάτων. Η έρευνα ανοίγει τον δρόμο για περαιτέρω βελτιώσεις στα συστήματα προτάσεων για διαδικτυακή εκπαίδευση.

Το άρθρο «Recipe Recommendation With Hierarchical Graph Attention Network» [15] προτείνει τη χρήση γράφων για τη σύσταση συνταγών, με σκοπό την αντιμετώπιση των προκλήσεων που αντιμετωπίζουν οι χρήστες όταν αναζητούν μια ικανοποιητική συνταγή στον υπερβολικό όγκο των διαθέσιμων πόρων συνταγών στο διαδίκτυο. Οι συγγραφείς προτείνουν την κατασκευή ενός γράφου συνταγών με τρεις τύπους κόμβων (συστατικό, συνταγή και χρήστης) και τέσσερις τύπους σχέσεων που τους συνδέουν (συστατικό-συστατικό, συνταγή-συστατικό, συνταγή-συνταγή και χρήστης-συνταγή). Στη συνέχεια, παρουσιάζουν το μοντέλο ΗΓΑΠ, Ιεραρχικό Γράφημα Δικτύου Προσοχής, (HGAT Hierarchical Graph Attention Network), το οποίο χρησιμοποιεί διάφορες μονάδες νευρωνικών

δικτύων για να κωδικοποιήσει το ιστορικό των συνταγών του χρήστη, το περιεχόμενο των συνταγών και τις σχέσεις μεταξύ των κόμβων του γράφου. Τέλος, προτείνουν έναν αλγόριθμο πρόβλεψης σκορ και μια αντικειμενοστραφή συνάρτηση αξιολόγησης για τη βελτιστοποίηση του μοντέλου. Τα κύρια συμπεράσματα του άρθρου είναι πως η χρήση γράφων και η κατανόηση των συνδέσεων μεταξύ χρηστών, συνταγών και συστατικών είναι σημαντική για την κατανόηση των προτιμήσεων των χρηστών σε συνταγές. Το μοντέλο HGAT είναι ικανό να αξιοποιήσει τις πληροφορίες των συνταγών και τις σχέσεις μεταξύ των κόμβων του γράφου για να προτείνει συνταγές που αντιστοιχούν στις προτιμήσεις και την ιστορική συμπεριφορά των χρηστών. Τα πειραματικά αποτελέσματα δείχνουν την υπεροχή του μοντέλου HGAT σε σχέση με άλλες μεθόδους για τη σύσταση συνταγών.

Η εργασία «Σχεδίαση Και Ανάπτυξη Διαδικτυακής Πλατφόρμας Διοργάνωσης Τουρνουά Παιχνιδιών» [16] αφορά την δημιουργία ενός παιχνιδιού τουρνουά με χρήση τεχνολογιών για την ανάπτυξη του frontend και του backend της εφαρμογής. Η μελέτη της εργασίας αυτής ήταν περισσότερο για το κομμάτι του σχεδιασμού της εφαρμογής με χρήση τεχνολογιών όπως React, Node.js καθώς και με επικοινωνία με βάση δεδομένων καθώς παρόμοιες τεχνολογίες και τεχνικές θα εφαρμοστούν για την εφαρμογή της παρούσας εργασίας. Αντίστοιχα, για τον ίδιο σκοπό μελετήθηκε και η εργασία με τίτλο «Διαδικτυακή εφαρμογή χρησιμοποιώντας React και Django» [17]. Στην συγκεκριμένη γίνεται περιγραφή της κατασκευής ενός διαδικτυακού καταστήματος με χρήση τεχνολογιών όπως η React για το κομμάτι του frontend, Django και mongoDB για το backend. Χρήση γίνεται και του Rest Api. Στην εργασία αναλύονται η αρχιτεκτονική και οι τεχνικές που χρησιμοποιήθηκαν για την επίτευξη του στόχου.

ΚΕΦΑΛΑΙΟ 3: ΜΕΘΟΔΟΛΟΓΙΑ

Στο παρόν κεφάλαιο θα εξεταστεί η μεθοδολογία του σχεδιασμού και της ανάπτυξης της εφαρμογής καθώς και οι στόχοι, προβλήματα και τα εμπόδια που προέκυψαν κατά τις όλες φάσεις της εξέλιξής της.

3.1 ΣΤΟΧΟΣ ΚΑΙ ΤΑ ΥΠΟΠΡΟΒΛΗΜΑΤΑ

Στα πλαίσια αυτής της εργασίας σκοπός ήταν η δημιουργία μιας fullstack εφαρμογής με χρήση ενός μοντέλου μηχανικής μάθησης καθώς και της NLP με θέμα έναν εξατομικευμένο ευρετή γευμάτων. Για να επιτευχθεί ο στόχος αυτός πρέπει το πρόβλημα να διασπαστεί σε επιμέρους προβλήματα προκειμένου η επιμέρους επίλυσή τους να οδηγήσει στην επίλυση του τελικού προβλήματος, της δημιουργίας της εφαρμογής.

Όπως Αναφέρθηκε μια full-stack εφαρμογή αποτελείται από το frontend και το backend. Για το frontend έγινε η επιλογή της γλώσσας **JavaScript** με την βιβλιοθήκη **React** ενώ για το back-end η **Node.js** και **Express.js** επίσης της οικογένειας της JavaScript. Οι εφαρμογές που αλληλοεπιδρούν με χρήστες απαιτούν και μια βάση δεδομένων για την αποθήκευση των επιλογών των χρηστών καθώς και τα στοιχεία σύνδεσής τους. Για τον σκοπό αυτόν έγινε χρήση της **Mongodb**. Έπειτα χρειάζεται κάποιο μοντέλο μηχανικής μάθησης ειδικά εκπαιδευμένο πάνω σε συνταγές φαγητών καθώς η εκπαίδευση εκ νέου είναι αρκετά πιο δύσκολο και απαιτεί τεράστιο όγκο δεδομένων όπως αναφέρθηκε στην ενότητα 2.6.

Έτσι διακρίνονται 5 υπό προβλήματα:

1. Την διεπαφή του χρήστη, frontend
2. Την διαχείριση των δεδομένων από και προς τον χρήστη καθώς και την επεξεργασία τους στο back-end.
3. Την αποθήκευση και ανάκτηση των δεδομένων προς και από την βάση δεδομένων ανάλογα τις ενέργειες του χρήστη π.χ σύνδεση, αναζήτηση συνταγής, διαγραφή ιστορικού κτλ.
4. Την εύρεση και χρήση κάποιου μοντέλου μηχανικής μάθησης.
5. Την επικοινωνία όλων των παραπάνω συστημάτων για την αποτελεσματική εμπειρία του χρήστη.

3.2 ΠΡΩΙΜΑ ΣΤΑΔΙΑ

Κατά τα πρώτα στάδια της εφαρμογής υπήρχε η προτίμηση και επιθυμία στην εύρεση ενός προ εκπαιδευμένου μοντέλου μηχανικής μάθησης, για την αναζήτηση συνταγών βάσει των υλικών και προτιμήσεων του χρήστη, με δυνατότητα της ενσωμάτωσή του με χρήση της Node.Js καθώς το κομμάτι του backend ήταν ήδη φτιαγμένο με την χρήση αυτής. Έπειτα δοκιμάστηκαν διάφορες εγκαταστάσεις

πακέτων και βιβλιοθηκών όπως η TensorFlow.js η οποία περιέχει πακέτα μοντέλων μηχανικής μάθησης αντίστοιχων της TensorFlow python, με την χρήση του διαχειριστή πακέτων της Node.js, **Npm** (Node Packet Manager) χωρίς όμως επιτυχία. Αυτό οδήγησε στην ιδέα της σύνδεσης της εφαρμογής γραμμένης σε Node.js με αρχεία γραμμένα σε Python τα οποία θα περιείχαν το μοντέλο μηχανικής μάθησης και θα επικοινωνούσαν με την κύρια εφαρμογή. Έτσι έγινε η επιλογή του μοντέλου **t5-recipe-generator** της **Hugging face** το οποίο βασίζεται στο ευρύτερο **t5-base** μοντέλο.

3.3 ΕΜΠΟΔΙΑ ΚΑΤΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ

Κατά την υλοποίηση της εφαρμογής παρουσιάστηκαν διάφορα προβλήματα που έχρηζαν επίλυσης προκυμμένου να προχωρήσει στάδιο η εφαρμογή. Ένα από αυτά ήταν πώς το σύστημα θα διατηρεί μια συνεδρία σύνδεσης ενός χρήστη σε όλην την διάρκεια μέχρι να επιλεγεί εσκεμμένα να αποσυνδεθεί.

Για την αντιμετώπιση του προβλήματος υπάρχουν ποικίλα λύσεις. Ωστόσο καθώς ο στόχος της εργασίας δεν είναι η βέλτιστη χρήση τεχνολογιών για την υλοποίηση μιας διαδικτυακής εφαρμογής αλλά περισσότερο η χρήση μηχανικής μάθησης σε σύγχρονες εφαρμογές, επιλέχθηκε η χρήση της **localStorage** της JavaScript. Η **localStorage** αποτελεί γρήγορη και αποτελεσματική λύση για αποθήκευση δεδομένων σε μορφή string στον διακομιστή. Η αποθήκευση επιτυγχάνεται με την χρήση των **setItem** δίνοντας ένα αναγνωριστικό κλειδί σαν παράμετρο ακολουθούμενο από την μεταβλητή που επιθυμούμε να αποθηκεύσουμε. Το κλειδί χρησιμεύει για την ανάκτηση του δεδομένου μέσω της **getItem**. Έτσι αποθηκεύοντας το email αλλά και τον κωδικό του χρήστη κατά την σύνδεσή του στον διακομιστή είναι δυνατή η διατήρηση μιας σύνδεσης. Άλλα δεδομένα όπως οι εγγραφές κάποιου χρήστη ή οι επιλογές του αποθηκεύονται στην βάση δεδομένων.

Παράλληλα με το 1^ο πρόβλημα πρόκυπτε και το θέμα της προσβασιμότητας, δηλαδή πώς θα απαγορευόταν η πρόσβαση στους χρήστες που δεν έχουν πραγματοποιήσει σύνδεση ή εγγραφή. Λύση δίνει η βιβλιοθήκη της React, **React Router DOM**. Η συγκεκριμένη είναι μια δημοφιλής βιβλιοθήκη για τη διαχείριση περιήγησης στην πλευρά του πελάτη. Επιλύει αρκετά προβλήματα σχετικά με τη διαχείριση και οργάνωση της περιήγησης εντός μιας Single Page Application (SPA).

Η React Router παρέχει μεθόδους για **protected routes**, όπου routes θεωρούμε κάθε σελίδα πχ σελίδα σύνδεσης, σελίδα εγγραφής, κύρια σελίδα κα., και protected αυτές που για να έχουμε πρόσβαση χρειάζεται κάποια κατάσταση (state) να είναι αληθής. Η κατάσταση αυτή θα μπορούσε να είναι αν υπάρχει αποθηκευμένο στο localStorage η πληροφορία ότι ο εκάστοτε χρήστης είναι συνδεδεμένος ή αν τα στοιχεία του χρήστη υπάρχουν καταχωρημένα στην βάση δεδομένων.

ΚΕΦΑΛΑΙΟ 4: ΤΟ ΣΥΣΤΗΜΑ ΜΟΥ

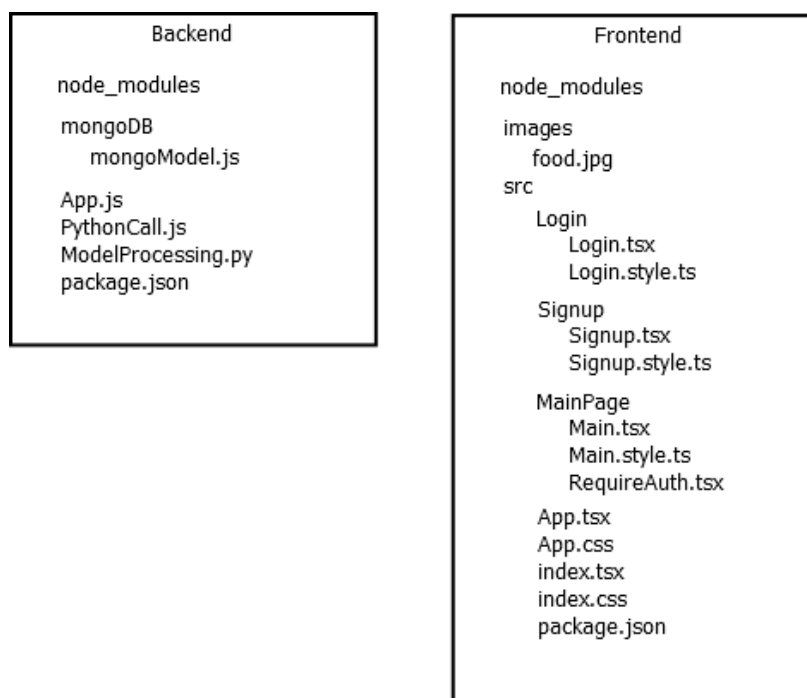
Η ανάπτυξη ενός διαδικτυακού συστήματος ή εφαρμογής μπορεί να αποδειχθεί περίπλοκη και μακροσκελή διαδικασία καθώς αποτελείται από πολλές διαφορετικές διαδικασίες η κάθε μια η οποία μπορεί να απαιτεί διαφορετικές τεχνολογίες για την επίτευξή της. Για την διευκόλυνση την ανάπτυξης, συνηθίζεται η επιλογή τεχνολογιών και βιβλιοθηκών με ευρύ κοινό και υποστήριξη ώστε να υπάρχουν όσο το δυνατόν λιγότερα προβλήματα. Ακολουθούνται, επίσης, τεχνικές και μεθοδολογίες για την βελτίωση της κατανόησης και συγγραφής του κώδικα, όπως μια σωστή δομή των αρχείων της εφαρμογής.

4.1 ΔΟΜΗ ΑΡΧΕΙΩΝ ΕΦΑΡΜΟΓΗΣ

Για την δημιουργία της εφαρμογής, το project, χωρίστηκε σε 2 φακέλους, ένα για το Backend και ένα για το Frontend. Αρχικά, σε κάθε εφαρμογή δημιουργήθηκε ένα **package.json** αρχείο περιέχοντας τις βιβλιοθήκες σε επιθυμητές εκδόσεις. Αυτός ο φάκελος μπορεί να δημιουργηθεί αυτόματα με την εντολή **npm init** περιέχοντας τις απολύτως βασικές βιβλιοθήκες, όπως της React, ωστόσο προσθέτουμε και τις δικές μας ανάλογα με τι χρειάζεται να υλοποιηθεί. Έπειτα με την εντολή **npm install** δημιουργεί ένα φάκελο **node_modules** ο οποίος περιέχει τις εγκαταστημένες βιβλιοθήκες ακολουθώντας τις εκδόσεις που έχουν δηλωθεί στο package.json.

Στον φάκελο Backend υπάρχει το αρχείο App.js το οποίο περιέχει την κύρια λογική του server καθώς και το αρχείο **PythonCall.js** το οποίο αναλαμβάνει την επικοινωνία μεταξύ του server και του μοντέλου γραμμένο σε γλώσσα Python. Έπειτα στο **ModelProcessing.py** φιλοξενείται το μοντέλο μηχανικής μάθησης μαζί με τις όλες διεργασίες της NLP και της απάντησης προς το PythonCall.js. Τέλος στο **MongoModel.js** περιλαμβάνονται όλα τα απαραίτητα για την σύνδεση με την βάση δεδομένων της MongoDB.

Στον φάκελο Frontend υπάρχει ο υπό φάκελος **src** ο οποίος περιλαμβάνει όλες τις κύριες σελίδες της εφαρμογής καθώς και ο φάκελος **images** για τις εικόνες. Στον src ξεχωρίζουν τα αρχεία **index.tsx**, με τα styles του (**index.css**), και **App.tsx** με τα styles του (**App.css**). Ακολουθούν οι φάκελοι για τις σελίδες **Login**, **Signup** και **MainPage**. Στους πρώτους δύο περιλαμβάνεται η σελίδα για την σύνδεση του χρήστη και για την εγγραφή του μαζί με τις δηλώσεις των στιλιστικών αντίστοιχα, **Login.tsx** και **Login.css**, **Signup.style.ts** και **Signup.style.ts**. Στον MainPage φάκελο υπάρχουν τα αρχεία **MainPage.tsx** και **MainPage.style.ts**, απαραίτητα για την κύρια σελίδα της εφαρμογής αφού ο χρήστης συνδεθεί επιτυχώς. Τέλος στον ίδιο υπό φάκελο βρίσκεται και το **RequireAuth.tsx** απαραίτητο για την επίτευξη της εισόδου του χρήστη στην εφαρμογή.



Σχήμα 4.1: Δομή αρχείων εφαρμογής

4.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΦΑΡΜΟΓΗΣ

Για την δημιουργία της εφαρμογής του προσωποποιημένου ευρετή γευμάτων χρησιμοποιήθηκε για το κομμάτι του Frontend **React** σε **TypeScript**, **MaterialUI** για την χρήση βασικών έτοιμων συνιστωσών (components) της εφαρμογής όπως πεδία κειμένου (textfields) για την εισαγωγή κειμένου, καθώς και **Styled Components** για την επεξεργασία των στυλιστικών της εφαρμογής. Για την περιήγηση στις σελίδες της εφαρμογής έγινε χρήση της βιβλιοθήκης **React Router Dom** και για τις κλήσεις στα endpoints προς τον server χρησιμοποιήθηκε το **Axios**. Ενώ για το Backend χρειάστηκε μόνο η **Mongoose**. Στο κομμάτι της Python προκυμμένου να λειτουργήσει το μοντέλο χρειάστηκε να εγκατασταθεί ένα περιβάλλον **Conda** το οποίο επιτρέπει την εγκατάσταση βιβλιοθηκών python μέσω του pip. Συγκεκριμένα για το κομμάτι του NLP χρησιμοποιήθηκαν οι βιβλιοθήκες **AutoTokenizer** και **FlaxAutoModelForSeq2SeqLM** της **Transformers** ενώ για το μοντέλο μηχανικής μάθησης t5 χρειάστηκε να εγκατασταθεί η PyTorch.

4.2.1 Η ΒΙΒΛΙΟΘΗΚΗ REACT

Η React [18] είναι μια δημοφιλής JavaScript βιβλιοθήκη για τη δημιουργία περιβάλλοντα χρήστη (User Interfaces) σε ιστοσελίδες και εφαρμογές. Αναπτύχθηκε αρχικά από το Facebook και είναι ανοιχτού κώδικα, που σημαίνει ότι ο κώδικας είναι διαθέσιμος για το κοινό και μπορεί να τον χρησιμοποιήσει και να τροποποιήσει. Η React χρησιμοποιεί την έννοια των components, τα οποία είναι ανεξάρτητα κομμάτια κώδικα που συνδυάζονται για να δημιουργήσουν μια εφαρμογή. Αυτό επιτρέπει

στους προγραμματιστές να δημιουργήσουν επαναχρησιμοποιήσιμα κομμάτια κώδικα, κάνοντας τον σχεδιασμό και τη συντήρηση των εφαρμογών πιο αποτελεσματικό και οργανωμένο.

4.2.2 Η ΒΙΒΛΙΟΘΗΚΗ TYPESCRIPT

Η TypeScript [19] είναι μια γλώσσα προγραμματισμού ανοιχτού κώδικα που αναπτύχθηκε από τη Microsoft. Είναι ένα υπερ-σύνολο της JavaScript, που σημαίνει ότι η TypeScript περιλαμβάνει όλες τις δυνατότητες της JavaScript και προσθέτει επιπλέον χαρακτηριστικά. Ένα από τα βασικά χαρακτηριστικά της TypeScript είναι η υποστήριξη δηλώσεων τύπων όπως string, number, boolean, κάτι που η JavaScript δεν υποστηρίζει από μόνης της. Ο καθορισμός τύπων μπορεί να εφαρμοστεί από τους προγραμματιστές στις μεταβλητές, τις παραμέτρους και τις επιστροφές των συναρτήσεων. Αυτό βοηθά στην ανίχνευση σφαλμάτων κατά τη διάρκεια του προγραμματισμού, αντί να περιμένουμε να εκτελεστεί ο κώδικας για να τα ανακαλύψουμε. Η TypeScript επίσης βελτιώνει την παραγωγικότητα και τη συντήρηση του κώδικα, καθώς τα εργαλεία που χρησιμοποιούν την TypeScript μπορούν να παρέχουν καλύτερες συμβουλές και αυτόματη συμπλήρωση κώδικα.

4.2.3 Η ΒΙΒΛΙΟΘΗΚΗ MATERIALUI

Το Material-UI [20] είναι μια δημοφιλής βιβλιοθήκη για τη δημιουργία User Interfaces στην React. Βασίζεται στις αρχές του Material Design, ενός ολοκληρωμένου συστήματος κατευθυντήριων γραφικών και αλληλεπιδράσεων που αναπτύχθηκε από την Google. Η βιβλιοθήκη Material-UI παρέχει μια σειρά από προκαθορισμένα στοιχεία UI και components, όπως κουμπιά, μενού, καρτέλες, εικονίδια, και πολλά άλλα. Αυτά τα στοιχεία μπορούν να προσαρμοστούν και να ενσωματωθούν στις React εφαρμογές, για να κατασκευάσουν γρήγορα και εύκολα υψηλής ποιότητας UI.

4.2.4 Η ΒΙΒΛΙΟΘΗΚΗ STYLED COMPONENTS

Τα Styled Components [21] είναι μια βιβλιοθήκη για την React, η οποία επιτρέπει την καλύτερη διαχείριση και οργάνωση των CSS (Cascading Style Sheets) μέσω της χρήσης JavaScript. Τα Styled Components επιτρέπουν την δημιουργία στυλιστικών προσαρμοσμένων στοιχείων UI, που έχουν τα CSS τους ενσωματωμένα ευθέως στον κώδικά τους. Η βασική ιδέα πίσω από τα Styled Components είναι η δημιουργία JavaScript components που περιλαμβάνουν τα στυλ που αφορούν συγκεκριμένα αυτά τα στοιχεία. Αυτό επιτρέπει την απομάκρυνση των στυλιστικών από τα εξωτερικά CSS αρχεία και την καλύτερη ενσωμάτωσή τους στον κώδικα της εφαρμογής. Κάθε στοιχείο UI που δημιουργείται με Styled Components έχει τα δικά του στυλ, τα οποία είναι απομονωμένα από τα υπόλοιπα στοιχεία της εφαρμογής, εξασφαλίζοντας έτσι τη συνέπεια και τη μη επικάλυψη στυλ.

4.2.5 Η ΒΙΒΛΙΟΘΗΚΗ REACT ROUTER DOM

Η React Router DOM [22] είναι μια βιβλιοθήκη για τη δημιουργία και τη διαχείριση της πλοήγησης σε εφαρμογές React για το web. Συνδυάζεται με τη βασική βιβλιοθήκη React Router για να παρέχει εργαλεία και συνιστώσες που επιτρέπουν τη δημιουργία μιας ομαλής και δυναμικής πλοήγησης στις εφαρμογές. Παρέχει components όπως το <BrowserRouter>, <Route>, <Switch> και <Link>, που επιτρέπουν τη δημιουργία πλοήγησης μεταξύ διαφορετικών σελίδων, συνιστωσών και διαδρομών (routes) μέσα στην εφαρμογή React. Οι διαδρομές (routes) που χρησιμοποιούνται στο React Router DOM είναι βασισμένες στην ιεραρχία των συνιστωσών και μπορούν να είναι δυναμικές, προσφέροντας μεγάλη ευελιξία κατά τη διαμόρφωση της πλοήγησης. Η βιβλιοθήκη αυτή είναι πολύ χρήσιμη σε Single-Page Applications (SPA), όπου όλη η πληροφορία φορτώνεται δυναμικά χωρίς την ανανέωση της ιστοσελίδας.

4.2.6 Η ΒΙΒΛΙΟΘΗΚΗ MONGOOSE

Η Mongoose [23] είναι μια βιβλιοθήκη της Node.js που παρέχει μια στρωματοποίηση (abstraction layer) για την επικοινωνία και την εργασία με τη βάση δεδομένων MongoDB. Η Mongoose απλοποιεί τη διαδικασία σύνδεσης, επεξεργασίας και επικοινωνίας με τη βάση δεδομένων μέσω της χρήσης σχημάτων και μοντέλων. Τα σχήματα (schemas), στο Mongoose, καθορίζουν τη δομή των εγγράφων που θα αποθηκευτούν στη βάση δεδομένων. Μπορούν να περιλαμβάνουν τους τύπους των πεδίων, τους περιορισμούς, τις προεπιλεγμένες τιμές, τις επικυρώσεις (validation) και άλλες πληροφορίες που σχετίζονται με τα δεδομένα. Τα μοντέλα (models) από την άλλη, είναι αντικείμενα που παρέχονται από το Mongoose και συνδέονται με ένα συγκεκριμένο σχήμα. Τα μοντέλα επιτρέπουν τη δημιουργία, την αναζήτηση, την ενημέρωση και τη διαγραφή εγγράφων στη βάση δεδομένων, καθώς και την εκτέλεση άλλων λειτουργιών. Επιτρέπουν επίσης τη δημιουργία μεθόδων που μπορούν να εφαρμοστούν στα αντικείμενα ενός μοντέλου, παρέχοντας έτσι προσαρμοσμένη λειτουργικότητα και επικοινωνία με τη βάση δεδομένων.

4.2.7 Η ΒΙΒΛΙΟΘΗΚΗ AXIOS

Το Axios [24] είναι μια βιβλιοθήκη της JavaScript που χρησιμοποιείται για την εκτέλεση HTTP αιτημάτων από τον πελάτη (client) προς τον διακομιστή (server) και αντίστροφα. Η βιβλιοθήκη αυτή παρέχει μια εύκολη στη χρήση και ευέλικτη API για τη διαχείριση των αιτημάτων, όπως GET, POST, PUT, DELETE και άλλα.

4.2.8 Η ΒΙΒΛΙΟΘΗΚΗ TRANSFORMERS

Τα Transformers [25] είναι ένα οικοσύστημα που προσφέρεται από την Hugging Face, η οποία είναι μία από τις πιο δημοφιλείς βιβλιοθήκες για εργασίες σχετικές με την Τεχνητή Νοημοσύνη και τη Μηχανική Μάθηση. Το οικοσύστημα Transformers περιλαμβάνει προ-εκπαιδευμένα μοντέλα, τα οποία μπορούν να χρησιμοποιηθούν για διάφορες εργασίες στον τομέα της φυσικής γλώσσας, όπως αναγνώριση ομιλίας, μετάφραση, σύνταξη κειμένου, κατανόηση γλώσσας κ.λπ. Οι Transformers

βασίζονται στην αρχιτεκτονική Transformer [8], η οποία έχει γίνει πολύ δημοφιλής λόγω της αποδοτικότητάς της και της ικανότητάς της να επιτυγχάνει υψηλή ακρίβεια σε διάφορες εργασίες. Το GPT-3 και το BERT είναι δύο παραδείγματα προ-εκπαιδευμένων μοντέλων που βασίζονται σε αυτή την αρχιτεκτονική και είναι διαθέσιμα μέσω της βιβλιοθήκης Transformers της Hugging Face.

Η αρχιτεκτονική Transformer εισήχθη για πρώτη φορά το 2017 από τους Vaswani και συνεργάτες στο άρθρο "Attention Is All You Need" [43]. Το Transformer είναι μια αρχιτεκτονική νευρωνικού δικτύου βαθιάς μάθησης, που σχεδιάστηκε για να αντιμετωπίσει προβλήματα φυσικής γλώσσας, όπως μετάφραση, συνεπαγωγή κειμένου, αναπαράσταση κειμένου και πολλά άλλα.

Η βασική ιδέα πίσω από την αρχιτεκτονική Transformer είναι η χρήση του μηχανισμού "self-attention", που επιτρέπει στο μοντέλο να αναγνωρίζει τις σχέσεις και τις αλληλεπιδράσεις μεταξύ των λέξεων σε μια πρόταση. Αυτός ο μηχανισμός προσφέρει μια πιο ευέλικτη και παράλληλη προσέγγιση στην επεξεργασία κειμένου σε σύγκριση με τις παραδοσιακές αρχιτεκτονικές, όπως τα αναδρομικά νευρωνικά δίκτυα (RNNs) και τα LSTM.

Τα RNNs είναι μια κατηγορία νευρωνικών δικτύων που έχουν εσωτερική μνήμη (ή κρυφή κατάσταση) και είναι σε θέση να επεξεργάζονται ακολουθίες δεδομένων με διαδοχικά βήματα. Τα RNNs χρησιμοποιούν την κρυφή κατάσταση για να αποθηκεύουν πληροφορίες από προηγούμενα βήματα και να επηρεάσουν τις προβλέψεις στα επόμενα βήματα.

Τα LSTM είναι μια παραλλαγή των RNNs, που εισήχθη για πρώτη φορά από τους Hochreiter και Schmidhuber το 1997 [54]. Τα LSTM έχουν σχεδιαστεί για να αντιμετωπίζουν το πρόβλημα της εξαφάνισης της κλίσης (vanishing gradient) που συναντάται στα τυπικά RNNs. Αυτό το πρόβλημα σχετίζεται με τη δυσκολία των RNNs στην εκμάθηση μακροπρόθεσμων εξαρτήσεων μεταξύ των στοιχείων της ακολουθίας. Τα LSTM διαθέτουν μια πιο περίπλοκη εσωτερική δομή σε σύγκριση με τα τυπικά RNNs, που τους επιτρέπει να αποθηκεύουν, να ενημερώνουν και να ανακτούν πληροφορίες με μεγαλύτερη ακρίβεια κατά τη διάρκεια της επεξεργασίας ακολουθιών. Η κύρια ιδέα πίσω από τα LSTM είναι η εισαγωγή της λεγόμενης "κυψέλης μνήμης" (memory cell), η οποία είναι σε θέση να αποθηκεύει πληροφορίες για μεγάλα διαστήματα χρόνου με ελάχιστες παρεμβολές.

4.2.9 Η ΒΙΒΛΙΟΘΗΚΗ AUTOTOKENIZER

Η AutoTokenizer [26] παρέχεται από την βιβλιοθήκη Transformers της Hugging Face είναι μια κλάση που διευκολύνει τη δημιουργία tokenizer, ανεξάρτητα από το είδος του προ-εκπαιδευμένου μοντέλου που χρησιμοποιείται. Οι tokenizers είναι υπεύθυνοι για τη μετατροπή των ακατέργαστων κειμένων σε δομές δεδομένων που μπορούν να κατανοηθούν από τα μοντέλα Transformers, όπως τα BERT, GPT-2, RoBERTa και άλλα. Η AutoTokenizer παρέχει μια απλή διεπαφή για την αυτόματη φόρτωση του κατάλληλου tokenizer ανάλογα με το όνομα του μοντέλου που προσδιορίζεται. Αυτό σημαίνει ότι δεν χρειάζεται να γνωρίζετε ποιος tokenizer αντιστοιχεί σε κάθε μοντέλο, καθώς η AutoTokenizer θα το επιλέξει αυτόματα

4.2.10 Η ΒΙΒΛΙΟΘΗΚΗ FLAXAUTOMODELFORSEQ2SEQLM

Η FlaxAutoModelForSeq2SeqLM [27] από τη βιβλιοθήκη Transformers της Hugging Face είναι μια βοηθητική κλάση που διευκολύνει τη δημιουργία και φόρτωση μοντέλων Seq2Seq (Sequence-to-Sequence) σε περιβάλλον JAX/Flax. Τα Seq2Seq (Sequence-to-Sequence) αναφέρεται σε μια κατηγορία αλγορίθμων και μοντέλων μηχανικής μάθησης που λειτουργούν μετατρέποντας μια ακολουθία εισόδου σε μια ακολουθία εξόδου. Τα Seq2Seq μοντέλα είναι πολύ δημοφιλή σε εφαρμογές φυσικής γλώσσας, όπως μετάφραση γλώσσας, περίληψη κειμένου, δημιουργία κειμένου, απάντηση ερωτήσεων και άλλα. Ενώ το Flax είναι μια βιβλιοθήκη ανοιχτού κώδικα που σχεδιάστηκε για να κάνει την ανάπτυξη και την εκπαίδευση νευρωνικών δικτύων ευκολότερη και πιο αποτελεσματική στην πλατφόρμα JAX. Το JAX είναι ένα πακέτο της Google για την υλοποίηση και βελτιστοποίηση αριθμητικών αλγορίθμων, το οποίο παρέχει αυτόματα διαφορική και προσεκτική βελτιστοποίηση των υπολογιστικών πόρων για μεγάλη κλίμακα προβλήματα μηχανικής μάθησης.

4.2.11 Η ΒΙΒΛΙΟΘΗΚΗ PYTORCH

Η PyTorch [28] είναι μια βιβλιοθήκη μηχανικής μάθησης ανοιχτού κώδικα που αναπτύχθηκε από την Facebook's AI Research Lab (FAIR) και είναι ειδικά σχεδιασμένη για εφαρμογές βαθιάς μάθησης, όπως τα νευρωνικά δίκτυα. Χαρακτηρίζεται από την ευελιξία και την απόδοση που παρέχει στους ερευνητές και τους προγραμματιστές. Ένα από τα βασικά χαρακτηριστικά της PyTorch είναι το δυναμικό υπολογιστικό γράφημα, το οποίο επιτρέπει την εκτέλεση και την τροποποίηση των νευρωνικών δικτύων κατά τη διάρκεια της εκπαίδευσης.

TypeScript	4.9.5
React	18.2.0
MaterialUI	5.11.10
Axios	1.3.4
Styled Components	5.3.6
React Router Dom	6.8.1
Mongoose	6.10.0
Transformers	4.27.3
PyTorch	1.12.0

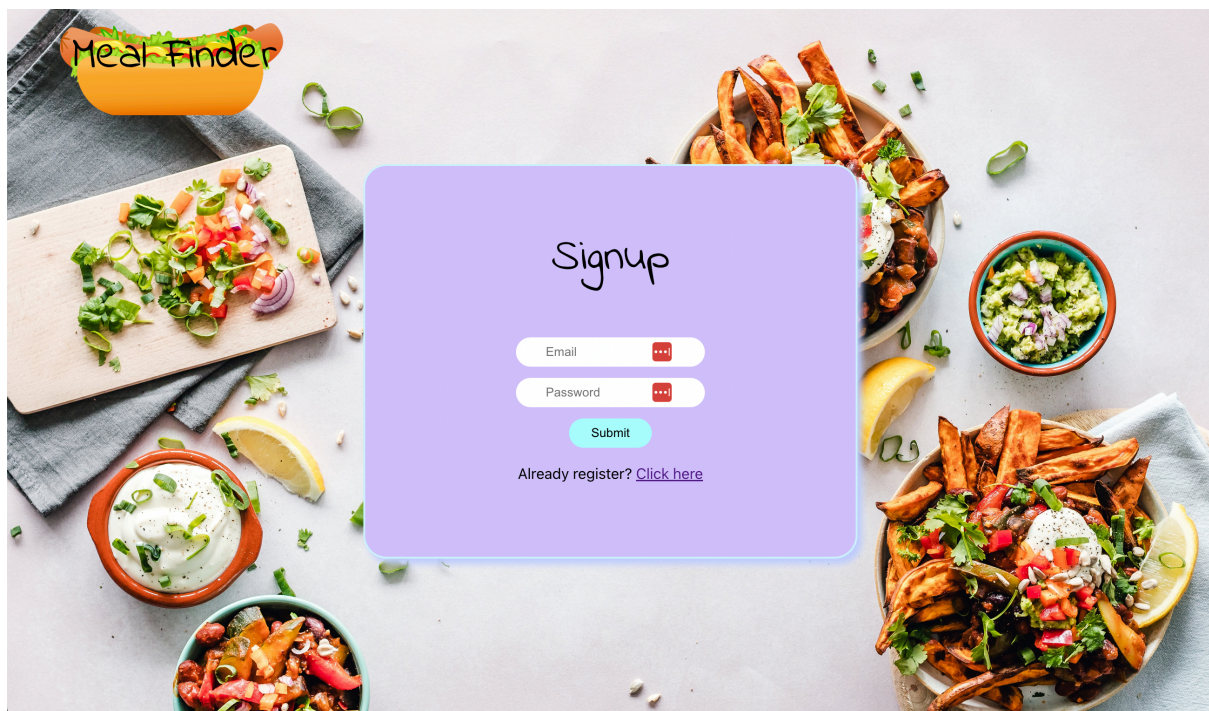
Πίνακας 4.1: Εκδόσεις βιβλιοθηκών εφαρμογής

4.3 ΕΜΠΡΟΣΘΙΟ ΑΚΡΟ (FRONTEND)

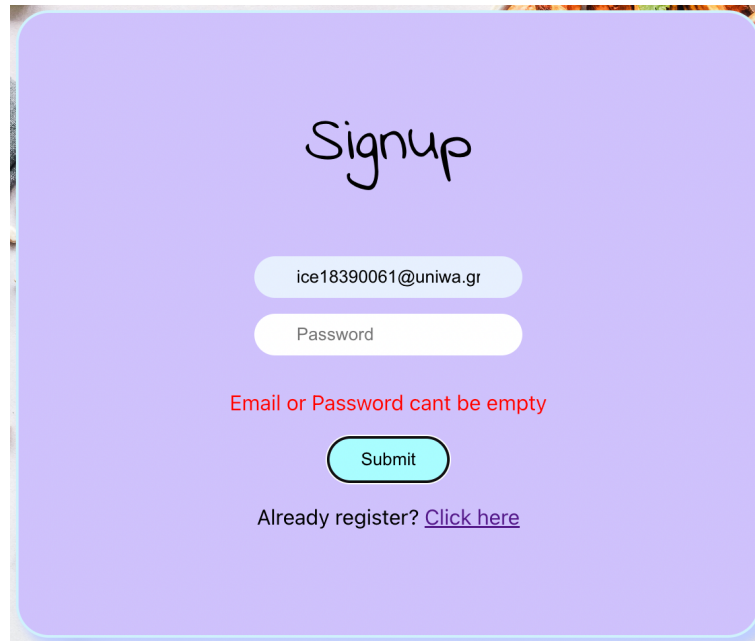
Για το Frontend κομμάτι της εφαρμογής χωρίστηκε σε τρεις κύριες σελίδες, η σελίδα εγγραφής, η σελίδα σύνδεσης και η κύρια σελίδα. Αρχικά για να επιτευχθεί η σελίδα της εγγραφής χρησιμοποιήθηκε μια απλή φόρμα για την εισαγωγή των στοιχείων του χρήστη. Για να ξεκινήσει το Frontend κομμάτι της εφαρμογής αρκεί να τρέξει η εντολή **npm start** και η εφαρμογή θα αρχίσει να λειτουργεί. Αυτό αναλογεί μόνο στο Frontend κομμάτι καθώς για το Backend υπάρχει άλλη αντίστοιχη εντολή.

4.3.1 ΣΕΛΙΔΑ ΕΓΓΡΑΦΗΣ

Για την δημιουργία της έγινε χρήση 3 components, δύο textFields και ένα button. Αφού ο χρήστης εισάγει στοιχεία τόσο στο πρώτο πεδίο όσο και στο δεύτερο, τότε μπορεί να πατήσει το κουμπί “submit” και τα δεδομένα θα εγγραφούν στην βάση δεδομένων αφού πρώτα σταλούν στο Backend. Σε περίπτωση που ο χρήστης παραλείψει κάποιο πεδίο, θα ειδοποιηθεί με ένα κόκκινο μήνυμα και η επιλογή του κουμπιού δεν θα έχει κάποιο αποτέλεσμα. Η σελίδα αποτελείται από τον κύριο τίτλο της εφαρμογής πάνω αριστερά και ένα πλαίσιο με τα κύρια components στο κέντρο μαζί με τον τίτλο της σελίδας. Για τα παραπάνω στιλιστικά όπως χρώματα, τοποθετήσεις των συνιστωσών και τα μεγέθη, έχουν υλοποιηθεί με την χρήση Styled Components



Εικόνα 4.1: Η σελίδα εγγραφής χρηστών

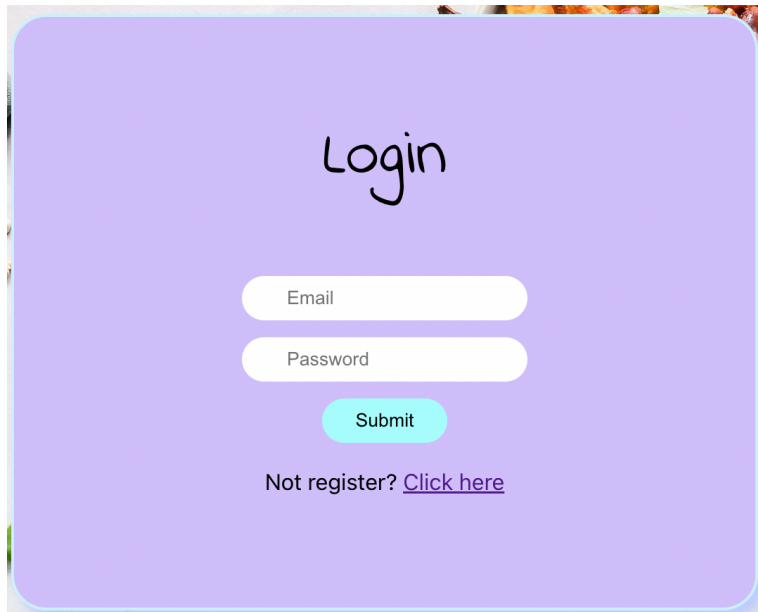


Εικόνα 4.2: Εμφάνιση μηνύματος ειδοποίησης προς τον χρήστη

Δίνεται επίσης η επιλογή στον χρήστη να μεταβεί στην σελίδα σύνδεσης σε περίπτωση που έχει ήδη εγγραφεί.

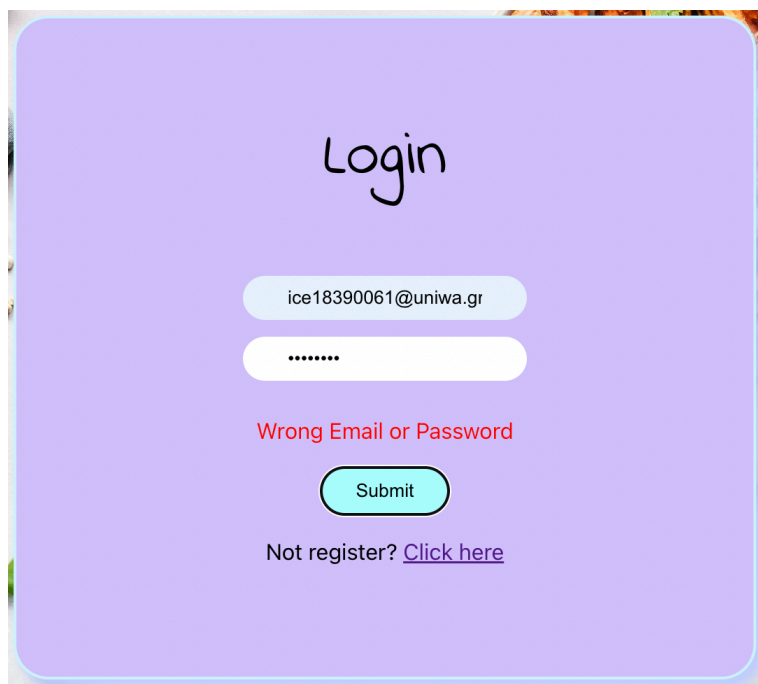
4.3.2 ΣΕΛΙΔΑ ΣΥΝΔΕΣΗΣ

Η σελίδα Σύνδεσης μοιάζει αρκετά με την σελίδα Εγγραφής με τις μόνες διαφορές, στην εμφάνιση, στον τίτλο. Ενώ στο κομμάτι της επικοινωνίας με το Backend, θα στείλει τα δεδομένα, Email και κωδικός του χρήστη, ώστε αυτό με την σειρά του να ελέγξει αν τα στοιχεία που δόθηκαν υπάρχουν στην βάση δεδομένων. Σε περίπτωση που υπάρχουν τότε απαντάει στην Login.tsx σελίδα που αιτήθηκε, ένα μήνυμα επιβεβαίωσης και έπειτα αποθηκεύεται στο localStorage το email του χρήστη και ο χρήστης οδηγείται στην κύρια σελίδα με την χρήση της React Router Dom. Σε διαφορετική περίπτωση, που δεν υπάρχει δηλαδή τα στοιχεία καταγεγραμμένα στη Βάση δεδομένων, τότε το Backend απαντάει κατάλληλα στην Login.tsx και έπειτα εμφανίζεται στον χρήστη ανάλογο μήνυμα πως τα στοιχεία είναι λανθασμένα. Επιλογή δίνεται σε περίπτωση που ο χρήστης δεν έχει εγγραφεί, να μεταβεί στην σελίδα Εγγραφής.



The image shows a login page with a purple background. At the top center, the word "Login" is written in a black, handwritten-style font. Below it, there are two white input fields with rounded corners. The first field is labeled "Email" and the second is labeled "Password". Below these fields is a light blue button with the word "Submit" in white. At the bottom, there is a link that says "Not register? [Click here](#)".

Εικόνα 4.3: Η σελίδα σύνδεσης χρηστών



The image shows the same login page as in Figure 4.3, but with an error message. The "Email" field now contains the text "ice18390061@uniwa.gr" and the "Password" field contains a series of dots. Below the input fields, the text "Wrong Email or Password" is displayed in red. The "Submit" button and the "Not register? [Click here](#)" link are still present at the bottom.

Εικόνα 4.4: Προσπάθεια λανθασμένης εισόδου

4.3.3 ΚΥΡΙΑ ΣΕΛΙΔΑ

Η κύρια σελίδα είναι προσβάσιμη μόνο αφού ο χρήστης έχει πραγματοποιήσει είσοδο. Έτσι η πρόσβαση σε αυτήν δεν πρέπει να είναι δυνατή σε διαφορετική περίπτωση ακόμη και αν ο χρήστης πληκτρολογήσει ακριβώς την διεύθυνση (URL) στον περιηγητή του. Για να επιτευχθεί το παραπάνω βοήθησε η βιβλιοθήκη React Router Dom.

Συγκεκριμένα, στο αρχείο App.tsx περιέχονται όλες οι διαδρομές της εφαρμογής (Routes). Ένα Route είναι ένα component της βιβλιοθήκης Router Dom και σε αυτό μπορούμε να δηλώσουμε το μονοπάτι που ο χρήστης θα ακολουθεί προκειμένου να μεταβεί σε μια σελίδα, καθώς και την ίδια την σελίδα σε μορφή component για παράδειγμα έχουμε το component προς την σελίδα εγγραφής `<Route path="/signup" element={<Signup />} />`. Έτσι χρειαζόμαστε τέσσερα Routes τα οποία είναι για την σελίδα Σύνδεσης, Εγγραφής, της κύριας σελίδας και έναν γενικότερο επιπλή ο οποίος μπαίνει στο τέλος και θα αναλαμβάνει κάθε διαδρομή που δεν αντιστοιχεί σε μια από τις παραπάνω που αναφέρθηκαν. Σε αυτήν την περίπτωση γίνεται ανακατεύθυνση στην σελίδα Σύνδεσης.

Για την επίτευξη της πρόσβασης στην κύρια σελίδα υπό συνθήκη δημιουργήθηκε ένα component το οποίο εξετάζει αν είναι συνδεδεμένος ο χρήστης μέσω της localStorage και αν είναι τότε επιτρέπει την πρόσβαση στην κύρια σελίδα. Αυτή είναι μια εύκολη τεχνική για έλεγχο πρόσβασης σε σελίδες βάση συνθηκών ή ρόλων ενώ παλιότερα δραστικότερο ρόλο είχε το Backend για τον έλεγχο.

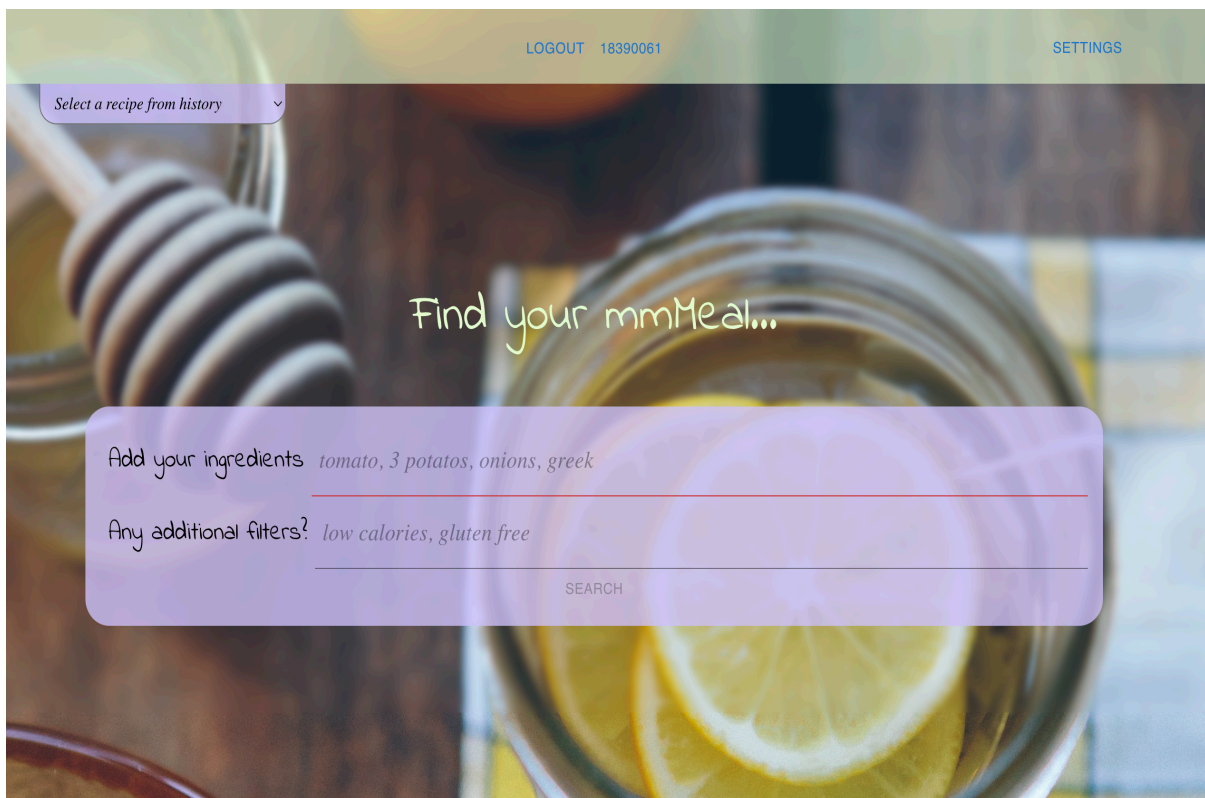
Αφού ο χρήστης έχει συνδεθεί με τα σωστά στοιχεία στην εφαρμογή, μεταβαίνει πλέον στην κύρια σελίδα της εφαρμογής. Εκεί ο χρήστης μπορεί να εισάγει τα υλικά της επιθυμίας του καθώς και πιθανές επιπλέον προτιμήσεις σε δύο textfields και κατά την επιλογή του search button, η εφαρμογή θα προτείνει μια συνταγή με βάση τα εισαγόμενα. Επιλογή έχει επίσης να δει το ιστορικό των συνταγών που έχουν παραχθεί επιλέγοντας πάνω αριστερά την αναπτυσσόμενη λίστα (dropdown). Οι επιπλέον προτιμήσεις που επιλέγονται στο δεύτερο textfield αποθηκεύονται στην εφαρμογή και θα αξιοποιηθούν μετέπειτα σε κάθε άλλη αναζήτηση. Με αυτόν τον τρόπο ο χρήστης δεν χρειάζεται να ξανά δηλώνει τις προτιμήσεις του. Σε περίπτωση που ο χρήστης δεν επιθυμεί να αξιοποιηθούν οι παλιές του προτιμήσεις, έχει την δυνατότητα επιλέγοντας στις ρυθμίσεις πάνω δεξιά κατά την εμφάνιση ενός μενού να επιλέξει την διαγραφή των παλιών προτιμήσεων μέσω της «Διαγραφή ειδικών φίλτρων» (Delete Special Filter). Στο ίδιο μενού εμφανίζεται και η επιλογή «Διαγραφή Ιστορικού» (Delete History) η οποία διαγράφει το ιστορικό αποτελεσμάτων συνταγών και κατά το άνοιγμα του dropdown, για το ιστορικό, θα είναι άδειο. Διατίθεται η επιλογή της αποσύνδεσης στο κέντρο προς τα πάνω της σελίδας ως «Αποσύνδεση» (Logout). Κατά την επιλογή της ο χρήστης θα αποσυνδεθεί και θα επιστρέψει στην σελίδα Σύνδεσης.

Κατά την επιλογή του χρήστη της επιλογής αναζήτησης (Search) αφού έχει εισάγει δεδομένα τουλάχιστον στο πρώτο textfield, τα δεδομένα θα σταλούν στο backend και θα φορτωθεί σε ένα νέο πλαίσιο η προτεινόμενη συνταγή. Μέχρι να γίνει η εύρεση της συνταγής, ένα εικονίδιο φόρτωσης εμφανίζεται ειδοποιώντας τον χρήστη πως το αίτημα είναι σε εξέλιξη. Ενώ σε περίπτωση που γίνει

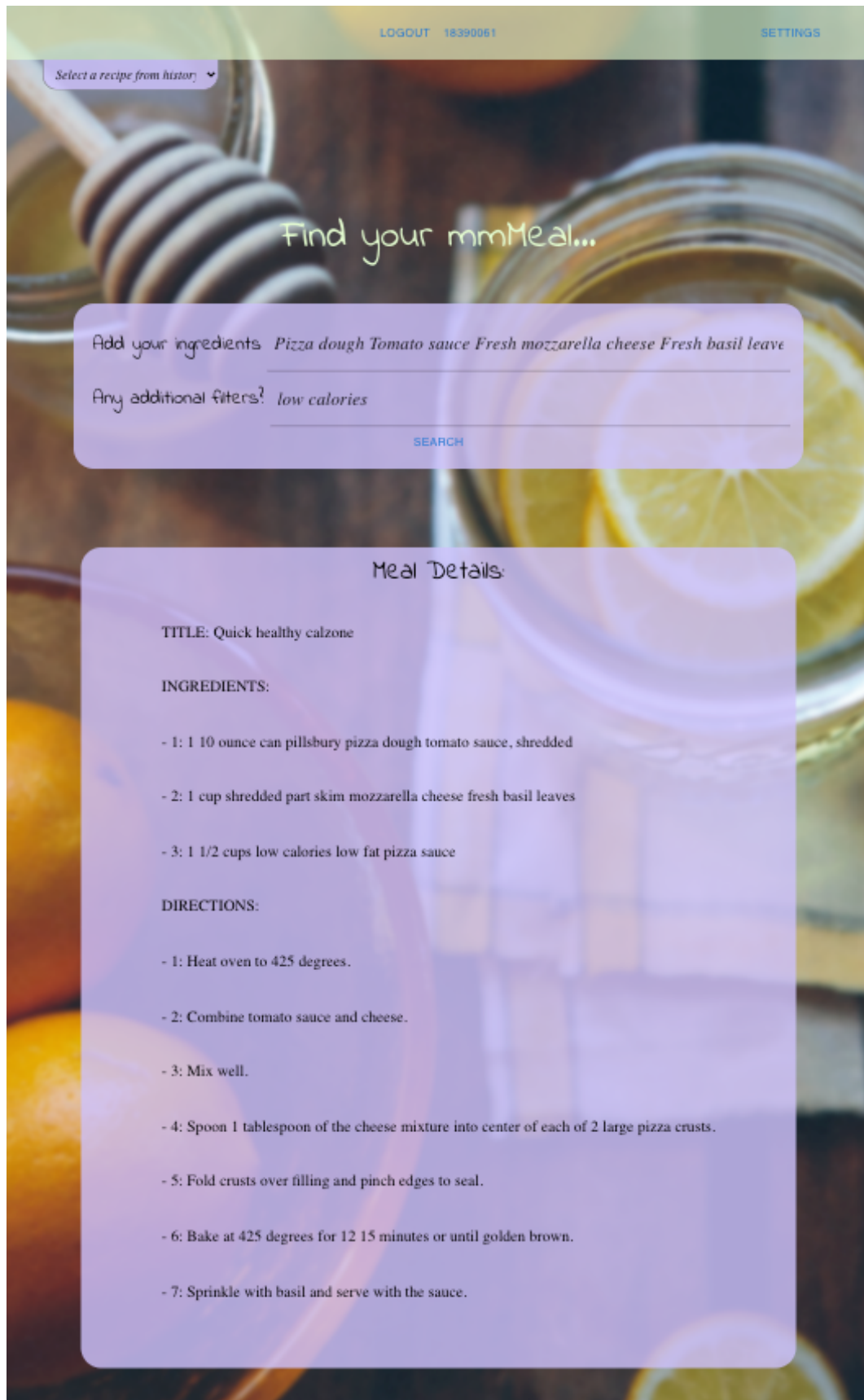
επιλογή μιας συνταγής από το dropdown τότε το πλαίσιο με την συνταγή θα εμφανιστεί άμεσα κάτω από την φόρμα δεδομένων.

Η κάθε συνταγή χωρίζεται σε τρία μέρη. Στον τίτλο (title) όπου αναγράφεται ο τίτλος της συνταγής, στα υλικά (ingredients) όπου αναγράφονται τα υλικά που χρειάζονται για την κατασκευή της συνταγής και στις οδηγίες (directions) όπου αναγράφονται τα βήματα εκτέλεσης.

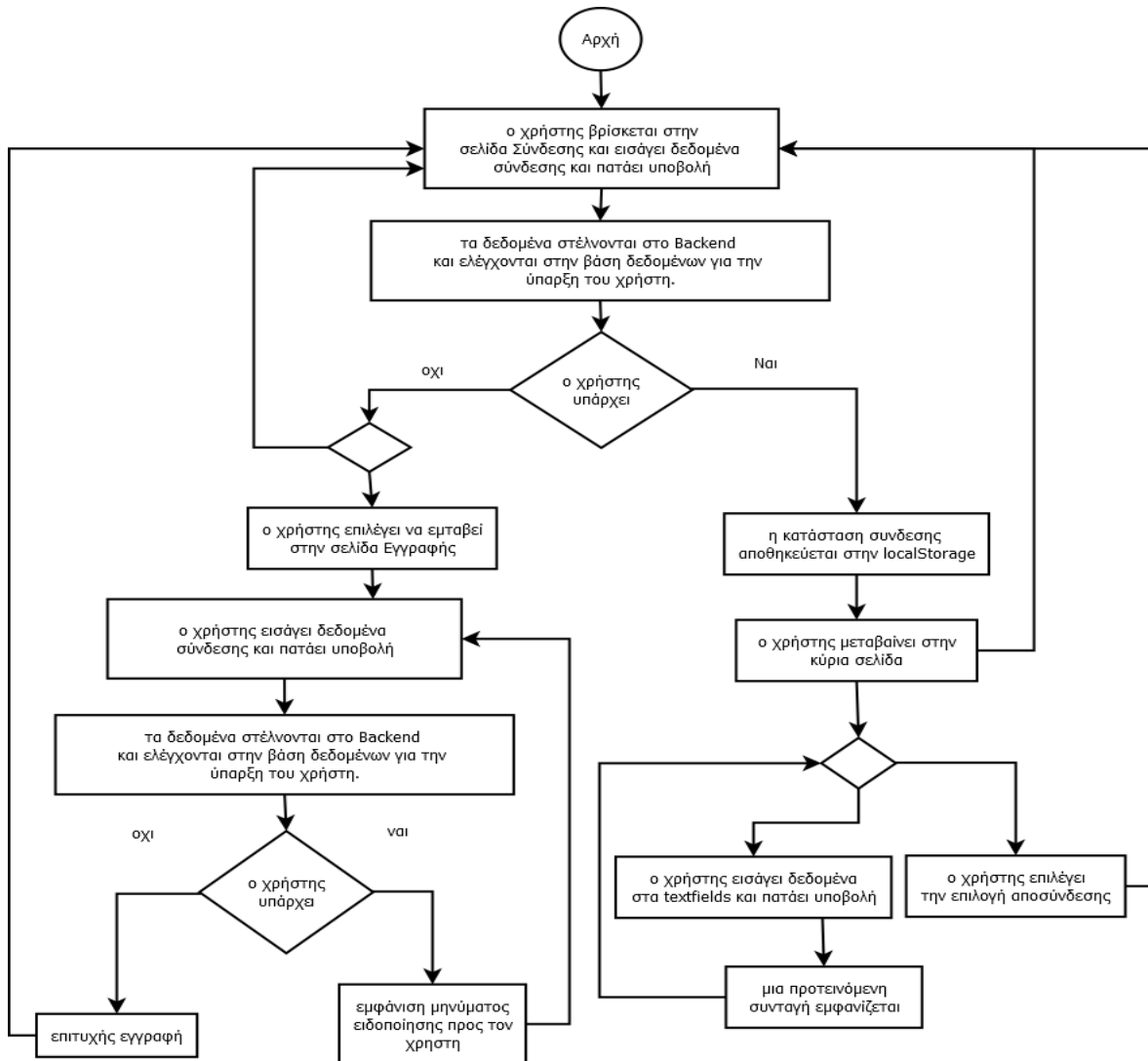
Αντίστοιχα με τις προηγούμενες σελίδες, Σύνδεσης και Εγγραφής, όλα τα components δηλαδή τα buttons, dropdown, menu, textfields έχουν χρησιμοποιηθεί μέσω της βιβλιοθήκης MaterialUI ενώ τα στιλιστικά έχουν πραγματοποιηθεί με την χρήση Styled Components. Στο υπόβαθρο της σελίδας φαίνεται μια εικόνα με σκοπό της να οικειοποιηθεί τον χρήστη με το αντικείμενο της εφαρμογής, της συνταγές. Στην εικόνα έχει εφαρμοστεί φίλτρο θολώματος ώστε η έμφαση να δίνεται στις επιλογές ως προς τον χρήστη και όχι στην ίδια την εικόνα.



Εικόνα 4.5: Κύρια σελίδα της εφαρμογής



Εικόνα 4.6: Κύρια σελίδα της εφαρμογής με εμφάνιση συνταγής



Σχήμα 4.2: Διάγραμμα εφαρμογής Frontend κομματιού

4.4 ΠΙΣΩ ΑΚΡΟ (BACKEND)

Ιδιαίτερα σημαντικό κομμάτι της εφαρμογής, αλλά και κάθε εφαρμογής, είναι το Backend το οποίο αναλαμβάνει την κύρια λογική της εφαρμογής καθώς και την επεξεργασία των δεδομένων. Σε κάθε σελίδα της εφαρμογής υπάρχει επικοινωνία με αυτό με σκοπό την ταυτοποίηση του χρήστη ή την διεκπεραίωση κάποιας ενέργειά του όπως την εύρεση συνταγής ή διαγραφή ιστορικού και άλλων. Σε αυτήν την ενότητα παρουσιάζεται η δομή και η ανάλυση του τι συμβαίνει στα κύρια βήματα καθώς και τα στάδια της επικοινωνίας με τα διάφορα αρχεία.

4.4.1 ΓΕΝΙΚΗ ΔΟΜΗ ΤΟΥ BACKEND

Η κύρια δομή του Backend περιέχεται στο αρχείο App.js το οποίο επικοινωνεί με τις σελίδες του Frontend. Ακολουθούν τα αρχεία PythonCall.js και ModelProcessing.py τα οποία το πρώτο αναλαμβάνει την επικοινωνία του App.js με την Python και το δεύτερο, την χρήση του μοντέλου

μηχανικής μάθησης. Περιλαμβάνεται επίσης η δήλωση και των μοντέλων και σχημάτων για την επικοινωνία με την βάση δεδομένων τα οποία θα γίνει αναφορά στο κεφάλαιο 5.3.2.

Όπως αναφέρθηκε το App.js διαχειρίζεται την κύρια επικοινωνία της εφαρμογής τόσο με το Frontend όσο και με το προ εκπαιδευμένο μοντέλο. Συγκεκριμένα με το Frontend λαμβάνει και στέλνει δεδομένα μέσω της βιβλιοθήκης axios και του πρωτοκόλλου HTTP με τις μεθόδους GET, POST.

Η μέθοδος GET χρησιμοποιείται για την ανάκτηση δεδομένων από έναν διακομιστή. Όταν πληκτρολογείτε μια διεύθυνση URL στον περιηγητή σας ή κάνετε κλικ σε έναν σύνδεσμο, γίνεται αίτηση GET για να ανακτήσει τη σελίδα. Οι πληροφορίες της αίτησης GET εμφανίζονται στην ίδια τη διεύθυνση URL.

Η μέθοδος POST χρησιμοποιείται για να στείλει δεδομένα σε έναν διακομιστή. Αυτό συμβαίνει, για παράδειγμα, κατά την υποβολή μιας φόρμας σε μια ιστοσελίδα. Τα δεδομένα της αίτησης POST δεν εμφανίζονται στη διεύθυνση URL, αλλά μεταδίδονται στο σώμα της αίτησης (request body).

Προκειμένου κάποια σελίδα του Frontend να επικοινωνήσει πρέπει να κάνει αίτηση με μια από τις παραπάνω μεθόδους, POST, GET, σε μορφή διεύθυνσης URL `http://localhost:8000/path`. Η ανάλυση του URL αυτού είναι

http://: Είναι το πρωτόκολλο που χρησιμοποιείται. Σε αυτήν την περίπτωση, πρόκειται για το HTTP (HyperText Transfer Protocol), που είναι το πρωτόκολλο που χρησιμοποιείται για την αποστολή και λήψη πληροφοριών στο διαδίκτυο

localhost: Αυτό είναι το όνομα κεντρικού υπολογιστή (hostname) ή το ο τομέας (domain). Στην περίπτωση αυτή, είναι η «localhost», που σημαίνει το ίδιο το μηχάνημα που τρέχει τον περιηγητή ή την εφαρμογή που χρησιμοποιείται.

:8000: Είναι ο αριθμός της θύρας (port). Σε αυτήν την περίπτωση, χρησιμοποιείται η θύρα 8000. Οι θύρες επιτρέπουν σε διάφορες υπηρεσίες να τρέχουν στον ίδιο υπολογιστή με μια μεθοδολογία για την καθοδήγηση της κίνησης στη σωστή υπηρεσία.

/path: Αυτή είναι η διαδρομή (path) που ακολουθείται στον server για να φτάσει στον συγκεκριμένο πόρο που ζητάτε. Στην περίπτωση αυτή, η διαδρομή είναι "/path".

Έτσι το κύριο μέρος του URL που επαναλαμβάνεται είναι το `http://localhost:8000/` και ακολουθεί ένα διαφορετικό κάθε φορά path ανάλογα την ενέργεια αιτήματος που απαιτείται, για παράδειγμα σύνδεσης ή εγγραφής.

Συγκεκριμένα, τα αιτήματα που δέχεται το Backend από το Frontend λαμβάνονται μέσω έξι διευθύνσεων

1. /login: Το αίτημα προέρχεται από την σελίδα Σύνδεσης και είναι τύπου POST και στο σώμα (body) του αιτήματος περιλαμβάνονται το email και ο κωδικός του χρήστη. Στο κομμάτι αυτό, ελέγχεται αν τα παραπάνω στοιχεία υπάρχουν στην βάση δεδομένων και αναλόγως ο server ανταποκρίνεται με το κατάλληλο μήνυμα.

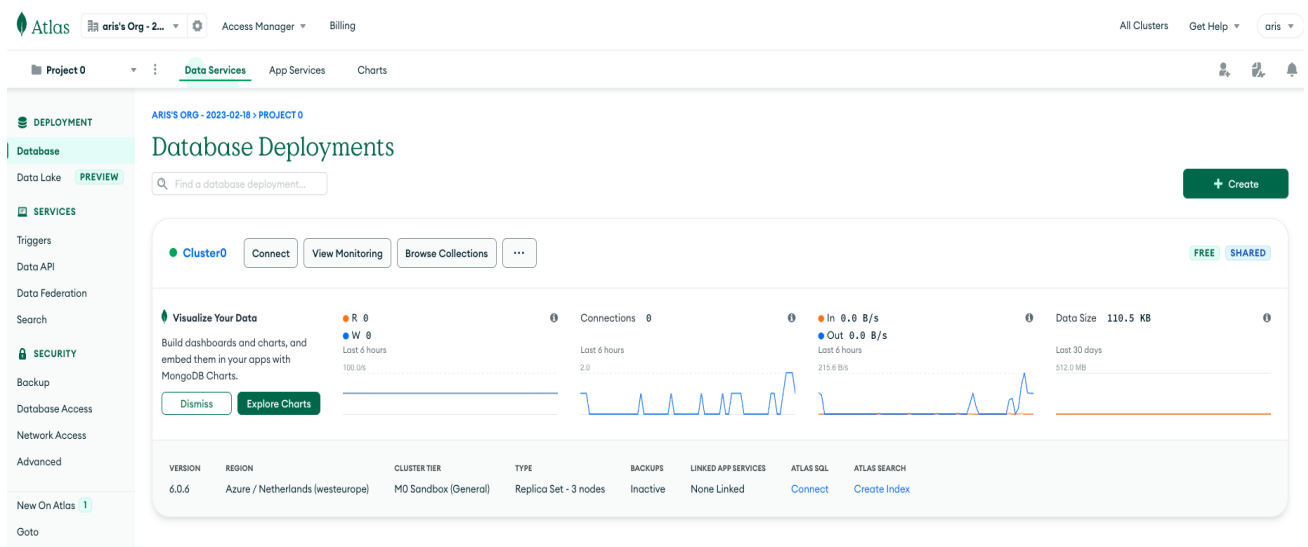
- 2. /signup:** Το αίτημα προέρχεται από την σελίδα Εγγραφής. Αντίστοιχα με το login, είναι τύπου POST και στο body περιλαμβάνονται πάλι το email και κωδικός του χρήστη. Σκοπός είναι να ελεγχθεί αν υπάρχει ίδια εγγραφή βάσει στοιχείων που δόθηκαν και σε περίπτωση που δεν υπάρχει, να προστεθεί αλλιώς να το backend απαντάει πως υπάρχει ήδη η εγγραφή.
- 3. /home:** Το αίτημα προέρχεται από την κύρια σελίδα και αναλαμβάνει να λάβει τα δεδομένα (υλικά για κάποια συνταγή) που έχει εισάγει ο χρήστης καθώς και την αποστολή αυτών στο αρχείο rython μέσω του αρχείου PythonCall.js. Το αίτημα είναι τύπου POST και στο body περιλαμβάνεται το email του χρήστη και το input του. Ελέγχεται αν υπάρχει αυτός ο χρήστης με βάση το email και φέρνει τα specialFilters που έχει εισάγει στο παρελθόν ο χρήστης. Τα φίλτρα αυτά θα προστεθούν στην αναζήτηση του χρήστη και θα προσπελαστούν στο rython μοντέλο. Η απάντηση από το μοντέλο αποθηκεύεται στο ιστορικό του συγκεκριμένου χρήστη.
- 4. /history:** Το αίτημα είναι τύπου GET και προέρχεται από την κύρια σελίδα και ρόλος του είναι να ανακτά το ιστορικό συνταγών βάση το email του χρήστη.
- 5. /specialFilters:** Το αίτημα είναι τύπου POST και προέρχεται από την κύρια σελίδα. Στο Body περιλαμβάνονται το email και τα ειδικά φίλτρα που έχει προσθέσει. Βάση του email ενημερώνεται προσθετικά η εγγραφή των ειδικών φίλτρων στην βάση δεδομένων. Έτσι αν παλιότερα είχε γίνει η αναζήτηση με φίλτρο χαμηλές θερμίδες (low calories) και σε επόμενη αναζήτηση ο χρήστης χρησιμοποιούσε το φίλτρο χωρίς γλουτένη (gluten free), τότε σε μεταγενέστερες αναζητήσεις θα γινόταν χρήση των φίλτρων ως low calories, gluten free μέχρις ότου ο χρήστης επέλεγε διαγραφή των φίλτρων.
- 6. /deleteHistory:** Το αίτημα είναι τύπου POST και προέρχεται από την κύρια σελίδα. Ρόλος του είναι με βάση το email που δέχεται στο body να διαγράψει το ιστορικό αναζητήσεων συνταγών.
- 7. /deleteSpecialFilters:** Το αίτημα είναι τύπου POST και προέρχεται από την κύρια σελίδα. Ρόλος του είναι με βάση το email που δέχεται στο body να διαγράψει το ιστορικό ειδικών φίλτρων που έχει εισάγει ο χρήστης στο παρελθόν.

Το αρχείο PythonCall.js αναλαμβάνει την επικοινωνία του App.js με το ModelProcessing.py, όπου περιέχεται το μοντέλο μηχανικής μάθησης. Εσωτερικά αποτελείται από μια συνάρτηση η οποία δέχεται το τελικό input του χρήστη, συνδυασμένο δηλαδή από την αναζήτηση και τα ειδικά φίλτρα, και με την συνάρτηση spawn() της Node.js, δημιουργεί μια νέα διεργασία «παιδί» καλώντας το αρχείο ModelProcessing.py και περνώντας του το παραπάνω input. Ακολουθούν τρεις μέθοδοι οι οποίες αναλαμβάνουν τις απαντήσεις από την διεργασία, την περίπτωση σφάλματος και την περίπτωση τερματισμού. Μια μεταβλητή, έστω output, αποθηκεύει τις απαντήσεις του μοντέλου και σε περίπτωση τερματισμού αποστέλλει το τελικό αποτέλεσμα πίσω στο App.js.

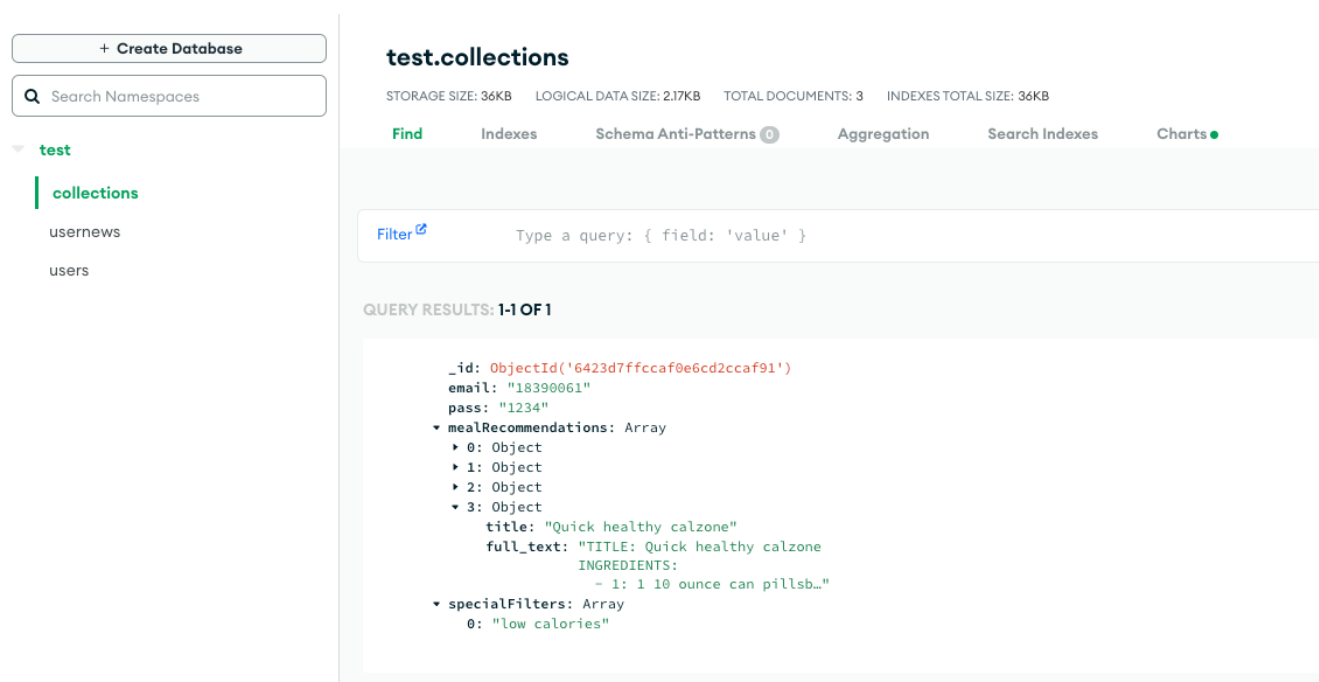
5.4.2 Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ MONGODB

Όπως έχει προαναφερθεί, για την βάση δεδομένων της εφαρμογής έγινε χρήση της MongoDB μιας μη σχεσιακής βάσης NoSQL, που αποθηκεύει τα δεδομένα σε έγγραφα JSON (JavaScript Object Notation), γνωστά ως BSON (Binary JSON). Για την προβολή και διαχείριση της βάσης παρέχεται η σελίδα διαχειριστή της MongoDB, Atlas. Στην σελίδα αυτή παρέχονται διάφορες λειτουργίες και στατιστικά για την κίνηση των αναγνώσεων και των εγγραφών στην βάση. Από αυτές διακρίνονται η ενότητα Database όπου παρουσιάζονται τα cluster της βάσης. Ένα cluster αναφέρεται σε ένα σύνολο από διακομιστές που συνεργάζονται για να αποθηκεύσουν και να διαχειριστούν τα δεδομένα.

Στο cluster μπορούμε να δούμε την επίδοση σε αυτό, δηλαδή τους ρυθμούς χρήσης του cluster από τις εφαρμογές που την χρησιμοποιούν. Σημαντικό είναι επίσης οι συλλογές (collections) της βάσης όπου αποθηκεύονται τα δεδομένα. Στην εικόνα 5.8 διακρίνονται τρεις συλλογές με την πρώτη ως «collections» να χρησιμοποιείται για την εφαρμογή της παρούσας εργασίας. Στην συγκεκριμένη φαίνεται μια εγγραφή χρήστη με παιδιά _id, email, pass, mealRecommendations και specialFilters. Το πρώτο γνώρισμα είναι μοναδικό και εφαρμόζεται αυτόματα από την βάση ώστε να ξεχωρίζουν οι εγγραφές. Ακολουθεί το email και το password του χρήστη που χρησιμοποιείται για να συνδεθεί στην εφαρμογή ενώ στον πίνακα mealRecommendations αποθηκεύεται το ιστορικό αναζητήσεων. Τέλος στο specialFilters πίνακα αποθηκεύονται τα επιπλέον φίλτρα που επιθυμεί ο χρήστης για παράδειγμα low calories στην συγκεκριμένη περίπτωση. Κατά την εγγραφή ενός χρήστη, μια νέα εγγραφή στην βάση με μοναδικό _id θα προστεθεί.



Εικόνα 4.7: Σελίδα διαχειριστή Atlas της βάσης MongoDB



Εικόνα 4.8: Οι συλλογές της βάσης

Προκυμμένου να επικοινωνήσει η εφαρμογή με την βάση δεδομένων και να μπορεί να αρχίσει να δημιουργεί και να ενημερώνει εγγραφές θα πρέπει να μεσολαβήσει η βιβλιοθήκη Mongoose που έγινε αναφορά στην ενότητα 4.2.6. Στο αρχείο του Backend, mongoose.js υλοποιείται όλη η αναγκαία δομή προκειμένου να στηθεί η επικοινωνία με την βάση και το αρχείο App.js να μπορεί να υλοποιεί CRUD ενέργειες δηλαδή δημιουργία, διάβασμα, ενημέρωση και διαγραφή εγγραφών (Create, Read, Update, Delete). Όπως προ αναφέρθηκε απαιτείται η δήλωση ενός ή περισσότερων schema καθώς και ενός μοντέλου. Το schema δημιουργείται για να ορίσει την δομή των αντικείμενων που χρειάζεται και στην συγκεκριμένη περίπτωση απαιτούνται 2 δομές αντικείμενων που θα αποθηκεύονται στην βάση, ένα για τον χρήστη και τα στοιχεία του και ένα για τις συνταγές.

Ακολουθεί το σχήμα για τις συνταγές το οποίο αποτελείται από τον τίτλο(title) και το κύριο σώμα της συνταγής δηλαδή τα υλικά και οι οδηγίες(full_text) const recipeSchema = new Schema({title: String, full_text: String,}). Έπειτα για το σχήμα του χρήστη αποτελείται από πεδία για το email, τον κωδικό password, τα επιπλέον φίλτρα specialFilters, και το ιστορικό συνταγών mealRecommendations με τα δύο πρώτα να ακολουθούνται από την σημαία (flag) ως απαιτούμενα (required) const userSchema = new Schema({email: { type: String, required: true }, password: { type: String, required: true }, specialFilters: { type: [String], default: [] }, mealRecommendations: [recipeSchema]}). Τέλος η εντολή const collection = mongoose.model("collection", userSchema) δημιουργεί μια συλλογή στην βάση δεδομένων χρησιμοποιώντας την μέθοδο mongoose.model() με το όνομα «collection», όπως και φαίνεται στην εικόνα 5.8, και χρησιμοποιεί το userSchema σχήμα. Απαιτούμενο είναι επίσης ο ορισμός της σύνδεσης προς ένα cluster μέσω ενός συνδέσμου που η σελίδα Atlas παρέχει για την σύνδεση με την εφαρμογή.

4.4.3 ΤΟ ΠΕΡΙΒΑΛΛΟΝ CONDA

Όπως αναφέρθηκε στην ενότητα 5.2, προκειμένου να συνδυαστούν οι βιβλιοθήκες της `python` σε μια `Node.js` εφαρμογή χρειάστηκε η εγκατάσταση του `Conda` περιβάλλοντος. Το `Conda` είναι ένα ανοιχτού κώδικα σύστημα διαχείρισης πακέτων για διάφορες γλώσσες προγραμματισμού δημιουργώντας ένα ξεχωριστό περιβάλλον για την εγκατάσταση, εκτέλεση και ενημέρωση πακέτων. Για την δημιουργία ενός περιβάλλοντος χρειάζεται η εντολή **`conda create --name myenv`** και με την εντολή **`conda activate myenv`** το περιβάλλον τίθεται σε χρήση ενώ με την εντολή **`conda list`** επιτρέπει την επίβλεψη των εγκαταστημένων πακέτων και βιβλιοθηκών.

Για τον σκοπό της εφαρμογής της εργασίας οι παραπάνω εντολές είναι αρκετές ενώ το `Conda` προσφέρει μεγάλη γκάμα επιλογών και διευκολύνσεων. Αφού έγινε η δημιουργία ενός ανεξάρτητου περιβάλλοντος μέσα στον ίδιο φάκελο που στεγάζεται το πρότζεκτ, με την χρήση του διαχειριστή πακέτων της `python`, `pip`, εγκαταστάθηκαν οι απαραίτητες βιβλιοθήκες για την σωστή λειτουργία του μοντέλου μηχανικής μάθησης. Το `Conda` κατά την εγκατάστασή του σε έναν φάκελο, θα δημιουργήσει ένα `.env` φάκελο περιέχοντας τα απαραίτητα αρχεία για την λειτουργία του περιβάλλοντος.

4.5 Η ΔΟΜΗ ΚΑΙ ΤΟ ΜΟΝΤΕΛΟ T5-RECIPE-GENERATOR

Στο αρχείο `ModelProcessing.py` υπάρχουν οι απαραίτητες εντολές για την προσθήκη, διαμόρφωση και χρήση του μοντέλου `T5-recipe-generator`. Το μοντέλο έχει εγκατασταθεί μέσα στους φακέλους της εφαρμογής σε μορφή αρχείων από την επίσημη σελίδα της `Hugging Face`, όχι μέσω χρήσης του `packet manager (NPM)`, και χρησιμοποιείται άμεσα μέσα στον κώδικα. Στο ίδιο αρχείο καθορίζονται οι παράμετροι, οι προ επεξεργασία της εισόδου από τον χρήστη καθώς και το `tokenization` αυτής. Τέλος ετοιμάζεται το αποτέλεσμα αφού χωριστεί σε τίτλο, απαιτούμενα υλικά και οδηγίες εκτέλεσης.

4.5.1 Η ΔΟΜΗ ΤΟΥ ΑΡΧΕΙΟΥ

Για την επιτυχή εκτέλεση του μοντέλου ακολουθείται η εξής δομή. Αρχικά, γίνεται η εισαγωγή των απαιτούμενων βιβλιοθηκών, συμπεριλαμβανομένων των `sys`, το οποίο περιλαμβάνει βασικές μεθόδους για την επικοινωνία με το σύστημα όπως ανάγνωση παραμέτρων κατά το κάλεσμα του αρχείου, `AutoTokenizer` και `FlaxAutoModelForSeq2SeqLM` από τη βιβλιοθήκη `transformers`. Έπειτα δηλώνονται οι μεταβλητές `tokenizer` και `model` οι οποίες αρχικοποιούνται με τον κατάλληλο `tokenizer` αυτόματα, όπως έχει αναφερθεί για την `AutoTokenizer` στην 5.2.9, και το μοντέλο `t5-recipe-generator` εκχωρώντας το από τον φάκελο που έχει ληφθεί και χρησιμοποιώντας το μέσω της `FlaxAutoModelForSeq2SeqLM`. Τέλος δηλώνονται οι παράμετροι για την διαμόρφωση του μοντέλου, οι οποίες θα περιγράφουν στην ενότητα 4.5.2.

Για το τρέξιμο των αρχείων `python` δηλώνεται μια κύρια συνάρτηση με όνομα `__main__` από όπου ξεκινάει η ροή του προγράμματος. Εκεί αρχικά λαμβάνεται το πρώτο όρισμα από την είσοδο στο κατά το κάλεσμα του αρχείου από το `App.js` όπου και περνιέται η είσοδος από τον χρήστη. Έπειτα καλείται με την παραπάνω είσοδο ως όρισμα η συνάρτηση `recommend_meal()`. Αυτή έχει σκοπό να λάβει τα παραγόμενα αποτελέσματα και να δημιουργήσει την απαιτούμενη μορφή τίτλου-υλικών-οδηγιών η οποία θα αποσταλεί πίσω στο `PythonCall.js`. Πριν αρχίσει την διαμόρφωση αρχικά απαιτείται το αποτέλεσμα το οποίο λαμβάνει από την συνάρτηση `generation_function()` περνώντας ως όρισμα την είσοδο του χρήστη.

Η συνάρτηση `generation_function` αποτελείται από διάφορα στάδια που εκτελούνται για να παράγει το επιθυμητό κείμενο συνταγής. Κατά το πρώτο στάδιο, η είσοδος της συνάρτησης ελέγχεται για το είδος της. Αν είναι μια λίστα, διατηρείται αναλλοίωτη, αλλιώς μετατρέπεται σε μια λίστα που περιέχει μόνο την είσοδο. Αυτή η προετοιμασία είναι σημαντική για την εξασφάλιση συνέπειας στην επεξεργασία της εισόδου. Στη συνέχεια, δημιουργείται μια νέα λίστα με τα στοιχεία της αρχικής εισόδου, αλλά προσθέτοντας ένα πρόθεμα «items» μπροστά από κάθε στοιχείο. Το πρόθεμα αυτό είναι σημαντικό για τη σωστή λειτουργία του μοντέλου. Η επόμενη διαδικασία είναι η μετατροπή της εισόδου σε αναγνωριστικά (`input_ids`) και μάσκα προσοχής (`attention_mask`) μέσω του `tokenizer`. Ο `tokenizer` αναλαμβάνει να διαχωρίσει την είσοδο σε κατάλληλα τμήματα (`tokens`) και να τα αντιστοιχίσει σε αριθμητικές αναπαραστάσεις. Τα `input_ids` αντιπροσωπεύουν τα αναγνωριστικά και η `attention_mask` δηλώνει ποια μέρη του κειμένου πρέπει να ληφθούν υπόψη κατά την επεξεργασία.

Στο επόμενο στάδιο, γίνεται η παραγωγή του κειμένου συνταγής με βάση τα `input_ids` και την `attention_mask`. Χρησιμοποιείται το μοντέλο `model.generate` για να παράξει το κείμενο. Οι παράμετροι παραγωγής, από το `generation_kwargs` που θα περιγράψει στην ενότητα 5.5.2. Το παραγόμενο κείμενο συνταγής υπόκειται σε επεξεργασία μέσω της συνάρτησης `target_postprocessing`. Κατά την επεξεργασία αυτή, αφαιρούνται τα ειδικά `tokens` που προστίθενται από τον `tokenizer`, καθώς και άλλες ορισμένες ειδικές ακολουθίες. Αυτή η επεξεργασία εξασφαλίζει ότι το παραγόμενο κείμενο είναι καθαρό και κατανοητό από τον χρήστη. Το τελικό παραγόμενο κείμενο συνταγής επιστρέφεται από τη συνάρτηση `generation_function` και μπορεί να χρησιμοποιηθεί για εμφάνιση ή περαιτέρω επεξεργασία.

4.5.2 ΠΑΡΑΜΕΤΡΟΙ ΜΟΝΤΕΛΟΥ

Οι παράμετροι του μοντέλου γίνονται σε ένα αντικείμενο `generation_kwargs` όπου δηλώνονται οι διάφοροι παράμετροι μαζί με τις τιμές. Συγκεκριμένα έγινε χρήση των εξής:

`"max_length": 2000` και `"min_length": 64` τα οποία δηλώνουν το μέγιστο και το ελάχιστο παραγόμενο αριθμό σε `tokens`, με τιμές 2000 και 64.

`"no_repeat_ngram_size": 6` χρησιμοποιείται για να ελέγξει την επανάληψη ομάδων λέξεων (`n-grams`) στο παραγόμενο κείμενο. Ένα `n-gram` αναφέρεται σε μια συνεχόμενη ακολουθία λέξεων μήκους `n`. Με

την παράμετρο "no_repeat_ngram_size": 6, το μοντέλο θα αποφεύγει να παράγει κείμενο που περιέχει επαναλαμβανόμενα 6-grams.

"do_sample": True Αυτή η παράμετρος ενεργοποιεί το δειγματοληψία κατά τη δημιουργία. Όταν ορίζεται σε True, το μοντέλο θα παίρνει τυχαία δείγματα από την κατανομή πιθανοτήτων του επόμενου τεκμηρίου αντί να επιλέγει πάντα το τεκμήριο με την υψηλότερη πιθανότητα. Η δειγματοληψία εισάγει τυχαιότητα και μπορεί να οδηγήσει σε πιο ποικίλες και δημιουργικές εξόδους.

"top_k": 50: Αυτή η παράμετρος χρησιμοποιείται μαζί με την do_sample. Ελέγχει τον αριθμό των δυνητικών επόμενων tokens που θα ληφθούν υπόψη κατά τη διαδικασία της δειγματοληψίας (sampling). Σε αυτήν την περίπτωση, το μοντέλο θα λάβει υπόψη τα 50 tokens με τις υψηλότερες πιθανότητες κατά τη δειγματοληψία. Ένα υψηλότερο top_k αυξάνει την ποικιλία του παραγόμενου κειμένου.

"top_p": 0.9: Η παράμετρος "top_p" χρησιμοποιείται για να ελέγξει πόσο από τη συνολική πιθανότητα των επόμενων λέξεων θα ληφθεί υπόψη κατά τη δειγματοληψία. Όταν ορίζεται η τιμή του "top_p", το μοντέλο θα επιλέξει τις λέξεις που συνοδεύονται από το σύνολο πιθανοτήτων που αντιστοιχούν στο κατώτατο όριο που έχει οριστεί. Αυτή η παράμετρος χρησιμοποιείται σε συνδυασμό με το "do_sample". Όταν θέτουμε την τιμή του "top_p", η διαδικασία δειγματοληψίας θα επιλέξει το ελάχιστο δυνατό σύνολο tokens του οποίου το συνολικό πιθανοτικό άθροισμα υπερβαίνει το κατώφλι που ορίσαμε. Δηλαδή, τα tokens με τις υψηλότερες πιθανότητες θα συγκεντρωθούν και θα εξαλείφονται σταδιακά όσο αυξάνεται το συνολικό τους πιθανοτικό άθροισμα. Για παράδειγμα, αν ορίσουμε το "top_p" σε 0.9, το μοντέλο θα επιλέξει το ελάχιστο δυνατό σύνολο tokens του οποίου το συνολικό πιθανοτικό άθροισμα θα υπερβαίνει το 0.9. Αυτό σημαίνει ότι η δειγματοληψία θα εστιάσει στις πιο πιθανές επιλογές, αποκλείοντας την εξέταση των πιο απίθανων επιλογών. Αυτό βοηθάει στην παραγωγή πιο συγκεντρωμένων και συνεκτικών αποτελεσμάτων.

"temperature": 1: Η παράμετρος αυτή ελέγχει τον βαθμό της τυχειότητας κατά τη δειγματοληψία. Ένα υψηλότερο temperature (π.χ. 1) καθιστά τη δειγματοληψία πιο τυχαία, ενώ ένα χαμηλότερο temperature (π.χ. 0.8) την καθιστά πιο συγκεντρωμένη και προσδιορισμένη. Ένα temperature της τιμής 1 σημαίνει ότι οι αρχικές πιθανότητες του μοντέλου χρησιμοποιούνται χωρίς καμία κλιμάκωση.

4.5.3 MONTEΛΟ T5

Το μοντέλο T5 (Text-to-Text Transfer Transformer) [9] είναι ένα από τα προηγμένα μοντέλα που βασίζονται στην αρχιτεκτονική Transformer. Αναπτύχθηκε από την εταιρεία Google Research και έχει αποδειχθεί εξαιρετικά αποτελεσματικό σε πολλές γλωσσικές εργασίες. Το μοντέλο βασίζεται σε μια τυπική αρχιτεκτονική Transformer με κωδικοποιητή και αποκωδικοποιητή, και προσαρμόζεται σε διάφορες εργασίες. Το βασικό μοντέλο αποτελείται από 12 μπλοκ σε κάθε κωδικοποιητή και

αποκωδικοποιητή. Το μοντέλο έχει συνολικά περίπου 220 εκατομμύρια παραμέτρους και χρησιμοποιεί πιθανότητα απόρριψης (dropout) 0,1 για την κανονικοποίηση.

Η διαδικασία εκπαίδευσης περιλαμβάνει προεκπαίδευση του μοντέλου σε ένα μεγάλο σύνολο δεδομένων, χρησιμοποιώντας έναν απλό στόχο αποκατάστασης λαθών και τη μέθοδο της μέγιστης πιθανοφάνειας. Χρησιμοποιείται ο αλγόριθμος βελτιστοποίησης AdaFactor και το μοντέλο προεκπαιδύεται για έναν συγκεκριμένο αριθμό επαναλήψεων. Η προεκπαίδευση χρησιμοποιεί μήκος ακολουθίας 512 και μέγεθος παρτίδας 128 ακολουθιών. Μετά την προεκπαίδευση, το μοντέλο προσαρμόζεται για έναν συγκεκριμένο αριθμό επαναλήψεων στις εργασίες μετάδοσης, χρησιμοποιώντας σταθερό ρυθμό μάθησης. Αποθηκεύονται έλεγχοι σε τακτά διαστήματα και επιλέγεται ο έλεγχος που έχει την υψηλότερη απόδοση επικύρωσης.

Όσον αφορά το λεξιλόγιο, χρησιμοποιείται το SentencePiece για την τομοποίηση του κειμένου σε κομμάτια WordPiece. Το λεξιλόγιο αποτελείται από 32,000 κομμάτια λέξεων και καλύπτει πολλές γλώσσες, συμπεριλαμβανομένων των αγγλικών, γερμανικών, γαλλικών και ρουμανικών. Το λεξιλόγιο χρησιμοποιείται τόσο για την είσοδο όσο και για την έξοδο του μοντέλου, επιτρέποντας μεταφραστικές εργασίες μεταξύ αυτών των γλωσσών.

Τα κομμάτια WordPiece είναι μια μέθοδος τομοποίησης του κειμένου που χρησιμοποιείται στην επεξεργασία φυσικής γλώσσας και έχει σκοπό έχει να χωρίσει τις λέξεις ενός κειμένου σε μικρότερα κομμάτια, τα οποία ονομάζονται subwords. Η διαδικασία τομοποίησης με κομμάτια WordPiece είναι αποτέλεσμα ενός μοντέλου μηχανικής μάθησης που εκπαιδύεται πάνω σε ένα μεγάλο σύνολο κειμένων. Κατά την εκπαίδευση, το μοντέλο μάθαινει να αναγνωρίζει τα κοινά subwords και να δημιουργεί ένα λεξιλόγιο από αυτά. Στην πράξη, η διαδικασία λειτουργεί ως εξής: Αρχικά, το κείμενο διαιρείται σε λέξεις. Στη συνέχεια, οι λέξεις αυτές χωρίζονται σε υπολέξεις (subwords) με βάση το λεξιλόγιο των κομματιών WordPiece. Κάθε subword αποτελεί μια μονάδα με ενιαίο νόημα, αλλά μπορεί να είναι μικρότερο από μια πλήρη λέξη.

Η χρήση των κομματιών WordPiece έχει οφέλη για την επεξεργασία φυσικής γλώσσας. Πρώτον, επιτρέπει την αντιμετώπιση άγνωστων λέξεων ή σπάνιων λέξεων, καθώς μπορεί να δημιουργήσει την αναπαράστασή τους από τα κοινά subwords. Δεύτερον, επιτρέπει την αποτύπωση μιας μεγαλύτερης ποικιλίας λέξεων στο λεξιλόγιο, παρέχοντας μεγαλύτερη γενικότητα και ευελιξία στο μοντέλο.

Η αρχιτεκτονική Transformer χρησιμοποιεί την ιδέα της προσοχής (attention) [8] για να αντιληφθεί τις συσχετίσεις μεταξύ των λέξεων σε ένα κείμενο, και αυτή η ιδέα ενσωματώνεται και στο T5 για την επίτευξη υψηλής απόδοσης.

Η ιδέα της προσοχής αποτελεί ένα σημαντικό κομμάτι στον τομέα της φυσικής γλώσσας και της μηχανικής μάθησης. Αναφέρεται σε έναν μηχανισμό που επιτρέπει σε ένα μοντέλο να εστιάζει την

προσοχή του σε συγκεκριμένα τμήματα μιας πληροφορίας και να αντιλαμβάνεται τις συσχετίσεις μεταξύ τους. Η προσοχή προέρχεται από την αναγνώριση του γεγονότος ότι κατά την επεξεργασία ενός κειμένου ή μιας πληροφορίας, διάφορα τμήματα έχουν διαφορετική σημασία και συμβολή στην τελική κατανόηση. Αντί να αντιμετωπίζουμε ολόκληρη την πληροφορία με ίση βαρύτητα, η προσοχή επιτρέπει στο μοντέλο να επιλέγει ποια τμήματα είναι πιο σημαντικά και να επεξεργαστεί ανάλογα τις πληροφορίες που παρέχονται. Η ιδέα της προσοχής έχει διευρύνει τις δυνατότητες των μοντέλων μηχανικής μάθησης, επιτρέποντάς τους να κατανοήσουν και να παράγουν πιο σύνθετα κείμενα με βάση τη σημασία των λέξεων και των συνδέσεων τους.

Ένα παράδειγμα εφαρμογής της προσοχής είναι ο μηχανισμός προσοχής στα μοντέλα Transformer. Σε αυτά τα μοντέλα, η προσοχή επιτρέπει στο μοντέλο να εστιάζει σε συγκεκριμένες λέξεις ή τμήματα της πρότασης κατά την επεξεργασία μιας άλλης λέξης. Με αυτόν τον τρόπο, το μοντέλο μπορεί να αναγνωρίζει τις συνδέσεις μεταξύ των λέξεων και να αποδίδει κατάλληλα βάρη στην κάθε λέξη ανάλογα με τη σημασία της.

Το βασικό σκεπτικό πίσω από το T5 είναι να εκπαιδευτεί ένα μοντέλο που μπορεί να μεταφέρει γνώση από έναν τύπο κειμένου σε άλλον. Αντί να εκπαιδεύεται για μια συγκεκριμένη εργασία, το T5 εκπαιδεύεται να μάθει να μεταφράζει την περιγραφή μιας εργασίας σε κείμενο καθώς και τη λύση της εργασίας. Με αυτόν τον τρόπο, το μοντέλο είναι σε θέση να αντιμετωπίσει πολλαπλές εργασίες και να προσαρμοστεί σε διάφορα προβλήματα.

Η αρχιτεκτονική του T5 μοντέλου περιλαμβάνει μια στοίβα από επαναλαμβανόμενα μπλοκ Transformer. Κάθε μπλοκ περιλαμβάνει ένα επίπεδο προσοχής και ένα πλήρως συνδεδεμένο επίπεδο. Τα επίπεδα αυτά επιτρέπουν στο μοντέλο να αναλύει το πλήρες περιεχόμενο και τη σημασία της πρότασης. Η προσοχή επιτρέπει στο μοντέλο να εστιάζει σε συγκεκριμένα τμήματα της πρότασης και να κατανοεί τις συσχετίσεις μεταξύ τους.

Για να εκπαιδευτεί το T5 [9], χρησιμοποιήθηκαν μεγάλα σύνολα δεδομένων που καλύπτουν διάφορες γλωσσικές εργασίες, όπως μετάφραση, απόδοση κειμένου και αναγνώριση οντοτήτων. Με αυτήν την εκπαίδευση, το T5 έχει επιδόσεις υψηλού επιπέδου σε πολλές από αυτές τις εργασίες. Διαθέτει επίσης επιμέρους μοντέλα ανάλογα με την χρήση που απαιτείται, αυτά είναι:

- T5-base: Είναι το βασικό μοντέλο του T5 με μεγάλη ικανότητα γενίκευσης και αποδοτικότητα.
- T5-large: Πρόκειται για μια αυξημένη έκδοση του T5 με μεγαλύτερο αριθμό παραμέτρων, που το καθιστούν ιδανικό για πιο απαιτητικές γλωσσικές εργασίες.
- T5-3B: Πρόκειται για μια ακόμα μεγαλύτερη έκδοση του T5 με 3 δισεκατομμύρια παραμέτρους, προσφέροντας ακόμα μεγαλύτερη ικανότητα κατανόησης και παραγωγής κειμένου.

4.5.4 ΜΕΤΑΦΟΡΑ ΜΑΘΗΣΗΣ

Η μεταφορά γνώσης (transfer learning)[8] αναφέρεται στη διαδικασία με την οποία η γνώση που έχει αποκτηθεί σε ένα πρόβλημα ή σε μια περιοχή μπορεί να εφαρμοστεί ή να μεταφερθεί σε ένα διαφορετικό πρόβλημα ή περιοχή. Στην ουσία, επιτρέπει τη χρήση της υπάρχουσας γνώσης για να βοηθήσει στην επίλυση νέων προβλημάτων ή να βελτιώσει τις επιδόσεις σε κάποιον άλλο τομέα.

Ένας από τους κύριους τρόπους που μπορεί να εφαρμοστεί είναι η μεταφορά γνώσης μέσω της αναλογίας. Σε αυτήν την περίπτωση, αξιοποιούνται οι ομοιότητες μεταξύ δύο προβλημάτων ή δύο πεδίων γνώσης και εφαρμόζεται η γνώση που έχει αποκτηθεί στο ένα για να βοηθήσει στην επίλυση του άλλου. Για παράδειγμα, η γνώση της οδήγησης ενός αυτοκινήτου, μπορεί να εφαρμοστεί για την μάθηση της οδήγησης ενός μοτοποδήλατου. Ένας άλλος τρόπος μεταφοράς γνώσης είναι η μεταφορά μέσω της απαραίτητης γνώσης. Σε αυτήν την περίπτωση, η γνώση που έχει αποκτηθεί σε ένα πρόβλημα χρησιμοποιείται για να ανακτηθεί η απαραίτητη γνώση που απαιτείται για την επίλυση ενός νέου προβλήματος. Για παράδειγμα, η ανάπτυξη ενός συστήματος αναγνώρισης προτύπων για την αναγνώριση γραφής, η γνώση αυτή, μπορεί να χρησιμοποιηθεί για την άντληση απαραίτητων πληροφοριών για την αναγνώριση κειμένου από φωτογραφίες.

Η μεταφορά γνώσης είναι επίσης συχνά συνδεδεμένη με τη χρήση τεχνητών νευρωνικών δικτύων και αλγορίθμων μηχανικής μάθησης. Με τη χρήση αυτών των τεχνικών, η γνώση μπορεί να αναλυθεί και να αντιπροσωπευθεί ως μοντέλα που μπορούν να μεταφερθούν, τροποποιηθούν και επαναχρησιμοποιηθούν σε διάφορα προβλήματα.

4.5.5 T5-RECIPE-GENERATOR

Το T5-RECIPE-GENERATOR χρησιμοποιεί την τεχνική της μεταφοράς μάθησης (transfer learning). Αρχικά, το βασικό μοντέλο T5 εκπαιδεύεται σε έναν ευρύ φάσμα γλωσσικών εργασιών μέσω μεγάλων συνόλων δεδομένων.

Μετά την εκπαίδευση του βασικού μοντέλου, το T5-RECIPE-GENERATOR εκπαιδεύεται περαιτέρω σε ένα ειδικευμένο σύνολο δεδομένων που αφορά συνταγές. Αυτό το ειδικευμένο σύνολο δεδομένων περιέχει συνταγές και τις αντίστοιχες περιγραφές τους. Κατά την εκπαίδευση, το μοντέλο προσπαθεί να μάθει τη συσχέτιση μεταξύ της περιγραφής μιας συνταγής και της ίδιας της συνταγής, έτσι ώστε να μπορεί να δημιουργήσει νέες συνταγές με βάση την περιγραφή που του δίνεται.

Με αυτόν τον τρόπο, το T5-RECIPE-GENERATOR αξιοποιεί τη γενική γνώση που έχει αποκτήσει από την εκπαίδευση σε άλλες γλωσσικές εργασίες και την ειδικεύει στην παραγωγή συνταγών. Αυτή η μεταφορά μάθησης του επιτρέπει να παράγει νέες συνταγές με βάση τις προηγούμενες γνώσεις που έχει αποκτήσει.

Απόδοση του ρ :

Το T5-RECIPE-GENERATOR έχει επιδείξει εντυπωσιακή απόδοση στη δημιουργία νέων συνταγών. Οι παραγόμενες συνταγές είναι συνήθως ποιοτικές και ακολουθούν τις προδιαγραφές που δίνονται στην περιγραφή. Το μοντέλο είναι σε θέση να αναπαράγει συνταγές με ακρίβεια και να παράγει ποικιλία αποτελεσμάτων, προσφέροντας διάφορες πιθανές εκδοχές μιας συνταγής βάσει της περιγραφής που του δίνεται.

Model	COSIM	WER	ROUGE-2	BLEU	METEOR
RecipeNLG	0.5723	1.2125	0.1354	0.1164	0.2309
T5-RECIPE-GENERATOR	0.7282	0.7613	0.2470	0.3245	0.4150

Πίνακας 4.2: Μετρικές αξιολόγησης απόδοσης μοντέλου από [29]

Στον πίνακα 4.2 αναγράφονται μετρικές που χρησιμοποιούνται συνήθως για την αξιολόγηση της απόδοσης μοντέλων επεξεργασίας φυσικής γλώσσας ειδικά σε εργασίες όπως μετάφραση μηχανής, συνοπτική ανάλυση κειμένου και δημιουργία κειμένου.

1. COSIM (Ομοιότητα συνημίτονου, Cosine Similarity) [31]: Μετρά το συνημίτονο της γωνίας μεταξύ δύο διανυσμάτων, προσδιορίζοντας την ομοιότητα μεταξύ τους. Στο πλαίσιο του NLP, χρησιμοποιείται συχνά για να μετρήσει την ομοιότητα μεταξύ των προβλέψεων και των αναφορών.
2. WER (Ποσοστό σφάλματος λέξης, Word Error Rate) [35]: Υπολογίζει το ποσοστό των λανθασμένων ή λείποντων λέξεων στο προβλεπόμενο κείμενο σε σχέση με το αναφερόμενο κείμενο. Χαμηλότερες τιμές υποδεικνύουν καλύτερη απόδοση.
3. ROUGE-2 [30]: Μετρά την επικάλυψη των 2-gram (διαδοχικές ακολουθίες δύο λέξεων) μεταξύ του προβλεπόμενου και του αναφερόμενου κειμένου. Χρησιμοποιείται συχνά για να αξιολογήσει την ποιότητα των συστημάτων περίληψης κειμένου ή των αποτελεσμάτων μηχανικής μετάφρασης.
4. BLEU (διαγλωσσική αξιολόγηση υπολοίπων, Bilingual Evaluation Understudy) [30]: Μετρά την ακρίβεια των n-gram (διαδοχικές ακολουθίες n λέξεων) στο προβλεπόμενο κείμενο σε σχέση με το αναφερόμενο κείμενο. Το BLEU χρησιμοποιείται ευρέως στις εργασίες μηχανικής μετάφρασης για να αξιολογήσει την ομοιότητα μεταξύ των προβλέψεων και των επιθυμητών μεταφράσεων.
5. METEOR (Μέτρηση για την αξιολόγηση της μετάφρασης με ρητή παραγγελία, Metric for Evaluation of Translation with Explicit ORdering) [34]: είναι μια αυτόματη μετρική για την αξιολόγηση της μηχανικής μετάφρασης που προτάθηκε για πρώτη φορά από τους Banerjee και

Lavie το 2005. Το METEOR βασίζεται στην ιδέα της αντιστοίχισης n-gram μεταξύ της μηχανής-παραγόμενης μετάφρασης και των ανθρώπινων αναφορών μεταφράσεων. Ωστόσο, το METEOR λαμβάνει επίσης υπόψη την πτύχωση λέξεων, τη συνωνυμία λέξεων και τη σειρά των λέξεων στη μετάφραση.

Το T5-RECIPE-GENERATOR υποστηρίζει προηγμένες λειτουργίες και χαρακτηριστικά που επεκτείνουν την λειτουργικότητά του. Μερικά παραδείγματα περιλαμβάνουν:

1. **Προσαρμογή των συνταγών ανάλογα με τις μερίδες:** Μπορεί να προσαρμόσει αυτόματα τις συνταγές ανάλογα με τον αριθμό των μερίδων που επιθυμεί ο χρήστης. Έτσι, μπορεί να προσαρμόσει την ποσότητα των συστατικών και τη διαδικασία μαγειρέματος, παρέχοντας συνταγές για οποιοδήποτε αριθμό μερίδων.
2. **Προτάσεις υποκατάστατων συστατικών:** Το μοντέλο μπορεί να προτείνει υποκατάστατα συστατικών για τις συνταγές, εάν κάποιο συστατικό δεν είναι διαθέσιμο ή δεν επιθυμείται. Αυτό επιτρέπει στους χρήστες να προσαρμόσουν τις συνταγές σύμφωνα με τις ανάγκες τους ή τη διαθεσιμότητα των συστατικών.
3. **Διαχείριση διαιτητικών προτιμήσεων:** Είναι ικανό να λαμβάνει υπόψη διαιτητικές προτιμήσεις των χρηστών, όπως χορτοφαγική, vegan, χαμηλής περιεκτικότητας σε γλουτένη, και να προσφέρει συνταγές που ταιριάζουν σε αυτές τις προτιμήσεις. Έτσι, εξασφαλίζει ότι οι χρήστες λαμβάνουν προτάσεις συνταγών που είναι συμβατές με τη διατροφή τους.

ΚΕΦΑΛΑΙΟ 5: ΑΞΙΟΛΟΓΗΣΗ

Στόχος του τελικού αποτελέσματος του συστήματος είναι μια εφαρμογή φιλική προς τον χρήστη στην οποία θα μπορεί εύκολα και γρήγορα να δέχεται απαντήσεις για συνταγές που επιθυμεί να φτιάξει.

5.1 ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Φτάνοντας στο τελικό στάδιο της εφαρμογής δόθηκαν κάποιες είσοδοι και η εφαρμογή ανταποκρίθηκε ως εξής:

Χρήση 1^η

Υλικά: Olive oil, Tomatoes, Cucumbers, Feta cheese, Red onions, Kalamata olives, Lemon juice, Oregano, Garlic, Chickpeas,

Αποτέλεσμα:

TITLE: Caesar salad

INGREDIENTS:

- 1: 1 tablespoon olive oil to taste
- 2: 3 cups lettuce shredded, romaine
- 3: 2 cups tomatoes cucumbers, chopped into 1/2 inch cubes
- 4: 1 cup feta crumbled
- 5: 3/4 cups chopped red onions 1/4 cup
- 6: 1/2 cups diced green or red capsicum
- 7: 1/2 cups chopped pitted olives
- 8: 1/4 cups lemon juice
- 9: 1/4 teaspoons dried oregano garlic chickpeas

DIRECTIONS:

- 1: Mix all ingredients together in a bowl.
- 2: Season with salt and pepper.
- 3: Cover and chill in the refrigerator for at least 30 minutes.
- 4: Toss well before serving.

Δοκιμή με διάφορα μεσογειακά υλικά, φαίνεται να έχει σωστή ανταπόκριση.

Χρήση 2^η

Υλικά: Ground beef Tortillas Black beans Corn Bell peppers Jalapenos Cheddar cheese Salsa Cumin Chili powder

Αποτέλεσμα:

TITLE: Salsa rellenos

INGREDIENTS:

- 1: Ground beef tofu
- 2: Ground beef tofu
- 3: Ground black beans corn bell peppers jalapenos cabots black beans, chopped into 1/2 inch cubes
- 4: Grated cheese and crumbled parmesan
- 5: Spices for salsa

DIRECTIONS:

- 1: Make the salsa, add cheese and breadcrumbs and mix well.
- 2: Spread on hamburger bun and top with a dollop of salsa and crumbled parmesan cheese.
- 3: Dip in salsa.
- 4: Enjoy

Δοκιμή με διάφορα υλικά για μεξικάνικο φαγητό χωρίς κόμματα. Φαίνεται πως ενώ προτάθηκε μια μεξικάνικη συνταγή, το μοντέλο δεν ήταν σε θέση να διαχωρίσει πλήρως τα υλικά.

Χρήση 3^η

Δοκιμή χωρίζοντας τα υλικά με κόμματα.

Υλικά: Ground beef, Tortillas, Black beans, Corn Bell, peppers, Jalapenos, Cheddar cheese, Salsa Cumin, Chili powder

Αποτέλεσμα:

TITLE: Salsa cheese enchilada casserole

INGREDIENTS:

- 1: 1 lb ground beef
- 2: 2 boxes of swirl n tortellini
- 3: 1 large can of corn
- 4: 1 can black beans
- 5: 1 can of mexi corn
- 6: 3 bell peppers
- 7: 2 jalapenos
- 8: 1 lb sharp cheddar or mexican cheese
- 9: 1 cup salsa
- 10: 1 1/2 tbsp cumin
- 11: 1 1/2 tbsp chili powder

DIRECTIONS:

- 1: Cook ground beef with chili powder and cumin until cooked through.
- 2: Drain grease and set aside.
- 3: In a medium sized saucepan over medium heat, bring black beans to a simmer.
- 4: Add corn and simmer for 5 minutes, set aside.
- 5: In a 9 x 13 baking dish, layer one cup of meat, one cup of cheese, and one cup of salsa.
- 6: Then add a layer of tortellini, then peppers, salsa, and so on until all used up.
- 7: Cover with remaining cheese and bake at 350 for 30 45 minutes.
- 8: Enjoy

Δοκιμή ίδιων υλικών χωρισμένων με κόμματα. Φαίνεται να επιστρέφει ορθή απάντηση

Χρήση 4^η

Υλικά: Chicken breast, Broccoli, Carrots, Soy sauce, Ginger, Garlic, Sesame oil, Green onions, Rice noodles

Αποτέλεσμα:

TITLE: Quick san francisco chicken noodle stir fry

INGREDIENTS:

- 1: 1 lb boneless skinless chicken breast, sliced into thin strips

- 2: 1/2 head broccoli, chopped into bite size pieces
- 3: 1 cup carrots, julienned
- 4: 1/4 cup low sodium soy sauce
- 5: 1 teaspoon ginger
- 6: 1 teaspoon garlic powder
- 7: 1 teaspoon sesame oil
- 8: 3 green onions, sliced into thin rounds
- 9: 1/2 lb rice noodles

DIRECTIONS:

- 1: Slice your chicken into thin strips and set aside.
- 2: Add a little bit of sesame oil to your wok and start to heat up. add chicken and stir fry until no longer pink. then add vegetables and stir fry.
- 3: Add soy sauce, ginger, garlic powder and sesame oil to chicken and vegetables. stir fry for 3 minutes. add the green onion and the rice noodles. stir fry for 2 more minutes.
- 4: Garnish with more green onion, if desired.

Δοκιμή υλικών για ασιατικό φαγητό. Φαίνεται να έχει σωστή απόδοση

Χρήση 5^η

Υλικά: Penne pasta, Italian sausage, Mushrooms, Spinach, Parmesan cheese, Tomato sauce, Basil, Crushed red pepper flakes, Garlic, Olive oil

Αποτέλεσμα:

TITLE: Italian sausage stroganoff

INGREDIENTS:

- 1: 1 packages uncooked pasta
- 2: 1 lb bulk italian sausage bob evans
- 3: 1 packages mushrooms
- 4: 1/2 packages spinachach
- 5: 1/2 packages parmesan cheese
- 6: 1 can tomato sauce
- 7: 1 tsp basil
- 8: 1/2 tsp crushed red pepper flakes
- 9: 1/4 tsp garlic powder
- 10: 1/4 cup olive oil

DIRECTIONS:

- 1: Put pasta in a pot of water and boil for 10 minutes.
- 2: Slice sausage and mushrooms.
- 3: Cook sausage in a skillet until brown.
- 4: Add tomato sauce, basil, pepper flakes, and garlic.
- 5: Let simmer for about 10 minutes.
- 6: Add mushrooms and spinach.
- 7: Simmer 5 minutes.
- 8: Add pasta and stir to combine.
- 9: Sprinkle with parmesan cheese.
- 10: Serve.
- 11: Enjoy.

Δοκιμή με υλικά για ιταλικό φαγητό, φαίνεται να έχει σωστή απόδοση.

Χρήση 6^η

Υλικά: Eggs, Bacon, Oatmeal, Maple syrup, Greek yogurt, Berries,

Αποτέλεσμα:

TITLE: Breakfast cereal burgers

INGREDIENTS:

- 1: 4 eggs
- 2: 4 pieces cooked bacon
- 3: 4 teaspoons oatmeal, ground into fine powder
- 4: 2 tablespoons maple syrup
- 5: 2 tablespoons yogurt, greek style
- 6: 4 pieces cooked ham or ham, slices toasted
- 7: 4 pieces crumbled ritz crackers
- 8: 4 pieces ritz toast, cut into small cubes
- 9: 4 pieces peeled berry medley, sliced
- 10: 4 pieces mixed fresh fruit medley
- 11: 4 pieces napoleon, sliced
- 12: 4 pieces sliced pear, strawberries, drained

DIRECTIONS:

- 1: 1. in a small mixing bowl, whisk eggs with yogurt, honey and bacon until blended. set aside.
- 2: 2. place a sheet of parchment paper on a baking sheet.
- 3: 3. preheat oven to 200 c, 400 f, gas mark 6.

- 4: 4. grease a large, rimmed baking sheet with oil.
- 5: 5. spoon oatmeal onto baking sheet.
- 6: 6. form a well in the middle and pour in egg mixture.
- 7: 7. add ground nasturtiums and stir to combine.
- 8: 8. form a 9x13 rectangle and bake in preheated oven for 25 minutes.
- 9: 9. carefully remove from oven and let cool completely.
- 10: 10. cut into 12 slices.
- 11: 11. arrange on prepared baking sheet and bake for 5 minutes, or until golden brown.
- 12: 12. transfer to a serving platter.
- 13: 13. top with sliced nasturtiums and fruit.

Δοκιμή υλικών για πρωινό. Δεν έχουν οριστεί ποσότητες και το μοντέλο αναλαμβάνει να προτείνει το ίδιο δοσολογίες. Ωστόσο δεν φαίνονται ιδιαίτερα σωστά. Επίσης στις οδηγίες υπάρχει διπλή αρίθμηση.

Χρήση 7^η

Υλικά: 2 Eggs, 4 slices of Bacon, 100gr of Oatmeal, Maple syrup, 1 Greek yogurt, 6 Berries

Αποτέλεσμα:

TITLE: Breakfast breakfast scotch eggs

INGREDIENTS:

- 1: 4 large or 4 small eggs
- 2: 4 4 slices of bacon or 1 slice of ham
- 3: 1 tablespoon 100gr of oatmeal
- 4: 1 tablespoon maple syrup
- 5: 1 tablespoon 1 tablespoon of greek yogurt
- 6: 1 tablespoon of chopped 6 berries

DIRECTIONS:

- 1: Preheat the oven to 200c/400f/gas mark 6.
- 2: Line a baking tray with baking paper.
- 3: Break the eggs into a bowl and whisk them together.
- 4: Place the bacon or ham in a small pan and cover with water.
- 5: Bring to the boil and simmer for about 10 minutes.

- 6: Remove the bacon or ham from the pan and place on the baking tray.
- 7: Pour the egg mixture over the bacon or ham and sprinkle with the oatmeal.
- 8: Bake for about 15 minutes or until the egg is cooked through.
- 9: Remove from the oven and drizzle with the maple syrup.
- 10: Serve with the yogurt and the chopped berries.

Δοκιμή ίδιων υλικών για πρωινό αλλά με επιθυμητές δοσολογίες. Το μοντέλο φαίνεται να επιστρέφει σωστές οδηγίες.

Χρήση 8^η

Υλικά: 2 Eg, 4 slices of Bacon, 100gr of Oatmeal, 1 Greek yogurt, 6 Berries

Αποτέλεσμα:

TITLE: Two cheese stroganoff

INGREDIENTS:

- 1: 1 slice of 3 cheese 2 slices of 2 slices of 2 cheese, 2 slices of 2 cheese, 2 slices of 2 slices of 2 cheese, 2 slices of 2 cheese, 1 slice of 2 slices of, 1 slice of 2 cheese, 1 slice of 2 cheese, 1 slice of 2 cheese, 1 piece of 2 slices of 2 cheese, 1 piece of 2 cheese, 1 slice of 2 cheese, 1 piece of 2 cheese, 1 piece of 2 cheese, 1 piece of 2 cheese,

Δοκιμή με λανθασμένες λέξεις για ως υλικά. Εδώ το μοντέλο χρειάστηκε περίπου 2:20 λεπτά για να δώσει κάποιο αποτέλεσμα, το οποίο δεν φαίνεται πως είναι λάθος. Στις υπόλοιπες δοκιμές χρειάστηκε κατά μέσο όρο 30 δευτερόλεπτα για την απάντηση από το μοντέλο.

Χρήση 9^η

Υλικά: 2 eggs

Αποτέλεσμα

TITLE: Quick easy fried eggs

INGREDIENTS:

- 1: 1 2 eggs per person

DIRECTIONS:

- 1: Boil eggs in salted water until they are just done.
- 2: You can cook them right away or set in a bowl and refrigerate or freeze them.

- 3: If you re making for brunch, if you re making for lunch, just break the yolks and then stir them together.

Δοκιμή με 1 μόνο υλικό, φαίνεται να ανταποκρίνεται σωστά.

5.2 ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΧΡΗΣΗΣ ΑΠΟ ΧΡΗΣΤΕΣ

Για την αξιολόγηση της εφαρμογής, δόθηκε σε διάφορα άτομα από διαφορετικές ηλικιακές ομάδες και φύλα. Συγκεκριμένα, παρουσιάσαμε την εφαρμογή σε έξι άτομα. Τα άτομα κλήθηκαν να απαντήσουν ερωτήσεις σχετικά με διάφορες πτυχές της εφαρμογής, όπως η λειτουργικότητα, η ευχρηστία, η απόδοση και η αισθητική. Είχαν την ευκαιρία να περιηγηθούν στις διάφορες λειτουργίες της εφαρμογής και να δώσουν τη γνώμη τους μέσω προκαθορισμένων ερωτημάτων και αξιολογήσεων. Είναι σημαντικό να οριστούν οι παραπάνω όροι ώστε να μπορούν να απαντηθούν εύστοχα.

Η **λειτουργικότητα** αναφέρεται στην ικανότητα ενός συστήματος ή μιας εφαρμογής να εκτελεί τις λειτουργίες για τις οποίες έχει σχεδιαστεί. Απαντά στο ερώτημα "Τι μπορεί να κάνει αυτό το σύστημα;". Για παράδειγμα, ένα σύστημα διαχείρισης πελατών πρέπει να μπορεί να διαχειρίζεται την προσθήκη, την ενημέρωση και τη διαγραφή πελατών. Ενώ η εφαρμογή της εργασίας θα πρέπει να επιστρέφει σχετικές συνταγές γευμάτων βάσει δεδομένων υλικών και όχι μόνο.

Η **ευχρηστία** αφορά το πόσο εύκολο είναι για τον χρήστη να χρησιμοποιήσει ένα σύστημα ή μια εφαρμογή. Αφορά τον σχεδιασμό της διεπαφής χρήστη, την οργάνωση των λειτουργιών και τη γενική εμπειρία του χρήστη. Ο στόχος είναι να προσφέρει μια άνετη, αποτελεσματική και ευχάριστη εμπειρία στον χρήστη. Παραδείγματα για βελτίωση της ευχρηστίας μπορεί να είναι η απλότητα της διεπαφής, η κατανοητή δομή του μενού ή η παροχή ευανάγνωστων και κατανοητών μηνυμάτων.

Η **απόδοση** είναι η απόδοση ενός συστήματος ή μιας εφαρμογής σε σχέση με την ταχύτητα, την απόκριση και την αποτελεσματικότητα. Αφορά την αποτελεσματική εκτέλεση των λειτουργιών και τη γενική απόδοση του συστήματος σε σχέση με την κατανάλωση πόρων όπως ο χρόνος, η μνήμη και οι υπολογιστικοί πόροι.

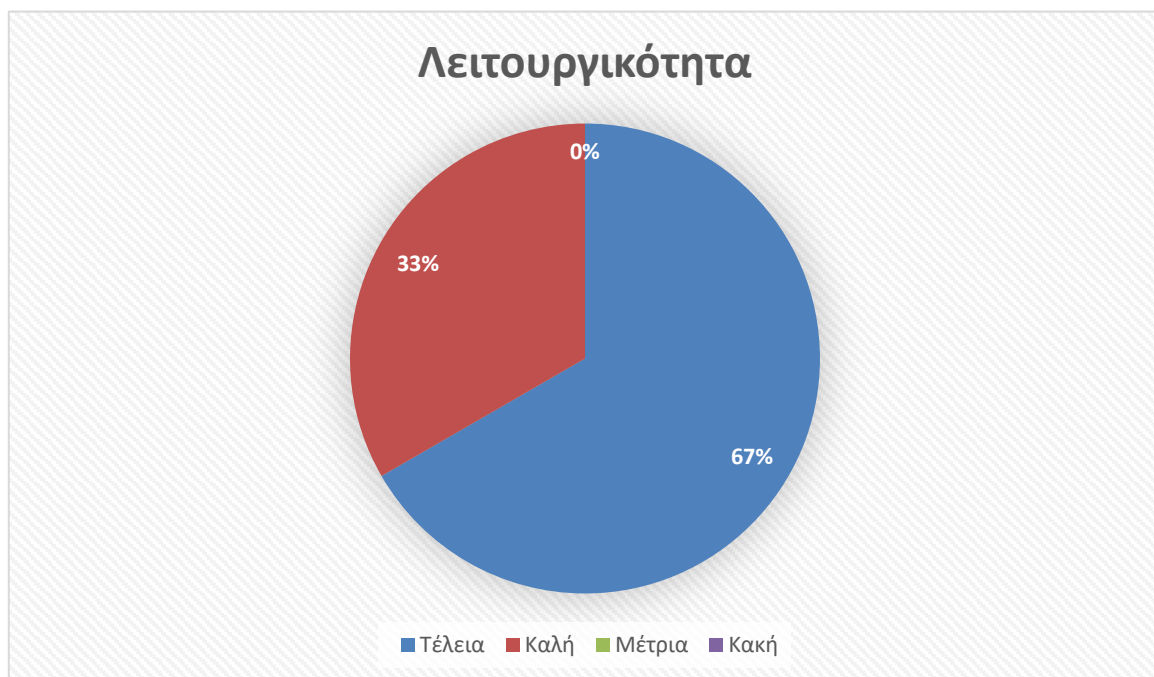
Ως **αισθητική** αναφέρεται στην αντίληψη της ομορφιάς και της αρμονίας σε σχέση με ένα σύστημα ή μια εφαρμογή. Στο πλαίσιο του λογισμικού και της σχεδίασης διεπαφής χρήστη, η αισθητική περιλαμβάνει τον σχεδιασμό του γραφικού περιβάλλοντος, την επιλογή χρωμάτων, τη χρήση γραφικών, τον τρόπο διάταξης των στοιχείων και γενικά την εμφάνιση και την αίσθηση του συστήματος.

5.2.1 ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ

Τα ερωτήματα που θέτονται στο πλαίσιο της εφαρμογής σχετικά με την λειτουργικότητα είναι:

- Το σύστημα μπορεί να αναλάβει την εγγραφή, την σύνδεση και αποσύνδεση των χρηστών χωρίς πρόβλημα;
- Το σύστημα προτείνει μια συνταγή βάσει των δεδομένων υλικών;
- Το σύστημα διατηρεί το ιστορικό των συνταγών και των προτιμήσεων αλλά και την διαγραφή αυτών κατά επιλογήν του χρήστη;

Τέσσερα από τα έξι άτομα έμειναν απόλυτα ικανοποιημένα από την λειτουργικότητα της εφαρμογής ενώ τα υπόλοιπα δύο ενώ έμειναν επαρκώς ικανοποιημένα, θα ήθελαν να έβλεπαν περαιτέρω λειτουργίες και επιλογές. Λειτουργίες που προτάθηκαν ήταν για παράδειγμα η προσθήκη ερωτηματολογίου κατά την εγγραφή με διατροφικές ερωτήσεις για μια πιο προσωποποιημένη εμπειρία. Άλλη λειτουργία ήταν η εφαρμογή να προτείνει παραπάνω από μία συνταγές σε κάθε αναζήτηση ώστε ο χρήστης να μπορεί να διαλέξει ανάμεσα σε αυτές όποια του ταιριάζει καλύτερα.

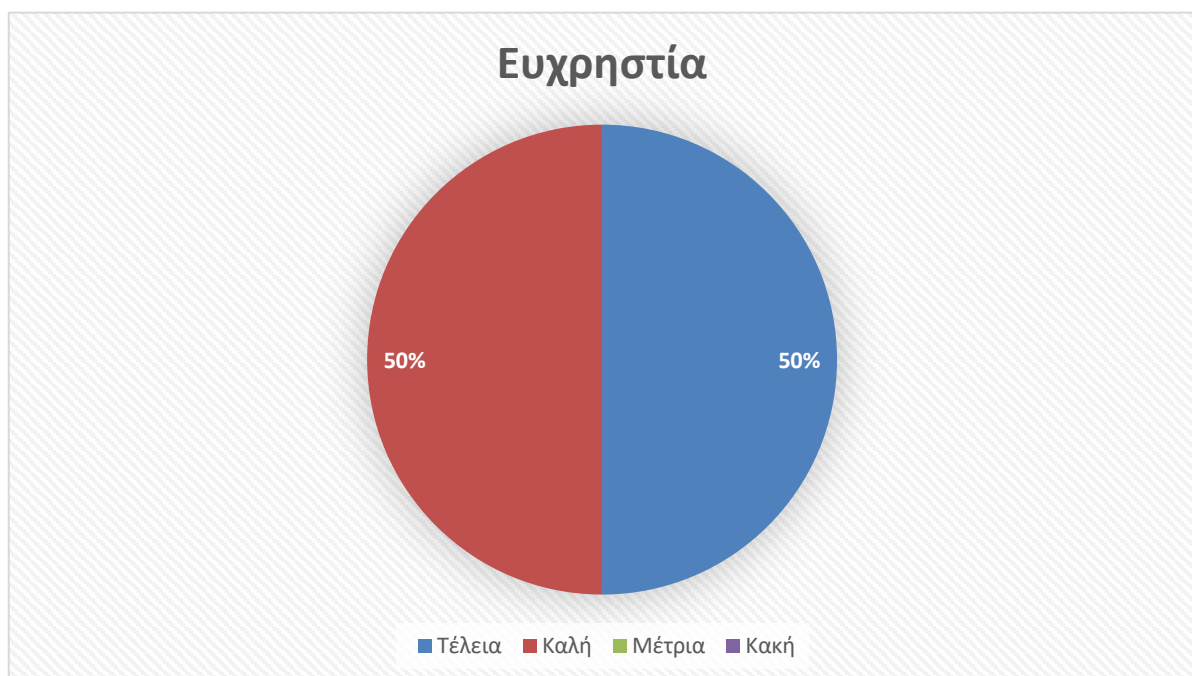


5.2.2 ΕΥΧΡΗΣΤΙΑ

Κάποια από τα βοηθητικά ερωτήματα ως προς τον χρήστη για την ευχρηστία

- Πόσο εύκολο ή δύσκολο ήταν να μάθει ο χρήστης να χρησιμοποιεί την εφαρμογή;
- Ποιες λειτουργίες ή δυνατότητες της εφαρμογής θα ήθελε ο χρήστης να βελτιωθούν ως προς την ευχρηστία;
- Πώς θα αξιολογούσε ο χρήστης τη γενική εμπειρία χρήσης της εφαρμογής;

Ως προς την ευχρηστία τα ποσοστά ήταν ισόβαθμα με τρία άτομα να δηλώνουν ως «τέλεια» την εμπειρία τους σε αυτό το κομμάτι και τα άλλα τρία ως «καλή» με προτάσεις όπως περισσότερα μηνύματα ως προς τον χρήστη από την εφαρμογή για την κατάσταση. Τέτοια είναι τα μηνύματα σφαλμάτων ή ενημέρωσης της προόδου αναζήτησης της συνταγής.

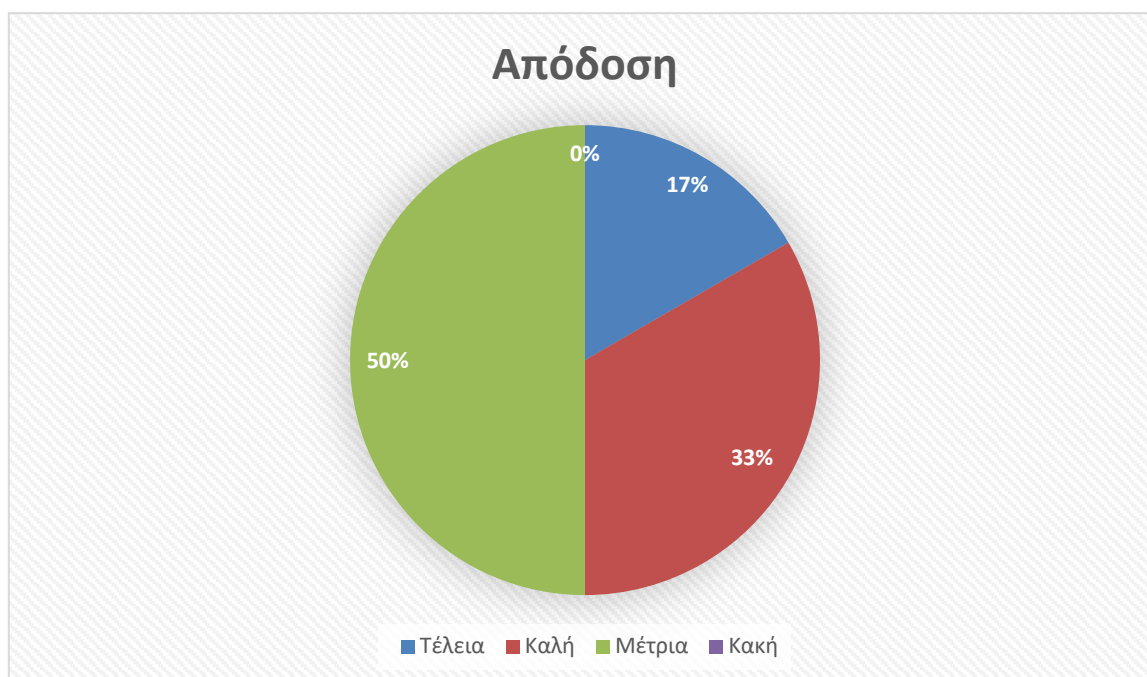


5.2.3 ΑΠΟΔΟΣΗ

- Πόσο ακριβείς και συναφείς ήταν οι προτεινόμενες συνταγές με τα υλικά που εισήχθησαν;
- Πώς θα αξιολογούσατε την ταχύτητα ανταπόκρισης του συστήματος στην πρόταση συνταγών;
- Πώς θα αξιολογούσατε την ποιότητα των πληροφοριών που παρέχονται για κάθε συνταγή, όπως οδηγίες μαγειρέματος, χρόνος προετοιμασίας και υλικά;

- Πώς θα αξιολογούσατε την ποικιλία των προτεινόμενων συνταγών ανάλογα με τα εισαγόμενα υλικά;
- Είχατε κάποια προβλήματα με την απόδοση του συστήματος κατά την αναζήτηση συνταγών;
- Ποιες βελτιώσεις θα προτεινάτε για να κάνετε την εμπειρία χρήσης του συστήματος ακόμη καλύτερη;

Ως προς την απόδοση η εφαρμογή έλαβε μικρότερη αξιολόγηση σε σχέση με τις προηγούμενες. Συγκεκριμένα ένα από τα έξι άτομα αξιολόγησε ως «τέλεια», δύο άτομα ως «καλή» ενώ τα υπόλοιπα τρία ως «μέτρια». Κοινό σχόλιο υπήρχε ο μεγάλος χρόνος αναμονής μέχρι την παραγωγή κάποιας συνταγής αφού αυτός συνήθως ήταν στα 30 δευτερόλεπτα. Άλλο σχόλιο ήταν πως σε κάποιες συνταγές υπήρχαν λανθασμένες λέξεις που δεν έβγαζαν νόημα. Αξίζει να σημειωθεί πως τα άτομα με τις μεγαλύτερες αξιολογήσεις είχαν εισάγει περισσότερα υλικά στην εφαρμογή σε σχέση με τα υπόλοιπα άτομα.

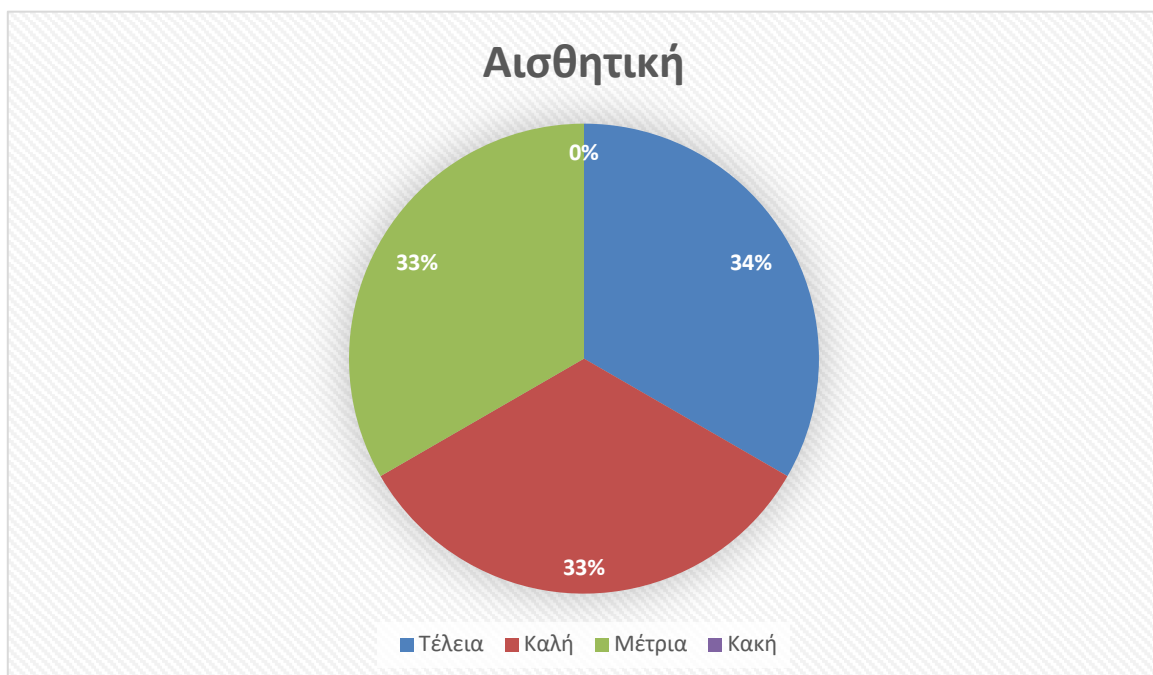


5.2.4 ΑΙΣΘΗΤΙΚΗ

- Πώς θα αξιολογούσατε το γενικό σχεδιασμό της εφαρμογής;
- Πώς θα περιέγραφατε την αισθητική του περιβάλλοντος της εφαρμογής;

- Η χρωματική παλέτα του συστήματος ήταν ευχάριστο και εναρμονισμένο;
- Πώς αξιολογείτε την επιλογή γραμματοσειράς και μεγέθους γραμματοσειράς του συστήματος;
- Πώς θα αξιολογούσατε τη χρήση γραφικών ή εικονογράφησης της εφαρμογής;
- Πώς σας επηρεάζει η αισθητική του συστήματος στην κατανόηση και την απόδοση των πληροφοριών σχετικά με τις συνταγές;
- Πώς θα αξιολογούσατε τον γενικό σχεδιασμό των καρτελών ή των διεπαφών που παρουσιάζουν τις συνταγές;
- Ποιες βελτιώσεις θα προτεινάτε για να βελτιώσετε την αισθητική του συστήματος πρότασης συνταγών βάσει των υλικών;

Για το κομμάτι της αισθητικής υπήρξαν πάλι μικτές αξιολογήσεις με 2 άτομα να αξιολογούν ως «τέλεια», δύο ως «καλή» και δύο ως «μέτρια». Προτάθηκαν αλλαγές στην μικρογραφία της εφαρμογής καθώς και στα πλαίσια σύνδεσης-εγγραφής. Επίσης αλλαγή στην χρωματική παλέτα καθώς και στον τρόπο εμφάνισης της συνταγής. Τέλος η τοποθέτηση των μενού επιλογής για το ιστορικό ή των ρυθμίσεων θα μπορούσε να είναι σε διαφορετικό μέρος και να γίνει χρήση των ίδιων components.



ΚΕΦΑΛΑΙΟ 6: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ

6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Η εφαρμογή Εξατομικευμένη εύρεση γευμάτων είναι ένα παράδειγμα του πόσο εύκολο είναι στις μέρες μας προσθήκη λειτουργιών που παρέχονται από την μηχανική μάθηση σε κάθε διαδικτυακή σελίδα. Τα προ – εκπαιδευμένα μοντέλα καθώς και η εξέλιξη της επεξεργασίας φυσικής γλώσσας διευκολύνουν ιδιαίτερα την δημιουργία τέτοιων εφαρμογών αρκετά πιο γρήγορα από την εκπαίδευση εκ νέου κάποιου μοντέλου.

Η χρήση της React και της Node.js μεταξύ άλλων, έκανε εφικτή την δημιουργία της εφαρμογής σε σχετικά μικρό χρόνο και χωρίς ιδιαίτερα προβλήματα πέρα από αυτών που αναφέρθηκαν στην ενότητα 3.3. Ο συνδυασμός όλων των τεχνολογιών με στόχο το κάτι λειτουργικό και χρήσιμο αποτέλεσε μια ευχάριστη εμπειρία κατά την υλοποίηση της εφαρμογής και προσκομίσθηκαν αρκετές γνώσεις και πρακτικές.

Το κύριο χαρακτηριστικό της εφαρμογής είναι το ίδιο το μοντέλο μηχανικής μάθησης καθώς αποτελεί την καρδιά της ανάκτησης συνταγών. Έτσι το μεγαλύτερο ποσοστό της απόδοσής της οφείλεται στο ίδιο το μοντέλο. Παρατηρήθηκε κατά τις δοκιμές αλλά και στην αξιολόγηση πως στις περιπτώσεις που η είσοδος των υλικών ήταν ποικίλα καθώς και ορθογραφικά σωστά, το σύστημα εμφάνιζε πιο εύστοχες και ορθές συνταγές χωρίς. Σε αντίθετες περιπτώσεις που τα υλικά ήταν χωρίς σωστή ορθογραφία και σύνταξη, το μοντέλο μπερδευόταν και δεν επέστρεφε σωστά κατανοητά αποτελέσματα. Στις περιπτώσεις που τα υλικά ήταν λίγα τότε η απόφαση για την πρόταση της συνταγής ήταν καθαρά στο μοντέλο και θα ήταν κάποια που θα περιείχε το υλικό αυτό (βλέπε τελευταίο παράδειγμα χρήσης της εφαρμογής 5.1).

6.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΞΕΛΙΞΕΙΣ

Οι βελτιώσεις που μπορούν να γίνουν είναι πολλές. Στόχος δεν ήταν μια τέλεια με κάθε λειτουργία διαδικτυακή εφαρμογή αλλά ένα λειτουργικό σύστημα που θα μπορούσε να παρέχει μια βασική απόδοση για τον στόχο καθώς ο συνδυασμός υλοποίησης άρτιων εφαρμογών καθώς και χρήση μηχανικής μάθησης είναι δύο μεγάλα πεδία από μόνα τους.

Σε μελλοντικές εξελίξεις, λαμβάνοντας υπόψη και τα σχόλια στις αξιολογήσεις της εφαρμογής, θα μπορούσε να υπάρξει εστίαση σε βελτίωση του συστήματος τόσο στο frontend, στο backend αλλά και στο μοντέλο.

Συγκεκριμένα στο frontend, βάσει σχολίων, αλλαγές στο σχεδιαστικό της εφαρμογής, για παράδειγμα τα πλαίσια της σελίδας της σύνδεσης και εγγραφής, αλλά και της χρωματικής παλέτας. Προσθήκη αναδυόμενων μηνυμάτων κατάστασης βοηθώντας τους χρήστες να ξέρουν τι πήγε λάθος ή ποια είναι η παρούσα κατάσταση.

Στο κομμάτι του backend, μελλοντικές προσθήκες θα ήταν περεταίρω λειτουργίες στην εφαρμογή όπως πρόταση παραπάνω από 1 συνταγή την φορά καθώς και βελτίωση της ασφάλειας με την εξαναγκασμό του χρήστη στην προσθήκη δυνατών κωδικών προσβάσεων καθώς και χρήση κρυπτογραφίας.

Στο κομμάτι του μοντέλου το κύριο πρόβλημα είναι η αργή απόδοση (30 δευτερόλεπτα κατά μέσο όρο) αλλά και η ανάγκη για απολύτως ορθής εισαγωγής δεδομένων-υλικών προκειμένου μιας σωστής πρότασης συνταγής. Για το πρόβλημα αυτό, μελλοντικά θα μπορούσε το μοντέλο να αντικατασταθεί με κάποιο πιο εξελιγμένο το οποίο θα υποστηρίζει και τα ελληνικά, ή την εκπαίδευση και παραμετροποίηση εξαρχής ενός μοντέλου.

Μια σημαντική λειτουργία στην εφαρμογή θα ήταν η πρόταση συνταγών βάσει παλιών αναζητήσεων. Η εφαρμογή δηλαδή να συνεχίσει να αναζητά συνταγές ακόμη και χωρίς ο χρήστης να έχει εσκεμμένα αναζητήσει κάποια. Με τον τρόπο αυτό αυξάνεται η εξατομίκευση στον εκάστοτε χρήστη. Δεύτερη λειτουργία είναι η προσθήκη αξιολόγησης των συνταγών από τον κάθε χρήστη με αποτέλεσμα οι υψηλές σε βαθμολογίες συνταγές ή παρόμοιες να προτείνονται συχνότερα στους χρήστες αφού θα έχει αξιολογηθεί από τον ίδιο ή και άλλους χρήστες. Με τις παραπάνω αλλαγές η εφαρμογή θα εξελισσόταν σε άλλο επίπεδο και παρείχε μια καλύτερη εμπειρία στους χρήστες.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh, 2015, “Recommendation Systems: Principles, Methods and Evaluation”, Egyptian Informatics Journal, Vol. 16, Issue 3, November, pp. 261–273.
- [2] Michael Aaron Dennis, “Encyclopaedia Britannica”, 2023, <https://www.britannica.com/biography/Tim-Berners-Lee>
- [3] Murugesan S., “Web Application Development: Challenges And The Role Of Web Engineering”, 2008, https://link.springer.com/chapter/10.1007/978-1-84628-923-1_2
- [4] Natesan, Abirami N, “A Detailed Study of Client-Server and its Architecture”, Nov 2019, https://www.researchgate.net/publication/337469571_A_Detailed_Study_of_Client-Server_and_its_Architecture
- [5] Thomas W. Edgar, David O. Manz, “Research Methods for Cyber Security”, 2017
- [6] Beverly Park Woolf, “Building Intelligent Interactive Tutors”, 2009
- [7] Sah, Shagan, “ Machine Learning: A review of Learning Types”, July 2020
- [8] Xu Han, Zhengyan Zhang et al, “**Pre-trained models: Past, present and future**”, **AI Open, 2021**
- [9] Colin Raffel et al., “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”, 2020
- [10] A. M. TURING, “COMPUTING MACHINERY AND INTELLIGENCE”, Mind, Volume LIX, Issue 236, October 1950, Pages 433–460, <https://academic.oup.com/mind/article/LIX/236/433/986238>
- [11] Ernst, G. & Newell, A. “GPS: A Case Study in Generality and Problem Solving. New York: Academic Press.”, 1969
- [12] Chavan, Pallavi Amod, “FeedMeRight-Recipe Recommendation System”, 2020, <https://scholarworks.calstate.edu/concern/theses/pg15bg98f>
- [13] Harry Zisopoulos et al. "Content-Based Recommendation Systems" November 2008

- [14] Estrela, David & Batista, Sergio & Martinho, Diogo & Marreiros, Goreti. (2017). A Recommendation System for Online Courses. 195-204.
- [15] Y. Tian, C. Zhang, R. Metoyer, and N. V. Chawla, “Recipe Recommendation With Hierarchical Graph Attention Network,” *Frontiers*, Dec. 16, 2021. <https://www.frontiersin.org/articles/10.3389/fdata.2021.778417/full>
- [16] Πάβελ, Αντρέι-Έντουαρντ, «Σχεδίαση και ανάπτυξη διαδικτυακής πλατφόρμας διοργάνωσης τουρνουά παιχνιδιών», 2022, <https://polynoe.lib.uniwa.gr/xmlui/handle/11400/3031>
- [17] Μάρκου, Βασίλειος, “Web application using React and Django”, 2021, <https://polynoe.lib.uniwa.gr/xmlui/handle/11400/651>
- [18] React, <https://react.dev/>
- [19] TypeScript, <https://www.typescriptlang.org/>
- [20] Material UI, <https://mui.com/>
- [21] Styled Components, <https://styled-components.com/>
- [22] React Router Dom, <https://reactrouter.com/en/main>
- [23] Mongoose, <https://mongoosejs.com/>
- [24] Axios, <https://axios-http.com/docs/intro>
- [25] Transformers, <https://huggingface.co/docs/transformers/index>
- [26] AutoTokenizer, [Auto Classes \(huggingface.co\)](https://huggingface.co/docs/transformers/index#auto-tokenizer)
- [27] FlaxAutoModelForSeq2SeqLM, [Auto Classes \(huggingface.co\)](https://huggingface.co/docs/transformers/index#flax-auto-model-for-seq2-seq-lm)
- [28] Pytorch, <https://pytorch.org/>
- [29] “flax-community/t5-recipe-generation · Hugging Face,” *flax-community/t5-recipe-generation · Hugging Face*. <https://huggingface.co/flax-community/t5-recipe-generation>
- [30] Kathrin Blagec et al, “A global analysis of metrics used for measuring performance in natural language processing”
- [31] James Briggs, “Similarity Metrics in NLP”, Towards Data Science, Apr 2021, <https://towardsdatascience.com/similarity-metrics-in-nlp-acc0777e234c>

- [32] Priya B et al., “An Analysis of the Applications of Natural Language Processing in Various Sectors”, 2021, https://www.researchgate.net/publication/355332692_An_Analysis_of_the_Applications_of_Natural_Language_Processing_in_Various_Sectors
- [33] Rachel Wolff, “11 NLP Applications & Examples in Business”, 2020, <https://monkeylearn.com/blog/natural-language-processing-applications/>
- [34] Banerjee, Satanjeev, Lavie, Alon “METEOR: A Metric for Evaluation of Machine Translation. In Proceedings of the 4th ACL Workshop on Statistical Machine Translation”, 2005,
- [35] Derouet M, Renals S. “A new approach to word error rate estimation in speech recognition”, 1996
- [36] CERN, "A short history of the Web," CERN, [Online]. Available: <https://home.cern/science/computing/birth-web/short-history-web>.
- [37] S. Fowler and V. Stanwick, Web Application Design Handbook: Best Practices for Web-Based Software. San Francisco, CA, USA: Morgan Kaufmann, 2004.
- [38] G. Rossi, O. Pastor, D. Schwabe, and L. Olsina, Eds., Web Engineering: Modelling and Implementing Web Applications. London, UK: Springer, 2008.
- [39] B. M. Al Saeed, M. Alsmadi et al., "Designing and Developing A Web Application for Tourism," International Journal of Scientific Research in Science and Technology, 2020.
- [40] S. Murugesan, "Web Application Development: Challenges and the Role of Web Engineering," [Online]. Available: https://doi.org/10.1007/978-1-84628-923-1_2
- [41] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," RFC 2616, June 1999. [Online]. Available: <https://dl.acm.org/doi/abs/10.17487/RFC2616>
- [42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019

- [43] A. Vaswani *et al.*, “Attention Is All You Need,” *arXiv.org*, Jun. 12, 2017. <https://arxiv.org/abs/1706.03762v5>
- [44] “Papers with Code - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding | Papers With Code*. <https://paperswithcode.com/paper/bert-pre-training-of-deep-bidirectional>
- [45] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 1997. <https://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.8.1735>
- [46] “Home v6.11.2,” *Home v6.11.2 | React Router*. <https://reactrouter.com/en/main>
- [47] “flax-community/t5-recipe-generation Hugging Face,” <https://huggingface.co/flax-community/t5-recipe-generation>
- [48] D. Khurana, A. Koli, K. Khatter, and S. Singh, “Natural language processing: state of the art, current trends and challenges,” *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 3713–3744, Jul. 2022, doi: 10.1007/s11042-022-13428-4.
- [49] “The Power of Natural Language Processing,” *Harvard Business Review*, Apr. 19, 2022. <https://hbr.org/2022/04/the-power-of-natural-language-processing>
- [50] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 328-339.
- [51] A. Le Glaz, Y. Haralambous, D.-H. Kim-Dufor, P. Lenca, R. Billot, T. C. Ryan, J. Marsh, J. DeVyllder, M. Walter, S. Berrouguet, and C. Lemey, "Machine Learning and Natural Language Processing in Mental Health: Systematic Review," *JMIR Publications*, vol. 23, no. 5, May 2021.
- [52] A. F. Smeaton, “Using NLP or NLP Resources for Information Retrieval Tasks,” *Text, Speech and Language Technology*, pp. 99–111, 1999, doi: 10.1007/978-94-017-2388-6_4.
- [53] R. van de Schoot, S. Depaoli, R. King, et al., "Bayesian statistics and modelling," *Nat. Rev. Methods Primers*, vol. 1, no. 1, p. 1, 2021. doi: 10.1038/s43586-020-00001-2.
- [54] Hochreiter S, Schmidhuber J. “Long short-term memory. *Neural Computation*”, 1997

