

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

ΔΗΜΙΟΥΡΓΙΑ ΨΗΦΙΑΚΟΥ ΒΟΗΘΟΥ ΓΙΑ ΤΟ ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εισηγητής: Ιωάννης Βλαχάκης, ΔΕ20420

Επιβλέπων: Κωνσταντίνος Παναγιωτάκης, Καθηγητής

©

2026



HELLENIC MEDITERRANEAN UNIVERSITY

SCHOOL OF MANAGEMENT AND ECONOMICS SCIENCE

DEPARTMENT OF MANAGEMENT SCIENCE AND TECHNOLOGY

DEVELOPING A DIGITAL ASSISTANT FOR THE DEPARTMENT OF MANAGEMENT SCIENCE AND TECHNOLOGY

DIPLOMA THESIS

Student : Ioannis Vlahakis, DE20420

Supervisor : Konstantinos Panagiotakis, Professor

©

2026

Υπεύθυνη Δήλωση *: Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Διοικητικής Επιστήμης και Τεχνολογίας του ΕΛ.ΜΕ.ΠΑ.*

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία πραγματεύεται τον σχεδιασμό και την υλοποίηση ψηφιακού βοηθού για το Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας του ΕΛΜΕΠΑ. Στόχος είναι η αυτοματοποιημένη παροχή ακαδημαϊκών και διοικητικών πληροφοριών στους φοιτητές. Η λύση αναπτύχθηκε με νέο κώδικα σε Python και FastAPI, με διεπαφή σε HTML, CSS και JavaScript, και στρώμα γνώσης σε SQLite. Η βασική λογική του συστήματος βασίζεται σε κανονικοποίηση εισόδου, κανόνες ανίχνευσης πρόθεσης, ιεραρχημένη δρομολόγηση αιτημάτων και ελαφρά διαχείριση κατάστασης για στοχευμένες διευκρινίσεις. Η βάση περιλαμβάνει δεδομένα οδηγού σπουδών, διαδικασιών και στοιχείων επικοινωνίας, ενώ η παρουσίαση υποστηρίζει επιλογές και ασφαλείς συνδέσμους. Η αξιολόγηση κάλυψε δοκιμές μονάδας και ολοκλήρωσης, ελέγχους ορθότητας αποκρίσεων και βασική ανθεκτικότητα σε σφάλματα επικοινωνίας. Τα αποτελέσματα δείχνουν ότι η κανονοκεντρική προσέγγιση είναι κατάλληλη για καλά ορισμένα ερωτήματα και προσφέρει διαφάνεια και εύκολη συντήρηση, με περιορισμούς σε ασαφείς διατυπώσεις που απαιτούν ενίσχυση της βάσης γνώσης και της ανίχνευσης πρόθεσης.

Λέξεις Κλειδιά : Ψηφιακός βοηθός, Chatbot, Επεξεργασία Φυσικής Γλώσσας, FastAPI, SQLite

ABSTRACT

This diploma thesis presents the design and implementation of a digital assistant for the Department of Management Science and Technology of Hellenic Mediterranean University (HMU). The primary objective is the automated provision of academic and administrative information to students. The system is implemented in Python with FastAPI, a web interface in HTML, CSS and JavaScript, and a SQLite knowledge repository. Business logic relies on input normalization, rule based intent detection, hierarchical request routing, and lightweight dialogue state for clarifications. The database stores information from the study guide, procedures and contact details, while the interface supports option buttons and safe link rendering. Evaluation included unit and integration checks, validation of response correctness for frequent queries, and resilience to communication failures. Results indicate that the rule centered approach is effective and transparent for well defined questions, while ambiguous inputs highlight the need for broader coverage, improved intent recognition and systematic expansion of the knowledge base.

Key Words: Digital assistant, Chatbot, Natural Language Processing, FastAPI, SQLite

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ 1

ABSTRACT 2

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ 3

Λίστα εικόνων ..... **Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.**

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ 8

ΕΙΣΑΓΩΓΗ 9

ΚΕΦΑΛΑΙΟ 1 12

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ 12

1.1 Ορισμός και Ιστορική Εξέλιξη της ΤΝ 12

1.2 Μηχανική Μάθηση 15

1.3 Επεξεργασία Φυσικής Γλώσσας (NLP) 16

1.4 Η Εφαρμογή της ΤΝ στην Τριτοβάθμια Εκπαίδευση 17

ΚΕΦΑΛΑΙΟ 2 20

ΨΗΦΙΑΚΟΙ ΒΟΗΘΟΙ 20

2.1 Ορισμός Ψηφιακών Βοηθών (Chatbots & Virtual Assistants) 20

2.2 Κατηγορίες Chatbots 21

2.3 Αρχιτεκτονική των Chatbots....23

2.4 Πλεονεκτήματα και Περιορισμοί....25

ΚΕΦΑΛΑΙΟ 3 28

ΔΗΜΙΟΥΡΓΙΑ ΨΗΦΙΑΚΟΥ ΒΟΗΘΟΥ ΓΙΑ ΤΟ ΤΜΗΜΑ ΔΕΤ 28

3.1 Σκοπός και Ανάλυση Απαιτήσεων 28

3.2 Σχεδίαση Συστήματος....30

3.3 Τεχνολογίες και Εργαλεία Ανάπτυξης....38

3.4 Υλοποίηση και Διασφάλιση Ποιότητας....44

3.5 Δοκιμές και Αξιολόγηση 49

ΚΕΦΑΛΑΙΟ 4 51

ΣΥΜΠΕΡΑΣΜΑΤΑ 51

4.1 Συμπεράσματα 51

4.2 Περιορισμοί και Προβλήματα 52

4.3 Προτάσεις για Μελλοντική Εξέλιξη 53

ΒΙΒΛΙΟΓΡΑΦΙΑ 55

Α. ΞΕΝΟΓΛΩΣΣΗ 55

Β. ΕΛΛΗΝΙΚΗ....59

ΠΑΡΑΡΤΗΜΑ Α....61

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

Εικόνα 1: Σύγκριση αποτελεσμάτων αναζήτησης (δημιουργία με χρήση gemini).....12

Εικόνα 2: Χρονοδιάγραμμα Παγκόσμιας Χρήσης AI(δημιουργία με χρήση gemini) 13

Εικόνα 3: Βασικές κατηγορίες της Μηχανικής Μάθησης.....15

Εικόνα 4: Απεικόνιση της Επεξεργασίας Φυσικής Γλώσσας (NLP) (δημιουργία με χρήση gemini).....16

Εικόνα 5: Διάγραμμα διάσπασης του NLP σε NLU και NLG (δημιουργία με χρήση gemini) 17

Εικόνα 6: Σχηματική απεικόνιση των βασικών λειτουργιών του ψηφιακού βοηθού (δημιουργία με χρήση gemini notebooklm).....18

Εικόνα 7: Κατηγορίες Chatbot (δημιουργία με χρήση gemini).....21

Εικόνα 8: Αρχιτεκτονική των Chatbots (δημιουργία με χρήση gemini) 23

Εικόνα 9 : Πλεονεκτήματα και Περιορισμοί (δημιουργία με χρήση gemini) 26

Εικόνα 10: Ανάλυση σκοπού και απαιτήσεων (δημιουργία με χρήση gemini).....28

Εικόνα 11: UI του chatbot 30

Εικόνα 12: Τμήμα κώδικα HTML & CSS.....31

Εικόνα 13: Τμήμα κώδικα detect intent.....32

Εικόνα 14: Κατάσταση pending 33

Εικόνα 15: Αλλαγή θεματολογίας 34

Εικόνα 16: Πίνακες της Βάσης Δεδομένων.....35

Εικόνα 17: Εγγραφές του Πίνακα det_course από τον οδηγό σπουδών.....36

Εικόνα 18: Scrollable συζήτηση (α).....37

Εικόνα 19: Scrollable συζήτηση (β).....37

Εικόνα 20: Παράδειγμα εφαρμογής FastAPI(κώδικας + συνθεση δύο εικόνων με χρήση gemini).....39

Εικόνα 21: Βάση δεδομένων και εργαλεία υποδομής (δημιουργία με χρήση gemini).....44

Εικόνα 22: detect_indent test (unit test).....47

Εικόνα 23: parsing test(unit test).....47

Εικόνα 24: Api_chat test (integration tests).....48

Εικόνα 25: Παράδειγμα χρήσης Ψηφιακού βοηθού(α).....50

Εικόνα 26: Παράδειγμα χρήσης Ψηφιακού βοηθού (β).....50

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

Ελληνικές συντομογραφίες

ΔΕΤ – Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας

ΕΛ.ΜΕ.ΠΑ – Ελληνικό Μεσογειακό Πανεπιστήμιο

ΤΝ – Τεχνητή Νοημοσύνη

Αγγλικές συντομογραφίες

Ajax – Asynchronous JavaScript and XML

API – Application Programming Interface

CSS – Cascading Style Sheets

DBMS – Database Management System

ELK – Elasticsearch, Logstash, Kibana

FAQ – Frequently Asked Questions

HTML – HyperText Markup Language

HTTP – Hypertext Transfer Protocol

JSON – JavaScript Object Notation

NLP – Natural Language Processing

POMDP – Partially Observable Markov Decision Process

REST – Representational State Transfer

SQL – Structured Query Language

UI – User Interface

UX – User Experience

ΕΙΣΑΓΩΓΗ

Η Τεχνητή Νοημοσύνη (ΤΝ) έχει εξελιχθεί ραγδαία τα τελευταία χρόνια και θεωρείται μια από τις σημαντικότερες τεχνολογικές εξελίξεις αυτού του αιώνα που έχει φέρει επανάσταση στη λειτουργία οργανισμών και εκπαιδευτικών ιδρυμάτων (Russell and Norvig, 2021). Ειδικότερα, οι τεχνολογίες αυτοματοποιημένης απόκρισης αποτελούν πλέον καταλυτικό παράγοντα για τη βελτιστοποίηση της διάχυσης της πληροφορίας και τον περιορισμό των γραφειοκρατικών διαδικασιών.

Τα Ανώτατα Εκπαιδευτικά Ιδρύματα αντιμετωπίζουν έναν διαρκώς αυξανόμενο όγκο πληροφοριών που σχετίζονται με πολυδιάστατες ακαδημαϊκές και διοικητικές διαδικασίες. Η επέκταση των ψηφιακών υπηρεσιών και η συσσώρευση κανονιστικών δεδομένων ενισχύουν την ανάγκη για αποτελεσματικούς μηχανισμούς υποστήριξης (Selwyn, 2016). Οι ευφυείς ψηφιακοί βοηθοί είναι σε θέση να ενισχύσουν την επιχειρησιακή αποτελεσματικότητα (Zawacki-Richter et al., 2019).

Στο Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας (ΔΕΤ), οι φοιτητές αναζητούν συνήθως πληροφορίες για το πρόγραμμα σπουδών, τις δηλώσεις μαθημάτων και τις εξετάσεις. Η πληροφόρηση αυτή, παρέχεται συνήθως μέσω της επίσημης ιστοσελίδας του τμήματος ή της γραμματείας, γεγονός που δεν εξασφαλίζει πάντα άμεση και εύκολη πρόσβαση. Λαμβάνοντας υπόψη τα παραπάνω, η υιοθέτηση συστημάτων αυτοματοποιημένης επικοινωνίας κρίνεται απόλυτα συμβατή με τη φυσιογνωμία του Τμήματος.

Σκοπός της παρούσας πτυχιακής εργασίας είναι η δημιουργία ενός ψηφιακού βοηθού για το Τμήμα ΔΕΤ, ο οποίος θα λειτουργεί ως ένα αποτελεσματικό εργαλείο διασύνδεσης, παρέχοντας αξιόπιστες απαντήσεις σε συχνά ερωτήματα που έχουν οι φοιτητές. Η υλοποίηση της διαδικασίας αποσκοπεί στη βελτίωση της παροχής ακαδημαϊκών και διοικητικών πληροφοριών προς τους φοιτητές, μέσω μιας αυτοματοποιημένης και άμεσα προσβάσιμης ψηφιακής υπηρεσίας.

Για την επίτευξη του σκοπού αυτού, τίθενται οι εξής επιμέρους στόχοι:

- Η ανάλυση των πληροφοριακών αναγκών που χρειάζονται οι φοιτητές, σύμφωνα με τη λήψη ακαδημαϊκών και διοικητικών πληροφοριών.
- Η συλλογή και οργάνωση δεδομένων με βάση τις πηγές του Τμήματος (οδηγός σπουδών, κανονισμοί και εξετάσεις).
- Η σχεδίαση και την υλοποίηση του ψηφιακού βοηθού με την χρήση των κατάλληλων τεχνολογιών εργαλείων ανάπτυξης.
- Η εκπαίδευση του συστήματος στη διαχείριση ερωτήσεων φυσικής γλώσσας.
- Η αξιολόγηση της αποτελεσματικότητας του βοηθού ως προς την εγκυρότητα των απαντήσεων.
- Και η δημιουργία προτάσεων για μελλοντική βελτίωση και επέκταση του προγράμματος.

Για την υλοποίηση του έργου, επιλέχθηκε η προσέγγιση της ιδιόκτητης ανάπτυξης κώδικα (custom development), διασφαλίζοντας με αυτόν τον τρόπο τον πλήρη έλεγχο της αρχιτεκτονικής του συστήματος και τη δυνατότητα μελλοντικής επεκτασιμότητας. Ο πυρήνας του βοηθού αναπτύχθηκε σε γλώσσα Python, αξιοποιώντας εξειδικευμένες βιβλιοθήκες για την λογική των συνομιλιών, ενώ για τη διαχείριση δεδομένων χρησιμοποιήθηκε η βάση SQLite Browser Database. Η διεπαφή χρήστη (front-end), σχεδιάστηκε με τις τεχνολογίες HTML, CSS και JavaScript, προσφέροντας μια σύγχρονη και διαδραστική εμπειρία περιήγησης.

Η παρούσα εργασία δομείται σε τέσσερα κεφάλαια, ακολουθώντας μια πορεία από την θεωρητική ανάλυση της ΤΝ έως τεχνολογική υλοποίηση του έργου. Στο πρώτο κεφάλαιο, αναλύεται ο ορισμός της ΤΝ και η σύντομη ιστορική εξέλιξη, καθώς και η ενσωμάτωση της στην Τριτοβάθμια Εκπαίδευση. Το δεύτερο κεφάλαιο, αναφέρεται στα συστήματα ψηφιακών βοηθών και πιο συγκεκριμένα στις κατηγορίες τους, στα πλεονεκτήματα αλλά και στους περιορισμούς που παρέχουν τα συστήματα αυτοματοποιημένης εξυπηρέτησης. Στο τρίτο κεφάλαιο παρουσιάζεται αναλυτικά η διαδικασία δημιουργίας του εξατομικευμένου βοηθού για το Τμήμα ΔΕΤ αλλά και ο σκοπός της, καθώς και τα εργαλεία και τεχνολογίες που

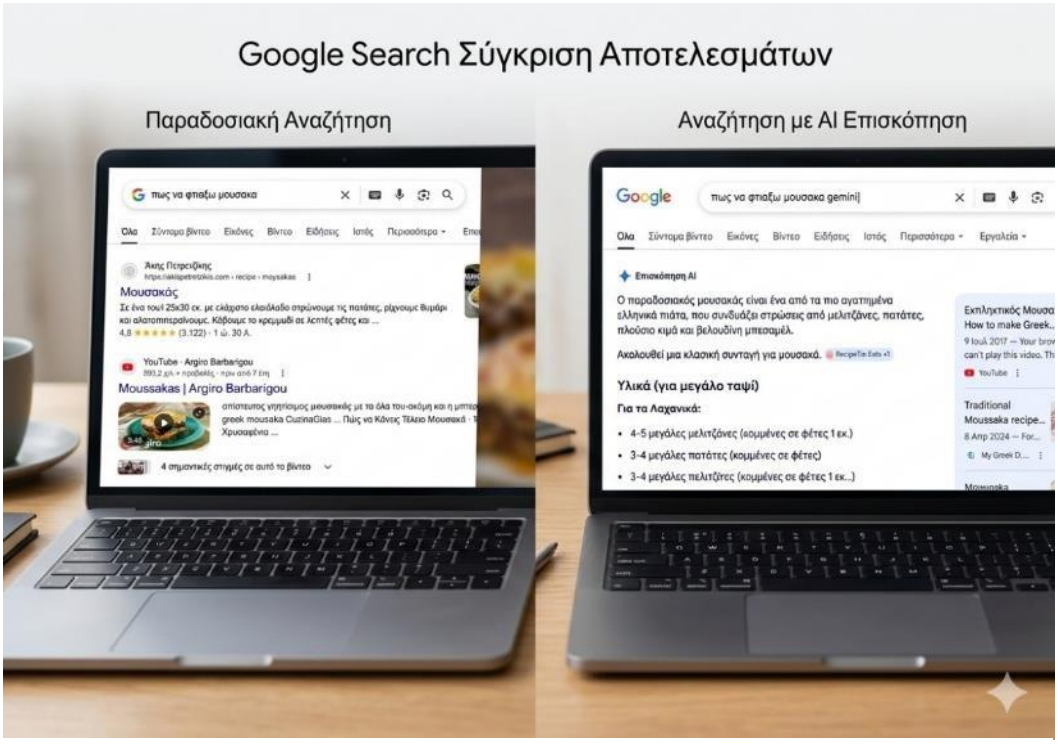
χρησιμοποιήθηκαν. Τέλος, η μελέτη ολοκληρώνεται με την παράθεση των τελικών συμπερασμάτων και προτάσεων για μελλοντική ανάπτυξη.

ΚΕΦΑΛΑΙΟ 1

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

1.1 Ορισμός και Ιστορική Εξέλιξη της ΤΝ

Όπως αναφέρθηκε στην Εισαγωγή, η ΤΝ έχει σημειώσει σημαντική ανάπτυξη τα τελευταία έτη και αποτελεί έναν από τους πιο βασικούς τομείς της Πληροφορικής. Στην καθημερινότητά μας, παρατηρείται η ευρεία χρήση της μέσω εφαρμογών και μηχανών



Εικόνα 1: Σύγκριση αποτελεσμάτων αναζήτησης (δημιουργία με χρήση gemini)

αναζήτησης, οι οποίες διευκολύνουν την επικοινωνία και επιτρέπουν ταχύτερη πρόσβαση σε πληροφορίες.

Η νοημοσύνη ορίζεται ως η ικανότητα ενός οργανισμού ή συστήματος να μαθαίνει, να κατανοεί, να λύνει προβλήματα και να λαμβάνει αποφάσεις (Negnevitsky, 2022). Η ΤΝ, με την σειρά της, ορίζεται ως ο επιστημονικός τομέας που στοχεύει στη σχεδίαση και

υλοποίηση τεχνητών συστημάτων ικανών να εκτελούν εργασίες, όπου αν πραγματοποιούνταν από ανθρώπους, θα απαιτούσαν χρήση νοημοσύνης (Negnevitsky, 2022).

Η ιστορική εξέλιξη της Τεχνητής Νοημοσύνης ξεκινά από τις πρώτες προσπάθειες μοντελοποίησης της ανθρώπινης συλλογιστικής, μέσω συμβολικών και αλγοριθμικών μεθόδων. Στα πρώτα στάδια ανάπτυξης του πεδίου, το ενδιαφέρον μετατοπίστηκε προς τη δημιουργία γενικών αλγορίθμων, οι οποίοι βασίζονταν σε λογικές διεργασίες και τεχνικές εξερεύνησης του χώρου των λύσεων. Οι προσεγγίσεις αυτές παρείχαν θεωρητικό υπόβαθρο, αλλά αποδείχθηκαν περιορισμένες στην εφαρμογή τους σε σύνθετα και δυναμικά περιβάλλοντα, λόγω της υψηλής υπολογιστικής πολυπλοκότητας και της δυσκολίας στη



Εικόνα 2: Χρονοδιάγραμμα Παγκόσμιας Χρήσης AI(δημιουργία με χρήση gemini)

διαχείριση εξειδικευμένης γνώσης (Negnevitsky, 2022).

Η πρόοδος της έρευνας οδήγησε στην δημιουργία των έμπειρων συστημάτων, τα οποία σχεδιάστηκαν για να ψηφιοποιήσουν την εξειδικευμένη γνώση. Με τη χρήση κανόνων, τα συστήματα αυτά μπορούσαν να αναπαράγουν τη διαδικασία λήψης αποφάσεων ενός ειδικού. Η συγκεκριμένη προσέγγιση ανέδειξε τη σημασία της αναπαράστασης και της οργάνωσης της γνώσης, επιτρέποντας την επίλυση προβλημάτων σε συγκεκριμένους τομείς. Παρά τη σημαντική τους συμβολή, τα έμπειρα συστήματα παρουσίασαν δυσκολίες κατά την

απόκτηση γνώσης, στη συντήρηση των βάσεων κανόνων και στη γενίκευση της λειτουργίας τους σε νέα προβλήματα (Negnevitsky, 2022).

Παράλληλα, αναπτύχθηκαν υπολογιστικά μοντέλα εμπνευσμένα από φυσικές διαδικασίες, όπως ο εξελικτικός υπολογισμός και οι γενετικοί αλγόριθμοι. Οι μέθοδοι αυτές βασίζονται σε μηχανισμούς επιλογής, προσαρμογής και επαναληπτικής βελτιστοποίησης, προσομοιώνοντας διαδικασίες της φυσικής εξέλιξης. Μέσω αυτών, έγινε δυνατή η αντιμετώπιση πολύπλοκων προβλημάτων βελτιστοποίησης και μη γραμμικής αναζήτησης, τα οποία ήταν δύσκολο να επιλυθούν με παραδοσιακές μεθόδους (Negnevitsky, 2022).

Κομβικό σημείο στην ιστορική εξέλιξη αποτέλεσε η επανεμφάνιση των τεχνητών νευρωνικών δικτύων και η ανάδειξη της μηχανικής μάθησης. Η προσέγγιση αυτή επιτρέπει στα συστήματα να εξάγουν γνώση απευθείας από τα δεδομένα και να βελτιστοποιούν την απόδοσή τους μέσω της εκπαιδευτικής διαδικασίας, υπερβαίνοντας την ανάγκη για ρητό προγραμματισμό. Η εξέλιξη αυτή έδειξε τη μετάβαση από το συμβολικό μοντέλο σε μοντέλα που βασίζονται στα δεδομένα (Negnevitsky, 2022).

Τέλος, η ανάπτυξη τεχνικών διαχείρισης αβεβαιότητας, όπως η ασαφής λογική, επέτρεψε την αναπαράσταση και επεξεργασία ασαφών ή μη αυστηρά ορισμένων εννοιών. Με αυτόν τον τρόπο, έγινε δυνατή η μοντελοποίηση καταστάσεων όπου οι πληροφορίες είναι ελλιπείς ή ασαφείς, προσεγγίζοντας περισσότερο τον τρόπο που οι άνθρωποι λαμβάνουν αποφάσεις σε πραγματικές συνθήκες (Negnevitsky, 2022).

Συνοψίζοντας, ο κλάδος της ΤΝ έχει διανύσει μια μεγάλη πορεία, μεταβαίνοντας από τη στατική μοντελοποίηση της σκέψης σε ευέλικτα συστήματα που μαθαίνουν εμπειρικά. Η ικανότητά τους να ενσωματώνουν γνώση και να προσαρμόζονται σε νέα δεδομένα, τα καθιστά απαραίτητα εργαλεία που μετασχηματίζουν την έρευνα αλλά και την καθημερινή μας εμπειρία.

1.2 Μηχανική Μάθηση

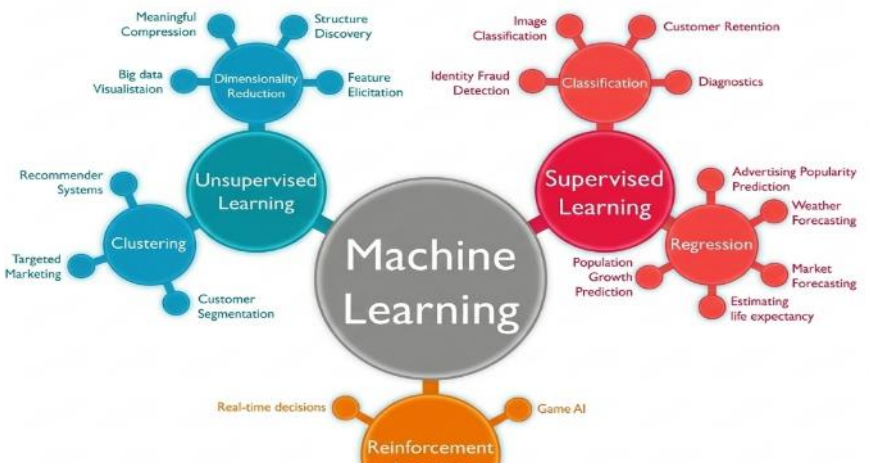
Η Μηχανική Μάθηση αποτελεί την βασική προσέγγιση της σύγχρονης ΤΝ, καθώς επιτρέπει τη δημιουργία συστημάτων που προσαρμόζονται σε νέα δεδομένα και λαμβάνουν αποφάσεις ακόμη και σε συνθήκες αβεβαιότητας (Russell & Norvig, 2021). Η δυνατότητα αυτή βασίζεται στη χρήση στατιστικών μεθόδων, οι οποίες εντοπίζουν πρότυπα στα δεδομένα και επιτρέπουν στο σύστημα να εφαρμόζει τη γνώση του και σε νέες, άγνωστες καταστάσεις (Murphy, 2012).

Η Μηχανική Μάθηση υλοποιείται με διαφορετικούς τρόπους, ανάλογα με την διαδικασία εκπαίδευσης του συστήματος. Οι τρεις βασικότερες κατηγορίες είναι:

Η Επιβλεπόμενη Μάθηση, όπου το σύστημα αντλεί γνώση από προσημειωμένα δεδομένα. Σε αυτή την προσέγγιση, κάθε είσοδος συνοδεύεται από την αντίστοιχη επιθυμητή έξοδο, επιτρέποντας στον αλγόριθμο να μάθει κανόνες αντιστοίχισης (Murphy, 2012).

Η Μη Επιβλεπόμενη Μάθηση, στην οποία το σύστημα καλείται να εντοπίσει αυτόνομα ενδιαφέροντα πρότυπα ή δομές σε δεδομένα που δεν φέρουν ετικέτες. Ο κύριος στόχος δεν είναι η πρόβλεψη μιας συγκεκριμένης τιμής, αλλά η ανακάλυψη της εσωτερικής οργάνωσης των δεδομένων (Jordan & Mitchell, 2015) και η Ενισχυτική Μάθηση, η οποία αφορά συστήματα που μαθαίνουν να λαμβάνουν αποφάσεις μέσω της αλληλεπίδρασης με ένα δυναμικό περιβάλλον. Η μάθηση καθοδηγείται ανταμοιβής (Russell & Norvig, 2021).

Η πρακτική υλοποίηση των παραπάνω κατηγοριών βασίζεται στη χρήση εξειδικευμένων υπολογιστικών βιβλιοθηκών και εργαλείων. Πιο συγκεκριμένα, η χρήση ολοκληρωμένων συνόλων εργαλείων (toolkits), όπως το Dlib-ml, διασφαλίζει την απαραίτητη υπολογιστική



Εικόνα 3: Βασικές κατηγορίες της Μηχανικής Μάθησης

απόδοση και αξιοπιστία, προσφέροντας βελτιστοποιημένες υλοποιήσεις αλγορίθμων που ανταποκρίνονται σε πραγματικές εφαρμογές (King, 2009).

Συμπερασματικά, η απόδοση των παραπάνω μεθοδολογιών εξαρτάται από την ποιότητα της προεπεξεργασίας των δεδομένων και την ικανότητα γενίκευσης του μοντέλου σε νέα στοιχεία. Η συγκεκριμένη διαδικασία ολοκληρώνεται με τον έλεγχο της αξιοπιστίας των αποτελεσμάτων, διασφαλίζοντας την ορθή λειτουργία του συστήματος σε πραγματικές συνθήκες (Murphy, 2012).

1.3 Επεξεργασία Φυσικής Γλώσσας (NLP)

Η Επεξεργασία Φυσικής Γλώσσας (NLP), είναι τομέας της ΤΝ που επιτρέπει στα υπολογιστικά συστήματα να κατανοούν, να επεξεργάζονται και να ερμηνεύουν την ανθρώπινη γλώσσα. Μέσω αυτών των τεχνικών, οι υπολογιστές μπορούν να εξάγουν πληροφορίες και νόημα από κείμενα ή ομιλίες (Adamopoulou & Moussiades, 2020). Η λειτουργία της συνδυάζει την αναγνώριση λέξεων και την κατανόηση του σημασιολογικού



Εικόνα 4: Απεικόνιση της Επεξεργασίας Φυσικής Γλώσσας (NLP)
(δημιουργία με χρήση gemini)

τους περιεχομένου.

Οι πρόσφατες μελέτες δείχνουν ότι η χρήση αρχιτεκτονικών βαθιάς μάθησης (Deep Learning), αντί για συστήματα που βασίζονταν σε στατικούς κανόνες, επέτρεψε στις μηχανές να κατανοούν το ευρύτερο πλαίσιο μιας συνομιλίας (Deng & Liu, 2018). Συμπερασματικά, η εξέλιξη αποδεικνύει ότι τα υπολογιστικά συστήματα έχουν τη δυνατότητα να

αντιλαμβάνονται το νόημα και το πλαίσιο των συνομιλιών, διευκολύνοντας την ανάπτυξη πιο ευφυών εφαρμογών.

Η λειτουργική δομή της Επεξεργασίας Φυσικής Γλώσσας διακρίνεται σε δύο βασικές λειτουργικές κατευθύνσεις:

1) την Κατανόηση Φυσικής Γλώσσας (NLU), η οποία επικεντρώνεται στην υπολογιστική ερμηνεία της εισόδου του χρήστη
2) και την Παραγωγή Φυσικής Γλώσσας (NLG), η οποία στοχεύει στη δημιουργία σωστά δομημένων και νοηματικά ορθών απαντήσεων (Zhao et al., 2023).

Η συγκεκριμένη διάκριση θεωρείται σημαντική καθώς συμβάλλει στην αντιμετώπιση της γλωσσικής ασάφειας, επιτρέποντας αρχικά την κατανόηση των συμφραζομένων (NLU) και στην συνέχεια την ορθή παραγωγή πληροφορίας (Zhao et al., 2023)

1.4 Η Εφαρμογή της ΤΝ στην Τριτοβάθμια Εκπαίδευση

Η εφαρμογή της ΤΝ στην Τριτοβάθμια Εκπαίδευση συμβάλλει στη διαμόρφωση ενός πιο ευέλικτου μαθησιακού περιβάλλοντος, όπου η διδασκαλία προσαρμόζεται στις ανάγκες των φοιτητών. Παράλληλα, τα συστήματα αυτά δημιουργούνται ώστε να προσφέρουν μια αποτελεσματικότερη οργάνωση, καθώς και στη λειτουργία των εκπαιδευτικών ιδρυμάτων.



Εικόνα 5: Διάγραμμα διάσπασης του NLP σε NLU και NLG (δημιουργία με χρήση gemini)

Η αξιοποίηση της ΤΝ βασίζεται στην ανάπτυξη συστημάτων που ενσωματώνουν τη γνώση και τις διδακτικές πρακτικές ειδικών, επιδιώκοντας την εξατομίκευση της μαθησιακής διαδικασίας (Woolf, 1991). Στο πλαίσιο της ψηφιακής εξέλιξης, η υιοθέτηση προσαρμοστικών μορφών μάθησης εντάσσεται στις βασικές κατευθύνσεις ανάπτυξης των πανεπιστημιακών ιδρυμάτων, ενισχύοντας τη μαθησιακή απόδοση και την ακαδημαϊκή



Εικόνα 6: Σχηματική απεικόνιση των βασικών λειτουργιών του ψηφιακού βοηθού (δημιουργία με χρήση gemini notebooklm)

υποστήριξη των φοιτητών κατά τη διάρκεια των σπουδών τους (Alexander et al., 2019).

Η εξέλιξη αυτή ενισχύεται με τη χρήση προηγμένων γλωσσικών μοντέλων, τα οποία επιτρέπουν την άμεση ανάλυση και δημιουργία ακαδημαϊκού περιεχομένου. Η ενσωμάτωση ευφυών εργαλείων παρέχει στους φοιτητές τη δυνατότητα επίλυσης σύνθετων προβλημάτων και εξατομικευμένης καθοδήγησης, ενισχύοντας την αποτελεσματικότητα του μαθησιακού περιβάλλοντος (Koubaa et al., 2023). Επιπλέον, η ενσωμάτωση αυτών των τεχνολογιών συμβάλλει στη βελτίωση της εκπαιδευτικής διαδικασίας και στην διαχείριση διοικητικών καθηκόντων (Alsharhan et al., 2023; Yang et al., 2024).

Ωστόσο, η αξιοποίηση της ΤΝ στην εκπαίδευση συνοδεύεται από σημαντικές προκλήσεις, που αφορούν την δεοντολογία, την ασφάλεια των προσωπικών δεδομένων και τη διαφάνεια των αλγορίθμων (Alexander et al., 2019). Η αντιμετώπιση των ζητημάτων

αυτών αποτελεί βασική προϋπόθεση για την υπεύθυνη και αποτελεσματική χρήση της στον εκπαιδευτικό τομέα.

ΚΕΦΑΛΑΙΟ 2

ΨΗΦΙΑΚΟΙ ΒΟΗΘΟΙ

2.1 Ορισμός Ψηφιακών Βοηθών (Chatbots & Virtual Assistants)

Ψηφιακοί βοηθοί ορίζονται ως υπολογιστικά συστήματα διαλογικής αλληλεπίδρασης, τα οποία αξιοποιούν τη φυσική γλώσσα με σκοπό την υποστήριξη της διαδραστικής επικοινωνίας μεταξύ ανθρώπου και υπολογιστικού περιβάλλοντος. (Laranjo et al., 2018). Οι βοηθοί κατατάσσονται σε διάφορες υποκατηγορίες, όπως τα chatbots και τους virtual assistants, οι οποίες διακρίνονται ως προς τις λειτουργικότητες και το εύρος εφαρμογής τους (Adamopoulou & Moussiades, 2020; Laranjo et al., 2018).

Πιο συγκεκριμένα, chatbots ονομάζονται οι ψηφιακοί βοηθοί που σχεδιάζονται για τη διαλογική επικοινωνία μέσω κειμένου ή φωνής, με σκοπό την παροχή πληροφοριών ή εκτέλεσης υπηρεσιών σε πραγματικό χρόνο (Adamopoulou & Moussiades, 2020). Τα chatbots αποτελούν παραδείγματα συνομιλιακών πρακτόρων που προσομοιώνουν ανθρώπινη συνομιλία, διευκολύνοντας την άμεση διαχείριση αιτημάτων από τους χρήστες.

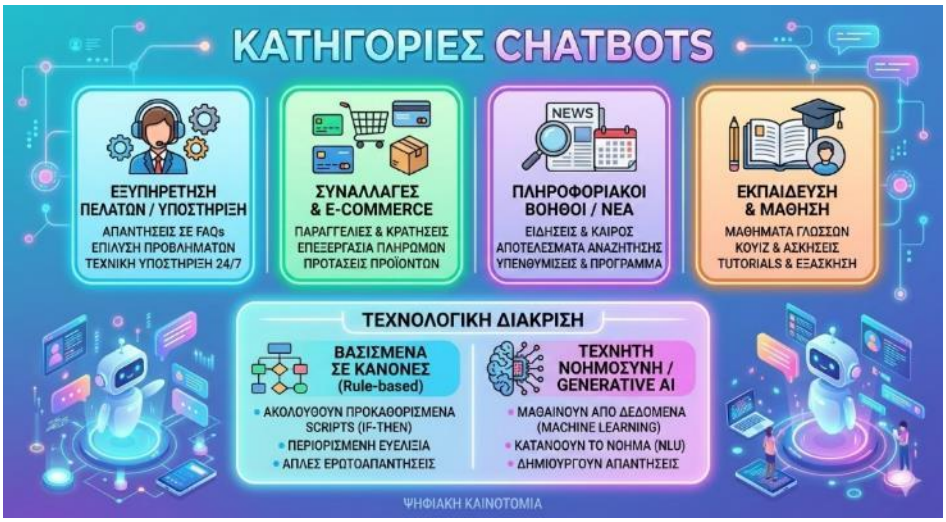
Τα virtual assistants, είναι εξελεγμένοι ψηφιακοί βοηθοί, οι οποίοι εκτός από την αλληλεπίδραση, εκτελούν σύνθετες εργασίες και προσαρμόζουν τις υπηρεσίες τους στις ανάγκες των χρηστών, χρησιμοποιώντας μηχανική μάθηση και ανάλυση δεδομένων (Laranjo et al., 2018).

Συμπερασματικά, οι ψηφιακοί βοηθοί αποτελούν εργαλεία για την υποστήριξη διαλόγου και εξατομικευμένης επικοινωνίας, με κάθε τύπο να προσφέρει διαφορετικές δυνατότητες στους χρήστες.

2.2 Κατηγορίες Chatbots

Τα chatbots κατατάσσονται σε διακριτές κατηγορίες βάσει συγκεκριμένων κριτηρίων, τα οποία σχετίζονται με το γνωστικό τους πεδίο, τον επιδιωκόμενο σκοπό λειτουργίας και τη μεθοδολογία διαμόρφωσης των απαντήσεών τους (Adamopoulou & Moussiades, 2020).

Σύμφωνα με το εύρος γνώσης που καλύπτουν, τα chatbots κατατάσσονται σε δύο βασικές κατηγορίες που αφορούν τα ανοικτού τύπου τα οποία έχουν σχεδιαστεί με τρόπο ώστε να υποστηρίζουν διάλογο γενικού περιεχομένου και δεν περιορίζονται σε συγκεκριμένο θεματικό πλαίσιο. Όπως και τα κλειστού τύπου, τα οποία επικεντρώνονται σε συγκεκριμένο γνωστικό αντικείμενο και αποδίδουν αποτελεσματικότερα όταν τα ερωτήματα εντάσσονται στο προκαθορισμένο πλαίσιο λειτουργίας τους (Adamopoulou & Moussiades, 2020). Η διαφοροποίηση αυτή αντανακλά το βάθος της παρεχόμενης γνώσης και τον συγκεκριμένο



Εικόνα 7: Κατηγορίες Chatbot (δημιουργία με χρήση gemini)

τομέα εφαρμογής του ψηφιακού βοηθού.

Ειδικότερα, στη βάση του επιδιωκόμενου σκοπού τους, διακρίνονται οι ακόλουθες μορφές chatbots:

- Ενημερωτικά chatbots**,** τα οποία παρέχουν προκαθορισμένες πληροφορίες όπως συστήματα FAQ.
- Συνομιλιακά chatbots, που στοχεύουν στη διατήρηση φυσικής ροής διαλόγου.
- Και Task-oriented chatbots, τα οποία έχουν σχεδιαστεί έτσι ώστε να εκτελούν συγκεκριμένες λειτουργίες ή διαδικασίες (Adamopoulou & Moussiades, 2020).

Η διάκριση αυτή αντικατοπτρίζει το επίπεδο λειτουργικής πολυπλοκότητας και τη φύση της παρεχόμενης υποστήριξης προς τον χρήστη.

Τέλος, μια θεμελιώδης διάκριση αφορά τον μηχανισμό επεξεργασίας δεδομένων και τη μεθοδολογία παραγωγής απόκρισης. Ανάλογα με την αρχιτεκτονική τους δομή, τα chatbots ταξινομούνται στις εξής τρεις κατηγορίες:

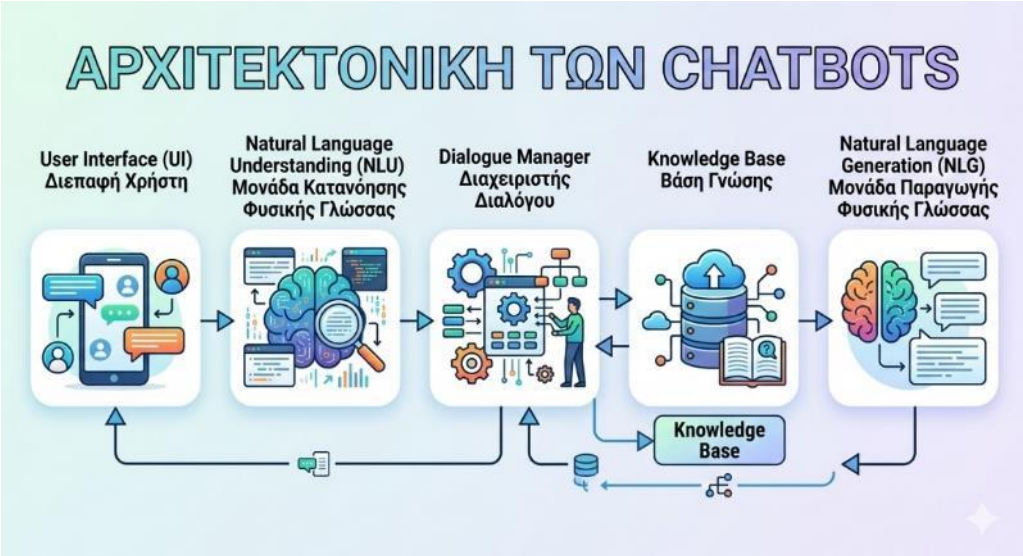
- Rule-based, τα οποία λειτουργούν μέσω προκαθορισμένων κανόνων και προτύπων αντιστοίχισης.
- Retrieval-based, τα οποία επιλέγουν την βέλτιστη απάντηση από προϋπάρχουσα βάση δεδομένων.
- Και Generative, τα οποία αξιοποιούν τεχνικές μηχανικής μάθησης για τη σύνθεση πρωτότυπων απαντήσεων σε πραγματικό χρόνο (Adamopoulou & Moussiades, 2020).

Η κατηγοριοποίηση αυτή αντανακλά τον βαθμό τεχνολογικής πολυπλοκότητας και αυτονομίας του συστήματος.

Συμπερασματικά, η ποικιλομορφία των chatbots καταδεικνύει ότι η επιλογή του κατάλληλου συστήματος εξαρτάται από τον επιδιωκόμενο συνδυασμό ευκολίας στη χρήση και λειτουργικής πολυπλοκότητας. Ανεξάρτητα από την κατηγορία στην οποία ανήκουν, η αποτελεσματικότητά τους κρίνεται τελικά από την ικανότητά τους να εξυπηρετούν με ακρίβεια τον σκοπό για τον οποίο σχεδιάστηκαν, γεφυρώνοντας το κενό στην επικοινωνία ανθρώπου και μηχανής.

2.3 Αρχιτεκτονική των Chatbots

Η αρχιτεκτονική ενός chatbot αναφέρεται στη δομή και την οργάνωση των επιμέρους υποσυστημάτων που επιτρέπουν την κατανόηση των εισροών του χρήστη, καθώς και την παραγωγή κατάλληλων αποκρίσεων. Στη βιβλιογραφία, τα συστήματα αυτά περιγράφονται



Εικόνα 8: Αρχιτεκτονική των Chatbots (δημιουργία με χρήση gemini)

ως πολυεπίπεδες δομές που συνδυάζουν μηχανισμούς επεξεργασίας φυσικής γλώσσας, διαχείρισης διαλόγου και παραγωγής απαντήσεων (Adamopoulou & Moussiades, 2020; Jurafsky & Martin, 2023). Παρά τις τεχνολογικές διαφοροποιήσεις μεταξύ των μοντέλων, η λειτουργική τους βάση παραμένει κοινή και στηρίζεται στα δομικά στοιχεία που αναλύονται στη συνέχεια.

Η αποτελεσματική λειτουργία ενός chatbot, εξαρτάται από τον συντονισμό διακριτών τεχνολογικών μονάδων, που αλληλεπιδρούν μεταξύ τους. Στο πλαίσιο της βιβλιογραφικής προσέγγισης, η αρχιτεκτονική του συστήματος συγκροτείται από πέντε κύριους πυλώνες. Πιο συγκεκριμένα, αφορούν την Διεπαφή Χρήστη (User Interface - UI), την Μονάδα Κατανόησης Φυσικής Γλώσσας (Natural Language Understanding – NLU), τον Διαχειριστή Διαλόγου (Dialogue Manager), την Βάση Γνώσης (Knowledge Base) και την Μονάδα Παραγωγής Φυσικής Γλώσσας (Natural Language Generation - NLG).

Η Διεπαφή Χρήστη (User Interface - UI), λειτουργεί ως το τμήμα διεπαφής (front-end) του συστήματος, μέσω του οποίου συλλέγονται τα δεδομένα εισόδου και εμφανίζεται η τελική απάντηση προς τον χρήστη (Adamopoulou & Moussiades, 2020). Η σχεδίαση της διεπαφής είναι καθοριστική, καθώς επηρεάζει άμεσα την εμπειρία του χρήστη (User Experience) και την αλληλεπίδραση με το σύστημα.

Η Μονάδα Κατανόησης Φυσικής Γλώσσας (Natural Language Understanding - NLU), αναλαμβάνει την μετατροπή του φυσικού κειμένου σε δομημένη πληροφορία. Χρησιμοποιώντας γλωσσικά πρότυπα και αλγορίθμους, καθορίζει την πρόθεση του χρήστη και εντοπίζει σημαντικά στοιχεία που απαιτούνται για την επεξεργασία (Jurafsky & Martin, 2023).

Από την άλλη, ο Διαχειριστής Διαλόγου (Dialogue Manager), λειτουργεί ως το πρόγραμμα που συντονίζει την συζήτηση μεταξύ χρήστη και chatbot. Σκοπός του είναι να ελέγχει την κατάσταση του διαλόγου (Dialogue State Tracking), καθώς και την λήψη αποφάσεων με βάση τους στόχους που έχουν τεθεί, διασφαλίζοντας την συνοχή της συζήτησης και τη διατήρηση του πλαισίου (contextual awareness) (Adamopoulou & Moussiades, 2020).

Η Βάση Γνώσης (Knowledge Base), είναι το σύστημα που συλλέγει ότι πληροφορία χρειάζεται έτσι ώστε να μπορεί να απαντήσει στον χρήστη. Σε αυτήν αποθηκεύονται τόσο στατικά δεδομένα και προκαθορισμένες απαντήσεις όσο και δυναμικές πληροφορίες που τροφοδοτούνται κατά τη διάρκεια της χρήσης. Η ύπαρξη μιας οργανωμένης βάσης γνώσης επιτρέπει στο σύστημα να προσδιορίζει τις κατάλληλες απαντήσεις, να διασφαλίζει συνοχή στις πληροφορίες και να υποστηρίζει την ακριβή επεξεργασία των αιτημάτων των χρηστών (Adamopoulou & Moussiades, 2020; Jurafsky & Martin, 2023).

Τέλος, η Μονάδα Παραγωγής Φυσικής Γλώσσας (Natural Language Generation - NLG), είναι το τελικό στάδιο όπου αφορά την οριστική διατύπωση της απάντησης που θα παρουσιαστεί στον χρήστη. Πιο συγκεκριμένα, η μονάδα αυτή αναλαμβάνει τη μετατροπή της εσωτερικής υπολογιστικής απόφασης σε φυσική γλώσσα, εφαρμόζοντας κανόνες σύνταξης και γραμματικής για την παραγωγή μιας κατανοητής απόκρισης (Jurafsky & Martin, 2023).

Συμπερασματικά, η αρχιτεκτονική των chatbots επιτρέπει τη μετατροπή της ανθρώπινης γλώσσας σε υπολογιστική δράση δίνοντας απαντήσεις στους χρήστες. Η ακρίβεια της βάσης γνώσης σε συνδυασμό με την ευφυΐα του διαχειριστή διαλόγου καθορίζουν την τελική αποτελεσματικότητα του συστήματος.

2.4 Πλεονεκτήματα και Περιορισμοί

Η αξιοποίηση των chatbots έχει ενισχυθεί σημαντικά τα τελευταία χρόνια, λόγω της αυξανόμενης ανάγκης για αυτοματοποίηση και άμεση παροχή πληροφοριών. Παρόλα αυτά η εφαρμογή τους συνοδεύεται από σημαντικά οφέλη αλλά και περιορισμούς. Η κατανόηση τόσο των πλεονεκτημάτων όσο και των προκλήσεων που παρουσιάζουν αποτελεί απαραίτητη προϋπόθεση για την ορθολογική και αποτελεσματική ενσωμάτωσή τους σε πραγματικά περιβάλλοντα χρήσης.

Τα πλεονεκτήματα που προσφέρουν τα chatbots είναι αρκετά όπως είναι η λειτουργική κλιμάκωση όπου τα chatbots έχουν την ικανότητα να διαχειρίζονται ταυτόχρονα έναν πρακτικά απεριόριστο αριθμό ερωτημάτων, προσφέροντας άμεσες αποκρίσεις χωρίς την ανάγκη αναμονής, κάτι που θα ήταν αδύνατο για έναν ανθρώπινο παράγοντα (Adamopoulou & Moussiades, 2020). Η λειτουργική αξιοπιστία που αφορά άλλο ένα πλεονέκτημα των chatbots, βασίζεται στην ικανότητά τους να παρέχουν τυποποιημένες αποκρίσεις, διασφαλίζοντας την ακεραιότητα της πληροφορίας ανεξάρτητα από τον όγκο των αιτημάτων (Hussain et al., 2019). Τέλος, η οικονομική αποδοτικότητα. όπου με την αυτοματοποιημένη επικοινωνιακή διαδικασία μειώνεται σημαντικά το λειτουργικό κόστος των οργανισμών, καθώς το σύστημα μπορεί να επιλύει το μεγαλύτερο ποσοστό των τυπικών ερωτημάτων (Hussain et al., 2019).

Αντίθετα, υπάρχουν και αρκετοί περιορισμοί χρησιμοποιώντας τα chatbots. Ένας από τους σημαντικότερους περιορισμούς θεωρούνται οι προκλήσεις στην κατανόηση συμφραζομενων διότι παρά τις εξελίξεις της Επεξεργασίας Φυσικής Γλώσσας, τα συστήματα συχνά δυσκολεύονται να αντιληφθούν το σαρκασμό, τα συναισθήματα ή πολύπλοκες γλωσσικές ιδιομορφίες, με αποτέλεσμα να κάνουν λάθοι διότι δεν καταλαβαίνουν αυτό που τους λέει ο χρήστης (Jurafsky & Martin, 2023). Τα chatbots πολλές φορές έχουν

περιορισμένη βάση γνώσης, με αποτέλεσμα η ποιότητα της απάντησης εξαρτάται αποκλειστικά από την εγκυρότητα και το εύρος των δεδομένων στα οποία έχει πρόσβαση το σύστημα. Ελλιπή ή λανθασμένα δεδομένα οδηγούν αναπόφευκτα σε ανακριβείς αποκρίσεις (Hussain et al., 2019). Τέλος, ένας ακόμη θεμελιώδης περιορισμός των chatbots είναι η παραγωγή ανακριβών ή κατασκευασμένων πληροφοριών που παρουσιάζονται ως πραγματικά γεγονότα. Αυτό συμβαίνει επειδή τα μοντέλα βασίζονται σε στατιστικές πιθανότητες για την πρόβλεψη της επόμενης λέξης και όχι σε πραγματική κατανόηση της αλήθειας (Ferrara, 2023).



Εικόνα 9 : Πλεονεκτήματα και Περιορισμοί (δημιουργία με χρήση gemini)



Συνοψίζοντας, τα πλεονεκτήματα των chatbots κρίνονται απαραίτητα για την σύγχρονη επικοινωνία, παρόλα αυτά οι περιορισμοί τους υπενθυμίζουν ότι η τεχνολογία αυτή έχει ακόμη περιθώρια βελτίωσης. Η προστασία της ιδιωτικότητας και η αποφυγή λανθασμένων αποκρίσεων παραμένουν οι κρισιμότερες προκλήσεις για την πλήρη αξιοποίησή τους.

ΚΕΦΑΛΑΙΟ 3

ΔΗΜΙΟΥΡΓΙΑ ΨΗΦΙΑΚΟΥ ΒΟΗΘΟΥ ΓΙΑ ΤΟ ΤΜΗΜΑ ΔΕΤ

3.1 Σκοπός και Ανάλυση Απαιτήσεων

Η ανάπτυξη του ψηφιακού βοηθού για το τμήμα Διοικητικής Επιστήμης και Τεχνολογίας (ΔΕΤ), βασίζεται στον προσδιορισμό ενός πλαισίου που στοχεύει στην ψηφιακή επικοινωνία του τμήματος με τους φοιτητές. Η ανάλυση απαιτήσεων αποτελεί κρίσιμο παράγοντα για ένα πληροφοριακό συστήματα, καθώς καθορίζει τις λειτουργικές και μη λειτουργικές προδιαγραφές, που θα καθοδηγήσουν τον σχεδιασμό και την υλοποίησή του (Pressman & Maxim, 2020).

Σκοπός της ανάπτυξης του ψηφιακού βοηθού, είναι η δημιουργία ενός αυτοματοποιημένου συστήματος chatbot, έτσι ώστε να παρέχετε άμεση και αξιόπιστη πληροφόρηση τις απορίες των φοιτητών. Με τον τρόπο αυτό, μειώνεται δραστικά ο φόρτος των καθημερινών αιτημάτων προς την Γραμματεία του Τμήματος, ενώ η πρόσβαση στις ακαδημαϊκές πληροφορίες γίνεται ευκολότερη στους φοιτητές χωρίς χρονικούς περιορισμούς. Η αξιοποίηση συνομιλιακών συστημάτων σε εκπαιδευτικά ιδρύματα έχει αποδειχθεί ότι συμβάλλει στην αποτελεσματικότερη διάχυση πληροφορίας και στην



Εικόνα 10: Ανάλυση σκοπού και απαιτήσεων (δημιουργία με χρήση gemini)

ενίσχυση της υποστήριξης των χρηστών (Adamopoulou & Moussiades, 2020).

Οι επιμέρους στόχοι του ψηφιακού βοηθού είναι:

- Η αυτοματοποιημένη παροχή έγκυρων ακαδημαϊκών και διοικητικών πληροφοριών προς τους φοιτητές.
- Η ανάπτυξη μηχανισμού αναγνώρισης πρόθεσης χρήστη για την ακριβή αντιστοίχιση ερωτήματος–απάντησης.
- Η δημιουργία δομημένων συζητήσεων που λειτουργούν ως οδηγός, προσφέροντας στον φοιτητή τις σωστές οδηγίες για κάθε γραφειοκρατικό ζήτημα που αναζητά.
- Η διασφάλιση ικανοποιητικού επιπέδου ακρίβειας των αποκρίσεων κατά τη φάση δοκιμών.
- Καθώς και η μείωση του φόρτου επαναλαμβανόμενων ερωτημάτων προς την γραμματεία.

Ο σαφής καθορισμός στόχων θεωρείται απαραίτητος για τη συστηματική αξιολόγηση πληροφοριακών συστημάτων (Pressman & Maxim, 2020).

Σύμφωνα με τη βιβλιογραφία, τα εκπαιδευτικά chatbots χρησιμοποιούνται κυρίως για την παροχή πληροφοριών σχετικών με διοικητικές διαδικασίες, ακαδημαϊκά προγράμματα και προθεσμίες (Adamopoulou & Moussiades, 2020; Laranjo et al., 2018). Οι βασικές κατηγορίες πληροφορίας που θα ενσωματωθούν στο αυτοματοποιημένο συστήματος περιλαμβάνουν:

- Πληροφορίες σχετικά με τον οδηγό σπουδών (π.χ. μαθήματα, δηλώσεις κλπ.)
- Διαδικασίες δηλώσεων και εξεταστικών περιόδων
- Κανονισμούς πτυχιακής εργασίας
- Κανονισμούς πρακτικής άσκησης
- Στοιχεία επικοινωνίας και ωράρια εξυπηρέτησης
- Πληροφορίες για δηλώσεις στον Morfea, το οποίο είναι σύστημα διαχείρισης αιτήσεων του Ελληνικού Μεσογειακού Πανεπιστημίου
- Και Μέλη του Τμήματος που αφορά τους καθηγητές.

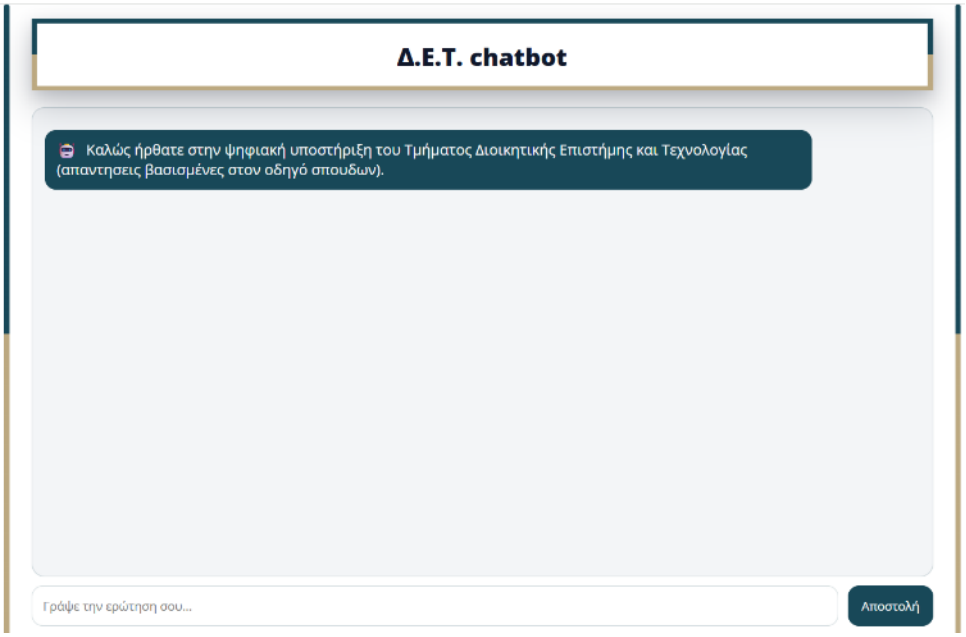
Με την ταξινόμηση αυτή, η βάση γνώσης αποκτά συγκεκριμένη δομή, γεγονός που επιτρέπει στο σύστημα να εξυπηρετεί τον χρήστη χωρίς καθυστερήσεις ή ασάφειες.

Με βάση τα παραπάνω, το σύστημα καλείται να να ικανοποιεί λειτουργικές απαιτήσεις που αφορούν διαχείριση ερωτημάτων και πιο συγκεκριμένα η αναγνώριση πρόθεσης χρήστη, απαντήσεις για τις παραπάνω πληροφορίες και η διαχείριση βασικών σεναρίων διαλόγου. Επίσης, κρίσιμες θεωρούνται και οι μη λειτουργικές απαιτήσεις, όπως η αξιοπιστία, η ταχύτητα απόκρισης, η ευχρηστία και η προστασία προσωπικών δεδομένων, ιδίως σε εκπαιδευτικά περιβάλλοντα (Laranjo et al., 2018).

3.2 Σχεδίαση Συστήματος

3.2.1 Αρχιτεκτονική και ροή επεξεργασίας αιτήματος (end-to-end)

Η αρχιτεκτονική του συστήματος περιλαμβάνει τρία βασικά επίπεδα, τα οποία αφορούν τη διεπαφή του περιηγητή (browser interface), τον εξυπηρετητή API (API Server), αλλά και την υποκείμενη λογική του συστήματος. Στο πρώτο επίπεδο, η εφαρμογή FastAPI παρέχει το αρχικό HTML και τα στατικά αρχεία μέσω του καταλόγου /static, ενώ ενσωματώνει τον χειριστή των αιτημάτων (router) (Bednarz & Miłosz (2025). Η εφαρμογή αναλαμβάνει τη διαμεσολάβηση μεταξύ χρήστη και συστήματος, διαχειρίζοντας τα



Εικόνα 11: UI του chatbot

εισερχόμενα αιτήματα και τις αντίστοιχες αποκρίσεις του εξυπηρετητή.

Στο δεύτερο επίπεδο, η λειτουργία εστιάζει στην επεξεργασία των δεδομένων και την επικοινωνία μεταξύ των υποσυστημάτων. Αρχικά, στο επίπεδο του φυλλομετρητή (client-side), το αρχείο ChatBotIndex.html φορτώνει τους απαραίτητους πόρους CSS και JavaScript, ενώ ενσωματώνει τη βιβλιοθήκη KaTeX για την ορθή μαθηματική απόδοση των αποκρίσεων (Jurafsky & Martin, 2023). Μόλις ο χρήστης εισάγει το ερώτημά του, η εφαρμογή ενεργοποιεί έναν μηχανισμό ασύγχρονης κλήσης API για τη διαβίβαση των δεδομένων στο

```html
<!doctype html>
<html lang="el">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1"/>
  <title>Α.Ε.Τ. chatbot</title>
  <link rel="icon" href="/favicon.ico" />
  <link rel="stylesheet" href="/static/theme.css" />
  <!-- KaTeX (math rendering) -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/katex@0.16.9/dist/katex.min.css">
  <script defer src="https://cdn.jsdelivr.net/npm/katex@0.16.9/dist/katex.min.js"></script>
  <script defer src="https://cdn.jsdelivr.net/npm/katex@0.16.9/dist/contrib/auto-render.min.js"></script>
</head>
<body>
  <div class="wrap">
    <header class="top">
      <h1>Α.Ε.Τ. chatbot</h1>
    </header>

    <main id="chat" class="chat"></main>

    <footer class="bar">
      <input id="msg" type="text" placeholder="Γράψε την ερώτηση σου..." autocomplete="off" />
      <button id="send">Αποστολή</button>
    </footer>
  </div>

  <script src="/static/chatbot.js?v=20260226_202704"></script>
</body>
</html>
```

```css
.msg.bot{
  margin-right: auto ;
  margin-left: 0 ;
  max-width: 85% ;
}
/* Links */
.msg a{
  text-decoration: underline ;
  text-underline-offset: 2px ;
  cursor: pointer ;
  pointer-events: auto ;
}
.msg.bot a{ color: #ffffff ; }
.msg.me  a{ color: #1f2937 ; }

/* Choices = ίδιά χρώμα με bot */
.choices{
  display:flex;
  flex-wrap:wrap;
  gap:10px;
  margin-top:10px;
}
.choice{
  background: var(--bot) ;
  border: 1px solid var(--bot) ;
  color: #ffffff ;
  padding: 10px 12px ;
  border-radius: 12px ;
  cursor:pointer ;
  font-size: 13px ;
  font-weight: 700 ;
```

Εικόνα 12: Τμήμα κώδικα HTML & CSS

υπολογιστικό υπόβαθρο (backend).

Η επεξεργασία του αιτήματος στον εξυπηρετητή FastAPI ακολουθεί μια διαρθρωμένη διαδικασία.Τα δεδομένα του χρήστη, όπως το περιεχόμενο του αιτήματος και το session_id, αποστέλλονται σε μορφή JavaScript Object Notation (JSON) (Benchmarking Web Frameworks, 2025). Το σύστημα, αφού επικυρώσει την ταυτότητα του πελάτη μέσω του client_key, προχωρά στην κανονικοποίηση (normalization) του κειμένου. Η διαδικασία αυτή περιλαμβάνει την αφαίρεση τόνων, τη μετατροπή σε πεζούς χαρακτήρες και την εξάλειψη περιττών διαστημάτων, διασφαλίζοντας την ομοιομορφία των δεδομένων εισόδου.

Στη συνέχεια, ενεργοποιείται ο μηχανισμός detect_intent, ο οποίος αποτελεί τον πυρήνα της λογικής του συστήματος. Ο μηχανισμός αυτός εφαρμόζει ιεραρχημένους κανόνες και ανάλυση οντοτήτων για την αναγνώριση της πρόθεσης του χρήστη. Με βάση την αναγνωρισμένη πρόθεση, το σύστημα προβαίνει στην ανάκτηση της πληροφορίας, είτε μέσω προκαθορισμένων στατικών δεδομένων είτε μέσω δυναμικών ερωτημάτων στη βάση

```python
def detect_intent(question: str) -> dict:
    parsed = parse_question(question)
    parsed["question"] = question

    normalized = normalize(question)
    parsed["normalized"] = normalized

    course_match = None  # cache for repo.find_course_in_text

    def get_course_match():
        nonlocal course_match
        if course_match is not None:
            return course_match

        # 1) raw
        course_match = repo.find_course_in_text(question)

        # 2) fallback: normalized
        if not (course_match and course_match.get("title")):
            course_match = repo.find_course_in_text(normalized)

        return course_match

    if not parsed.get("semester"):
        m = re.search(r"\b([1-9]|10)(?:ο|ου)?\s*εξαμην", normalized)
        if m:
            parsed["semester"] = int(m.group(1))

    if not parsed.get("entity") and ("μαθημ" in normalized or "course" in normalized):
        parsed["entity"] = "course"

    if not parsed.get("course_hint"):
        c = get_course_match()
        if c and c.get("title"):
            parsed["entity"] = "course"
            parsed["course_hint"] = c["title"]
```

Εικόνα 13: Τμήμα κώδικα detect intent

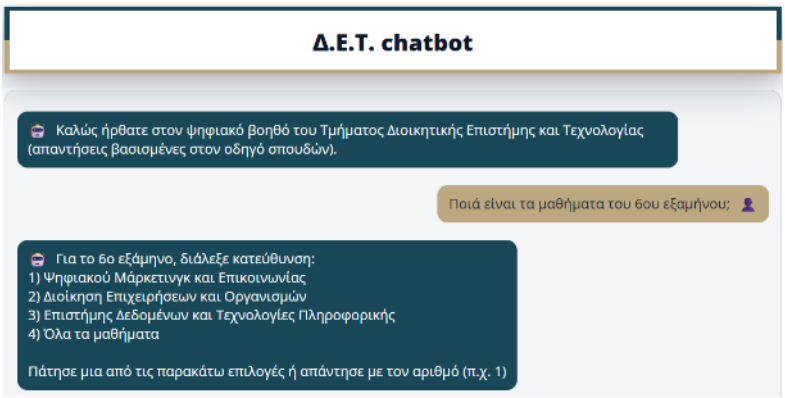
δεδομένων SQLite (Jurafsky & Martin, 2023).

Στο τρίτο επίπεδο, το οποίο αποτελεί το επίπεδο των δεδομένων (Data Tier), η εφαρμογή αλληλεπιδρά με τη βάση δεδομένων SQLite. Το επίπεδο αυτό είναι υπεύθυνο για την εμμονή των δεδομένων (data persistence), διασφαλίζοντας την αποθήκευση και την ανάκτηση του ιστορικού των συνομιλιών, καθώς και των απαραίτητων πληροφοριών για την παραγωγή των αποκρίσεων. Η επικοινωνία με τη βάση πραγματοποιείται μέσω δομημένων ερωτημάτων SQL, επιτρέποντας στο σύστημα να διαχειρίζεται αποτελεσματικά το περιεχόμενο και να διατηρεί τη συνέχεια της επικοινωνίας (session state) με τον χρήστη. Η συνολική ροή επεξεργασίας ενός αιτήματος από το επίπεδο διεπαφής έως την επιστροφή της τελικής απόκρισης, αποτυπώνεται στο Παράρτημα Α.

3.2.2 Διαχείριση διαλόγου με διατήρηση κατάστασης

Η διαχείριση της κατάστασης του διαλόγου βασίζεται σε έναν αποδοτικό μηχανισμό παρακολούθησης της αλληλεπίδρασης του χρήστη. Στο επίκεντρο της υλοποίησης βρίσκεται η δομή δεδομένων _STATE, ένα λεξικό (dictionary) το οποίο συσχετίζει το μοναδικό κλειδί πελάτη (client_key) με την τρέχουσα κατάσταση της συνομιλίας. Η δομή αυτή αποθηκεύει πληροφορίες για τα ενεργά αιτήματα και τους εκκρεμείς παραμέτρους (slots) που απαιτούν συμπλήρωση για την ολοκλήρωση μιας ενέργειας. Το σύστημα διατηρεί τη συνεχιζόμενη κατάσταση για 20 λεπτά και την επαναφέρει σε αρχική μορφή εάν δεν υπάρξει αλληλεπίδραση εντός αυτού του χρονικού διαστήματος (Bohus & Rudnicky, 2009). Η συγκεκριμένη στρατηγική διασφαλίζει τη συνέχεια του διαλόγου και την αποφυγή απώλειας πληροφορίας κατά τη διάρκεια της επικοινωνίας.

Η διαδικασία διαχείρισης διαλόγου βασίζεται σε δείκτες εκκρεμότητας (pending flags), οι οποίοι σηματοδοτούν την ανάγκη για συμπλήρωση συγκεκριμένων πεδίων

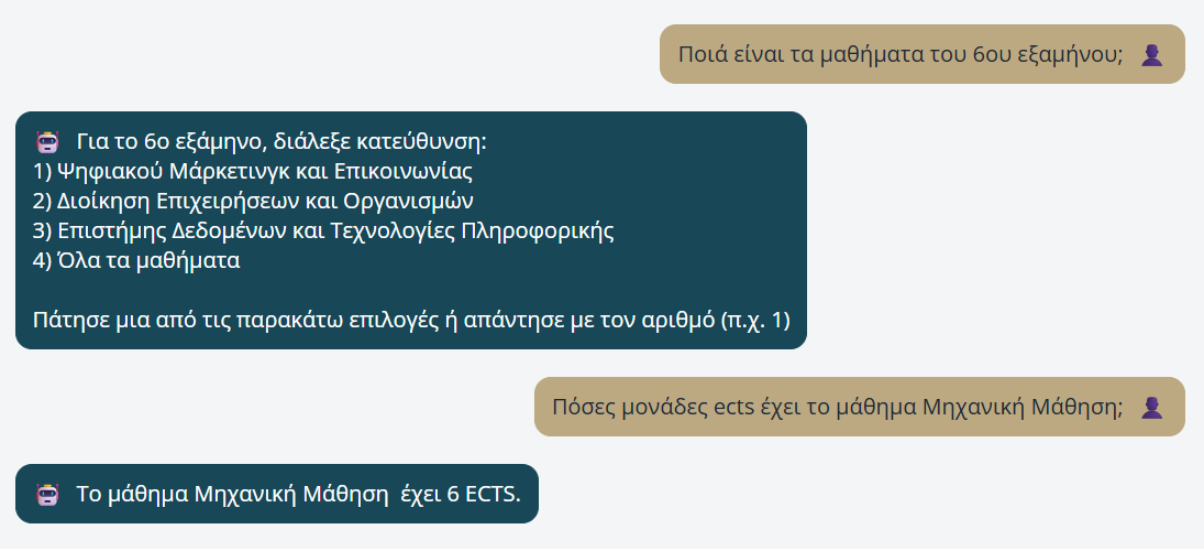


Εικόνα 14: Κατάσταση pending

δεδομένων. Κάθε δείκτης αντιστοιχεί σε μια μεταβλητή τύπου slot που πρέπει να πληρωθεί προκειμένου να ολοκληρωθεί το αίτημα. Για την κάλυψη αυτών των κενών, το σύστημα καθοδηγεί τον χρήστη να εισάγει την απαιτούμενη πληροφορία για το εκάστοτε αντικείμενο. Μόλις ολοκληρωθεί η πλήρωση των παραμέτρων, το σύστημα παρέχει τη ζητούμενη απόκριση, ακολουθώντας μια δομημένη ροή βημάτων βάσει προκαθορισμένων κανόνων (Young et al., 2013).

Επιπλέον, το σύστημα ενσωματώνει έναν μηχανισμό διαχείρισης πολλαπλών επιλογών (choices) για την ενίσχυση της ευελιξίας στην αλληλεπίδραση. Σε αυτό το πλαίσιο, ο χρήστης καλείται να επιλέξει ανάμεσα σε παραμέτρους που παρουσιάζονται ως διαδραστικά στοιχεία (κουμπιά) στη διεπαφή, διευκολύνοντας τη λήψη αποφάσεων και τη δρομολόγηση του διαλόγου σε συγκεκριμένα υποσυστήματα.

Στην περίπτωση που υφίσταται εκκρεμές αίτημα, η αρχιτεκτονική του συστήματος επιτρέπει την άμεση έναρξη νέας αλληλεπίδρασης χωρίς να παρεμποδίζεται η ροή των εισερχόμενων δεδομένων. Ο σχεδιασμός αυτός υποστηρίζει τη δυναμική αλλαγή θεματολογίας (context switching) και την απρόσκοπτη μετάβαση μεταξύ διαφορετικών ροών διαλόγου. Με αυτόν τον τρόπο, το σύστημα προσαρμόζεται ευέλικτα στις μεταβαλλόμενες ανάγκες του χρήστη, εξασφαλίζοντας την αποτελεσματική εξυπηρέτηση των αιτημάτων χωρίς χρονικές καθυστερήσεις, ακόμη και όταν υπάρχουν προγενέστερες εκκρεμότητες στην



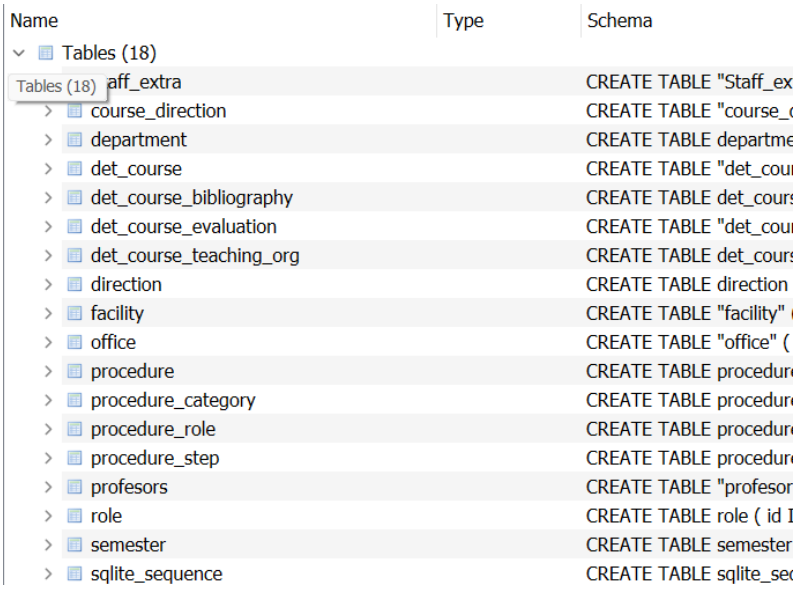
Εικόνα 15: Αλλαγή θεματολογίας

πλήρωση των παραμέτρων (McTear, 2004).

3.2.3 Στρώμα γνώσης και πρόσβαση δεδομένων

Το στρώμα γνώσης του συστήματος βασίζεται στη βάση δεδομένων SQLite, η οποία εξασφαλίζει υψηλή αποδοτικότητα στην αποθήκευση και την ανάκτηση των δεδομένων. Η βάση δεδομένων εντοπίζεται στο αρχείο det.db, ενώ η διασύνδεση πραγματοποιείται δυναμικά μέσω της βιβλιοθήκης sqlite3 module. Κάθε λειτουργία που απαιτεί πρόσβαση στο επίπεδο των δεδομένων ενεργοποιεί μια νέα σύνδεση, η οποία αξιοποιεί τον μηχανισμό Row factory. Ο συγκεκριμένος μηχανισμός επιτρέπει την ευχρηστία και την αποδοτικότητα στην αναζήτηση και χρήση των δεδομένων (Elmasri & Navathe, 2016).

Η αρχιτεκτονική της βάσης δεδομένων επιτρέπει την ανάκτηση πληροφοριών μέσω προκαθορισμένων ερωτημάτων, τα οποία αξιοποιούν πράξεις συνένωσης (JOIN operations) για τη συσχέτιση δεδομένων από πολλαπλούς πίνακες. Η αναζήτηση των ακαδημαϊκών μαθημάτων πραγματοποιείται με τη χρήση φίλτρων, όπως η κατεύθυνση σπουδών και τα υποχρεωτικά μαθήματα (mandatory flag), ώστε να περιορίζονται τα αποτελέσματα και να



Εικόνα 16: Πίνακες της Βάσης Δεδομένων

αποφεύγεται η υπερφόρτωση με αχρείαστα δεδομένα.

Τα δεδομένα που σχετίζονται με τα μαθήματα, ανακτώνται με εμπλουτισμένες εξαγωγές που διαμορφώνουν το τελικό κείμενο της απάντησης προς τον χρήστη. Αυτό επιτυγχάνεται μέσω της λειτουργίας find_course_in_text, η οποία αναζητά τον πλήρη κανονικοποιημένο τίτλο του μαθήματος στο αίτημα του χρήστη και επιστρέφει τα αποτελέσματα μόνο εφόσον πληρούνται συγκεκριμένα κριτήρια (Fowler, 2002). Η συγκεκριμένη μεθοδολογία εγγυάται τη συνοχή μεταξύ της αποθηκευμένης γνώσης και της τελικής πληροφορίας που παρουσιάζεται στον τελικό χρήστη.

Table: det_course

Filter in any column

	id	code	title	semester_id	ects	learning_outcomes	skills	content
	Fil...	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	0801.1.002.0	Εισαγωγή στην Οικονομική Θεωρία	1	6	Σκοπός του μαθήματος …	• Αυτόνομη Εργασία …	Σε επίπεδο μικροοικονομικής ανάλυσης …
2	2	0801.1.003.0	Οργάνωση και Διοίκηση Επιχειρήσεων	1	6	Η Διοίκηση ως αντικείμεν…	• Ομαδική Εργασία …	Τα περιεχόμενα του μαθήματος είναι: …
3	3	0801.1.004.0	Μαθηματική Ανάλυση	1	6	Μετά την επιτυχή …	• Αυτόνομη Εργασία …	• Ακολουθίες: Μονοτονία και Φράγμα/…
4	4	0801.1.005.0	Επιχειρησιακή Επικοινωνία	1	6	Σκοπός του μαθήματος …	• Εργαστηριακές Ασκήσεις …	Προσδιορισμός επικοινωνίας/οργανωσιακής
5	5	0801.2.001.0	Εισαγωγή στην πληροφορική	1	6	Το μάθημα αποτελεί βασικ…	• Αυτόνομη Εργασία …	• Ιστορική εξέλιξη ηλεκτρονικών …
6	6	0801.2.002.0	Πιθανότητες και Στατιστική	2	6	Μετά την επιτυχή …	• Αυτόνομη Εργασία …	Το περιεχόμενο του μαθήματος …
7	7	0801.2.003.0	Μαρκετινγκ	2	6	Σκοπός του μαθήματος …	• Αναζήτηση, ανάλυση και …	• Η κατανόηση του ρόλου του μάρκετινγκ …
8	8	0801.2.004.0	Αρχές Λογιστικής	2	6	Το μάθημα αποτελεί …	• Ατομικές Εργασίες κατά τη …	Τα περιεχόμενα του μαθήματος είναι:…
9	9	0801.2.005.0	Δομημένος Προγραμματισμός	2	6	Σκοπός του μαθήματος …	• Αυτόνομη Εργασία …	• Εισαγωγή στον προγραμματισμό και η …
10	10	0801.3.001.0	Γραμμική Άλγεβρα και Διακριτά …	2	6	Γνώσεις…	Αυτόνομη Εργασία Ομαδική …	1. Θεωρία συνόλων και αλγεβρικές δομές:
11	11	0801.3.002.0	Επιχειρησιακή Έρευνα	3	6	Το μάθημα αποσκοπεί να …	• Αυτόνομη Εργασία …	• Σύνδεση της Επιχειρησιακής Έρευνας με
12	12	0801.3.003.0	Εφαρμοσμένη Στατιστική	3	6	Μετά την επιτυχή …	• Αυτόνομη Εργασία …	• Ανασκόπηση περιγραφικής στατιστικής, …
13	13	0801.3.004.0	Αντικειμενοστραφής Προγραμματισμός	3	6	Σκοπός του μαθήματος …	• Αυτόνομη Εργασία …	• Εισαγωγικά στοιχεία γλωσσών …
14	14	0801.3.005.0	Αλγόριθμοι και Δομές Δεδομένων	3	6	Το μάθημα στοχεύει στη …	• Αυτόνομη Εργασία …	• Βασικές έννοιες δομών δεδομένων.…
15	15	0801.4.001.0	Χρηματοοικονομική Λογιστική	3	6	Το μάθημα αποτελεί …	• Ατομικές Εργασίες κατά τη …	Τα περιεχόμενα του μαθήματος είναι:…
16	16	0801.4.002.0	Χρηματοοικονομική Διοίκηση	4	6	Μετά την επιτυχή …	• Ατομικές Εργασίες κατά τη …	Το περιεχόμενο του μαθήματος περιλαμβάνε
17	17	0801.4.003.0	Λήψη Επιχειρηματικών Αποφάσεων	4	6	Η λήψη αποφάσεων είναι …	• Ομαδικές Εργασίες κατά τη …	Το περιεχόμενο του μαθήματος περιλαμβάνε
18	18	0801.4.004.0	Βάσεις Δεδομένων	4	6	Σκοπός του μαθήματος …	• Αυτόνομη Εργασία …	• Βασικές αρχές συστημάτων βάσεων …
19	19	0801.4.005.0	Μέθοδοι Βελτιστοποίησης	4	6	Απόκτηση γνώσεων σε …	• Αυτόνομη Εργασία …	• Βασικές έννοιες αναζήτησης: τοπικό και
20	20	0801.5.001.0	Σχεδιασμός και Βέλτιστη Εμπειρία …	4	6	Η ψηφιακή σχεδίαση έχει …	• Ομαδική Εργασία …	• Εισαγωγή στον σχεδιασμό και στις αρχές
21	21	0801.5.002.0	Διοίκηση Έργων και Προγραμμάτων	5	6	Σκοπός του μαθήματος …	• Αυτόνομη Εργασία …	• Επιχειρησιακός σχεδιασμός έργου: …
22	22	0801.5.003.0	Διοίκηση Ποιότητας	5	6	Η ποιότητα των προϊόντων…	• Ατομικές Εργασίες κατά τη …	• Ορισμός βασικών εννοιών της διαχείριση
23	23	0801.5.004.0	Μηχανική Μάθηση	5	6	Ο στόχος του μαθήματος …	• Αυτόνομη Εργασία …	• Εισαγωγή και επισκόπηση της Τεχνητής …

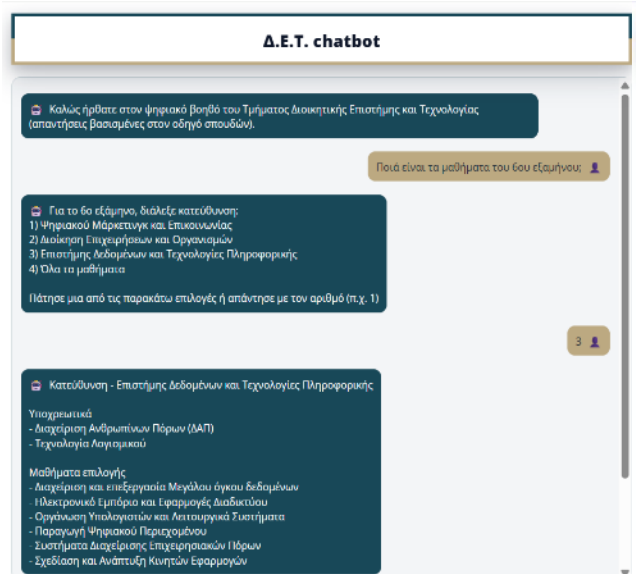
Εικόνα 17: Εγγραφές του Πίνακα det_course από τον οδηγό σπουδών

Για τη διασφάλιση της ακεραιότητας του συστήματος, η διαχείριση των δυναμικών αναγνωριστικών που σχετίζονται με τη βάση δεδομένων (όπως ονόματα πινάκων ή στηλών) πραγματοποιείται μέσω αυστηρών μηχανισμών προστασίας από επιθέσεις SQL Injection. Η υλοποίηση βασίζεται στη χρήση λευκών λιστών (whitelists) και τεχνικών παράθεσης (quoting), διασφαλίζοντας ότι μόνο έγκυρες και προεγκεκριμένες παράμετροι εκτελούνται στο επίπεδο της βάσης. Αυτή η προσέγγιση διασφαλίζει ότι το σύστημα παραμένει ασφαλές και αξιόπιστο, ενώ ταυτόχρονα διαχωρίζεται η επιχειρησιακή λογική από τις λειτουργίες πρόσβασης και διαχείρισης των δεδομένων (Silberschatz, Korth, & Sudarshan, 2019). Η

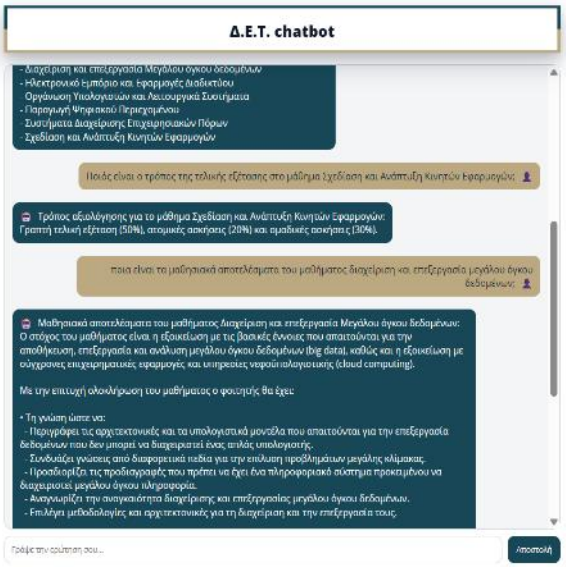
χρήση του SQLite ως αποθηκευτής για τις ερωτήσεις και τις απαντήσεις ευθυγραμμίζεται με την αρχή της απομόνωσης ευθυνών, η οποία εξασφαλίζει τη διαύγεια και τη συντηρησιμότητα του κώδικα.

3.2.4 Διεπαφή χρήστη και παρουσίαση πληροφορίας

Η διεπαφή χρήστη (User Interface) έχει σχεδιαστεί με γνώμονα την απλότητα και την αποτελεσματικότητα κατά την αλληλεπίδραση. Το κύριο αρχείο της διεπαφής είναι το ChatBotIndex.html, το οποίο αναλαμβάνει την οργάνωση και την παρουσίαση των μηνυμάτων, καθώς και των διαδραστικών επιλογών. Η σελίδα δομείται σε τρία κύρια τμήματα, τα οποία αφορούν την κεφαλίδα (header) για την ταυτότητα της εφαρμογής, την περιοχή συνομιλίας (chat) και το πεδίο εισαγωγής δεδομένων για την υποβολή των ερωτημάτων από τον χρήστη. Η περιοχή συνομιλίας διαθέτει ανεξάρτητο μηχανισμό κύλισης (scrollable), επιτρέποντας στον χρήστη να περιηγηθεί στο ιστορικό της συνομιλίας χωρίς περιορισμούς, ενώ η υπόλοιπη σελίδα παραμένει στατική για να αποφεύγεται η κύλιση κατά



Εικόνα 18: Scrollable συζήτηση (α)



Εικόνα 19: Scrollable συζήτηση (β)

λάθος (Shneiderman et al., 2016).

Η αισθητική της διεπαφής καθορίζεται μέσω CSS (Cascading Style Sheets) και βασίζεται σε μια χρωματική παλέτα (λευκό, μπλε και χρυσό). Η υλοποίηση αξιοποιεί μεταβλητές CSS (CSS variables), εξασφαλίζοντας τη συνεκτικότητα του σχεδιασμού και την ευκολία μελλοντικών προσαρμογών. Η διάταξη των μηνυμάτων είναι σαφής, με τη διαφορά μεταξύ των μηνυμάτων του χρήστη και του bot να γίνεται αντιληπτή μέσω διαφορετικών χρωμάτων και στοίχισης, καθώς και μέσω της χρήσης εικονιδίων. Αυτή η διάκριση διευκολύνει τον χρήστη στην κατανόηση της αλληλεπίδρασης και στη διαχείριση της ροής του διαλόγου (Norman, 2013).

Η αλληλεπίδραση με τη διεπαφή υποστηρίζεται από το αρχείο chatbot.js, το οποίο διαχειρίζεται την υποβολή του αιτήματος μέσω του πλήκτρου Enter και τη δυνατότητα εμφάνισης και απενεργοποίησης του κουμπιού κατά τη διάρκεια της επεξεργασίας. Η εφαρμογή πραγματοποιεί αυτόματη εστίαση (auto-scroll) στην τελευταία καταχώρηση της συνομιλίας για να διατηρεί τον χρήστη ενημερωμένο με την τελευταία απάντηση. Εάν προκύψει πρόβλημα επικοινωνίας όπως παραδείγματος σφάλματα πρωτοκόλλου HTTP, το σύστημα εμφανίζει σαφές μήνυμα σφάλματος για την ενημέρωση του χρήστη (Nielsen, 1994). Επίσης, μέσω του μηχανισμού linkify, οι υπερσύνδεσμοι (URLs), οι διευθύνσεις ηλεκτρονικού ταχυδρομείου (e mails) και οι αριθμοί τηλεφώνου μετατρέπονται σε ασφαλείς διαδραστικούς συνδέσμους, βελτιώνοντας την ευχρηστία της διεπαφής.

3.3 Τεχνολογίες και Εργαλεία Ανάπτυξης

3.3.1 Backend Τεχνολογίες

Η υλοποίηση του backend βασίζεται στη χρήση της γλώσσας προγραμματισμού Python, η οποία προσφέρει ευχρηστία, ισχυρές βιβλιοθήκες και υποστήριξη για ταχεία ανάπτυξη εφαρμογών. Το FastAPI χρησιμοποιείται για την υλοποίηση του REST API, λόγω των υψηλών επιδόσεων και της ευκολίας χρήσης του για την κατασκευή RESTful υπηρεσιών. Το FastAPI αξιοποιεί τις δυνατότητες του Python για την αποδοτική διαχείριση των αιτημάτων και της επεξεργασίας των δεδομένων. Ένα από τα κύρια πλεονεκτήματα της επιλογής του είναι η αυτόματη παραγωγή τεκμηρίωσης OpenAPI και Swagger

documentation, η οποία διευκολύνει σημαντικά τη διαδικασία αποσφαλμάτωσης (debugging) και τις δοκιμές των σημείων ελέγχου του API (Meyers, 2021).

Το FastAPI ενσωματώνει πλήρως την υποδομή async/await της Python, επιτρέποντας την ασύγχρονη επεξεργασία των εισερχόμενων αιτημάτων. Αυτή η δυνατότητα καθιστά το σύστημα εξαιρετικά αποδοτικό για περιβάλλοντα με υψηλό φόρτο εργασίας και απαιτήσεις χαμηλού χρόνου απόκρισης. Επιπλέον, το FastAPI συνεργάζεται με τη βάση δεδομένων SQLite μέσω του sqlite3 module, επιτρέποντας την αποθήκευση και ανάκτηση δεδομένων με δομημένο τρόπο (Meyers, 2021).

Για την ανάλυση και την προεπεξεργασία των εισερχόμενων δεδομένων, το σύστημα αξιοποιεί τις ενσωματωμένες βιβλιοθήκες της Python, re και unicodedata. Η βιβλιοθήκη re επιτρέπει τη χρήση κανονικών εκφράσεων (Regular Expressions) για τον εντοπισμό μοτίβων στο κείμενο, ενώ η unicodedata διασφαλίζει την κανονικοποίηση (normalization) των χαρακτήρων, εξαλείφοντας προβλήματα που προκύπτουν από τόνου ς ή διαφορετικές



Εικόνα 20: Παράδειγμα εφαρμογής FastAPI(κώδικας + συνθεση δύο εικόνων με χρήση gemini)

κωδικοποιήσεις.

Επιπλέον, η ενσωμάτωση εξειδικευμένων βιβλιοθηκών NLP (Natural Language Processing) επιτρέπει την αναγνώριση οντοτήτων και την εξαγωγή προθέσεων (intent recognition) με υψηλή ακρίβεια. Ο συνδυασμός αυτών των εργαλείων δημιουργεί έναν

ισχυρό μηχανισμό επεξεργασίας, ο οποίος μετατρέπει το αδόμητο κείμενο του χρήστη σε δομημένη πληροφορία, έτοιμη προς επεξεργασία από τη βάση δεδομένων. Ο συνδυασμός αυτών των τεχνολογιών παρέχει το κατάλληλο περιβάλλον για την ανάπτυξη σύγχρονων και υψηλής απόδοσης web εφαρμογών (Van Rossum, 2020).

3.3.2 Τεχνολογίες Διεπαφής

Η ανάπτυξη της διεπαφής χρήστη σχεδιάζεται με τη χρήση βασικών τεχνολογιών του web, δηλαδή HTML, CSS και JavaScript. Ο συνδυασμός αυτών των τεχνολογιών επιτρέπει την κατασκευή μιας δυναμικής και αλληλεπιδραστικής εμπειρίας χρήστη, που βασίζεται στις αρχές του αποκριτικού σχεδιασμού (responsive design) και της προσαρμογής στις ανάγκες των χρηστών. Η HTML (HyperText Markup Language) παρέχει τη δομή της σελίδας, με τη χρήση στοιχείων για την οργανωμένη διάταξη του περιεχομένου. Αυτά τα στοιχεία εξυπηρετούν την κατασκευή της βασικής δομής του chatbot, την περιοχή συνομιλίας και τη φόρμα εισαγωγής δεδομένων (Bertini et al., 2018).

Η μορφοποίηση της σελίδας επιτυγχάνεται μέσω του CSS (Cascading Style Sheets), το οποίο καθορίζει την εμφάνιση των στοιχείων της HTML και διασφαλίζει την αισθητική συνέπεια του συστήματος. Οι μεταβλητές CSS (CSS Custom Properties) χρησιμοποιούνται για την εύκολη προσαρμογή του στυλ και τη συντήρηση του κώδικα, επιτρέποντας την κεντρική διαχείριση των χρωμάτων, των διαστάσεων και της διάταξης των στοιχείων της σελίδας. Το CSS είναι υπεύθυνο για τη δημιουργία καθαρών, οργανωμένων οπτικών στοιχείων, όπως είναι η διάκριση των μηνυμάτων του χρήστη και του bot, ενώ διασφαλίζει τη βελτιστοποιημένη απεικόνιση σε συσκευές με διαφορετικά μεγέθη οθόνης (Sharma & Upadhyaya, 2019).

Η δυναμική αλληλεπίδραση του χρήστη με την εφαρμογή επιτυγχάνεται μέσω του JavaScript, το οποίο διαχειρίζεται τις ενέργειες του χρήστη, όπως την αποστολή μηνυμάτων ή την υποβολή επιλογών. Το JavaScript χρησιμοποιείται για να ενεργοποιεί τις διεπαφές, να συνδέει το frontend με το backend μέσω της αποστολής αιτημάτων API και να ενημερώνει την οθόνη χωρίς να απαιτεί ανανέωση της σελίδας (Ajax). Αυτή η τεχνική παρέχει ταχεία

ανταπόκριση και διασφαλίζει την αποτελεσματική αλληλεπίδραση του χρήστη με το σύστημα σε πραγματικό χρόνο (Fitzpatrick & Wolf, 2017).

Τέλος, για την επιστημονική εγκυρότητα της παρουσίασης, ενσωματώνεται η βιβλιοθήκη KaTeX. Πρόκειται για μια ταχύτατη μηχανή απόδοσης (math rendering engine), η οποία μετατρέπει τις συμβολικές εκφράσεις LaTeX σε οπτικά άρτιους μαθηματικούς τύπους απευθείας στον περιηγητή. Η επιλογή της KaTeX έναντι εναλλακτικών λύσεων βασίστηκε στην ανώτερη ταχύτητα επεξεργασίας και στη συμβατότητά της με τη δυναμική φύση της JavaScript διεπαφής.

Οι παραπάνω τεχνολογίες συνεργάζονται αρμονικά για την απόδοση του περιεχομένου, με έμφαση στη σωστή διαχείριση του ιστορικού συνομιλίας, την ενημέρωση των στοιχείων σε πραγματικό χρόνο και την απεικόνιση των επιλογών του χρήστη με τρόπο που διευκολύνει την αλληλεπίδραση και βελτιώνει την εμπειρία χρήστη.

3.3.3 Βάση Δεδομένων και Εργαλεία Υποδομής

Για τη διαχείριση της γνωσιακής βάσης χρησιμοποιήθηκε η SQLite, ένα ελαφρύ και ενσωματωμένο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Η βάση δεδομένων εντοπίζεται στο αρχείο det.db, το οποίο βρίσκεται στον ριζικό κατάλογο του έργου. Η σύνδεση με την βάση επιτυγχάνεται μέσω της βιβλιοθήκης sqlite3 της Python, χρησιμοποιώντας την τεχνική Row factory για τη διαχείριση των δεδομένων ως λεξικά, επιτρέποντας την ευχρηστία στην πρόσβαση στις στήλες της βάσης με ονομαστική αναφορά (Elmasri & Navathe, 2016). Η σύνδεση και η διαχείριση των ερωτημάτων πραγματοποιούνται μέσω συνάρτησης _get_conn, η οποία δημιουργεί και επιστρέφει την σύνδεση με τη βάση δεδομένων.

Το πρότυπο Repository (Repository Pattern) χρησιμοποιείται για να απομονώσει τις λειτουργίες πρόσβασης στα δεδομένα από την επιχειρησιακή λογική του συστήματος. Αυτή η προσέγγιση επιτρέπει την εύκολη συντήρηση του κώδικα, καθώς οι συναρτήσεις που αλληλεπιδρούν με τη βάση δεδομένων είναι απομονωμένες σε συγκεκριμένα αρχεία και μπορούν να αναπτύσσονται ανεξάρτητα από την υπόλοιπη εφαρμογή (Fowler, 2002). Με την τεχνική αυτή, το σύστημα εξασφαλίζει την επεκτασιμότητα και τη διαφάνεια στις λειτουργίες

που σχετίζονται με τη διαχείριση των δεδομένων. Επιπλέον, η χρήση της μνήμης cache με τη βιβλιοθήκη lru_cache για τον έλεγχο της ύπαρξης πινάκων στη βάση, βοηθά στην αποδοτικότητα της διαδικασίας και μειώνει την ανάγκη για επαναλαμβανόμενα ερωτήματα (Silberschatz et al., 2019).

Η διαχείριση των εξαρτήσεων του έργου πραγματοποιείται μέσω του αρχείου requirements.txt, το οποίο περιλαμβάνει όλες τις βιβλιοθήκες Python που απαιτούνται για την εκτέλεση της εφαρμογής. Η χρήση αυτού του αρχείου εξασφαλίζει τη δυνατότητα αναπαραγωγής του περιβάλλοντος ανάπτυξης και την εύκολη εγκατάσταση των βιβλιοθηκών μέσω του εργαλείου pip, ενισχύοντας τη συντηρησιμότητα και την ομαλή λειτουργία της εφαρμογής σε διάφορα περιβάλλοντα (Meyers, 2021).

3.3.3 Βάση Δεδομένων και Εργαλεία Υποδομής

Για τη διαχείριση της γνωσιακής βάσης χρησιμοποιήθηκε η SQLite, ένα ελαφρύ και ενσωματωμένο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Η βάση δεδομένων εντοπίζεται στο αρχείο det.db, το οποίο βρίσκεται στον ριζικό κατάλογο του έργου. Η σύνδεση με την βάση επιτυγχάνεται μέσω της βιβλιοθήκης sqlite3 της Python, χρησιμοποιώντας την τεχνική Row factory για τη διαχείριση των δεδομένων ως λεξικά, επιτρέποντας την ευχρηστία στην πρόσβαση στις στήλες της βάσης με ονομαστική αναφορά (Elmasri & Navathe, 2016). Η σύνδεση και η διαχείριση των ερωτημάτων πραγματοποιούνται μέσω συνάρτησης _get_conn, η οποία δημιουργεί και επιστρέφει την σύνδεση με τη βάση δεδομένων.

Το πρότυπο Repository (Repository Pattern) χρησιμοποιείται για να απομονώσει τις λειτουργίες πρόσβασης στα δεδομένα από την επιχειρησιακή λογική του συστήματος. Αυτή η προσέγγιση επιτρέπει την εύκολη συντήρηση του κώδικα, καθώς οι συναρτήσεις που αλληλεπιδρούν με τη βάση δεδομένων είναι απομονωμένες σε συγκεκριμένα αρχεία και μπορούν να αναπτύσσονται ανεξάρτητα από την υπόλοιπη εφαρμογή (Fowler, 2002). Με την τεχνική αυτή, το σύστημα εξασφαλίζει την επεκτασιμότητα και τη διαφάνεια στις λειτουργίες που σχετίζονται με τη διαχείριση των δεδομένων. Επιπλέον, η χρήση της μνήμης cache με τη

βιβλιοθήκη lru_cache για τον έλεγχο της ύπαρξης πινάκων στη βάση, βοηθά στην αποδοτικότητα της διαδικασίας και μειώνει την ανάγκη για επαναλαμβανόμενα ερωτήματα (Silberschatz et al., 2019).

Η διαχείριση των εξαρτήσεων του έργου πραγματοποιείται μέσω του αρχείου requirements.txt, το οποίο περιλαμβάνει όλες τις βιβλιοθήκες Python που απαιτούνται για την εκτέλεση της εφαρμογής. Η χρήση αυτού του αρχείου εξασφαλίζει τη δυνατότητα αναπαραγωγής του περιβάλλοντος ανάπτυξης και την εύκολη εγκατάσταση των βιβλιοθηκών μέσω του εργαλείου pip, ενισχύοντας τη συντηρησιμότητα και την ομαλή λειτουργία της εφαρμογής σε διάφορα περιβάλλοντα (Meyers, 2021).



Εικόνα 21: Βάση δεδομένων και εργαλεία υποδομής (δημιουργία με χρήση gemini)

3.4 Υλοποίηση και Διασφάλιση Ποιότητας

3.4.1 Ανάπτυξη της Λογικής Επιχειρησιακής Διαδικασίας

Η ανάπτυξη της επιχειρησιακής λογικής του συστήματος αποτελεί τον πυρήνα της λειτουργικότητας του, καθώς διαχειρίζεται την επεξεργασία των δεδομένων και την εκτέλεση των αιτημάτων του χρήστη. Η επιχειρησιακή λογική εκτελείται κυρίως μέσω του backend και επιτρέπει την αλληλεπίδραση με τις διάφορες υπηρεσίες του συστήματος, η ανάκτηση δεδομένων από τη βάση, και η απόδοση των κατάλληλων απαντήσεων στον χρήστη (Fowler, 2002).

Κεντρικό ρόλο στη διαδικασία αυτή κατέχει ο μηχανισμός detect_intent, ο οποίος αναλύει το κείμενο του χρήστη, κανονικοποιεί τα δεδομένα, και εκτελεί διαδικασίες που σχετίζονται με την αναγνώριση της πρόθεσης και της κατηγορίας του αιτήματος. Ο αλγόριθμος αυτός βασίζεται στην αναγνώριση συγκεκριμένων λέξεων-κλειδιών και οντοτήτων, συνδυάζοντας αυτές τις πληροφορίες για να εξάγει την ακριβή πρόθεση του χρήστη και να δρομολογήσει το αίτημα στην κατάλληλη υπηρεσία (Manning et al., 2008).

Η επεξεργασία των αιτημάτων ακολουθεί ένα ιεραρχικό μοντέλο, όπου τα δεδομένα πρώτα υποβάλλονται σε ανάλυση για την ανίχνευση της πρόθεσης και στη συνέχεια καταχωρούνται στην αντίστοιχη βάση δεδομένων μέσω των κατάλληλων SQL queries. Η λογική της επιχειρησιακής διαδικασίας περιλαμβάνει τον έλεγχο και την επεξεργασία αυτών των δεδομένων με χρήση της βάσης δεδομένων SQLite. Ο μηχανισμός αυτός εξασφαλίζει τη σωστή αποθήκευση, ανάκτηση και επεξεργασία των δεδομένων σύμφωνα με την ανάγκη της επιχειρησιακής διαδικασίας, ενώ ταυτόχρονα διασφαλίζει την ασφάλεια των δεδομένων μέσω τεχνικών ελέγχου εισόδου και παρακολούθησης των αιτημάτων (Elmasri & Navathe, 2016).

Συνοψίζοντας, η υλοποίηση της επιχειρησιακής λογικής εξασφαλίζει την αλληλεξάρτηση μεταξύ των διάφορων επιπέδων του συστήματος, από τη βάση δεδομένων έως την παρουσίαση των αποτελεσμάτων στην διεπαφή χρήστη. Με την εφαρμογή αυτών των μηχανισμών διασφαλίζεται η αποτελεσματικότητα και η ακρίβεια της διαδικασίας, καθώς και η καλή συντήρηση του συστήματος με τη χρήση καθαρών συναρτήσεων και τη διαχείριση των δεδομένων μέσω ενός ξεκάθαρου Repository Pattern (Fowler, 2002).

3.4.2 Διαχείριση Σφαλμάτων και Ανάκτηση

Η διαχείριση σφαλμάτων και η αποκατάσταση της ομαλής λειτουργίας αποτελούν κρίσιμες διαδικασίες για την εξασφάλιση της αξιοπιστίας του συστήματος. Σε επίπεδο διεπαφής, η εφαρμογή αντιμετωπίζει σφάλματα μέσω σαφών μηνυμάτων προς τον χρήστη, εξασφαλίζοντας ότι ο χρήστης ενημερώνεται για οποιοδήποτε πρόβλημα παρουσιαστεί κατά τη διάρκεια της αλληλεπίδρασης. Η προσέγγιση αυτή βελτιώνει την εμπειρία χρήστη, ελαχιστοποιώντας την αίσθηση απογοήτευσης σε περιπτώσεις αποτυχίας και προσφέροντας εναλλακτικές λύσεις για την αποκατάσταση της αλληλεπίδρασης (Shneiderman et al., 2016).

Η διαχείριση σφαλμάτων στο backend περιλαμβάνει την έγκαιρη αναγνώριση και την αποκατάσταση λαθών κατά την εκτέλεση των αιτημάτων. Τα σφάλματα καταγράφονται και αντιμετωπίζονται μέσω δομημένου exception handling, το οποίο επιτρέπει την αναγνώριση και την απομόνωση των προβλημάτων χωρίς να διακόπτεται η λειτουργία του συστήματος. Ο μηχανισμός αυτός διασφαλίζει ότι τα σφάλματα που ενδέχεται να προκύψουν κατά την

επεξεργασία δεδομένων, όπως προβλήματα σύνδεσης με τη βάση ή λάθη κατά την εκτέλεση SQL queries, δεν επηρεάζουν την συνολική απόδοση της εφαρμογής. Στις περιπτώσεις σφαλμάτων που επηρεάζουν τη βάση δεδομένων, οι συναρτήσεις αποκατάστασης προσπαθούν να εξασφαλίσουν την ακεραιότητα των δεδομένων και να επιστρέψουν τη διαδικασία σε κανονική ροή, είτε μέσω επανεκκίνησης των λειτουργιών είτε με την εφαρμογή fallback στρατηγικών (Meyers, 2021).

Επιπλέον, η ενσωμάτωση μηχανισμών καταγραφής σφαλμάτων (logging) αποτελεί θεμελιώδη πρακτική για την αποδοτική διάγνωση και την αντιμετώπιση δυσλειτουργιών σε μελλοντικές εκδόσεις. Η παρακολούθηση αυτών των καταγραφών επιτρέπει στους διαχειριστές να ανιχνεύσουν μοτίβα σφαλμάτων και να προβούν σε προληπτικές ενέργειες για την αποτροπή τους, ενισχύοντας την σταθερότητα του συστήματος μακροπρόθεσμα (Fitzpatrick & Wolf, 2017).

3.4.3 Διαδικασία Δοκιμών και Επαλήθευσης

Η διαδικασία δοκιμών και επαλήθευσης είναι θεμελιώδης για την εξασφάλιση της ποιότητας και της σωστής λειτουργίας του συστήματος. Οι δοκιμές μονάδας (unit tests) χρησιμοποιούνται για να ελέγξουν τις βασικές λειτουργίες του συστήματος, διασφαλίζοντας ότι κάθε μεμονωμένη μονάδα κώδικα, όπως οι συναρτήσεις και οι μέθοδοι εκτελεί τη λειτουργία της όπως αναμένεται. Αυτές οι δοκιμές εκτελούνται αυτόματα κατά τη διάρκεια του κύκλου ανάπτυξης, επιτρέποντας την έγκαιρη αναγνώριση σφαλμάτων και τη μείωση του κόστους συντήρησης (Beck, 2003).

```python
def test_normalize_variants(raw, expected):
    assert normalize(raw) == expected


def test_normalize_removes_tonos_and_lowercases():
    assert normalize("Άσκηση, Εξέταση, Βιβλιογραφία") == "ασκηση, εξεταση,


def test_normalize_collapses_internal_whitespace():
    assert normalize("ένα \n δύο\t\tτρία") == "ενα δυο τρια"


def test_keyword_collections_are_pre_normalized():
    all_collections = [FACILITY_KEYWORDS, ENTITY_KEYWORDS, FIELD_KEYWORDS]
    for collection in all_collections:
        for _, values in collection.items():
            for value in values:
                assert value == normalize(value)


def test_course_triggers_are_pre_normalized():
    for trigger in COURSE_TRIGGERS:
        assert trigger == normalize(trigger)


@pytest.mark.parametrize(
    "text, expected",
    [
        ("μαθηματα 6 εξαμηνου", 6),
        ("6ο εξαμηνο", 6),
        ("6ου εξαμηνου", 6),
        ("10ο εξαμηνο", 10),
        ("semester 7", 7),
        ("στο 8 semester υπαρχουν μαθηματα", 8),
    ],
)
def test_parse_semester_numeric_and_fallback_cases(text, expected):
    assert parse_semester(normalize(text)) == expected
```

Εικόνα 23: parsing test(unit test)

```python
def _patch_repo_find_course(monkeypatch, resolver):
    calls = []

    def fake_find_course_in_text(text):
        calls.append(text)
        return resolver(text)

    monkeypatch.setattr(di.repo, "find_course_in_text", fake_find_course_in_text
    return calls


def test_detect_intent_courses_by_semester_numeric(monkeypatch):
    _patch_repo_find_course(monkeypatch, lambda _: None)

    result = di.detect_intent("μαθηματα 6 εξαμηνου")

    assert result["intent"] == "courses_by_semester"
    assert result["confidence"] == 0.90
    assert result["entities"]["semester"] == 6
    assert result["entities"]["entity"] == "course"
    assert result["entities"]["question"] == "μαθηματα 6 εξαμηνου"
    assert result["entities"]["normalized"] == normalize("μαθηματα 6 εξαμηνου")


def test_detect_intent_courses_by_semester_ordinal(monkeypatch):
    _patch_repo_find_course(monkeypatch, lambda _: None)

    result = di.detect_intent("μαθηματα εκτα εξαμηνα")

    assert result["intent"] == "courses_by_semester"
    assert result["confidence"] == 0.90
    assert result["entities"]["semester"] == 6


def test_detect_intent_directions_info(monkeypatch):
    _patch_repo_find_course(monkeypatch, lambda _: None)

    result = di.detect_intent("ποιες κατευθυνσεις εχει το τμημα;")

    assert result["intent"] == "directions_info"
```

Εικόνα 22: detect_indent test (unit test)

Για τον έλεγχο της ολοκληρωμένης λειτουργικότητας του συστήματος, πραγματοποιήθηκαν δοκιμές ολοκλήρωσης (integration tests), οι οποίες εστιάζουν στην αλληλεπίδραση μεταξύ των επιμέρους επιπέδων, όπως η επικοινωνία του frontend με το API και η σύνδεση του backend με τη βάση δεδομένων SQLite. Οι δοκιμές ολοκλήρωσης επιβεβαιώνουν ότι το σύστημα λειτουργεί ως ενιαία μονάδα, διασφαλίζοντας τη σωστή ροή δεδομένων και επικοινωνίας μεταξύ των επιπέδων του συστήματος (Meyers, 2021). Στο σύστημα χρησιμοποιούνται βιβλιοθήκες όπως το pytest για την εκτέλεση αυτών των δοκιμών, που παρέχουν εύκολη συγγραφή, εκτέλεση και παρακολούθηση των αποτελεσμάτων.



```python
@pytest.fixture
def client():
    with TestClient(app) as c:
        yield c


def _install_stubs(monkeypatch, *, route_return=None, course_return=None):
    route_calls = []
    course_calls = []
    log_calls = []

    if route_return is None:
        route_return = {
            "answer": "OK",
            "intent": "fallback",
            "confidence": 0.5,
            "entities": {},
        }

    def fake_route_intent(q, client_key):
        route_calls.append({"q": q, "client_key": client_key})
        if callable(route_return):
            return route_return(q, client_key)
        return dict(route_return)

    def fake_find_course_in_text(q):
        course_calls.append(q)
        if callable(course_return):
            return course_return(q)
        return course_return

    def fake_log_turn(**kwargs):
        log_calls.append(kwargs)

    monkeypatch.setattr(chat_module, "route_intent", fake_route_intent)
    monkeypatch.setattr(chat_module, "find_course_in_text", fake_find_course_in_text)
    monkeypatch.setattr(chat_module, "log_turn", fake_log_turn)

    return route_calls, course_calls, log_calls
```

Εικόνα 24: Api_chat test (integration tests)

Επιπρόσθετα, η στρατηγική δοκιμών περιλαμβάνει επίσης τη δοκιμή απόδοσης (performance testing), με σκοπό την αξιολόγηση της ταχύτητας και της ικανότητας του συστήματος να διαχειρίζεται μεγάλο όγκο δεδομένων ή πολλαπλές ταυτόχρονες αιτήσεις. Οι δοκιμές απόδοσης διασφαλίζουν ότι το σύστημα μπορεί να ανταποκριθεί γρήγορα, ακόμη και σε περιβάλλοντα με υψηλό φόρτο εργασίας, και ότι η απόδοση του παραμένει σταθερή σε συνθήκες αυξημένης χρήσης (Fowler, 2002).

Τέλος, για την αυτοματοποίηση της διαδικασίας δοκιμών, υιοθετήθηκαν πρακτικές Συνεχούς Ενσωμάτωσης (Continuous Integration) με τη χρήση εργαλείων όπως το Jenkins. Το συγκεκριμένο εργαλείο επιτρέπει την αυτοματοποιημένη και επαναλαμβανόμενη εκτέλεση των δοκιμών (Unit & Integration tests) σε κάθε φάση του κύκλου ανάπτυξης. Η στρατηγική αυτή διασφαλίζει ότι το σύστημα υποβάλλεται σε συνεχή έλεγχο, εγγυώμενη ότι οι νέες προσθήκες ή αλλαγές στον κώδικα δεν προκαλούν παλινδρομήσεις (regressions) ή

δυσλειτουργίες στις υπάρχουσες δυνατότητες της εφαρμογής. Η ενσωμάτωση τέτοιων εργαλείων ενισχύει την αξιοπιστία του Bot και επιτρέπει την ταχεία διάθεση νέων εκδόσεων (deployment) με τη μέγιστη δυνατή ασφάλεια (Fitzpatrick & Wolf, 2017).

3.5 Δοκιμές και Αξιολόγηση

Η φάση της αξιολόγησης αποτέλεσε το τελικό στάδιο της ανάπτυξης, όπου το σύστημα υποβλήθηκε σε συστηματικούς ελέγχους για να διαπιστωθεί η συνέπειά του με τις λειτουργικές απαιτήσεις και η ορθή συμπεριφορά του σε πραγματικές αλληλεπιδράσεις. Η στρατηγική δοκιμών δεν περιορίστηκε στην ανίχνευση σφαλμάτων, αλλά κάλυψε τόσο την αξιοπιστία της ροής επεξεργασίας όσο και ζητήματα ασφάλειας και χρηστικότητας.

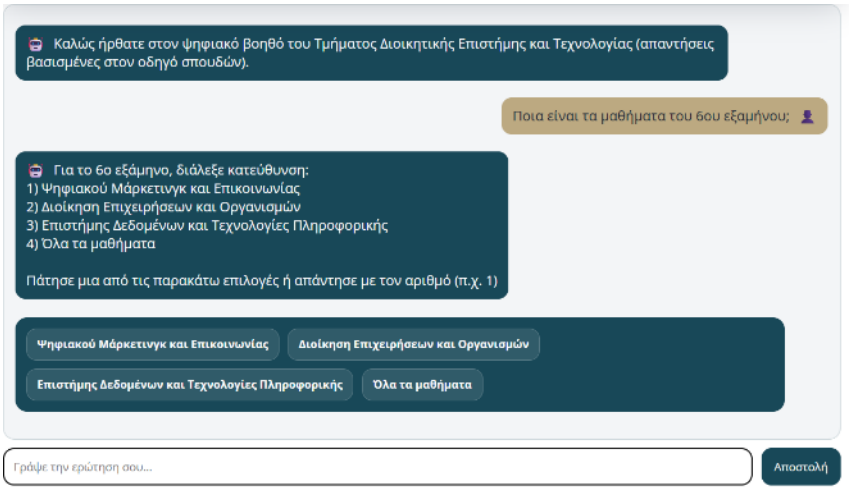
Μέσω δοκιμών μονάδας (unit tests) επαληθεύτηκε η σωστή λειτουργία κρίσιμων επιμέρους τμημάτων, όπως η κανονικοποίηση των εισερχομένων, η αντιστοίχιση προθέσεων (intent detection) και η ορθή επιστροφή απαντήσεων από τα επιμέρους modules. Παράλληλα, οι δοκιμές ολοκλήρωσης (integration tests) επιβεβαίωσαν την απρόσκοπτη συνεργασία frontend και backend, ελέγχοντας ότι οι κλήσεις API εκτελούνται σωστά και ότι η πληροφορία ρέει ομαλά από τον χρήστη προς το σύστημα και πίσω (Meyers, 2021).

Ιδιαίτερη έμφαση δόθηκε στη θωράκιση της πρόσβασης στη βάση δεδομένων και στον χειρισμό εισόδου χρήστη. Πραγματοποιήθηκαν δοκιμές ασφαλείας με “κακόβουλα” ή οριακά inputs ώστε να επιβεβαιωθεί ότι το σύστημα παραμένει ανθεκτικό απέναντι σε SQL injection και Cross-Site Scripting (XSS), κυρίως μέσω χρήσης παραμετροποιημένων ερωτημάτων κατά την πρόσβαση στη SQLite και ελέγχων/φιλτραρίσματος στην είσοδο πριν από την επεξεργασία (Elmasri & Navathe, 2016). Επιπλέον, ελέγχθηκε η ορθή λειτουργία του frontend σε σενάρια μη αναμενόμενων χαρακτήρων/μορφών εισόδου, ώστε να αποφεύγονται προβλήματα εμφάνισης ή αλληλεπίδρασης (Fitzpatrick & Wolf, 2017).

Οι δοκιμές απόδοσης (performance tests) επικεντρώθηκαν στη μέτρηση της ταχύτητας απόκρισης σε αντιπροσωπευτικά σενάρια χρήσης (π.χ. ερωτήσεις που επιστρέφονται από στατικές απαντήσεις έναντι ερωτήσεων που απαιτούν ανάκτηση από τη βάση). Η απόκριση βελτιώθηκε πρακτικά μέσω της οργάνωσης των συχνών/σταθερών απαντήσεων σε δομές μνήμης (π.χ. στατικά λεξικά/menus) και της περιορισμένης,

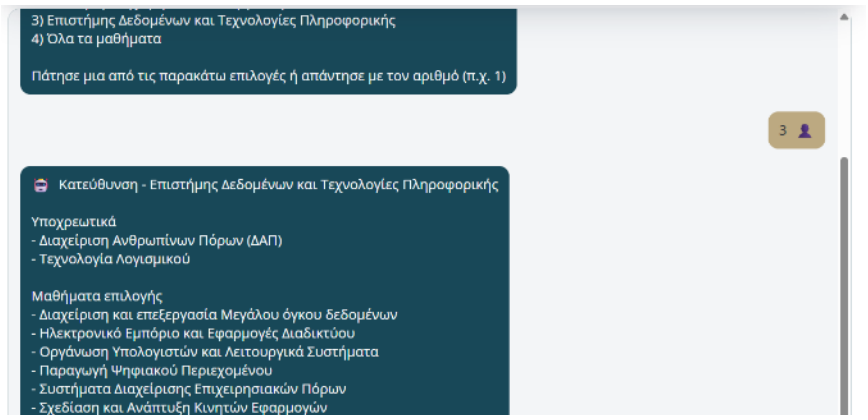
στοχευμένης πρόσβασης στη βάση μόνο όταν απαιτείται, μειώνοντας περιττές κλήσεις και καθυστερήσεις.

Τέλος, η αξιολόγηση ολοκληρώθηκε με έλεγχο εμπειρίας χρήστη (UX), όπου η παρατήρηση της αλληλεπίδρασης και η συλλογή ανατροφοδότησης οδήγησαν σε βελτιώσεις στη σαφήνεια των απαντήσεων, στη συνέπεια της διεπαφής και στον χειρισμό "δύσκολων" περιπτώσεων (π.χ. ασαφή ερωτήματα ή μη αναγνωρισμένες προθέσεις). Η συνολική αυτή προσέγγιση διασφάλισε ότι το σύστημα είναι σταθερό, ασφαλές και λειτουργικά αξιόπιστο,



Εικόνα 25: Παράδειγμα χρήσης Ψηφιακού βοηθού(α)

ενώ παραμένει φιλικό και κατανοητό για τον τελικό χρήστη (Shneiderman et al., 2016).



Εικόνα 26: Παράδειγμα χρήσης Ψηφιακού βοηθού (β)

ΚΕΦΑΛΑΙΟ 4

ΣΥΜΠΕΡΑΣΜΑΤΑ

4.1 Συμπεράσματα

Η ολοκλήρωση της παρούσας εργασίας επιβεβαιώνει ότι ο αρχικός σκοπός επιτεύχθηκε πλήρως, καθώς υλοποιήθηκε ένα λειτουργικό και ασφαλές σύστημα ικανό να διαχειρίζεται αυτοματοποιημένα τις πληροφοριακές ανάγκες των χρηστών. Η υλοποίηση επιβεβαίωσε ότι ένα σύστημα εξυπηρέτησης πληροφοριακών αναγκών μπορεί να επιτύχει υψηλή πρακτική χρησιμότητα όταν η επιχειρησιακή λογική ορίζεται με σαφείς κανόνες δρομολόγησης και προτεραιοποίησης. Η αξιοπιστία προκύπτει από την προβλέψιμη αντιστοίχιση ερωτημάτων σε συγκεκριμένες ροές απάντησης, γεγονός που μειώνει την πιθανότητα ασυνεπών αποκρίσεων και διευκολύνει τον έλεγχο ποιότητας.

Επιπλέον, ο διαχωρισμός σε επίπεδα επέτρεψε την ανεξάρτητη εξέλιξη της διεπαφής, των μηχανισμών ανίχνευσης πρόθεσης και της πρόσβασης δεδομένων. Η επιλογή του προτύπου αποθετηρίου (Repository) πάνω από τη SQLite συνέβαλε στη σαφή οριοθέτηση ευθυνών, συγκεντρώνοντας τις λειτουργίες ανάκτησης σε διακριτές συναρτήσεις και ενισχύοντας την ανθεκτικότητα του συστήματος. Παράλληλα, οι έλεγχοι ταυτοποίησης αναγνωριστικών και η συντηρητική πρόσβαση στη βάση υποστήριξαν την ασφάλεια και τη σταθερότητα.

Η διαχείριση κατάστασης, με περιορισμένη διάρκεια και σαφείς εκκρεμότητες, απέδειξε ότι είναι επαρκής για ανάγκες καθοδηγούμενης διευκρίνισης, χωρίς να απαιτείται σύνθετη μοντελοποίηση διαλόγου. Σε επίπεδο διεπαφής, η καθαρή οπτική διάκριση, η μετατροπή συνδέσμων σε λειτουργικά στοιχεία και η οργάνωση επιλογών διευκόλυναν την κατανόηση των απαντήσεων. Η προσέγγιση αυτή αποτελεί ένα πρότυπο ψηφιακού μετασχηματισμού για το τμήμα ΔΕΤ, αναδεικνύοντας πώς η τεχνολογική αυτοματοποίηση μπορεί να βελτιστοποιήσει τις εσωτερικές διαδικασίες και να αποσυμφορήσει το προσωπικό.

Τέλος, η εργασία αναδεικνύει ότι η διαφάνεια στον σχεδιασμό και η συντηρητική πρόσβαση στη βάση δεδομένων SQLite εξασφαλίζουν ένα σταθερό και επεκτάσιμο περιβάλλον. Η μελλοντική βελτίωση θα εστιάσει στη συστηματική επέκταση της βάσης γνώσης και στη βελτιστοποίηση της ανίχνευσης πρόθεσης, ενσωματώνοντας πιο ευέλικτους αλγορίθμους χωρίς να θυσιάζεται ο έλεγχος που προσφέρει η τρέχουσα υλοποίηση.

4.2 Περιορισμοί και Προβλήματα

Παρά την επιτυχή ενσωμάτωση των βασικών λειτουργιών, η ανάπτυξη του συστήματος ανέδειξε συγκεκριμένους περιορισμούς που επηρεάζουν το εύρος και την ευελιξία των αποκρίσεων. Η υιοθέτηση της κανονοκεντρικής προσέγγισης, αν και σταθερή, αποδεικνύεται ευάλωτη σε εισόδους που χαρακτηρίζονται από συντακτικές ασάφειες, παραλλαγές της φυσικής γλώσσας ή ελλιπή συμφραζόμενα. Στις περιπτώσεις όπου το αίτημα στερείται σαφών δεικτών πρόθεσης, η ακρίβεια της ανίχνευσης μειώνεται, αυξάνοντας την ανάγκη για διευκρινίσεις τις οποίες η τρέχουσα λογική δεν είναι πάντα σε θέση να διαχειριστεί πλήρως αυτοματοποιημένα.

Ένας δεύτερος περιορισμός εντοπίζεται στη βάση δεδομένων det.db. Παρότι το αποθετήριο (repository) διαθέτει μηχανισμούς για δυναμική ανάγνωση στηλών, η ποιότητα της απάντησης παραμένει άρρηκτα συνδεδεμένη με την πληρότητα των εγγραφών και τη συνέπεια του σχήματος. Τυχόν αποκλίσεις στην ονοματολογία των πινάκων ή κενά πεδία στη βάση γνώσης οδηγούν αναπόφευκτα σε ελλιπείς αποκρίσεις, μειώνοντας το συνολικό ποσοστό κάλυψης των ερωτημάτων.

Επιπλέον, η διαχείριση διαλόγου με ελαφρά κατάσταση και χρονικό όριο μπορεί να μην διατηρεί επαρκές πλαίσιο σε διαδοχικά αιτήματα, ειδικά όταν ο χρήστης αλλάζει γρήγορα θέμα. Αυτό ενδέχεται να προκαλέσει ανεπιθύμητες μεταβάσεις ροής ή να απαιτήσει επανάληψη πληροφορίας από τον χρήστη, αυξάνοντας το γνωστικό φορτίο.

Τέλος, εντοπίστηκαν ζητήματα παρατηρησιμότητας και συντήρησης. Η απουσία οργανωμένων μετρικών και συστηματικής καταγραφής δυσχεραίνει τη διάγνωση σφαλμάτων και την τεκμηριωμένη αξιολόγηση. Παράλληλα, η επέκταση του συνόλου κανόνων αυξάνει την πολυπλοκότητα δρομολόγησης και καθιστά πιο απαιτητικό τον έλεγχο παρενεργειών,

άρα απαιτείται αυστηρότερη πειθαρχία σε δοκιμές και τεκμηρίωση κατά τις επόμενες εκδόσεις.

4.3 Προτάσεις για Μελλοντική Εξέλιξη

Με βάση την τρέχουσα υλοποίηση και τα ευρήματα των δοκιμών, αναγνωρίζονται συγκεκριμένες περιοχές που προσφέρονται για περαιτέρω ανάπτυξη και βελτιστοποίηση. Αρχικά, η προσέγγιση βασισμένη σε κανόνες για την ανίχνευση προθέσεων και τη διαχείριση των διαλόγων μπορεί να επεκταθεί για να υποστηρίξει πιο πολύπλοκα σενάρια φυσικής γλώσσας, όπως η αναγνώριση της έννοιας πίσω από τις ερωτήσεις και η καλύτερη διαχείριση αμφισημιών. Η ενσωμάτωση αλγορίθμων μηχανικής μάθησης ή NLP για την ανάλυση προθέσεων μπορεί να ενισχύσει την ακρίβεια του συστήματος και να επιτρέψει την προσαρμογή σε νέες και πιο ποικίλες ερωτήσεις από τους χρήστες (Manning et al., 2008).

Δεύτερον, η υποδομή της βάσης δεδομένων, αν και λειτουργική, μπορεί να επεκταθεί α για να υποστηρίξει περιβάλλοντα με υψηλό φόρτο εργασίας. Η αναβάθμιση της βάσης δεδομένων σε ένα πιο ισχυρό σύστημα διαχείρισης βάσεων δεδομένων (DBMS), όπως το PostgreSQL, και η εισαγωγή τεχνικών όπως η κατανομή δεδομένων ή η χρήση μηχανισμών caching, θα προσφέρει ανώτερη απόδοση και απρόσκοπτη επεκτασιμότητα (Elmasri & Navathe, 2016).

Ένα ακόμη κρίσιμο βήμα είναι η ενσωμάτωση σύνθετων μηχανισμών καταγραφής (logging) και παρακολούθησης σφαλμάτων για την ενίσχυση της παρατηρησιμότητας του συστήματος. Η χρήση εργαλείων όπως το ELK Stack (Elasticsearch, Logstash, Kibana) θα επιτρέψει την ανάλυση μεγάλων ποσοτήτων δεδομένων σε πραγματικό χρόνο και τη γρήγορη ανίχνευση προβλημάτων, εξασφαλίζοντας τη διαρκή βελτίωση του συστήματος (Fitzpatrick & Wolf, 2017).

Επιπλέον, η βελτίωση της διεπαφής χρήστη μπορεί να περιλαμβάνει την εισαγωγή πιο προηγμένων δυνατοτήτων αλληλεπίδρασης, όπως η ενσωμάτωση φωνητικής αναγνώρισης ή η χρήση δυναμικών γραφικών για τη βελτίωση της εμπειρίας χρήστη. Η ενσωμάτωση εργαλείων ανάλυσης δεδομένων χρήστη, όπως η ανάλυση χρήσης και η παρακολούθηση της

αλληλεπίδρασης, μπορεί να προσφέρει πολύτιμα δεδομένα για την εξέλιξη και βελτίωση της διεπαφής (Shneiderman et al., 2016).

Τέλος, η χρήση αυτοματοποιημένων εργαλείων για την ανάπτυξη και τη συντήρηση των δοκιμών θα μπορούσε να μειώσει το χρόνο ανάπτυξης και να αυξήσει την ποιότητα του συστήματος, επιτρέποντας την ταχεία ενσωμάτωση νέων χαρακτηριστικών χωρίς να διακυβεύεται η αξιοπιστία του συστήματος.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Α. ΞΕΝΟΓΛΩΣΣΗ

Adamopoulou, E., & Moussiades, L. (2020). An overview of chatbot technology. In IFIP international conference on artificial intelligence applications and innovations (pp. 373-383). Springer, Cham.

Alexander, B., Ashford-Rowe, K., Barajas-Murphy, N., Dobbin, G., Knott, J., McCormack, M., Pomerantz, J., Seilhamer, R. and Weber, N. (2019) EDUCAUSE Horizon Report: 2019 Higher Education Edition. Louisville, CO: EDUCAUSE. Available at: https://www.learntechlib.org/p/208644/ [Accessed Feb. 2026].

Alsharhan, A., Al-Emran, M. and Shaalan, K. (2023) Chatbot adoption: A multiperspective systematic review and future research agenda. IEEE Transactions on Engineering Management.

Beck, K. (2003). Test-Driven Development: By Example. Addison-Wesley.

Bednarz, B., & Miłosz, M. (2025). Benchmarking the performance of Python web frameworks. Journal of Computer Sciences Institute, 36, 336-341. Available at:https://doi.org/10.35784/jcsi.5838

Benchmarking Web Frameworks. (2025). Benchmarking the performance of Python web frameworks. Journal of Computer Sciences Institute, 36, 336-341. Available at: https://www.researchgate.net/publication/396039112_Benchmarking_the_performance_of_Python_web_frameworks (Accessed: February 2026).

Bertini, E., Chaves, P., & Cauvin, M. (2018). Web design and interactivity: Principles and challenges. Springer.

Bohus, D., & Rudnicky, A. I. (2009). The RavenClaw dialog management framework: Architecture and systems. Computer Speech & Language, 23(3), 332–361.

Deng, L., & Liu, Y. (Eds.). (2018). Deep Learning in Natural Language Processing. Springer Singapore. Available at: https://books.google.gr/books?hl=el&lr=&id=y_lcDwAAQBAJ&oi=fnd&pg=PR5&dq=Natural+Language+Processing+-+NLP+book+&ots=ajwqd-NR79&sig=EEcqXjOYpsuGA9UDqk2lvUQbuVM&redir_esc=y#v=onepage&q=Natural%20Language%20Processing%20-%20NLP%20book&f=false

Elmasri, R., & Navathe, S. B. (2016). Fundamentals of Database Systems (7th ed.). Pearson.

Ferrara, E. (2023). Should ChatGPT be biased? Challenges and risks of large language models. First Monday, 28(11). Available at: https://arxiv.org/pdf/2304.03738 (Accessed: February 2026).

Fitzpatrick, C., & Wolf, J. (2017). Enhancing user experience with JavaScript: An in-depth study of web design patterns.

Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley.

Hussain, S., Ameri Sianaki, O., & Ababneh, N. (2019). A survey of conversational agents: Types, architecture, and applications. In Advanced Information Networking and Applications (pp. 489–501). Springer, Cham.

Jordan, M.I. & Mitchell, T.M. (2015) Machine learning: Trends, perspectives, and prospects. Science, 349(6245), pp. 255-260.

Jurafsky, D., & Martin, J. H. (2023). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (3rd ed. draft). Prentice Hall.

King, D.E. (2009) 'Dlib-ml: A Machine Learning Toolkit', Journal of Machine Learning Research, 10, pp. 1755-1758. Available at: https://www.academia.edu/87493970/Dlib_ml_A_Machine_Learning_Toolkit?sm=b&rhid=37959357101

Koubaa, A., Boulila, W., Ghouti, L., Alzahem, A., & Latif, S. (2023). Exploring ChatGPT capabilities and limitations: a survey. IEEE Access.

Laranjo, L., Dunn, A. G., Tong, H. L., Kocaballi, A. B., Chen, J., Bashir, R., Surian, D., Gallego, B., Magrabi, F., Lau, A. Y. S., & Coiera, E. (2018). Conversational agents in healthcare: a systematic review. Journal of the American Medical Informatics Association, 25(9), 1248-1258.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

McTear, M. F. (2004). Spoken Dialogue Technology: Toward the Conversational User Interface. Springer.

Meyers, S. (2021). FastAPI: The complete guide. O'Reilly Media.

Murphy, K.P. (2012) Machine Learning: A Probabilistic Perspective. Cambridge, MA: MIT Press. Available at: https://books.google.gr/books?hl=el&lr=&id=RC43AgAAQBAJ&oi=fnd&pg=PR7&dq=Kevin+Murphy+(2012)&ots=unmvexRp-6&sig=6zkVmdwQLqjp66xXJdHnRYgIyxU&redir_esc=y#v=onepage&q=Kevin%20Murphy%20(2012)&f=false

Nielsen, J. (1994). Usability Engineering. Morgan Kaufmann.

Norman, D. A. (2013). The Design of Everyday Things (Revised and expanded ed.). Basic Books.

Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach. 9th Edition, McGraw-Hill Education.

Russell, S. & Norvig, P. (2021) Artificial Intelligence: A Modern Approach. 4th edn. Hoboken, NJ: Pearson.

Selwyn, N. (2016) Education and Technology: Key Issues and Debates. 2nd edn. London: Bloomsbury Academic.

Sharma, S., & Upadhyaya, S. (2019). Web Technologies: HTML, CSS, JavaScript, and More. ResearchGate.

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the User Interface: Strategies for Effective Human-Computer Interaction (6th ed.). Pearson.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). Database System Concepts (7th ed.). McGraw-Hill Education.

Van Rossum, G. (2020). Python Documentation. Python Software Foundation.

Woolf, B. P. (1991) AI in Education (Technical Report 91-37). Amherst, MA: University of Massachusetts, Department of Computer and Information Science. Available at: [https://web.cs.umass.edu/publication/docs/1991/UM-CS-1991-037.pdf] [Accessed Feb. 2026].

Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., et al. (2024) Harnessing the power of LLMs in practice: A survey on ChatGPT and beyond. ACM Transactions on Knowledge Discovery from Data, 18(6), pp. 1-32.

Young, S., Gašić, M., Thomson, B., & Williams, J. D. (2013). POMDP-based statistical spoken dialog systems: A review. Proceedings of the IEEE, 101(5), 1160–1179.

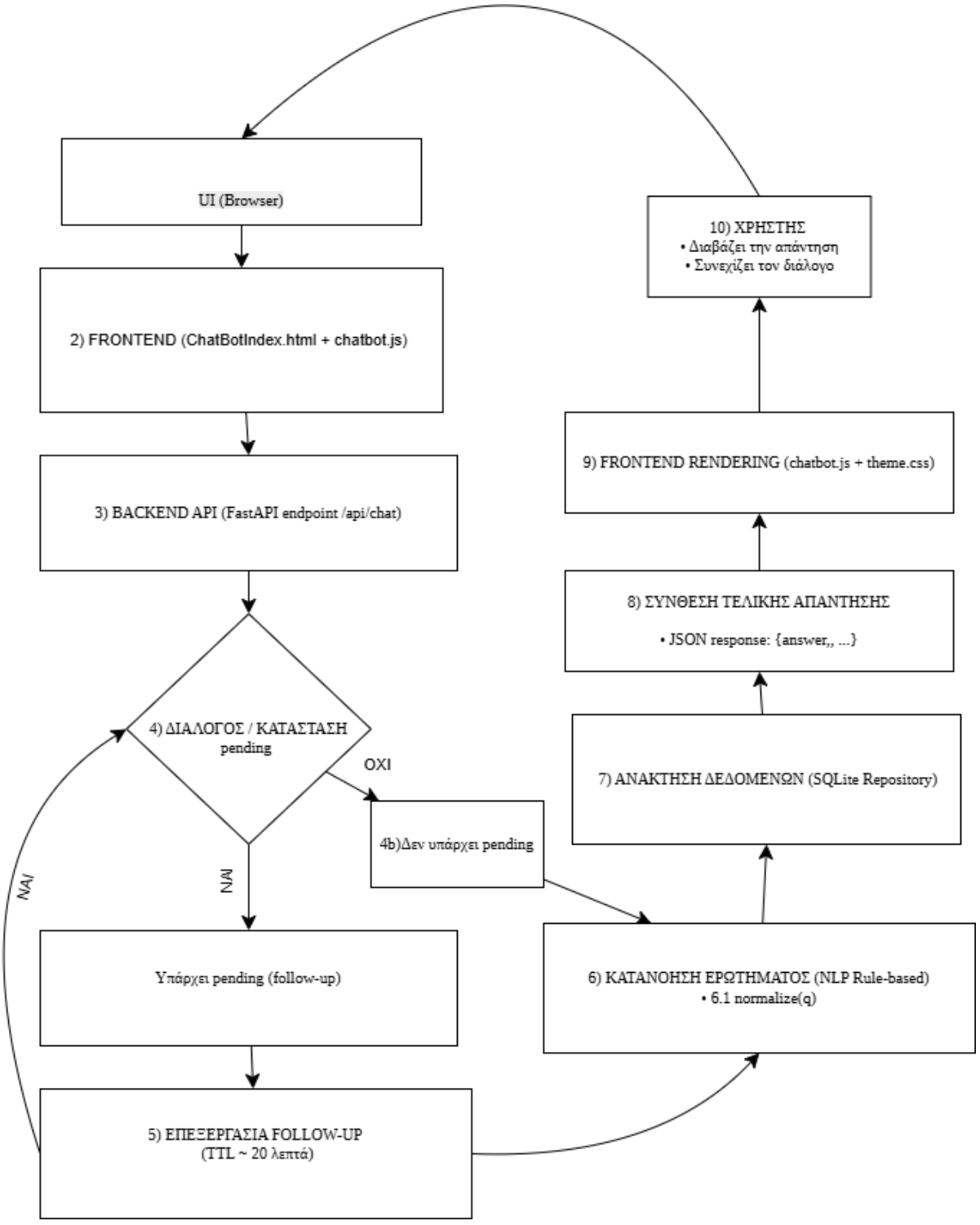
Zawacki-Richter, O., Marín, V.I., Bond, M. and Gouverneur, F. (2019) 'Systematic review of research on artificial intelligence applications in higher education – where are the educators?', International Journal of Educational Technology in Higher Education, 16(1), article 39. Available at: https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-019-0171-0 (Accessed: 14 February 2026).

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., ... & Wen, J. R. (2023). A survey of large language models. arXiv preprint arXiv:2303.18223

Β. ΕΛΛΗΝΙΚΗ

Negnevitsky, M., 2022. Τεχνητή Νοημοσύνη. 3η έκδ. Θεσσαλονίκη: Εκδόσεις Τζιόλα.

ΠΑΡΑΡΤΗΜΑ Α


UI (Browser)
2) FRONTEND (ChatBotIndex.html + chatbot.js)
3) BACKEND API (FastAPI endpoint /api/chat)
4) ΔΙΑΛΟΓΟΣ / ΚΑΤΑΣΤΑΣΗ pending
ΝΑΙ
ΟΧΙ
Υπάρχει pending (follow-up)
5) ΕΠΕΞΕΡΓΑΣΙΑ FOLLOW-UP (TTL ~ 20 λεπτά)
4b)Δεν υπάρχει pending
6) ΚΑΤΑΝΟΗΣΗ ΕΡΩΤΗΜΑΤΟΣ (NLP Rule-based)
• 6.1 normalize(q)
7) ΑΝΑΚΤΗΣΗ ΔΕΔΟΜΕΝΩΝ (SQLite Repository)
8) ΣΥΝΘΕΣΗ ΤΕΛΙΚΗΣ ΑΠΑΝΤΗΣΗΣ
• JSON response: {answer,, ...}
9) FRONTEND RENDERING (chatbot.js + theme.css)
10) ΧΡΗΣΤΗΣ
• Διαβάζει την απάντηση
• Συνεχίζει τον διάλογο