



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ CHAT ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΓΙΑ ΚΑΤΑΤΟΠΙΣΗ ΠΕΛΑΤΩΝ ΣΤΗΝ ΒΕΛΤΙΣΤΗ ΕΞΥΠΗΡΕΤΗΣΗ ΤΟΥΣ

ΚΑΤΣΩΝΗΣ ΓΕΩΡΓΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Κολομβάτσος Κωνσταντίνος
Αναπληρωτής Καθηγητής

Λαμία 27 Σεπτεμβρίου έτος 2026



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

DEVELOPMENT OF AN ARTIFICIAL
INTELLIGENCE CHAT APPLICATION FOR
GUIDING CUSTOMERS TOWARD OPTIMAL
SERVICE

GEORGE KATSONIS

FINAL THESIS

ADVISOR

Kolomvatsos Konstantinos
Associate Professor

Lamia September 27 year 2026

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 27/09/2026

Ο Δηλών.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Η τεχνολογία αποτελεί πλέον σημαντικό μέρος της καθημερινής ζωής, της εργασίας και της εκπαίδευσης. Τα κινητά τηλέφωνα και οι φορητοί υπολογιστές χρησιμοποιούνται καθημερινά, ενώ οι εφαρμογές που έχουν σχεδιαστεί για αυτές τις συσκευές γίνονται όλο και πιο απαιτητικές. Η παρούσα διπλωματική εργασία παρουσιάζει το PitsiBot, ένα conversational υβριδικό μοντέλο που βασίζεται στην τεχνητή νοημοσύνη και βοηθά τους χρήστες να βρίσκουν κατάλληλα προϊόντα, παρέχοντας προτάσεις σύμφωνα με τις προσωπικές τους ανάγκες και τις σύγχρονες τάσεις.

Η εφαρμογή αναπτύχθηκε με τη χρήση των HTML, CSS και JavaScript. Διαθέτει UI που θυμίζει μέσα κοινωνικής δικτύωσης, ώστε να είναι πιο οικεία και εύκολη στη χρήση. Η εφαρμογή τρέχει τοπικά με χρήση Node.js και χρησιμοποιεί δύο υπηρεσίες API, οι οποίες καλούνται από το πρόγραμμα με τη χρήση του Express. Η πρώτη API υπηρεσία, το Groq, διαχειρίζεται τη συνομιλία με τον χρήστη και συλλέγει τα δεδομένα. Στην συνέχεια καλείται το Serper API για να κάνει την διαδικτυακή αναζήτηση επιλογών, τα οποία και επιστρέφει σαν δεδομένα. Τέλος, για τη δημιουργία των προτάσεων, η πρώτη υπηρεσία εξετάζει δεδομένα από τοπική προσωρινή μνήμη cache και καταλόγους σε μορφή JSON, καθώς και τα αποτελέσματα αναζήτησης που επιστρέφει το Serper, έχοντας πάντα σαν οδηγό τα δεδομένα που έχει αντλήσει από τον χρήστη. Στη συνέχεια, η ίδια υπηρεσία καλείται να επιλέξει από τα δεδομένα που έχει στην διάθεσή της, τις τρεις καλύτερες προτάσεις με βάση την διαθεσιμότητα και την επικαιρότητα προϊόντων στην ελληνική αγορά.

Κατά τα πρώτα στάδια της ανάπτυξης δοκιμάστηκε η εφαρμογή σε πέντε κατηγορίες προϊόντων, όμως αργότερα περιορίστηκε σε δύο (laptop και κινητά τηλέφωνα) με μεγαλύτερη έμφαση στο prompt engineering για να παρέχει πιο αξιόπιστα αποτελέσματα. Σίγουρα η επέκταση σε περισσότερες κατηγορίες προϊόντων, χωρίς να μειωθεί η ακρίβεια των αποτελεσμάτων, θα μπορούσε να αποτελεί μελλοντική βελτίωση.

ABSTRACT

Technology is now an important part of everyday life, work, and education. Mobile phones and laptops are used every day, while the applications designed for these devices are becoming more demanding. This thesis presents PitsiBot, a conversational hybrid model based on AI. It helps users find suitable products by providing recommendations based on their personal needs and current trends.

The application was developed using HTML, CSS, and JavaScript. It has a user interface similar to social media, making it familiar and easy to use. The application runs locally using Node.js and uses two APIs, which are called by the program through Express. The first API service, Groq, manages the conversation with the user and collects the required information. The Serper API is then used to search the internet for possible product options and return the results as data. Finally, to create the recommendations, the first service examines data from a local cache and JSON catalogues, as well as the search results returned by Serper. Throughout this process, it uses the information collected from the user as a guide. The service then selects the three best recommendations from the available data, based on product availability and how up to date the products are in the Greek market.

During the early stages of development, the application was tested with five product categories. However, it was later limited to two categories with a stronger focus on prompt engineering in order to provide more reliable results. Expanding the application to include more product categories,

without reducing the accuracy of the results, could be a possible improvement in the future.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	I
ABSTRACT	III
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ.....	2
1.1 ΑΝΤΙΚΕΙΜΕΝΟ ΚΑΙ ΣΗΜΑΣΙΑ	2
1.2 ΣΤΟΧΟΙ ΕΡΓΑΣΙΑΣ	3
ΚΕΦΑΛΑΙΟ 2 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ.....	5
2.1 ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΚΑΙ ΜΕΓΑΛΑ ΓΛΩΣΣΙΚΑ ΜΟΝΤΕΛΑ	5
2.2 PROMPT ENGINEERING	6
2.3 ΣΥΣΤΗΜΑΤΑ ΣΥΣΤΑΣΗΣ ΚΑΙ CONVERSATIONAL RECOMMENDERS	7
ΚΕΦΑΛΑΙΟ 3 ΕΡΓΑΛΕΙΑ.....	9
3.1 ΓΛΩΣΣΕΣ ΑΝΑΠΤΥΞΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	9
3.1.A HTML.....	9
3.1.B CSS.....	10
3.1.Γ JAVASCRIPT	10
3.2 NODE.JS & EXPRESS	11
3.2.A NODE.JS.....	11
3.2.B EXPRESS.....	12
3.3 API	13
3.3.A GROQ.....	14
3.3.B SERPER	14
3.4 JSON CATALOG, CACHE.....	15
3.4.A ΚΑΤΑΛΟΓΟΣ JSON.....	15
3.4.B ΜΝΗΜΗ CACHE	16
ΚΕΦΑΛΑΙΟ 4 ΠΕΙΡΑΜΑΤΑ	17
4.1 OFFLINE TESTS	17
4.1.A ΤΕΣΤ 1. ΚΙΝΗΤΑ ΓΙΑ ΚΑΘΗΜΕΡΙΝΗ ΧΡΗΣΗ ΜΕΧΡΙ 300 ΕΥΡΩ.....	19
4.1.B ΤΕΣΤ 2. ΛΑΠΤΟΡ ΓΙΑ ΚΑΘΗΜΕΡΙΝΗ ΧΡΗΣΗ ΜΕΧΡΙ 600 ΕΥΡΩ.....	21
4.1.Γ ΤΕΣΤ 3. ΣΥΓΚΕΚΡΙΜΕΝΟ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ ΛΑΠΤΟΡ (ΚΑΡΤΑ ΓΡΑΦΙΚΩΝ 5060)	23
4.1.Δ ΤΕΣΤ 4. ΣΥΓΚΕΚΡΙΜΕΝΟ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ ΚΙΝΗΤΟ (ΑΠΟΘΗΚΕΥΤΙΚΟΣ ΧΩΡΟΣ 512GB)	25
4.1.Ε ΤΕΣΤ 5. ΣΕΝΑΡΙΟ (ΓΟΝΙΟΣ ΓΙΑ ΠΑΙΔΙ)	27
4.2 LIVE ΠΕΙΡΑΜΑΤΑ.....	28
4.2.A ΤΕΣΤ 1. ΚΙΝΗΤΑ ΓΙΑ ΚΑΘΗΜΕΡΙΝΗ ΧΡΗΣΗ ΜΕΧΡΙ 300 ΕΥΡΩ.....	29
4.2.B ΤΕΣΤ 2. ΛΑΠΤΟΡ ΓΙΑ ΚΑΘΗΜΕΡΙΝΗ ΧΡΗΣΗ ΜΕΧΡΙ 600 ΕΥΡΩ.....	31
4.2.Γ ΤΕΣΤ 3. ΣΥΓΚΕΚΡΙΜΕΝΟ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ ΛΑΠΤΟΡ (ΚΑΡΤΑ ΓΡΑΦΙΚΩΝ 5060)	33
4.2.Δ ΤΕΣΤ 4. ΣΥΓΚΕΚΡΙΜΕΝΟ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ ΚΙΝΗΤΟ (ΑΠΟΘΗΚΕΥΤΙΚΟΣ ΧΩΡΟΣ 512GB)	35
4.2.Ε ΤΕΣΤ 5. ΣΕΝΑΡΙΟ (ΓΟΝΙΟΣ ΓΙΑ ΠΑΙΔΙ)	37
4.3 ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΓΚΡΙΣΕΙΣ.....	38
4.3.A ΤΕΣΤ 1 ΑΠΟΤΕΛΕΣΜΑΤΑ.....	38

4.3.β ΤΕΣΤ 2 ΑΠΟΤΕΛΕΣΜΑΤΑ	39
4.3.γ ΤΕΣΤ 3 ΑΠΟΤΕΛΕΣΜΑΤΑ	39
4.3.δ ΤΕΣΤ 4 ΑΠΟΤΕΛΕΣΜΑΤΑ	39
4.3.ε ΤΕΣΤ 5 ΑΠΟΤΕΛΕΣΜΑΤΑ	39
4.3.στ ΣΥΝΟΛΙΚΗ ΑΠΟΤΙΜΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ	40
<u>ΚΕΦΑΛΑΙΟ 5 ΣΥΜΠΕΡΑΣΜΑΤΑ</u>	<u>40</u>
5.1 ΔΥΣΚΟΛΙΕΣ	40
5.2 ΣΥΜΠΕΡΑΣΜΑΤΑ	42
5.3 ΜΕΛΛΟΝΤΙΚΕΣ ΒΕΛΤΙΩΣΕΙΣ	44
<u>ΒΙΒΛΙΟΓΡΑΦΙΑ</u>	<u>45</u>

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

1.1 Αντικείμενο και Σημασία

Καθημερινά πλέον καλείται κανείς να επιλέξει ορισμένα τεχνολογικά εργαλεία είτε, για την σχολή ή το σχολείο, είτε για κάποιο χόμπι, είτε για την εργασία ή ακόμα και για κάποια μικρή ή μεγάλη επιχείρηση που μπορεί να διαχειρίζεται. Τα εργαλεία αυτά, το κόστος και η ποιότητα τους θα παίξουν καθοριστικό ρόλο στην πορεία κάθε τομέα. Στην αγορά υπάρχουν ποικίλα προϊόντα και οι διαφημιστικές στρατηγικές καταφέρνουν πολλές φορές να επηρεάσουν τις επιλογές που θα κάνει κάποιος. Επιπροσθέτως η δυσκολία μετάφρασης των τεχνολογικών προδιαγραφών σε δυνατότητες των εργαλείων, σε συνδυασμό με την ευκολία που έχουν προσφέρει οι ηλεκτρονικές αγορές, πολλές φορές οδηγούν σε μη αναστρέψιμες λάθος επιλογές. Πολλές ήδη υπάρχουσες εφαρμογές μπορούν να προσφέρουν συμβουλές για την επιλογή κάποιου εργαλείου, αλλά πολλές φορές λόγω ανεπαρκών πληροφοριών ή λάθος διατύπωσης ή επεξήγησης οδηγείται κανείς πάλι σε λανθασμένη απόφαση.

Η παρούσα εργασία με έναν συνδυασμό ήδη υπαρχόντων τεχνολογιών λαμβάνει θέση πάνω στο θέμα, με σκοπό τις όσο το δυνατόν καλύτερες και πιο εύστοχες προτάσεις για τον χρήστη. Η σημασία μιας σωστής επιλογής σημαίνει καλύτερη ποιότητα δουλείας σε λιγότερο χρόνο συνήθως. Μπορεί λοιπόν να συμπεράνει κανείς ότι αυτό σημαίνει λιγότερο κόστος για τον χρήστη είτε αυτό είναι ενέργεια που αλλιώς θα έπρεπε να καταβάλλει είτε κυριολεκτικά χρηματικά ποσά σε κάποιες περιπτώσεις.

Η ανάπτυξη στην τεχνολογία υποστήριξης και παροχής πληροφοριών μέσω μοντέλων τεχνίτης νοημοσύνης έχει ήδη συνεισφέρει αρκετά σε αυτό το

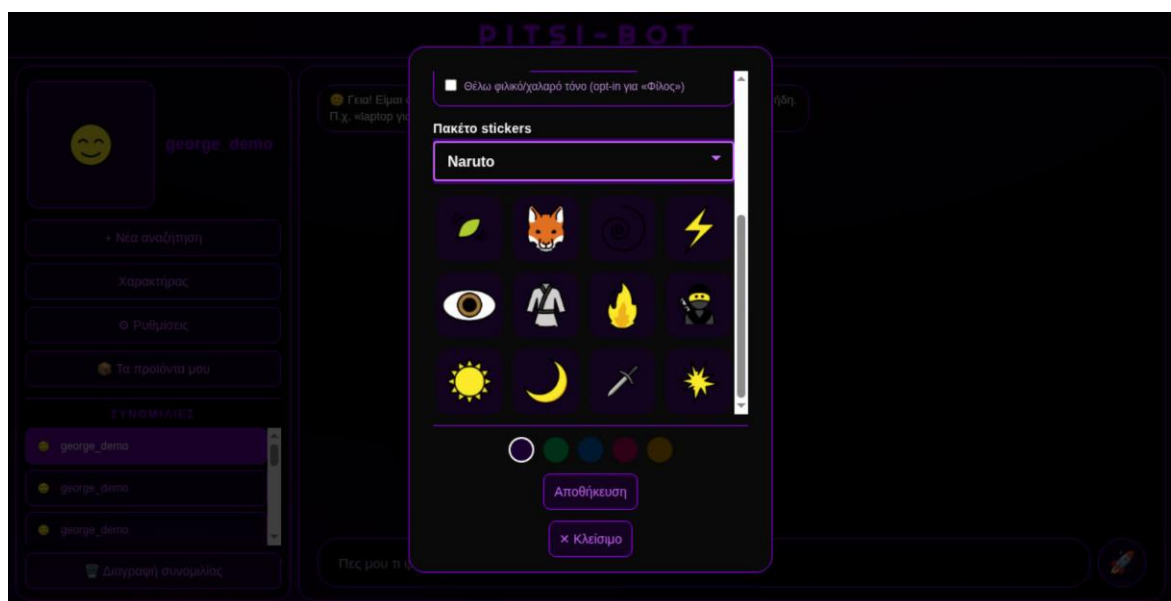
πρόβλημα. Είναι γεγονός, επίσης, πως οι περισσότεροι έχουν εξοικειωθεί με την εικόνα και την λογική των social media, πράγμα που στην παρούσα εργασία έχει χρησιμοποιηθεί για την καλύτερη εμπειρία του χρήστη. Για να επιτευχθεί όμως η λογική των social media, χρειάστηκε ένα LLM για τις συζητήσεις με τον χρήστη και την επιλογή προτάσεων και ένα για την αναζήτηση στο διαδίκτυο.

1.2 Στόχοι Εργασίας

Κύριος στόχος της εφαρμογής PitisBot είναι η δημιουργία ενός περιβάλλοντος με εύκολη πρόσβαση για όλους, με ενσωματωμένο σκεπτικό έρευνας για ορθολογική επιλογή τεχνολογικών εργαλείων ανάλογα με τις ανάγκες της καθημερινότητας κάποιου. Συνολικά όμως οι στόχοι της εργασίας χωρίζονται σε δυο κατηγορίες. Η μια σχετίζεται με την κατανόηση πληροφοριών από την επικοινωνία με τον χρήστη και τη σωστή επιλογή προτάσεων μέσα από τα αποτελέσματα που θα συναντάει. Ενώ η άλλη είναι η πιο φιλική προσέγγιση του χρήστη και ο ρόλος που μπορεί να έχει η εφαρμογή στην ζωή του.

Το PitsiBot καταφέρνει να πλησιάζει ολοένα και περισσότερο τον σκοπό του για εύστοχες προτάσεις και κατανόηση πληροφοριών, μέσα από την λογική για κατανόηση που έχει περαστεί με prompt engineering, από τους καταλόγους που έχουν φτιαχτεί χειροκίνητα για καλύτερη καθοδήγηση και από τον συνδυασμό των υπηρεσιών που καλούνται με API keys. Καταφέρνει να αντιδράσει ανάλογα με τις απαιτήσεις του χρήστη και του προτείνει εργαλεία που εύστοχα ταιριάζουν στην περιγραφή ακόμα και με λίγες πληροφορίες, πράγμα που αποτέλεσε κύριο στόχο καθ' όλη την διάρκεια του σχεδιασμού.

Επίσης ένας στόχος είναι να καταπολεμήσει το πρόβλημα τις απρόσωπης συνομιλίας. Παρατηρώντας κανείς τον σχεδιασμό της εφαρμογής, θα παρατηρήσει ότι ο χρήστης έχει την δυνατότητα να αλλάξει τον χαρακτήρα στον οποίο αναφέρεται για επιλογές, τόσο το όνομα και το ύφος του, όσο και το πρόσωπο του από μια μεγάλη γκάμα επιλογών. Αυτό έχει γίνει έτσι ώστε να βοηθήσει τον χρήστη να κάνει ακόμα πιο μοναδική και ιδιαίτερη την εμπειρία του. Ο χαρακτήρας αυτός τόσο μεταφορικά, όσο και λειτουργικά μέσα από prompt engineering παίρνει τον ρόλο του βοηθού τσέπης, ενός μικρού προσωπικού σύμβουλου, που είναι εκεί για τον χρήστη για να του παρέχει καθοδήγηση μέσα από την πληθώρα επιλογών που συναντά σε κάθε του έρευνα.



[Εικόνα 1] Επίδειξη σχεδιασμού χαρακτήρα.

ΚΕΦΑΛΑΙΟ 2 Βιβλιογραφική Επισκόπηση

2.1 Τεχνητή νοημοσύνη και μεγάλα γλωσσικά μοντέλα

Το παρόν πρόγραμμα κάνει χρήση μεγάλων γλωσσικών μοντέλων. Για να κατανοήσει κανείς τι είναι και που χρησιμεύουν πρέπει να ξέρει λίγα πράγματα για τα NLP ή αλλιώς Επεξεργασία Φυσικής Γλώσσας. Τα NLP είναι ένας ευρύς κλάδος της τεχνητής νοημοσύνης, ο οποίος επικεντρώνεται στην κατανόηση, την ερμηνεία και την ίδια την παραγωγή λέξεων και ανθρώπινης γλώσσας, από τους ηλεκτρονικούς υπολογιστές. Σκοπός της τεχνικής αυτής, ή καλύτερα του συνόλου των τεχνικών που χρησιμοποιούνται στα NLP πρότυπα, είναι να δοθεί στους υπολογιστές η δυνατότητα επεξεργασίας των λέξεων, είτε γραπτά είτε φωνητικά. Σημαντικό επίσης είναι να αναφερθεί, ότι όταν μιλάμε για κατανόηση και επεξεργασία δεν αναφερόμαστε μόνο στην σημασία της λέξης, αλλά στην ικανότητα του υπολογιστή να μπορέσει να εμπεδώσει και την πρόθεση και την διάθεση, στα Αγγλικά : intent και sentiment, πίσω από αυτή. Το NLP είναι ένας συνδυασμός από machine και deep learning πρότυπα και τεχνικές καθώς και από στατιστικά μοντέλα. Ο συνδυασμός αυτός έχει επιτρέψει στο NLP να γίνει βασικός παράγοντας στην εξέλιξη των μεταφραστικών προγραμμάτων, των GPS, των φωνητικών βοηθών και αναμφισβήτητα των chatbots.

Σημαντική εξέλιξη στην επεξεργασία φυσικής γλώσσας αποτέλεσε η αρχιτεκτονική Transformer, η οποία βασίζεται στον μηχανισμό attention και αποτέλεσε τη βάση για την ανάπτυξη πολλών σύγχρονων γλωσσικών μοντέλων (Vaswani et al., 2017).

Όσο για τα LLMs ή αλλιώς Μεγάλα Γλωσσικά Μοντέλα είναι ένα ισχυρό υποσύνολο μοντέλων NLP. Αποτελούν ουσιαστικά έναν τύπο αλγορίθμου τεχνητής νοημοσύνης, που χρησιμοποιεί διάφορες τεχνικές Deep Learning και τεράστιο μέγεθος δεδομένων. Τα LLMs χαρακτηρίζονται από τα εκτεταμένα δεδομένα εκπαίδευσης και την ικανότητα των υπολογιστών να εκτελούν ένα

ευρύ φάσμα γλωσσικών εργασιών με ελάχιστη εκπαίδευση (Brown et al., 2020). Έχουν έναν γιγαντιαίο αριθμό παραμέτρων που κυμαίνεται σε εκατομμύρια με δισεκατομμύρια. Κύριος σκοπός τους είναι να πραγματοποιούν πληθώρα εργασιών NLP, όπως η δημιουργία και η μετάφραση κειμένου καθώς και η ανάλυση συναισθήματος της γλώσσας. Μοντέλα όπως το Llama, το GPT και το Claude είναι παραδείγματα LLMs που έφεραν επανάσταση στις δυνατότητες των NLPs. Το πρόγραμμα χρησιμοποιεί έτοιμο μεγάλο γλωσσικό μοντέλο μέσω του Groq μέσα σε αυστηρά όρια, άρα τα LLMs που χρησιμοποιεί το πρόγραμμα είναι υπεύθυνα για τον διάλογο την κρίση και την επιλογή δεδομένων που έρχονται εξωτερικά.

2.2 Prompt Engineering

Το Prompt Engineering, στα Ελληνικά η Μηχανική Εντολών, είναι μια τεχνική τελειοποίησης των γλωσσικών μοντέλων της τεχνητής νοημοσύνης. Είναι η διαδικασία σχεδιασμού, διατύπωσης και βελτιστοποίησης των κειμένων, δηλαδή των prompts σε ένα μοντέλο AI, ώστε να αποσπαστούν οι επιθυμητές απαντήσεις και να ευθυγραμμιστούν με το ύφος της εκάστοτε συνομιλίας (Liu et al., 2023). Η τέχνη της δημιουργίας ενός σωστού και αποτελεσματικού prompt είναι πέρα από την απλή διατύπωση μιας ερώτησης. Συνδυάζει την παροχή σωστής ποσότητας πλαισίου, τον καθορισμό του τόνου και του ύφους, καθώς και την ιδανική δομή του επιθυμητού αποτελέσματος. Τα καλά δομημένα prompts καθοδηγούν ένα AI ή chatbot στο να παράγει ακριβέστερες και ιδανικές απαντήσεις στον χρήστη, κάνοντας έτσι την εμπειρία του καλύτερη. Κάποιες τεχνικές του prompt engineering λειτουργούν σε όλα τα μοντέλα, όπως το message role. Ωστόσο κάποιοι τύποι μοντέλων μπορεί να χρειάζονται διαφορετικές προσεγγίσεις στις εντολές τους

για την βέλτιστη παραγωγή αποτελεσμάτων, όπως τα συλλογιστικά μοντέλα. Η αποτελεσματικότητα των τεχνικών prompting μπορεί να διαφοροποιείται ανάλογα με το μοντέλο, την εργασία και τον τρόπο με τον οποίο διαμορφώνεται η είσοδος προς το γλωσσικό μοντέλο (Liu et al., 2023). Το PitsiBot χρησιμοποιεί αυτήν την τεχνική για να δώσει οδηγίες για την σύσταση προϊόντων, την ροή της συνομιλίας αλλά και την διαχείριση συγκεκριμένων περιπτώσεων. Ακολουθεί παράδειγμα prompt engineering για το μοντέλο επιλογής απαντήσεων και συνομιλίας.

```
877     const cap = budgetCap(budget, category);
878     const budgetRule = budget
879       ? `USER BUDGET: ${budget.from}-${budget.to} EUR${cap ? ` (host slack: allow up to
880       o ${cap} EUR)` : ''}. Propose ONLY products inside this range. Prefer inside [from, to
881       l; small overshoot only up to the slack cap if the host allows it. If nothing realisti
882       c fits, pick the closest IN-RANGE options you have and set budget_warning in Greek. NE
883       VER claim "δεν υπάρχουν..." while still returning out-of-range picks.`
884       : `NO USER BUDGET. Specs-first. Spread picks across price levels so the final 3
885       can be ordered cheap-expensive with ≥100 EUR gap between consecutive proposals. Larger
886       gaps (~300 / ~400) are ideal when the pool allows. If the pool cannot support ≥100 EU
887       R gaps, still order cheap-expensive and explain briefly in budget_warning – do not inv
888       ent prices or weak substitutes.`;
889
890     const data = await askJSON([
891       {
892         role: 'system',
893         content: `You rank products for a Greek electronics shopping assistant (Pits
894         iBot). You pick ONLY from a given list of refs – you never invent product names, price
895         s, or specs. Output JSON only. User-facing text (review, budget_warning, extras) MUST
896         be Greek. Scope: smartphones (kinito) and laptops only. Tablet is OUT of scope. Person
897         a/ tone: keep reviews concrete and helpful; UI may apply persona separately – do not ov
898         er-style.`
899       },
900       {
901         role: 'user',
902         content: `Target product: ${cat.aiName}.
903         User's own words: ${needs || 'not stated'}
904         Use-case profile: ${answers && answers.usage} || (soft && soft.usage) || 'custom' //
905         gaming | pro | mainstream | daily | kids | custom
906         Requirements (user explicitly asked): ${statedTxt}
907         ${softTxt ? `Soft preferences (inferred – nice to have, never a reject reason): ${softTx
908         t}` : ''}
909         ${budgetRule}
910
911         AVAILABLE PRODUCTS – the ONLY set you may choose from:
912         ${uniLines(entries)}
913
914         Tags:
915         [catalog+shop] = verified catalogue AND on sale now (most reliable).
916         [catalog] = known product, not in today's shop listing.
917         [shop only] = seen in shop listing, not yet in catalogue.
918
919         === USE-CASE SPEC BARS (requirements, not suggestions) ===
920         ${barsFor(category)}
921
922         === BUDGET, PRICE, GEN, STOCK (HARD) ===
923         - FINAL output size downstream = 3 (FINAL_COUNT). You may propose up to ${cfg.MAX_CANDID
924         ATEs}.
925         - If NO user budget: spread cheap → mid → stronger so consecutive final prices differ by
926         ≥100 EUR when possible (larger gaps OK).
927         - If user budget {from-to}: ONLY propose inside that range (host may allow small slack t
928         o to+slack).
929         - NEVER propose outside the range just to fill a slot.
930         - ALWAYS use live/list prices only. Prefer verified prices. Never invent a price.`
```

[Εικόνα 2] Παράδειγμα prompt engineering.

2.3 Συστήματα σύστασης και Conversational Recommenders

Τα Recommender Systems, ή αλλιώς Συστήματα Σύστασης, είναι προγράμματα και αλγόριθμοι λογισμικού που προτείνουν προϊόντα, υπηρεσίες ή περιεχόμενα στους χρήστες με βάση τις προτιμήσεις τους (Adomavicius & Tuzhilin, 2005). Στόχος τους είναι να αναλύουν

και να φιλτράρουν τεράστιο όγκο δεδομένων από το διαδίκτυο και να παρουσιάσουν στον χρήστη μια λίστα επιλογών οι οποίες είναι προσαρμοσμένες στις προσωπικές του προτιμήσεις, βάση των παρελθοντικών του προτιμήσεων. Τα συστήματα αυτά βασίζονται σε διάφορες μεθοδολογίες για να φτάσουν στα επιθυμητά αποτελέσματα, όπως το Content Based Filtering, όπου προτείνονται στον χρήστη επιλογές που είχε προτιμήσει στο παρελθόν, το Collaborative Filtering, όπου βασίζεται και στην συμπεριφορά άλλων χρηστών με παρεμφερείς προτιμήσεις, καθώς και υβριδικά συστήματα που συνδυάζουν και τα δύο παραπάνω. Σε γενικότερο επίπεδο, τα υβριδικά συστήματα σύστασης μπορούν να συνδυάζουν διαφορετικές τεχνικές σύστασης, με σκοπό την αξιοποίηση των πλεονεκτημάτων τους και τον περιορισμό των αδυναμιών κάθε επιμέρους προσέγγισης (Burke, 2002). Ενώ τα Conversational Recommender Systems, ή αλλιώς Συστήματα Συστάσεων μέσω Διαλόγου, είναι εξελιγμένα διαδραστικά συστήματα τεχνητής νοημοσύνης και συστημάτων σύστασης. Βασικό τους χαρακτηριστικό είναι η άμεση αλληλεπίδραση με τον χρήστη μέσω διαλόγου, φυσικής γλώσσας (Gao et al., 2021). Με αυτόν τον τρόπο ο χρήστης ανατροφοδοτεί το σύστημα με δεδομένα πέραν των παρελθοντικών του προτιμήσεων ή των προτιμήσεων άλλων χρηστών. Η αλληλεπίδραση αυτή επιτρέπει στο σύστημα να αξιοποιεί τις πληροφορίες και την ανατροφοδότηση που παρέχει ο χρήστης κατά τη διάρκεια του διαλόγου, προσαρμόζοντας ανάλογα τη διαδικασία σύστασης (Liu et al., 2020). Σε πραγματικό χρόνο το σύστημα προτείνει εξειδικευμένες προτάσεις με βάση τις ανάγκες του χρήστη που λαμβάνει ως δεδομένα μέσω διαλόγου. Τα Συστήματα Συστάσεων μέσω Διαλόγου κατανοούν και θυμούνται τις συνομιλίες με τον χρήστη για να βρουν το κατάλληλο πλαίσιο αναζήτησης και ύστερα αναζητούν και φιλτράρουν τις επιλογές για να καταλήξουν σε προτάσεις όσο το δυνατόν πιο κοντά στο βέλτιστο αποτέλεσμα για τον χρήστη. Το ενδιαφέρον για τα CRSs έχει αυξηθεί σημαντικά τα τελευταία χρόνια. Η εξέλιξη τους βασίζεται στην εκτεταμένη πρόοδο στην επεξεργασία φυσικής γλώσσας από τους

υπολογιστές και στην αυξημένη χρήση των chatbots. Βλέποντας κανείς την παρούσα εφαρμογή θα διαπιστώσει ότι ανήκει στην δεύτερη κατηγορία, καθώς αλληλοεπιδρά σημαντικά με το ελεύθερο κείμενο και τις διευκρινήσεις που παίρνει από αυτό.

ΚΕΦΑΛΑΙΟ 3 Εργαλεία

3.1 Γλώσσες ανάπτυξης της εφαρμογής

3.1.α HTML

Η HTML ή αλλιώς Hyper Text Markup Language είναι μια γλώσσα σήμανσης που περιγράφει τη σημασιολογική δομή ενός εγγράφου στον φυλλομετρητή μέσω στοιχείων και ιδιοτήτων, όχι αλγόριθμο υπολογισμού. Η HTML είναι το βασικότερο δομικό εργαλείο του διαδικτύου που καθορίζει την δομή και την πορεία του περιεχομένου. Κάθε ξεχωριστό στοιχείο στην συγκεκριμένη γλώσσα καθορίζεται με ετικέτες. Όπως φαίνεται στο παρακάτω παράδειγμα [Εικόνα 3], με την ίδια λογική που οροθετούνται τα κουμπιά σε αυτήν την περίπτωση, δομείτε και κάθε στοιχείο που βλέπει και μπορεί να αλληλοεπιδράσει ο χρήστης. Η τελική μορφή της εφαρμογής και οι λειτουργίες που θα εκτελεί δεν καθορίζονται στην συγκεκριμένο κώδικα. Η HTML είναι υπεύθυνη για την σωστή σύνδεση και επικοινωνία μεταξύ των υπολοίπων αρχείων κώδικα. Ακολουθεί παράδειγμα με την οριοθέτηση και την εδραίωση των κουμπιών στο side menu στην οθόνη της εφαρμογής.

```
33     <span id="bot-name">PitsiBot</span>
34 </div>
35 <button class="btn-purple" id="btn-new-chat">+ Νέα αναζήτηση</button>
36 <button class="btn-purple" id="btn-character">Χαρακτήρας</button>
37 <button class="btn-dark" id="btn-settings">⚙ Ρυθμίσεις</button>
38 <button class="btn-dark" id="btn-products">📦 Τα προϊόντα μου</button>
39 <h2 id="chats-label">ΣΥΝΟΜΙΛΙΕΣ</h2>
40 <div id="chat-history" role="list" aria-labelledby="chats-label"></div>
41 <button id="btn-delete-mode" class="btn-delete">🗑 Διαγραφή συνομιλίας</button>
42 </nav>
```

[Εικόνα 3] Παράδειγμα εισαγωγής menu με HTML.

3.1.β CSS

Η CSS ή στα ελληνικά Cascading Style Sheets είναι μια γλώσσα μορφοποίησης και γενικότερα είναι υπεύθυνη για την εμφάνιση όσων εισάγονται από την HTML. Δεν περιγράφει λογική, αλλά εφαρμόζει κανόνες με σειρά προτεραιότητας. Σε συνδυασμό με την HTML αποδίδουν το τελικό αποτέλεσμα που φτάνει μπροστά στον χρήστη, όπως τα χρώματα στο light και dark theme και πολλά άλλα. Σε εφαρμογή όπως το PitsiBot, το θεωρητικό όφελος είναι το σταθερό οπτικό λεξιλόγιο (κάρτες, κουμπιά, αντίθεση) και δυνατότητα αλλαγής εμφάνισης χωρίς επέμβαση στη λογική σύστασης. Ακολουθεί παράδειγμα μορφοποίησης όσων εισάγονται από την HTML στο παραπάνω παράδειγμα.

```
157 .btn-purple, .btn-dark, .btn-delete, .btn-delete-selected {
158     background: var(--btn-bg);
159     color: var(--accent);
160     border: var(--border-w) solid var(--border);
161     border-radius: 8px;
162     padding: 11px;
163     cursor: pointer;
164     font-size: 14px;
165     text-align: center;
166     font-family: inherit;
167     transition: background 0.15s, color 0.15s;
168 }
169 .btn-purple:hover, .btn-dark:hover,
170 .btn-delete:hover, .btn-delete-selected:hover {
171     background: var(--btn-hover);
172     color: var(--text);
173 }
174 .btn-delete { margin-top: auto; }
```

[Εικόνα 4] Παράδειγμα μορφοποίησης κουμπιού με CSS.

3.1.γ JAVASCRIPT

Η JavaScript είναι δυναμική, πολυ-παραδειγματική γλώσσα. Εκτελείται από μηχανή JS και έχει άμεση πρόσβαση στα APIs. Είναι υπεύθυνη για τον χειρισμό εισόδου, την επικοινωνία με διακομιστή, ενημέρωση κατάστασης

διεπαφής. Χειρίζεται γεγονότα χρήστη, επικοινωνεί με τον server καλεί τα APIs και γενικότερα χειρίζεται όλο το λειτουργικό κομμάτι όσων εισάγονται από την HTML. Στην παρούσα εργασία ο ρόλος της είναι να καλεί τα endpoints, να διαχειρίζεται την σύνδεση από τον λογαριασμό και την εγγραφή, να χτίζει δυναμικά τις κάρτες προτάσεων και τον πίνακα σύγκρισης και να στέλνει τα μηνύματα στον χρήστη, αλλά και πολλά άλλα. Σε αρχιτεκτονική πελάτη με σύμβουλο, ο ρόλος της JS είναι να μεταφέρει αίτημα ή απάντηση και να αποδίδει αποτελέσματα. Έτσι διαχωρίζεται σαφώς η UI λογική, από την λογική business του Node.js και του Express και των υπηρεσιών Τεχνητής Νοημοσύνης. Ακολουθεί παράδειγμα για την λειτουργία του μενού για σχεδιασμό χαρακτήρα όταν το σωστό κουμπί επιλεγθεί.

```
1916 // Listener για το κουμπί #btn-character
1917 $('btn-character').addEventListener('click', openCharacterModal);
1918
1919 // Η συνάρτηση που ανοίγει το modal
1920 function openCharacterModal() {
1921     pickedEmoji = profile?.emoji || '👤';
1922     pickedColor = profile?.color || '#1a0030';
1923     mergePersonaFromLocal(profile || {});
1924     pickedPersona = PERSONA_OK.has(profile?.persona) ? profile.persona : 'oudeteros';
1925     pickedFriendOptIn = !profile?.friendOptIn;
1926     if (pickedPersona === 'filos' && !pickedFriendOptIn) pickedPersona = 'oudeteros';
1927
1928     $('character-name').value = profile?.name || '';
1929     setEmojiContent($('character-preview'), pickedEmoji, { imgClass: 'emoji-img emoji-img-preview', alt: pickedEmoji });
1930     $('character-preview').style.background = pickedColor;
1931
1932     const cb = $('friend-optin');
1933     if (cb) cb.checked = pickedFriendOptIn;
1934     const radio = $('persona-${pickedPersona}') || $('persona-oudeteros');
1935     if (radio) radio.checked = true;
1936     syncFriendOptInUi();
}
```

[Εικόνα 5] Παράδειγμα λειτουργίας κουμπιού με JAVASCRIPT.

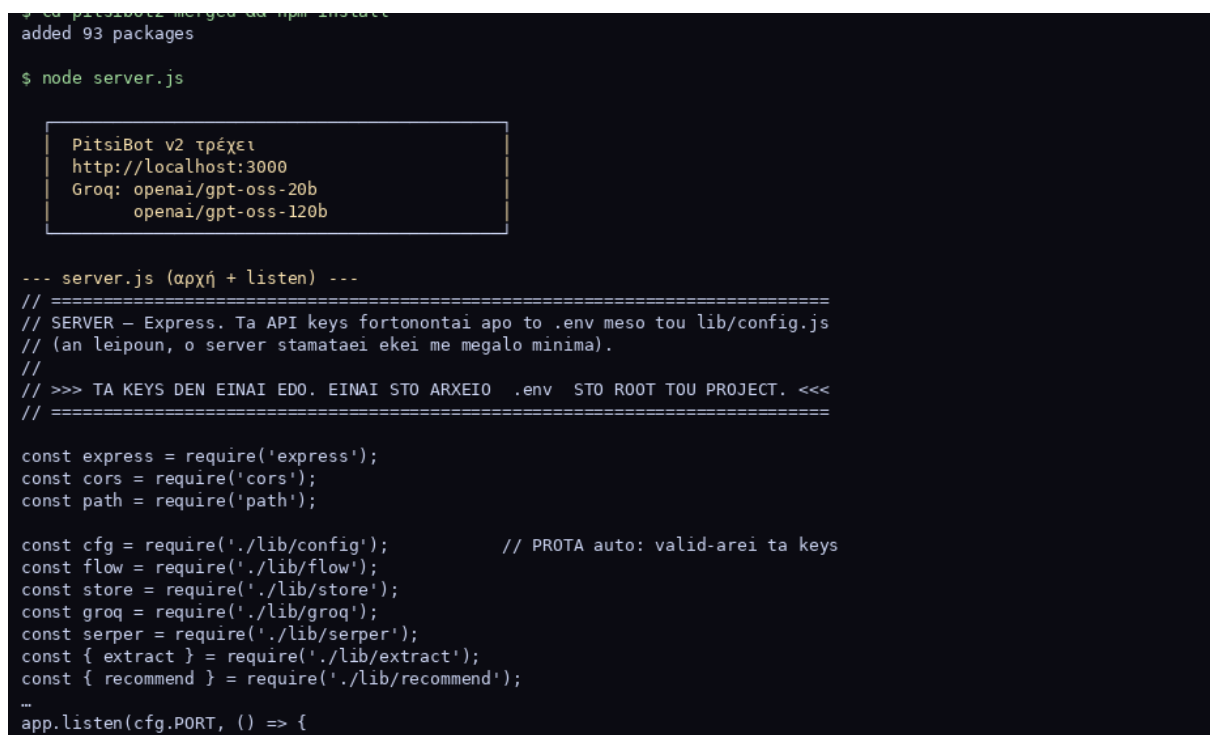
3.2 Node.JS & Express

3.2.α Node.JS

Είναι ένα περιβάλλον εκτέλεσης εντολών JAVASCRIPT εκτός του φυλλομετρητή, χτισμένο πάνω σε μηχανή v8. Σε γενικά πλαίσια επιτρέπει στον διακομιστή να γραφεί στην ίδια γλώσσα front-end με τα εργαλεία, σε

ασύγχρονο μοντέλο, κατάλληλο για πολλά ταυτόχρονα αιτήματα διαδικτύου. Είναι η πλατφόρμα που οργανώνει τις τεχνολογίες τεχνητής νοημοσύνης.

Πιο συγκεκριμένα τρέχει την εφαρμογή PitsiBot στον υπολογιστή, locally δηλαδή, παραδίδει τα στατικά αρχεία, εκτελεί την λογική και τέλος καλεί τα εξωτερικά API. Ακολουθεί στιγμιότυπο από εκκίνηση του server μέσω terminal και node.js



```
$ cd pitsibot2 merged && npm install
added 93 packages

$ node server.js

PitsiBot v2 τρέχει
http://localhost:3000
Groq: openai/gpt-oss-20b
openai/gpt-oss-120b

--- server.js (αρχή + listen) ---
// =====
// SERVER – Express. Τα API keys fortonontai apo to .env meso tou lib/config.js
// (an leipoun, o server stamataei ekei me megalo minima).
//
// >>> TA KEYS DEN EINAI ED0. EINAI STO ARXEIO .env STO ROOT TOU PROJECT. <<<
// =====

const express = require('express');
const cors = require('cors');
const path = require('path');

const cfg = require('./lib/config'); // PROTA auto: valid-arei ta keys
const flow = require('./lib/flow');
const store = require('./lib/store');
const groq = require('./lib/groq');
const serper = require('./lib/serper');
const { extract } = require('./lib/extract');
const { recommend } = require('./lib/recommend');
...
app.listen(cfg.PORT, () => {
```

[Εικόνα 6]Άνοιγμα server με node.js.

3.2.β Express

Το Express είναι ένα ελαφρύ framework για το Node.js, που ορίζει διαδρομές, οργανώνει το middleware και τις απαντήσεις JSON. Αντί να γράφει κάποιος χειροκίνητα λοιπόν, δηλώνει το αίτημα και το Express το δρομολογεί. Φαίνεται τόσο από το στιγμιότυπο που ακολουθεί όσο και από το παραπάνω ότι το Express είναι πρακτικά ο ενδιάμεσος που καλεί τα APIs, δομεί και οργανώνει όλη την επικοινωνία.

Στο ακόλουθο στιγμιότυπο φαίνεται ένα απόσπασμα από το αρχείο server.js το οποίο δείχνει το setup και τα routes που πρέπει να τηρήσει το Express.

```
Express — server.js (setup + routes)

10 const path = require('path');
11
12 const cfg = require('./lib/config'); // PROTA auto: valid-arei ta keys
13 const flow = require('./lib/flow');
14 const store = require('./lib/store');
15 const groq = require('./lib/groq');
16 const serper = require('./lib/serper');
17 const { extract } = require('./lib/extract');
18 const { recommend } = require('./lib/recommend');
19 const { chat } = require('./lib/chat');
20 const rl = require('./lib/ratelimit');
21
22 const app = express();
23 app.use(cors());
24 app.use(express.json({ limit: '256kb' }));
25
26 // Servaroume MONO ton fakelo /public — pote to root (gia na min ekteithei to .env)
27 // Dev-friendly: no-cache for html/js/css so character/delete UI updates show after deploy
28 app.use(express.static(path.join(__dirname, 'public'), {
29   setHeaders(res, filePath) {
30     if (/\. (html?|js|css)$/i.test(filePath)) {
31       res.setHeader('Cache-Control', 'no-cache');
32     }
33   }
34 }));
35
36 // ===== AUTH MIDDLEWARE =====
37 // Perimenei header: Authorization: Bearer <token>
38 function auth(req, res, next) {
39   const header = req.headers.authorization || '';
40   const username = store.verifyToken(header.replace(/^Bearer\s+/i, ''));
41   if (!username) return res.status(401).json({ error: 'Χρειάζεται σύνδεση' });
42   req.username = username;
43   next();
44 }
45
46 // ===== LOGGINGS =====
47 // LOGGINGS
48 // =====
49
50 app.post('/api/register', rl.auth, (req, res) => {
51   const { username, password } = req.body || {};
52   if (!username || !password) return res.status(400).json({ error: 'Συμπλήρωσε όλα τα πεδία' });
53   if (username.length < 3) return res.status(400).json({ error: 'Το username θέλει τουλάχιστον 3 χαρακτήρες' });
54   if (password.length < 8) return res.status(400).json({ error: 'Το password θέλει τουλάχιστον 8 χαρακτήρες' });
55   if (username.length > 40) return res.status(400).json({ error: 'Πολύ μεγάλο username' });
56   if (password.length > 200) return res.status(400).json({ error: 'Πολύ μεγάλο password' });
```

[Εικόνα 7] απόσπασμα με set up του express.

3.3 API

Τα APIs ή αλλιώς Application Programming Interfaces, είναι σύνολα κανόνων και ορισμών που επιτρέπουν σε διαφορετικά προγράμματα ή συστήματα να επικοινωνούν μεταξύ τους. Πιο συγκεκριμένα καθορίζουν τι μπορεί να ζητήσει το ένα πρόγραμμα από το άλλο, πως πρέπει να διατυπωθεί το αίτημα, και την μορφή της απάντησης. Όλα αυτά πραγματοποιούνται χωρίς να χρειάζεται να γνωρίζει το ένα σύστημα πως λειτουργεί το άλλο. Για να

αποκτήσει το ένα πρόγραμμα πρόσβαση σε ένα άλλο μέσω API υπηρεσιών χρησιμοποιούνται τα API keys .Τα κλειδιά αυτά λειτουργούν σαν μέσο ταυτοποίησης και εξουσιοδότησης για την χρήση αυτών των υπηρεσιών.

Στην παρούσα εφαρμογή χρησιμοποιήθηκαν με την χρήση API keys οι υπηρεσίες του Groq για την επικοινωνία με το Llama, ένα LLM για διαχείριση συζήτησης και οι υπηρεσίες του Serper, για παροχή αποτελεσμάτων από αναζητήσεις τύπου Google.

3.3.α Groq

Η Groq, είναι μια πλατφόρμα που προσφέρει πρόσβαση σε μεγάλα γλωσσικά μοντέλα με API keys. Η πλατφόρμα δεν αναζητά, αλλά δέχεται κείμενο και επιστρέφει απάντηση. Στην υλοποίηση της εργασίας ο ρόλος της ήταν να κατανοήσει τα μηνύματα που δέχεται από τον χρήστη να εντοπίζει κατηγορίες, χαρακτηριστικά και budget, μέσα από διάφορες επιλογές και διατυπώνει την απάντηση που θα δοθεί στον χρήστη στα ελληνικά. Χωρίς το Serper όμως εφευρίσκει μοντέλα ή τιμές, οπότε χρειάστηκε να δουλέψει σε συνδυασμό πάνω σε ήδη φιλτραρισμένες αναφορές. Το φαινόμενο αυτό σχετίζεται με τα λεγόμενα hallucinations των συστημάτων παραγωγής φυσικής γλώσσας, κατά τα οποία ένα μοντέλο μπορεί να παράγει πληροφορίες που δεν υποστηρίζονται από τα διαθέσιμα δεδομένα ή δεν ανταποκρίνονται σε πραγματικά γεγονότα (Ji et al., 2023).

3.3.β Serper

Το Serper είναι ένα API αναζήτησης που επιστρέφει αποτελέσματα τύπου Google σε JSON αρχείο .Στο PitsiBot χρησιμοποιήθηκε για να βρεθούν σελίδες από καταστήματα της ελληνικής αγοράς, για να γίνει το σωστό φιλτράρισμα

και να επιστρέφει εναλλακτικές με Links .Είναι απαραίτητο λοιπόν για την πιο ενημερωμένη λίστα επιλογών με βάση την επικαιρότητα.

3.4 JSON catalog, cache

3.4.α Κατάλογος JSON

Το JSON ή στα ελληνικά JavaScript Object Notation είναι μορφή κειμένου που χρησιμοποιείται για την αποθήκευση και τη μεταφορά δεδομένων. Ένα αρχείο JSON είναι ένα απλό αρχείο κειμένου με κατάληξη .json και τύπο MIME. Τα δεδομένα σε ένα αρχείο JSON οργανώνονται σε πεδία και τιμές. Η μορφή του θυμίζει τα αντικείμενα της JavaScript, όμως μπορεί να χρησιμοποιηθεί και από άλλες γλώσσες προγραμματισμού. Για αυτόν τον λόγο, χρησιμοποιείται συχνά σε διαδικτυακές εφαρμογές για την ανταλλαγή δεδομένων μεταξύ του server και του client. Για παράδειγμα, ο server μπορεί να στείλει δεδομένα σε μορφή JSON στον client. Ο client διαβάζει τα δεδομένα και τα εμφανίζει στην ιστοσελίδα. Η ίδια διαδικασία μπορεί να γίνει και αντίστροφα, όταν ο client στέλνει δεδομένα στον server.

Στη JavaScript, η επεξεργασία αρχείων JSON γίνεται με τη χρήση του αντικειμένου JSON. Με αυτόν τον τρόπο, τα δεδομένα μπορούν να μετατραπούν από κείμενο σε αντικείμενο και αντίστροφα. Στην παρούσα εφαρμογή, τα αρχεία JSON χρησιμοποιούνται για την αποθήκευση πληροφοριών προϊόντων, δεδομένων προσωρινής μνήμης και στοιχείων που σχετίζονται με τις συνομιλίες των χρηστών. Τα δεδομένα αποθηκεύονται σε συγκεκριμένα πεδία, όπως η κατηγορία, το όνομα, η τιμή και τα χαρακτηριστικά του προϊόντος. Η συγκεκριμένη δομή βοηθά την εφαρμογή να διαβάζει και να επεξεργάζεται τα δεδομένα με σταθερό τρόπο. Παράλληλα,

επιτρέπει στο γλωσσικό μοντέλο να χρησιμοποιεί οργανωμένες πληροφορίες αντί να βασίζεται μόνο σε ελεύθερο κείμενο.

```
1 {
2   "category": "kinito",
3   "count": 160,
4   "items": {
5     "kinito:samsung-galaxy-a26": {
6       "id": "kinito:samsung-galaxy-a26",
7       "name": "Samsung Galaxy A26 6GB/128GB 5G",
8       "brand": {
9         "value": "Samsung",
10        "source": "kotsovolos",
11        "at": "2026-09-06T20:15:46.295Z"
12      },
13      "specs": {
14        "5g": true,
15        "storage": 128,
16        "ram": 6
17      },
18      "price_min": {
19        "value": 242,
20        "source": "skroutz",
21        "at": "2026-09-19T19:15:14.625Z"
22      },
23      "release_year": {
24        "value": 2025,
25        "source": "kotsovolos",
26        "at": "2026-09-06T20:15:46.295Z"
27      },
28      "urls": {
29        "skroutz": "https://www.skroutz.gr/s/59363684/Samsung-Galaxy-A26-5G-Dual...",
30        "kotsovolos": "https://www.kotsovolos.gr/mobile-phones-gps/mobile-phones/sm..."
31      },
32      "shops": [
33        "skroutz",
34        "kotsovolos"
35      ]
36    }
37  }
38 }
```

[Εικόνα 8] Παράδειγμα λίστας JSON.

3.4.β Μνήμη cache

Η cache είναι μια προσωρινή μνήμη που αποθηκεύει αποτελέσματα τα οποία έχουν ήδη χρησιμοποιηθεί. Με αυτόν τον τρόπο, η εφαρμογή δεν χρειάζεται να εκτελεί ξανά την ίδια διαδικασία κάθε φορά. Έτσι, μειώνεται ο χρόνος απόκρισης και ο αριθμός των κλήσεων προς τις εξωτερικές υπηρεσίες.

Στην παρούσα εργασία, η cache αποθηκεύεται σε αρχείο JSON μέσα στον φάκελο των δεδομένων. Το αρχείο περιέχει αποτελέσματα προηγούμενων αναζητήσεων και πληροφορίες για προϊόντα που έχουν ήδη ελεγχθεί από την εφαρμογή. Όταν ο χρήστης πραγματοποιεί μια παρόμοια αναζήτηση, η εφαρμογή ελέγχει πρώτα την cache. Αν βρει τα απαραίτητα δεδομένα, τα

χρησιμοποιεί χωρίς να χρειαστεί νέα κλήση στο API. Αν δεν βρει τα δεδομένα ή αυτά δεν είναι πλέον επίκαιρα, πραγματοποιείται νέα αναζήτηση και η cache ενημερώνεται. Με αυτόν τον τρόπο, η εφαρμογή λειτουργεί πιο γρήγορα και χρησιμοποιεί λιγότερους πόρους. Ωστόσο, τα αποθηκευμένα δεδομένα μπορεί να μην είναι πάντα ενημερωμένα. Για αυτόν τον λόγο, θα πρέπει να ελέγχεται η ημερομηνία αποθήκευσής τους ή να γίνεται νέα αναζήτηση μετά από κάποιο χρονικό διάστημα.

```
1 {
2   "msi cyborg 15 a13vf": {
3     "at": "2026-09-06T13:01:16.054Z",
4     "name": "MSI Cyborg 15 A13VF",
5     "data": {
6       "links": {
7         "skroutz": "https://www.skroutz.gr/s/58405828/MSI-Cyborg-15-A13VF-15-6-FHD-144Hz-i7-13620H-16GB-512GB-SSD-GeForce-RTX-4060-No-OS-International-English-Keybaord.html",
8         "public": "https://www.public.gr/product/computers-and-software/laptops/msi-cyborg-15-a13vf-156-fhd-ips-core-i7-13620h16gb1tb-ssdgeforce-rtx-4060win11home-laptop/1986684",
9         "plaisio": "https://www.plaisio.gr/product/pc-perifereiaka/laptops/msi-cyborg-15-a13vf-1689gr-laptop-15-dot-6-ips-core-i7-13620h-16-gb-1-tb-rtx-4060-8-gb-windows-11-home_4684699",
10        "kotsovolos": "https://www.kotsovolos.gr/computing/laptop-tablet-ipad/notebook-macbook-ultrabook/298814-nb-msi-cyborg-15-i7-13620h-16-512-4060"
11      },
12      "count": 4,
13      "retrieved": false
14    }
15  }
16 }
```

[Εικόνα 9] Παράδειγμα προϊόν από cache.

ΚΕΦΑΛΑΙΟ 4 Πειράματα

4.1 Offline tests

Στην παρούσα ενότητα θα δούμε κάποια πειράματα και τεστ που έγιναν χρησιμοποιώντας μόνο τους καταλόγους JSON και τις πληροφορίες από την μνήμη cache. Μεγάλη σημασία στα αποτελέσματα έχει η ευστοχία των προτάσεων, η επικαιρότητα των προϊόντων και ο αποκλεισμός μη διαθεσίμων προϊόντων. Επίσης σημαντικές είναι η εγκυρότητα στις τιμές και η παρουσίαση των προϊόντων, ξεκινώντας από αυτό με το μικρότερο κόστος και κλείνοντας με το πιο ακριβό.

Στο πρώτο πείραμα που έγινε ζητήθηκε από το PitsiBot να παρουσιάσει επιλογές για κινητά τηλέφωνα, για καθημερινή και απλή χρήση, με περιορισμό στο budget να φτάνει μέχρι 300 ευρώ. Στο δεύτερο παράδειγμα ζητήθηκε ένας φορητός υπολογιστής, πάλι για καθημερινή χρήση με περιορισμό στο budget τα 600 ευρώ. Οι λόγοι που διεξάχθηκαν αυτά τα δυο πειράματα είναι δυο. Πρώτον επειδή το chatbot έχει ρητές οδηγίες μέσα από prompt

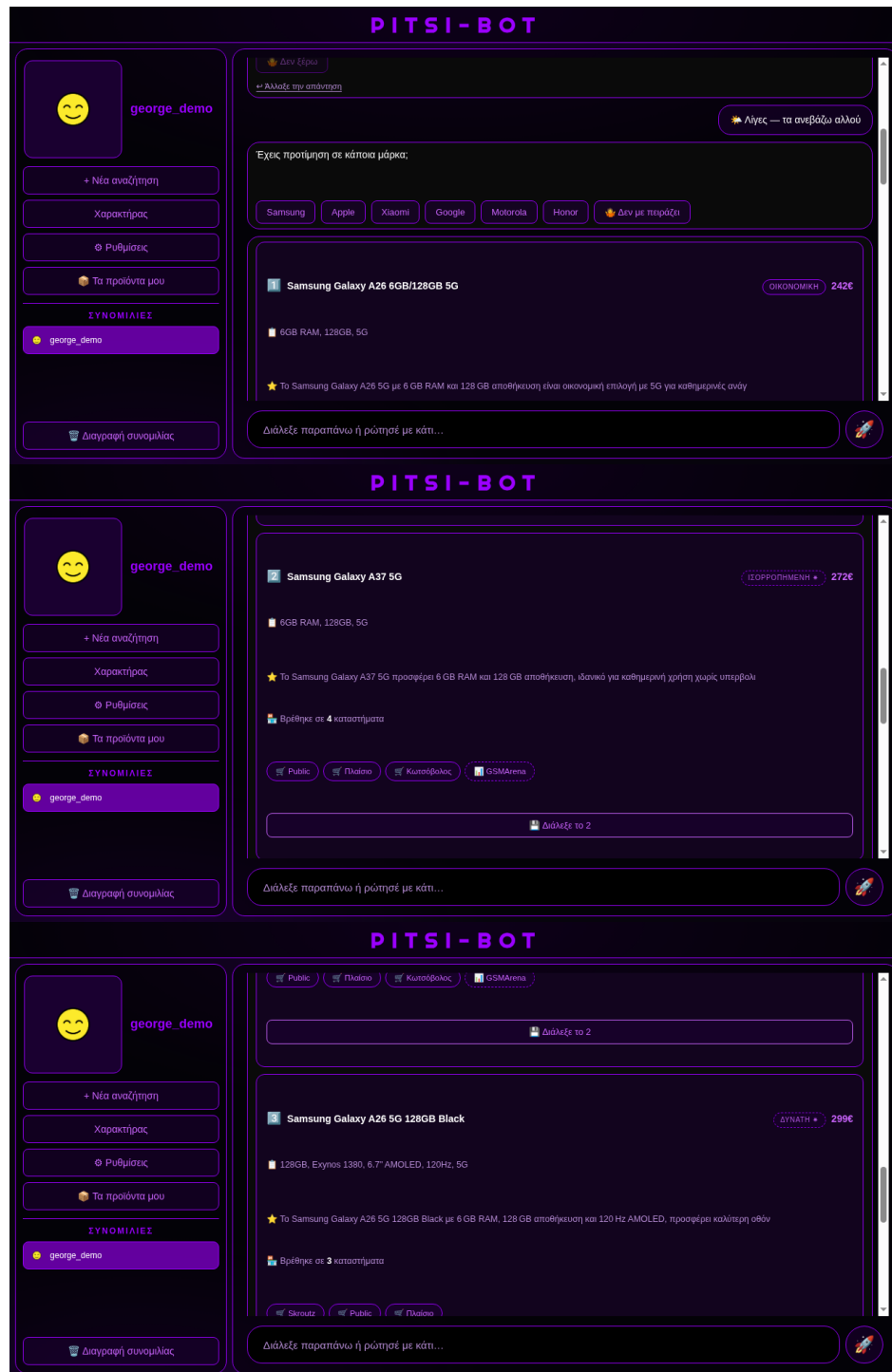
engineering να δίνει πάντα τρεις επιλογές, μια χαμηλού κόστους, μια μεσαίου και μια υψηλού αν δεν αναφέρει κανέναν περιορισμό ο χρήστης. Οπότε πρέπει να παρατηρηθεί πως θα λειτουργήσει όταν ο χρήστης βάλει κάποιον περιορισμό. Δεύτερον θα έπρεπε να βάλει τις τιμές σε αύξουσα σειρά. Όλα αυτά χωρίς να χάνει στην ευστοχία και σε όσα προαναφέρθηκαν.

Στο τρίτο πείραμα δοκιμάστηκε μια συνομιλία για την εύρεση φορητού υπολογιστή με ένα συγκεκριμένο χαρακτηριστικό σαν απαίτηση από τον χρήστη. Στο παράδειγμα αυτό το χαρακτηριστικό που δοκιμάστηκε ήταν η κάρτα γραφικών και πιο συγκεκριμένα ζητήθηκε RTX 5060. Ενώ στο τέταρτο πείραμα δοκιμάστηκε το ίδιο αλλά σε κινητή συσκευή. Το χαρακτηριστικό που ζητήθηκε σε αυτήν την περίπτωση ήταν ο αποθηκευτικός χώρος να είναι 512GB. Σε αυτά τα δυο πειράματα εξετάζεται το πως θα ανταποκριθεί σε πολύ περιοριστικά κριτήρια, καθώς και τα δυο αυτά χαρακτηριστικά είναι πολύ απαιτητικά και πιο δυσεύρετα στην αγορά. Στόχος είναι να παρέχει μόνο εύστοχες προτάσεις, ακόμα και αν προκύψουν λιγότερες από τρεις, καθώς κύριο κριτήριο στην επιλογή είναι η πλήρης κάλυψη των αναγκών του χρήστη.

Τέλος ζητήθηκε από το chatbot να βρει κινητές συσκευές και η μονή πληροφορία που δόθηκε είναι πως αφορά αγορά από γονιό για το παιδί του. Αυτό δεν δίνει κάποια ξεκάθαρη πληροφορία, αλλά είναι ένα από τα σενάρια που το PitsiBot πιθανά να κληθεί να αντιμετωπίσει. Για αυτόν τον λόγο πάλι μέσω prompt engineering έχει λάβει κάποιες οδηγίες για κάποιες από αυτές τις περιπτώσεις, οι οποίες πρέπει να ελεγχθούν.

Ακολουθούν οι προτάσεις και οι πίνακες σύγκρισης προτάσεων από τα τεστ 1-5 για τις offline αναζητήσεις.

4.1.α Τεστ 1. Κινητά για καθημερινή χρήση μέχρι 300 ευρώ



[Εικόνα 10] προτάσεις πρώτου τεστ.

4.1.β Τεστ 2. Laptop για καθημερινή χρήση μέχρι 600 ευρώ

**george_demo**

+ Νέα αναζήτηση

Χαρακτήρας

⚙ Ρυθμίσεις

🛒 Τα προϊόντα μου

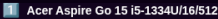
ΣΥΝΟΜΙΛΙΕΣ

● george_demo

● george_demo

🗑 Διαγραφή συνομιλίας

PITSI-BOT

**Acer Aspire Go 15 i5-1334U/16/512**

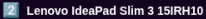
ΟΙΚΟΝΟΜΙΚΗ 549€

■ 16GB RAM, 512GB, i5-1334U, 15.6"

📦 Βρέθηκε σε 1 καταστήματα

🛒 Καταβολός

👤 Διάλεξε το 1

**Lenovo IdeaPad Slim 3 15IRH10**

ΔΥΝΑΤΗ 699€

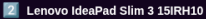
■ 8GB RAM, 512GB, i5-13420H, 15.3" IPS

📦 Βρέθηκε σε 5 καταστήματα

🛒 Skrautz 🛒 Καταβολός 🛒 Πλαιοιο 🛒 Public 🗒 review

👤 Διάλεξε το 1

PITSI-BOT

**Lenovo IdeaPad Slim 3 15IRH10**

ΔΥΝΑΤΗ 699€

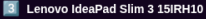
■ 8GB RAM, 512GB, i5-13420H, 15.3" IPS

📦 Βρέθηκε σε 5 καταστήματα

🛒 Skrautz 🛒 Καταβολός 🛒 Πλαιοιο 🛒 Public 🗒 review

👤 Διάλεξε το 1

PITSI-BOT

**Lenovo IdeaPad Slim 3 15IRH10**

ΔΥΝΑΤΗ 699€

■ 8GB RAM, 512GB, i5-13420H, 15.3" IPS

📦 Βρέθηκε σε 5 καταστήματα

🛒 Skrautz 🛒 Καταβολός 🛒 Πλαιοιο 🛒 Public 🗒 review

👤 Διάλεξε το 3

[Εικόνα 12] προτάσεις δεύτερου τεστ.

21



george_demo

+ Νέα αναζήτηση

Χαρακτήρας

⚙ Ρυθμίσεις

🛒 Τα προϊόντα μου

ΣΥΝΟΜΙΛΙΕΣ

 george_demo

 george_demo

🗑 Διαγραφή συνομιλίας

📊 Σύγκριση — έντονα ό,τι είναι καλύτερο σε κάθε γραμμή

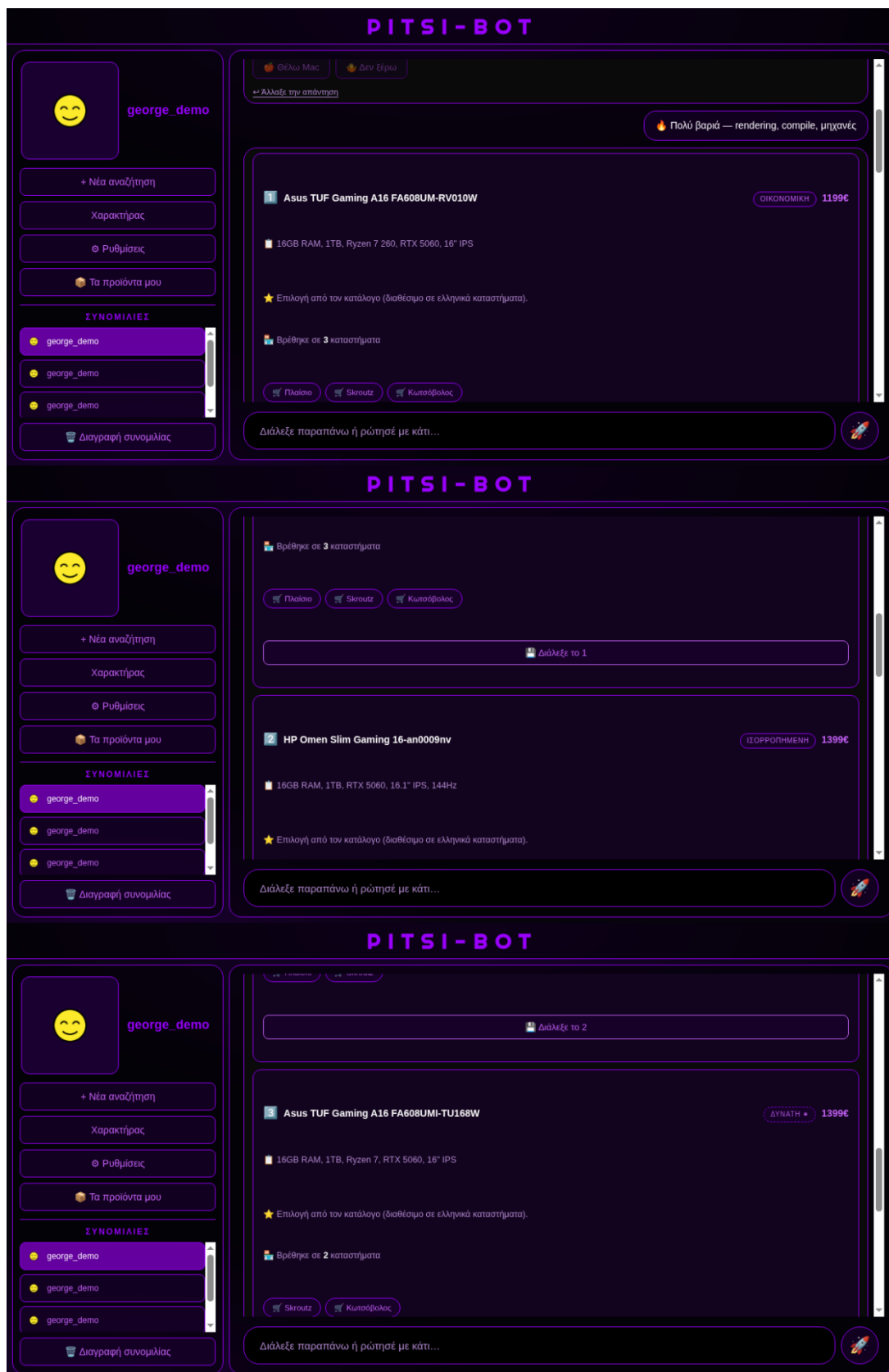
	1 Acer Aspire Go 15 i5-1334U/16/512	2 Lenovo IdeaPad Slim 3 15IRH10	3 Lenovo IdeaPad Slim 3 15IRH10
💡 Τιμή	549€ •	699€	699€
⚡ RAM	16GB •	8GB	8GB
🖥 Οθόνη	15.6"	15.3"	15.3"
🔥 Επεξεργαστής	i5-1334U	i5-13420H	i5-13420H
📦 Καταστήματα	1	5 •	5 •

Εμφανίζονται μόνο όσα διαφέρουν μεταξύ τους.

Διάλεξε παραπάνω ή ρώτησέ με κάτι... 

[Εικόνα 13] πλαίσιο σύγκρισης προτάσεων

4.1.γ Τεστ 3. Συγκεκριμένο χαρακτηριστικό Laptop (κάρτα γραφικών 5060)



[Εικόνα 14] προτάσεις τρίτου τεστ.



george_demo

+ Νέα αναζήτηση

Χαρακτήρας

⚙ Ρυθμίσεις

🛒 Τα προϊόντα μου

ΣΥΝΟΜΙΛΙΕΣ

👤 george_demo

Αναζήτηση

👤 george_demo

Αναζήτηση

👤 george_demo

Αναζήτηση

📄

Διαγραφή συνομιλίας

PITSI-BOT

1

Asus TUF Gaming A16 FA608UM-RV010W

2

HP Omen Slim Gaming 16-an0009nv

3

Asus TUF Gaming A16 FA608UM-TU168W

🔥 Τιμή	1199€	1399€	1399€
📺 Οθόνη	16"	16.1"	16"
🔄 Refresh	165Hz	144Hz	144Hz
🔥 Επεξεργαστής	Ryzen 7 260	—	Ryzen 7
📦 Καταστήματα	3	2	2

Εμφανίζονται μόνο όσα διαφέρουν μεταξύ τους.

📊

Τα κενά τιμής είναι 200€ / 0€ — στόχος ~250-350 και 300-500€.

🚀

Διάλεξε παραπάνω ή ρώτησέ με κάτι...

[Εικόνα 15] πλαίσιο σύγκρισης προτάσεων

24

4.1.δ Τεστ 4. Συγκεκριμένο χαρακτηριστικό κινητό (αποθηκευτικός χώρος 512GB)



george_demo

+ Νέα αναζήτηση

Χαρακτήρας

Ρυθμίσεις

Τα προϊόντα μου

ΣΥΝΟΜΙΛΙΕΣ

● george_demo

● george_demo

● george_demo

🗑️ Διαγραφή συνομιλίας

PITSI-BOT

Εχεις προτίμηση σε κάποια μάρκα;

SamsungAppleXiaomiGoogleMotorolaHonor🔥 Δεν με πειράζει

➡️ Άλλαξε την απάντηση

🔥 Δεν με πειράζει

 **Xiaomi Redmi Note 15 Pro+ 12GB/512GB 5G**

ΟΙΚΟΝΟΜΙΚΗ539€



Τιμή 539 €Πάνελ AMOLED 120 HzΜέγεθος οθόνης 6.3"

...

Διάλεξε παραπάνω ή ρώτησέ με κάτι...



george_demo

+ Νέα αναζήτηση

Χαρακτήρας

Ρυθμίσεις

Τα προϊόντα μου

ΣΥΝΟΜΙΛΙΕΣ

● george_demo

● george_demo

● george_demo

🗑️ Διαγραφή συνομιλίας

PITSI-BOT

 Skrautz

🖨️ Διάλεξε το 1

 **Xiaomi 17T Pro 12GB/512GB 5G**

ΙΣΟΠΡΟΫΗΜΕΝΗ949€



Τιμή 949 €Πάνελ AMOLED 120 HzΜέγεθος οθόνης 6.3"

★ Το Xiaomi 17T Pro με 512 GB αποθήκευση και 12 GB RAM είναι εντυπωσιακό για φωτογραφίες με υψηλή ανάλυση και ομ

📍 Βρέθηκε σε 9 καταστήματα

Διάλεξε παραπάνω ή ρώτησέ με κάτι...



george_demo

+ Νέα αναζήτηση

Χαρακτήρας

Ρυθμίσεις

Τα προϊόντα μου

ΣΥΝΟΜΙΛΙΕΣ

● george_demo

● george_demo

● george_demo

🗑️ Διαγραφή συνομιλίας

PITSI-BOT

🖨️ Διάλεξε το 2

 **Samsung Galaxy S26 Ultra 12GB/512GB Sky Blue**

ΔΥΝΑΤΗ1699€



★ Το Samsung Galaxy S26 Ultra 12 GB/512 GB προσφέρει κορυφαία κάμερα και ισχυρό επεξεργαστή για επαγγελματικές φ

📍 Βρέθηκε σε 1 καταστήματα

 Kitsapoulos

Διάλεξε παραπάνω ή ρώτησέ με κάτι...

[Εικόνα 16] προτάσεις τέταρτου τεστ.

george_demo

+ Νέα αναζήτηση

Χαρακτήρας

🔍 Ρυθμίσεις

🛒 Τα προϊόντα μου

ΣΥΝΟΜΙΛΙΕΣ

george_demo

george_demo

george_demo

📄 Διαγραφή συνομιλίας

1

Asus TUF Gaming A16 FA608UM-RV010W

2

HP Omen Slim Gaming 16-an0009nv

3

Asus TUF Gaming A16 FA608UM-TU168W

🔥 Τιμή	1199€	1399€	1399€
📺 Οθόνη	16"	16.1"	16"
🔄 Refresh	165Hz	144Hz	144Hz
🔥 Επεξεργαστής	Ryzen 7 260	—	Ryzen 7
📦 Καταστήματα	3	2	2

Εμφανίζονται μόνο όσα διαφέρουν μεταξύ τους.

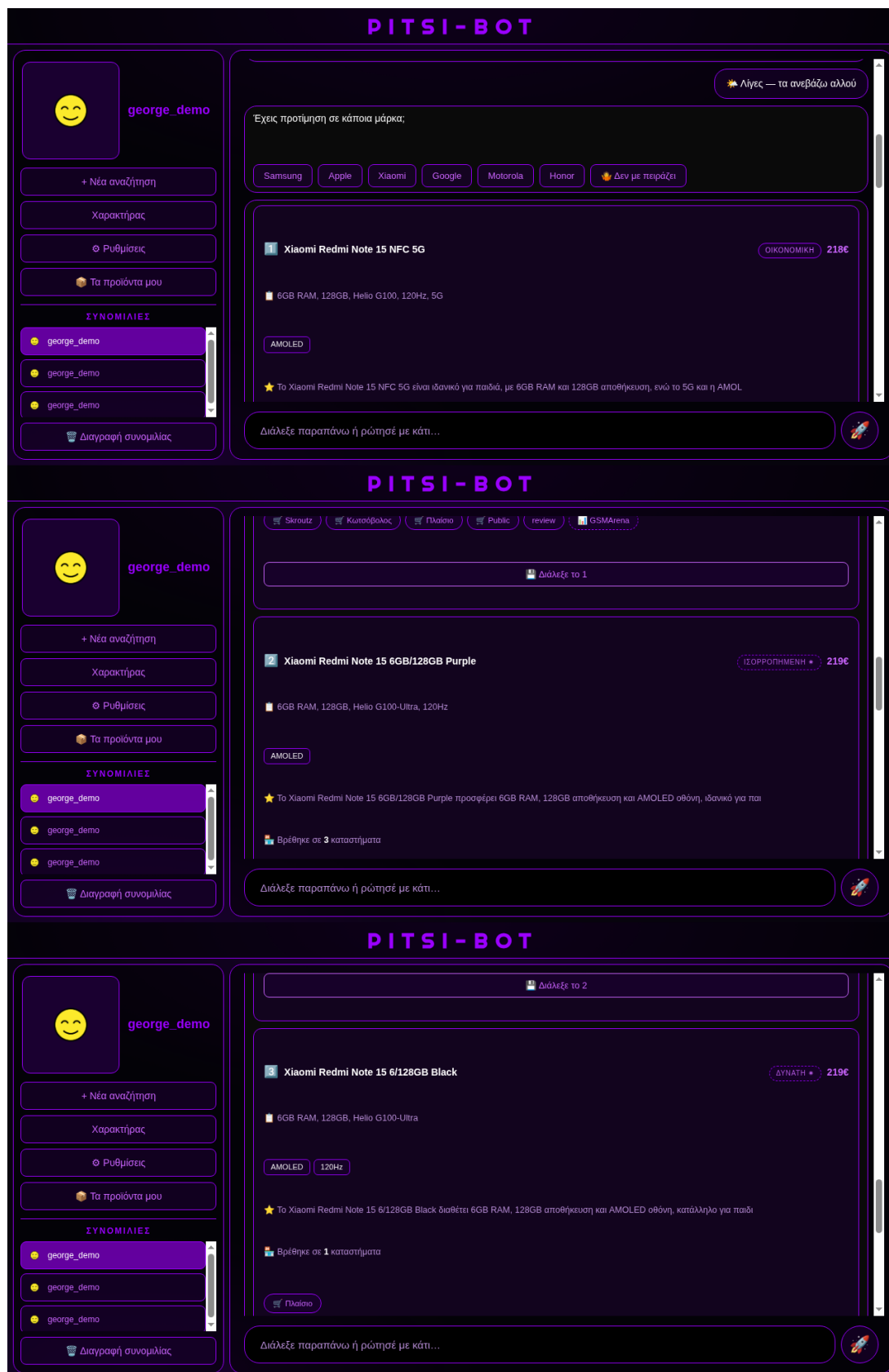
📌 Τα κενά τιμές είναι 200€ / 0€ — στόχος ~250-350 και 300-500€.

Διάλεξε παραπάνω ή ρώτησέ με κάτι...

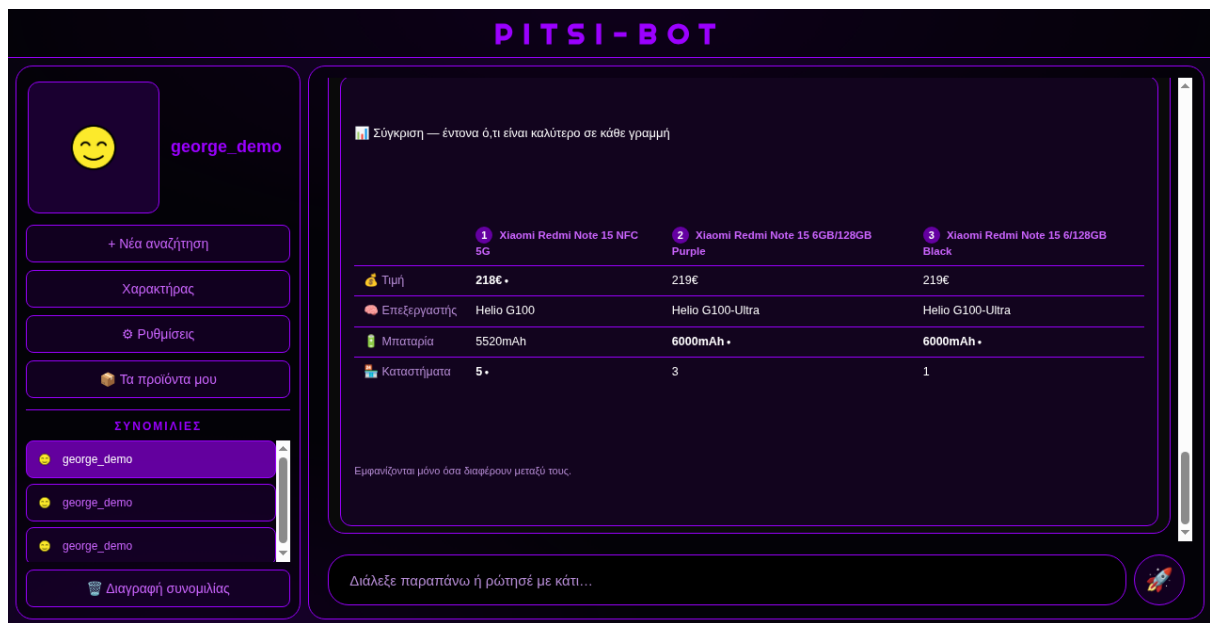
[Εικόνα 17] πλαίσιο σύγκρισης προτάσεων

26

4.1.ε Τεστ 5. Σενάριο (γονιός για παιδί)



[Εικόνα 18] προτάσεις πέμπτου τεστ.

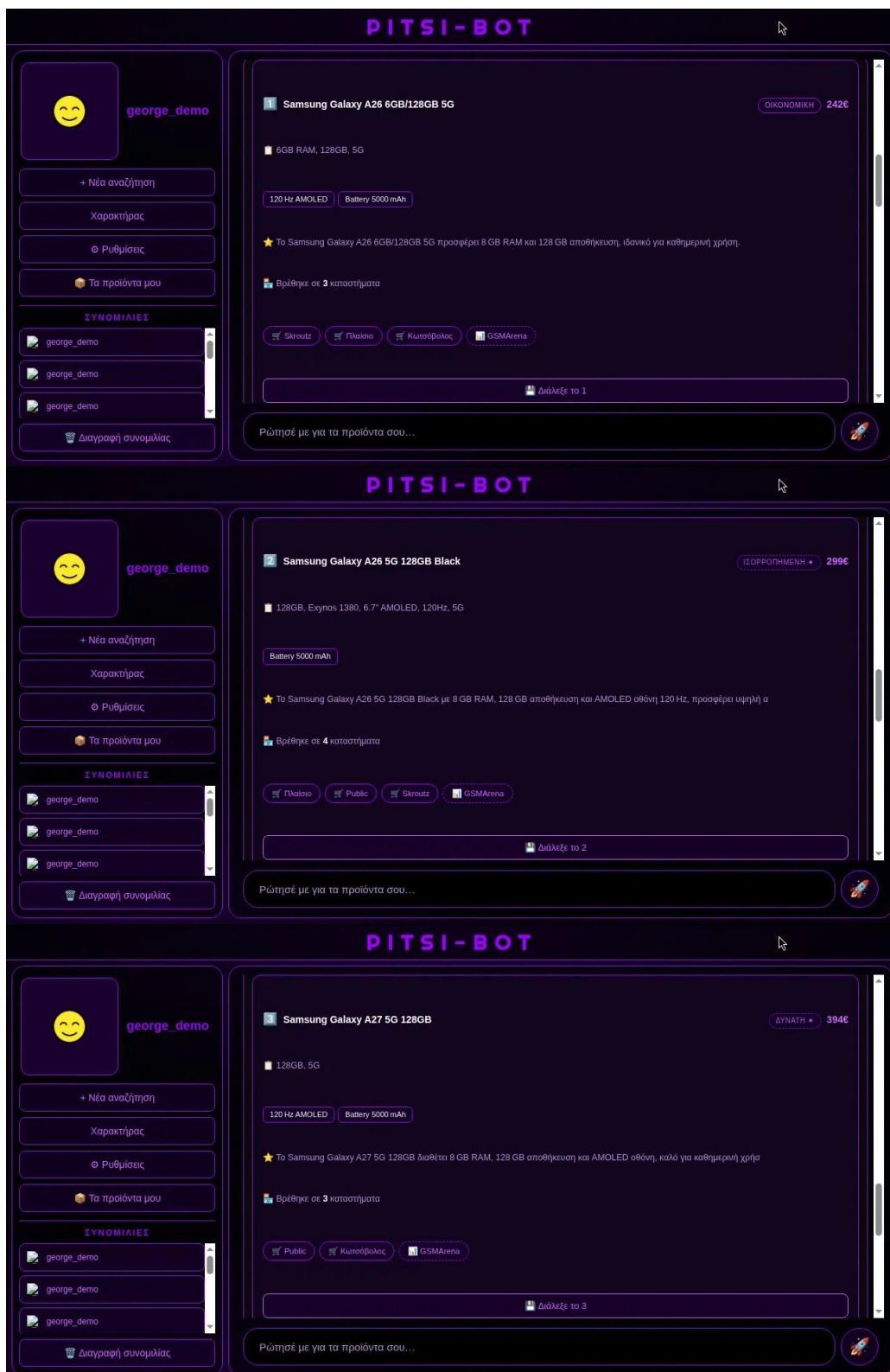


[Εικόνα 19] πλαίσιο σύγκρισης προτάσεων

4.2 Live πειράματα

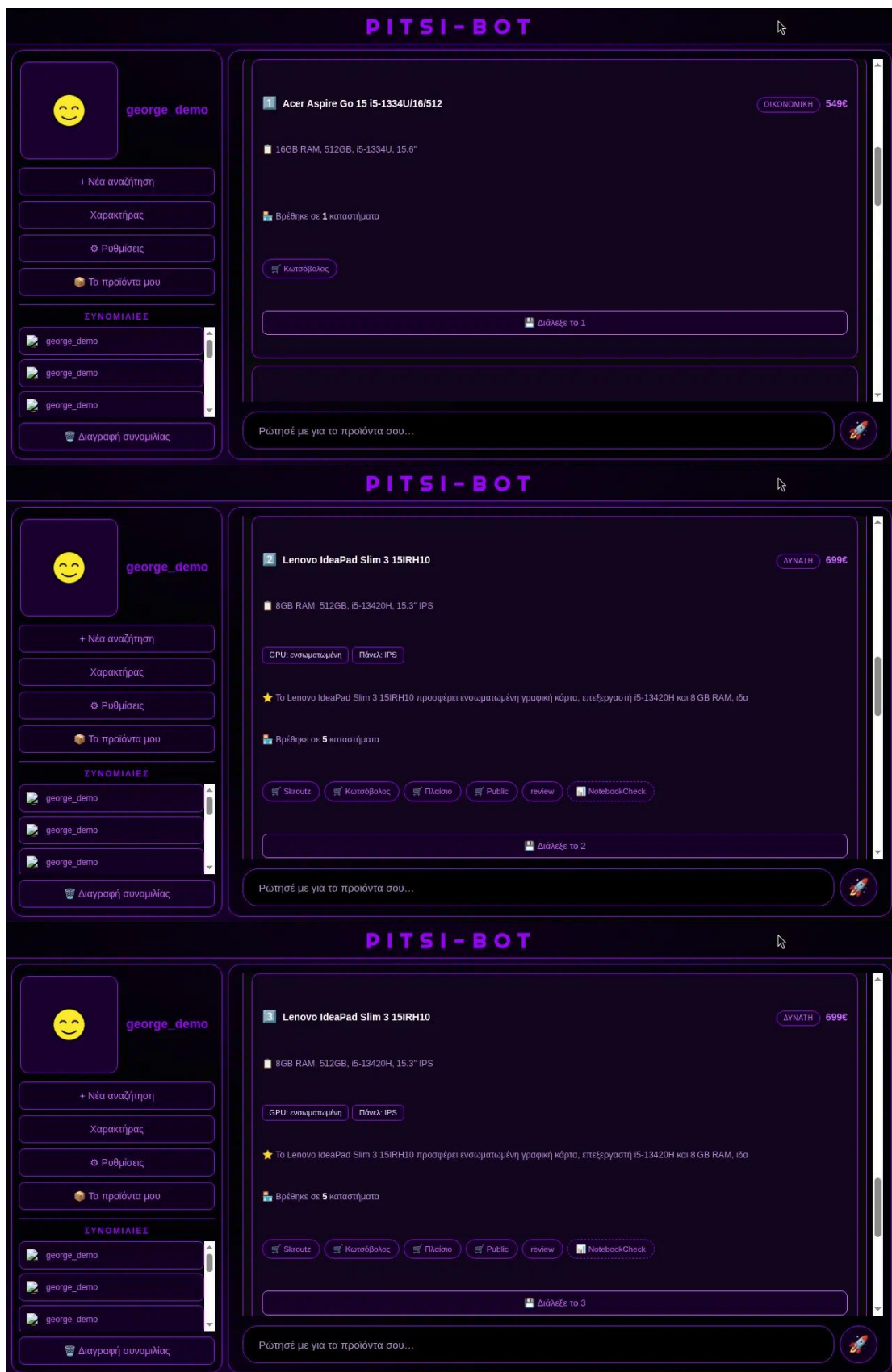
Στην παρούσα ενότητα θα παρουσιαστούν τα αποτελέσματα των αναζητήσεων όταν προστίθεται και η live αναζήτηση στα εργαλεία που χρησιμοποιεί το PitsiBot για να καταλήξει σε προτάσεις. Τα παραδείγματα που χρησιμοποιήθηκαν είναι τα ίδια ακριβώς με τα offline τεστ για καλύτερη παρατήρηση και σύγκριση αποτελεσμάτων. Παρομοίως, οι στόχοι και τα κριτήρια καθώς και οι λόγοι που έγινε κάθε πείραμα είναι επίσης οι ίδιοι. Ένα επιπλέον χαρακτηριστικό όμως που εξετάζεται σε αυτήν την ενότητα είναι ο έλεγχος της επικαιρότητας, η επιλογή των πιο σύγχρονων μοντέλων και η συνεργασία μεταξύ του Grog και του Serper.

4.2.α Τεστ 1. Κινητά για καθημερινή χρήση μέχρι 300 ευρώ



[Εικόνα 20] προτάσεις πρώτου τεστ.

4.2.β Τεστ 2. Laptop για καθημερινή χρήση μέχρι 600 ευρώ



[Εικόνα 22] προτάσεις δεύτερου τεστ.



george_demo

+ Νέα αναζήτηση

Χαρακτήρας

⚙ Ρυθμίσεις

🛒 Τα προϊόντα μου

ΣΥΝΟΜΙΛΙΕΣ

 george_demo

 george_demo

 george_demo

 Διαγραφή συνομιλίας

🏠 Σύγκριση — έντονα ό,τι είναι καλύτερο σε κάθε γραμμή

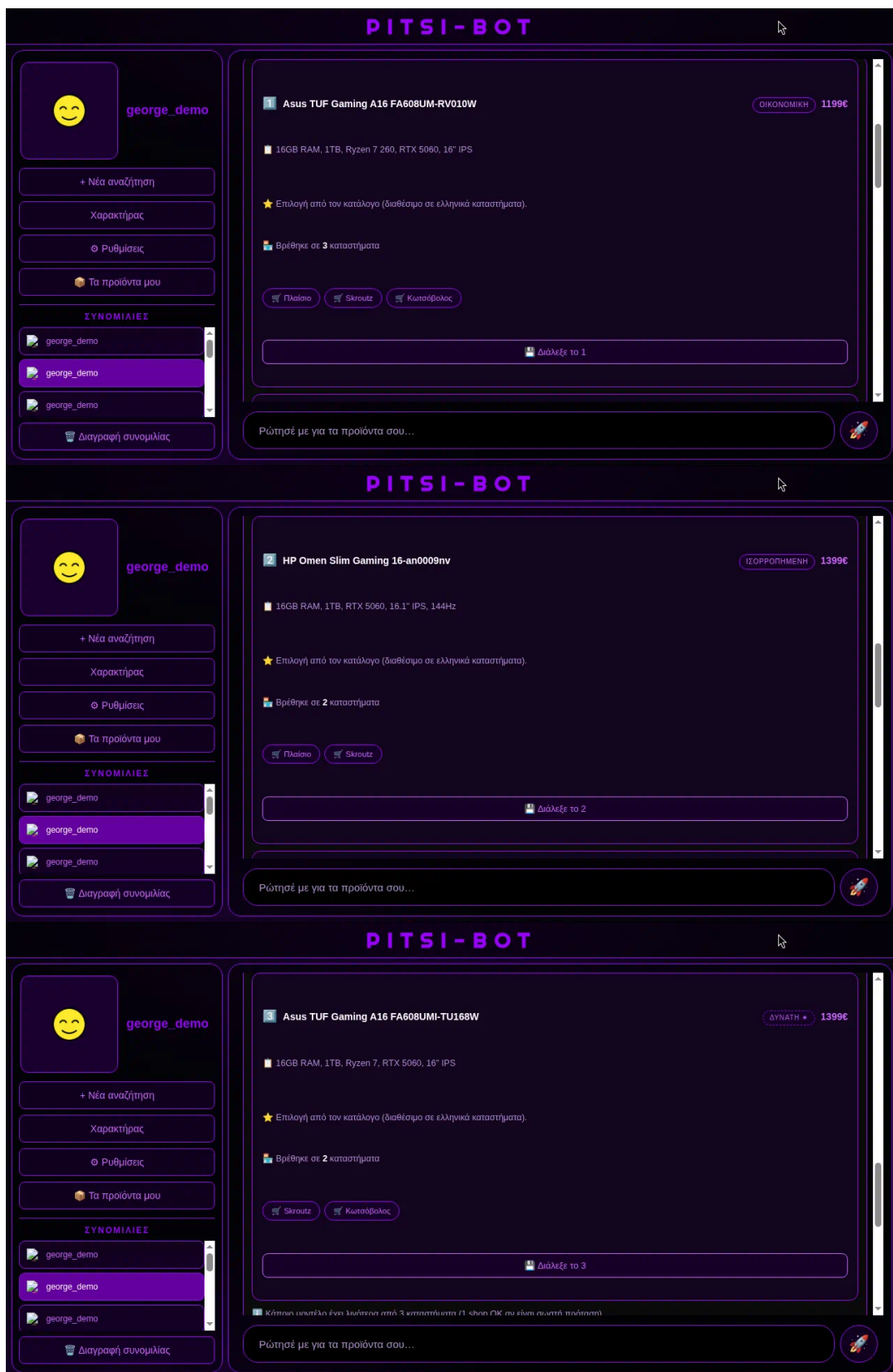
	1 Acer Aspire Go 15 i5-1334U/16/512	2 Lenovo IdeaPad Slim 3 15IRH10	3 Lenovo IdeaPad Slim 3 15IRH10
💰 Τιμή	549€ •	699€	699€
⚡ RAM	16GB •	8GB	8GB
🖥 Οθόνη	15.6"	15.3"	15.3"
🔥 Επεξεργαστής	i5-1334U	i5-13420H	i5-13420H
📦 Καταστήματα	1	5 •	5 •

Εμφανίζονται μόνο όσα διαφέρουν μεταξύ τους.

Διάλεξε παραπάνω ή ρώτησέ με κάτι... 

[Εικόνα 23] πλαίσιο σύγκρισης προτάσεων

4.2.γ Τεστ 3. Συγκεκριμένο χαρακτηριστικό Laptop (κάρτα γραφικών 5060)



[Εικόνα 24] προτάσεις τρίτου τεστ.

george_demo

+ Νέα αναζήτηση

Χαρακτήρας

Φ Ρυθμίσεις

Τα προϊόντα μου

ΣΥΝΟΜΙΛΙΕΣ

george_demo

george_demo

george_demo

Διαγραφή συνομιλίας

Σύγκριση — έντονα ό,τι είναι καλύτερο σε κάθε γραμμή

	1 Asus TUF Gaming A16 FA608UM-RV010W	2 HP Omen Slim Gaming 16-an0009nv	3 Asus TUF Gaming A16 FA608UM-TU168W
Τιμή	1199€	1399€	1399€
Οθόνη	16"	16.1"	16"
Refresh	165Hz	144Hz	144Hz
Επεξεργαστής	Ryzen 7 260	—	Ryzen 7
Καταστήματα	3	2	2

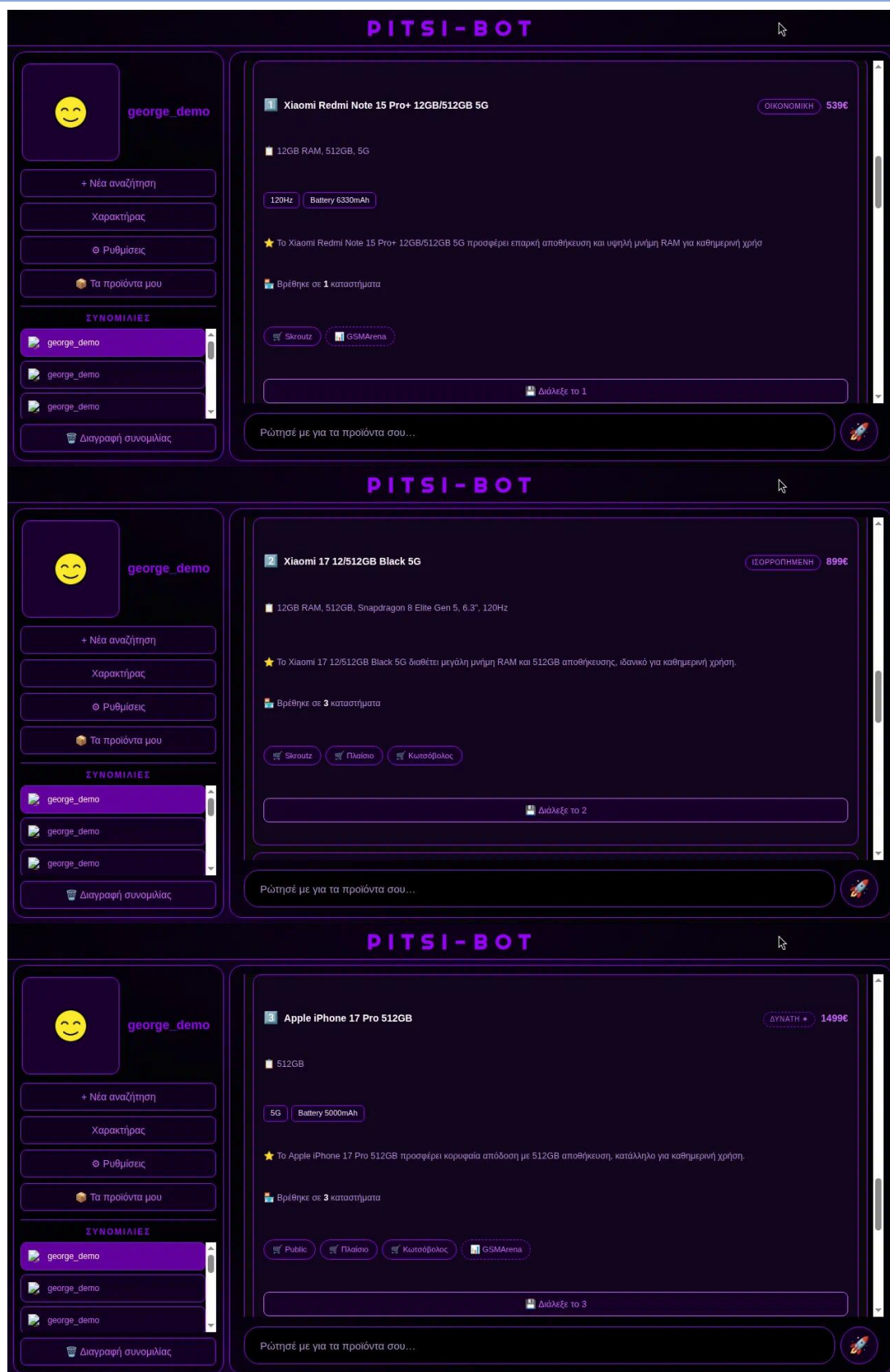
Εμφανίζονται μόνο όσα διαφέρουν μεταξύ τους.

Διάλεξε παραπάνω ή ρώτησέ με κάτι...

[Εικόνα 25] πλαίσιο σύγκρισης προτάσεων

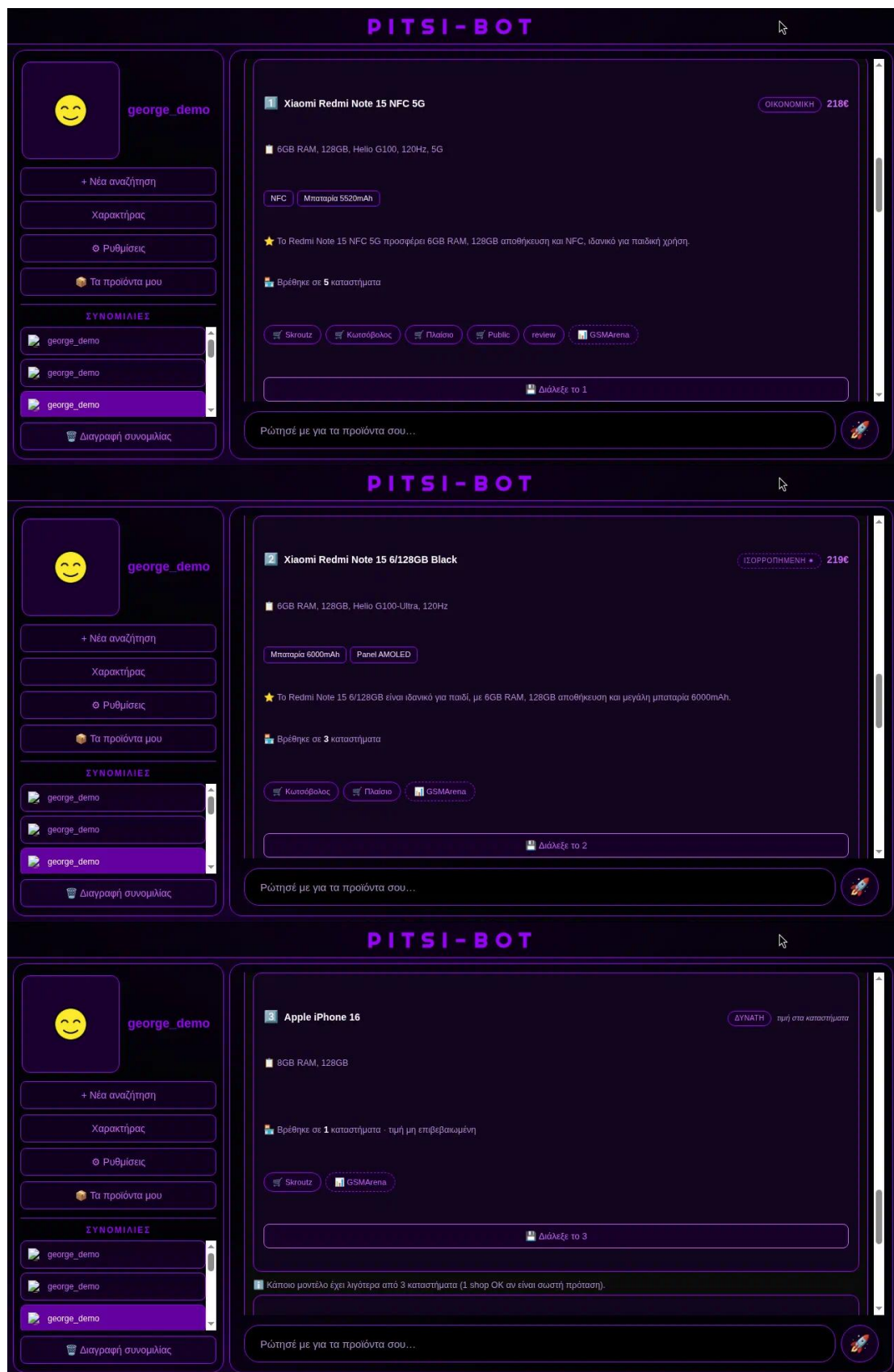
34

4.2.δ Τεστ 4. Συγκεκριμένο χαρακτηριστικό κινητό (αποθηκευτικός χώρος 512GB)

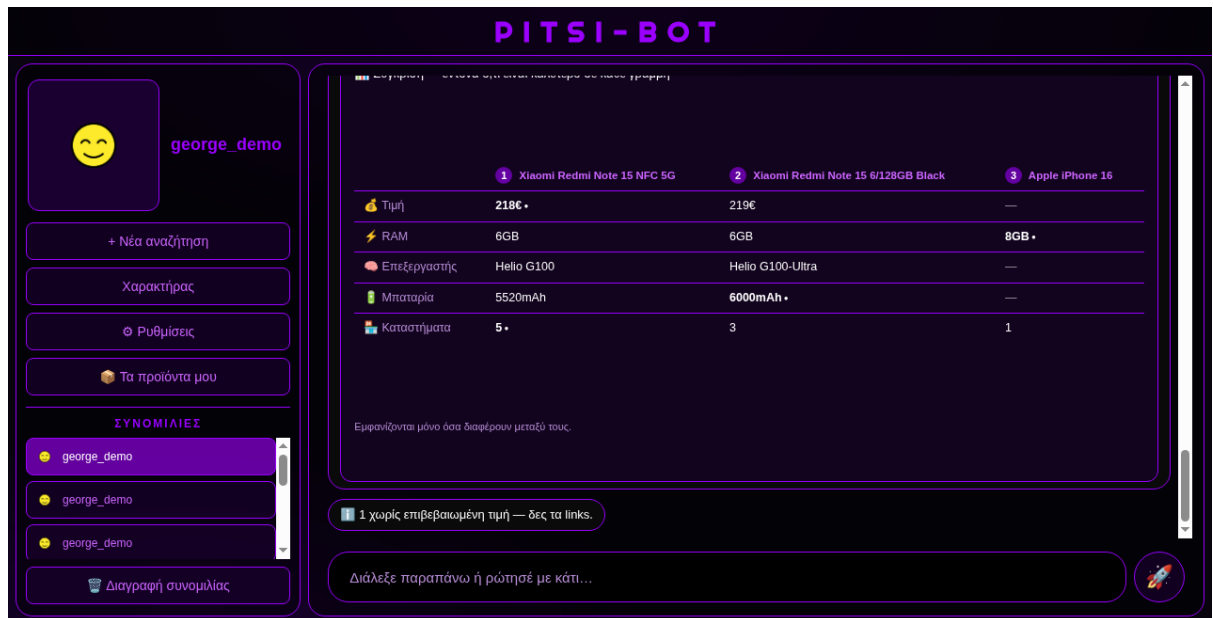


[Εικόνα 26] προτάσεις τέταρτου τεστ.

4.2.ε Τεστ 5. Σενάριο (γονιός για παιδί)



[Εικόνα 28] προτάσεις πέμπτου τεστ.



[Εικόνα 29] πλαίσιο σύγκρισης προτάσεων

4.3 Αποτελέσματα και συγκρίσεις

Τα αποτελέσματα δείχνουν πως η εφαρμογή λειτουργεί και τηρεί τους περισσότερους κανόνες που πρέπει να εφαρμόζονται. Επίσης φαίνεται η συνεργασία του Groq και του Serper να κυλάει ομαλά. Όπως ήταν αναμενόμενο όμως από την πολυπλοκότητα που προκύπτει από το εύρος επιλογών να δημιουργούνται κάποιες μικρές δυσλειτουργίες που χρειάζονται βελτίωση. Ακολουθεί αναλυτική εξήγηση αποτελεσμάτων.

4.3.α Τεστ 1 αποτελέσματα

Στο πρώτο πείραμα, στο offline 4.1.α, οι επιλογές που δόθηκαν ήταν σε αύξουσα σειρά και τηρείται το όριο που τέθηκε στο budget, όπως ήταν επιθυμητό. Επιπλέον ο πίνακας σύγκρισης φαίνεται πλήρης και κατατοπιστικός, όμως παρουσιάζει δύο σχεδόν ίδια προϊόντα, δηλαδή ίδιο μοντέλο με μεγαλύτερο αποθηκευτικό χώρο.

Στο ίδιο σενάριο, στα live τεστ 4.2.α, όμως διαφέρει κάτι άξιο παρατήρησης. Ενώ η αύξουσα σειρά λειτουργεί και σε αυτήν την περίπτωση, η τελευταία επιλογή ξεπερνά το όριο στο budget. Το γεγονός αυτό οφείλεται σε ελλείψεις οδηγίες στις υπηρεσίες αναζήτησης.

4.3.β Τεστ 2 αποτελέσματα

Στο δεύτερο πείραμα τα αποτελέσματα που προέκυψαν, τόσο από τα offline 4.1.β, όσο και από τα live πειράματα 4.2.β ήταν ακριβώς οι ίδιες επιλογές. Η αύξουσα σειρά ήταν και πάλι σωστά δομημένη, και ο πίνακας συγκρίσεων σχεδόν πλήρης. Όμως και στα δυο παρατηρήθηκε ένα καθαρό διπλότυπο, δηλαδή δυο επιλογές ήταν ακριβώς ίδιες. Πράγμα που προκύπτει πιθανά από την δύσκολη εξαγωγή πληροφοριών για τους φορητούς υπολογιστές, κάτι που παρατηρήθηκε γενικότερα κατά την διεξαγωγή των πειραμάτων.

4.3.γ Τεστ 3 αποτελέσματα

Στο τρίτο πείραμα τα αποτελέσματα που προέκυψαν, τόσο από τα offline, όσο και από τα live πειράματα ήταν ξανά κοινές επιλογές μεταξύ τους. Παρόλα αυτά οι προτάσεις που έκανε ήταν πολύ εύστοχες, δίνοντας σύγχρονα μοντέλα με αρκετές πληροφορίες στον πίνακα συγκρίσεων, χωρίς να χάσει σε κανένα το απαιτούμενο χαρακτηριστικό.

4.3.δ Τεστ 4 αποτελέσματα

Το τέταρτο πείραμα έφτασε και αυτό πολύ κοντά στο επιθυμητό αποτέλεσμα τόσο στα offline 4.1.δ, όσο και στα live 4.2.δ, δίνοντας διαφορετικά αποτελέσματα και όλα αρκετά εύστοχα χωρίς να χάσει πουθενά στο απαιτούμενο χαρακτηριστικό. Σημαντικό είναι επίσης να παρατηρηθεί ότι στα live αποτελέσματα, πρότεινε και μια επιλογή πολύ εύστοχη ως προς την επικαιρότητα, κάτι που έλειπε από τα offline αποτελέσματα.

4.3.ε Τεστ 5 αποτελέσματα

Στο τελικό από τα offline πειράματα 4.1.ε, το PitsiBot έκανε την δομή σωστά για ακόμα μια φορά, αλλά δυο αποτελέσματα ήταν το ίδιο μοντέλο σε διαφορετικό χρώμα. Το λάθος αυτό έχει παρατηρηθεί αρκετά κατά τον σχεδιασμό της εφαρμογής και αποτέλεσε μεγάλο εμπόδιο για την εξέλιξη της. Όπως φαίνεται από αυτήν την περίπτωση δεν έχει εξαλειφθεί εντελώς, καθώς ακόμα και οι μικρές διάφορες στο σύνολο των πληροφοριών είναι δύσκολο να μην το διαφοροποιήσουν. Στα live πειράματα 4.2.ε όμως παρατηρούμε τρεις σωστές

προτάσεις σωστά δομημένες με την τελική να ξεχωρίζει καθώς για ακόμα μια φορά συνάδει αρκετά με την επικαιρότητα.

4.3.στ Συνολική αποτίμηση αποτελεσμάτων

Συνολικά, τα πειράματα δείχνουν ότι το σύστημα μπορεί να παράγει σταθερά τρεις προτάσεις σωστά δομημένες τόσο στα offline όσο και στα live πειράματα. Φάνηκε να εντοπίζει πολύ καλά συγκεκριμένα χαρακτηριστικά και να προτείνει επιλογές αποκλειστικά και μόνο με αυτά. Στα live πειράματα επίσης παρατηρήθηκε, πως τείνει να τηρεί το κριτήριο της επικαιρότητας, πράγμα που διέφερε στα offline. Καταλαβαίνει κανείς λοιπόν μέσα από τα πειράματα ότι η εφαρμογή, αλλά και γενικότερα ένα chatbot που χρησιμοποιεί τεχνολογίες τεχνητής νοημοσύνης, χρειάζεται συνεχή τριβή, εκμάθηση και ενημέρωση. Απαραίτητο είναι επίσης το σωστό στήσιμο με prompt engineering και λίστες δεδομένων που να μπορεί να ανατρέξει ανά πάσα στιγμή. Ο συνδυασμός των δυο προσεγγίσεων που χρησιμοποιήθηκαν όμως έφερε τις τελικές προτάσεις πολύ κοντά στο επιθυμητό αποτέλεσμα.

ΚΕΦΑΛΑΙΟ 5 Συμπεράσματα

5.1 Δυσκολίες

Κατά την ανάπτυξη και την αξιολόγηση του PitsiBot παρουσιάστηκαν αρκετές δυσκολίες που επηρέασαν τόσο τον σχεδιασμό όσο και την ποιότητα των τελικών προτάσεων. Οι σημαντικότερες συνοψίζονται στα παρακάτω.

Η έλλειψη πλήρων και ομοιογενών πληροφοριών στην ελληνική αγορά, σε συνδυασμό με τις συχνές αποκλίσεις τιμών μεταξύ καταστημάτων, δυσκόλεψε σημαντικά την αξιόπιστη σύσταση. Το ίδιο μοντέλο εμφανιζόταν με διαφορετικά τεχνικά χαρακτηριστικά ή διαφορετική τιμή ανά πηγή, ενώ ορισμένα προϊόντα είχαν ελλιπή δεδομένα. Έτσι, πριν από κάθε πρόταση έπρεπε να γίνει διασταύρωση και φιλτράρισμα,

αλλιώς υπήρχε κίνδυνος να παρουσιαστεί λάθος τιμή ή μη διαθέσιμο προϊόν ως έγκυρη επιλογή. Επίσης η επιλογή έπρεπε να βασίζεται κυρίως στη διαθεσιμότητα και όχι σε δευτερεύοντα κριτήρια. Στην πράξη, μια πρόταση που ταίριαζε θεωρητικά στις ανάγκες του χρήστη αλλά ήταν εξαντλημένη δεν είχε πρακτική αξία. Η προτεραιότητα στη διαθεσιμότητα περιορίζει το σύνολο των επιλογών και συχνά οδηγούσε σε συμβιβασμούς προκειμένου να διατηρήσει η πρόταση την αξία της. Ο συντονισμός και η λειτουργία σε συνεργασία με εξωτερικά συστήματα, όπως μοντέλα γλώσσας, υπηρεσίες αναζήτησης, κατάλογοι και μηχανισμοί επαλήθευσης, αποδείχθηκε σύνθετος. Κάθε υποσύστημα έχει δική του καθυστέρηση, δικά του όρια και διαφορετική μορφή δεδομένων. Η ανάγκη για ροή παρόμοια με RAG, δηλαδή ανάκτηση σχετικών πληροφοριών από κατάλογο ή αναζήτηση και στη συνέχεια χρήση τους από το μοντέλο, προέκυψε ακριβώς επειδή το μοντέλο από μόνο του δεν γνωρίζει τις τρέχουσες τιμές και τη διαθεσιμότητα της ελληνικής αγοράς. Χωρίς αυτή τη σύνδεση, οι απαντήσεις θα παρέμεναν γενικές ή ανακριβείς. Έπειτα ο μεγάλος όγκος επιλογών προκαλούσε σύγχυση τόσο στη λογική επιλογής όσο και στο τελικό αποτέλεσμα που έβλεπε ο χρήστης. Όταν υπάρχουν δεκάδες ή εκατοντάδες παρόμοια μοντέλα, το σύστημα μπορεί εύκολα να προτείνει παραπλήσιες ή λιγότερο σχετικές επιλογές. Για τον λόγο αυτό κατέστη αναγκαία η συστηματική χρήση prompt engineering το οποίο περιείχε σαφείς οδηγίες, κριτήρια προτεραιότητας και δομημένη μορφή απάντησης, ώστε το μοντέλο να περιορίζεται σε λίγες, διαφοροποιημένες και αιτιολογημένες προτάσεις. Ένα ακόμα πρόβλημα που δημιούργησε ο μεγάλος όγκος πληροφοριών είναι η ανάγκη για συνειδητή θυσία της μεγάλης γκάμας προς όφελος εύστοχων προτάσεων. Αρχικά δηλαδή καταργήθηκαν κατηγορίες όπως εκτυπωτές και tablets και υστέρησε στα εργαλεία που παρέμεναν αντί να εμφανίζονται πολλές εναλλακτικές με μικρή διαφορά μεταξύ τους, επιλέχθηκε η παρουσίαση περιορισμένου αριθμού προτάσεων, με έμφαση στην ακρίβεια τιμής, στη διαθεσιμότητα και στην αντιστοίχιση στις ανάγκες. Η απόφαση αυτή βελτίωσε τη

χρηστικότητα, αλλά σήμαινε ότι ορισμένα αποδεκτά προϊόντα δεν εμφανίζονταν καν στον τελικό χρήστη.

Τέλος τα Live τεστ συχνά διαφωνούσαν με τα offline. Στα offline πειράματα οι πηγές ήταν ελεγχόμενες ή βασισμένες στον τοπικό κατάλογο, ενώ στα Live η αγορά άλλαζε σε πραγματικό χρόνο: τιμές, σύνδεσμοι και διαθεσιμότητα μεταβάλλονταν, και τα αποτελέσματα αναζήτησης περιλάμβαναν θόρυβο (αξεσουάρ, σελίδες κατηγορίας, παρωχημένα μοντέλα). Η διαφωνία αυτή δυσκόλεψε την αξιολόγηση, γιατί μια λύση που φαινόταν σωστή offline μπορούσε να αποτύχει ή να χρειαστεί πρόσθετους ελέγχους στο Live περιβάλλον. Για τον λόγο αυτό κρίθηκε απαραίτητο να εξεταστούν και οι δύο τρόποι δοκιμής και να συγκριθούν τα αποτελέσματά τους.

Συνολικά, οι παραπάνω δυσκολίες δεν ήταν μεμονωμένα τεχνικά προβλήματα, αλλά συνέπειες της προσπάθειας να συνδυαστεί τεχνητή νοημοσύνη με πραγματικά, μεταβλητά δεδομένα της ελληνικής αγοράς υπό περιορισμούς ακρίβειας και χρηστικότητας. Όπως και ο καθένας θα χρειάζεται εκπαίδευση για να μπορέσει να κάνει την σωστή έρευνα και επιλογή σε κάθε τομέα, έτσι και το PitsiBot χρειάστηκε ένα αυστηρό πλαίσιο κανόνων και μια σύνδεση και επικοινωνία μεταξύ των μέλλων του.

5.2 Συμπεράσματα

Η ανάπτυξη του PitsiBot έδειξε ότι η δημιουργία μιας εφαρμογής σύστασης προϊόντων δεν περιορίζεται μόνο στη χρήση ενός γλωσσικού μοντέλου. Χρειάζεται επίσης σωστή οργάνωση των δεδομένων, συχνό έλεγχος των αποτελεσμάτων και καλή επικοινωνία μεταξύ των διαφορετικών υπηρεσιών.

Αρχικά, διαπιστώθηκε ότι η εκπαίδευση ενός νέου μοντέλου ειδικά για την πρόταση προϊόντων δεν ήταν απαραίτητη για την παρούσα εργασία. Η διαδικασία θα απαιτούσε μεγάλο αριθμό δεδομένων από την ελληνική αγορά, αρκετό χρόνο και σημαντικούς υπολογιστικούς πόρους. Για τον λόγο αυτό, χρησιμοποιήθηκε ένα ήδη εκπαιδευμένο

γλωσσικό μοντέλο μέσω API, το Groq. Με αυτόν τον τρόπο, η εφαρμογή μπορούσε να επεξεργάζεται τα μηνύματα του χρήστη χωρίς να χρειάζεται εκπαίδευση ενός μοντέλου από την αρχή. Η συνομιλία αφορά την κατανόηση των αναγκών του χρήστη, τη συλλογή πληροφοριών και τη διατύπωση διευκρινιστικών ή ερωτήσεων. Η αναζήτηση αφορά την εύρεση προϊόντων, τον έλεγχο των τιμών, τη διαθεσιμότητα και την απομάκρυνση άσχετων αποτελεσμάτων. Οι δύο διαδικασίες πρέπει να συνεργάζονται, αλλά δεν πρέπει να συγχέονται. Αν η εφαρμογή συνομιλεί σωστά αλλά χρησιμοποιεί λανθασμένα δεδομένα, οι προτάσεις δεν είναι αξιόπιστες. Αντίστοιχα, αν η εφαρμογή βρίσκει σωστά προϊόντα αλλά δεν τα παρουσιάζει με κατανοητό τρόπο, η εμπειρία του χρήστη δεν είναι ικανοποιητική. Για τον λόγο αυτό, η τελική λειτουργία της εφαρμογής χωρίστηκε σε τρία βασικά στάδια τα οποία είναι η κατανόηση των αναγκών, η αναζήτηση και το φιλτράρισμα προϊόντων, και παρουσίαση των τελικών προτάσεων. Η χρήση γλωσσικού μοντέλου μαζί με εξωτερικές πηγές δεδομένων αποδείχθηκε χρήσιμη. Το μοντέλο μπορεί να κατανοεί το φυσικό κείμενο και να παρουσιάζει τις πληροφορίες με απλό τρόπο. Οι εξωτερικές πηγές, όπως το Serper και οι τοπικοί κατάλογοι, παρέχουν δεδομένα για τα προϊόντα. Με αυτόν τον συνδυασμό, το μοντέλο δεν βασίζεται μόνο στις γενικές γνώσεις του, αλλά χρησιμοποιεί και πληροφορίες που προέρχονται από την αγορά.

Τέλος, διαπιστώθηκε ότι η ποιότητα ενός βοηθού αγορών δεν εξαρτάται από τον αριθμό των προϊόντων που παρουσιάζει. Περισσότερη σημασία έχει οι προτάσεις να είναι σχετικές με τις ανάγκες του χρήστη, διαθέσιμες και εύκολες στην κατανόηση. Η περιορισμένη παρουσίαση επιλογών, ο διαχωρισμός της συνομιλίας από την αναζήτηση και ο συνδυασμός του γλωσσικού μοντέλου με πραγματικά δεδομένα αποτέλεσαν τα βασικά χαρακτηριστικά του PitsiBot.

5.3 Μελλοντικές βελτιώσεις

Υπάρχουν πολλοί τομείς στους οποίους θα μπορούσε να βελτιωθεί η εφαρμογή. Ένας από αυτούς είναι το εύρος των προϊόντων που καλύπτει καθώς τα κινητά και τα laptop δεν είναι τα μόνα που απαιτούν σωστή έρευνα και επιλογή από τον χρήστη. Κάποια επιπλέον προϊόντα θα μπορούσαν να είναι οι εκτυπωτές, οι οθόνες τηλεοράσεων και monitors αλλά και εργαλεία ηχογράφησης και βιντεοσκόπησης και πολλά αλλά. Η ευστοχία των προτάσεων χρειάζεται σίγουρα συνεχή επίβλεψη και βελτίωση. Επιπροσθέτως, θα μπορούσε να μετατραπεί το PitsiBot και σε application κινητής συσκευής για ακόμα πιο εύκολη πρόσβαση. Η χρήση περισσότερων και πιο ποιοτικών πόρων αναζήτησης και καλύτερος εξοπλισμός, όπως docking station δεδομένων και ξεχωριστών μηχανήματων για ερευνά , θα έπαιζε σίγουρα καθοριστικό ρόλο στην ανάπτυξη της εφαρμογής σε μεγαλύτερη κλίμακα, αλώστε όπως συμπεραίνεται από την παρούσα εργασία, τα εργαλεία με τα οποία υλοποιείται κάθε εργασία, κάθε τύπου, προσδίδουν όγκο και ταχύτητα εργασίας αλλά και ποιότητα.

Τέλος θα μπορούσε να προστεθεί ένα section επεξήγησης της απόφασης και τεχνολογικής ερμηνείας προς τον χρήστη έτσι ώστε να καταλαβαίνει την λογική και την ροή σκέψης στο πως επιλεγεί κανείς το εκάστοτε εργαλείο για μελλοντικές έρευνες που μπορεί να χρειαστεί να κάνει. Εκτός από αυτό, με ένα τμήμα τεχνικής υποστήριξης και υποβοήθησης για τον χρήστη αλλά και με ένα τμήμα ενδείξεων των ανάλογων αναλωσίμων, αξεσουάρ αλλά και καλωδίωσης για μέγιστη χρησιμότητα για τον χρήστη, θα μπορούσε όχι απλά να προσφέρει βοήθεια για το υπάρχον πρόβλημα αλλά και να το λύσει.

BIBΛΙΟΓΡΑΦΙΑ

- Adomavicius, G. and Tuzhilin, A., 2005 “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions”, *IEEE Transactions on Knowledge and Data Engineering*, 17(6), pp. 734–749.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S., 2020 “Language models are few-shot learners”, in *Advances in Neural Information Processing Systems*, 33, pp. 1877–1901.
- Burke, R. (2002) “Hybrid recommender systems: Survey and experiments”, *User Modeling and User-Adapted Interaction*, 12, pp. 331–370.
- Gao, C., Lei, W., He, X., de Rijke, M., & Chua, T.-S., 2021. “Advances and Challenges in Conversational Recommender Systems: A Survey.” *AI Open*, 2, 100–126.
- Ji, Z., et al. (2023). “Survey of Hallucination in Natural Language Generation.” *ACM Computing Surveys*, 55(12), Article 248, 1–38
- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G., 2023. “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing.” *ACM Computing Surveys*, 55(9), Article 195, 1–35.
- Liu, Z., Wang, H., Niu, Z.-Y., Wu, H., Che, W., & Liu, T., 2020. “Towards Conversational Recommendation over Multi-Type Dialogs.” In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1036–1049.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I., 2017. “Attention Is All You Need.” *Advances in Neural Information Processing Systems*, 30, 5998–6008.