

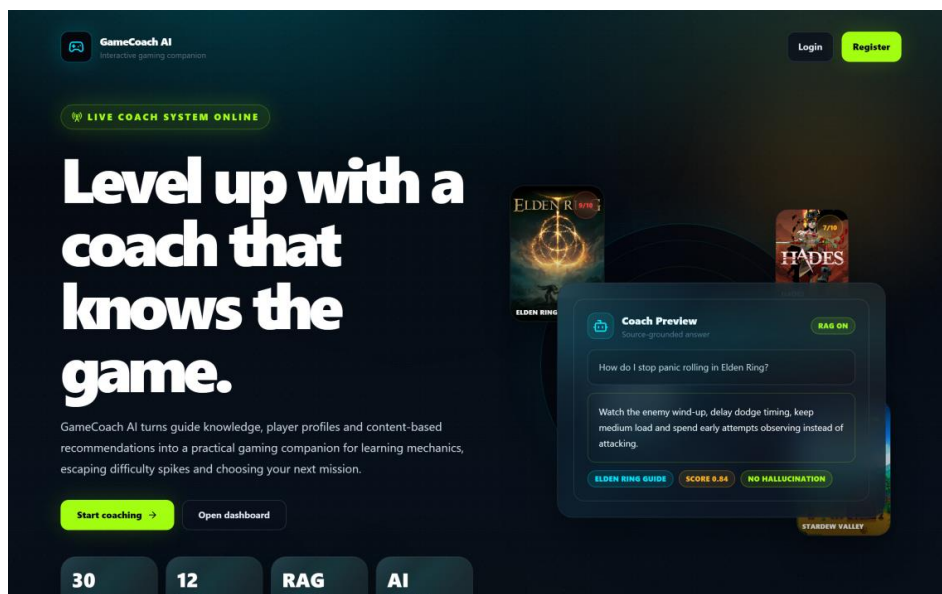
ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Σχεδίαση και Υλοποίηση Ευφυούς Συστήματος Υποστήριξης Παικτών
Βιντεοπαιχνιδιών με χρήση Μεγάλων Γλωσσικών Μοντέλων (LLMs) και
Τεχνικών Retrieval-Augmented Generation (RAG)»



Του φοιτητή
Αλέξανδρου Μίνγκα
Αρ. Μητρώου: 2019104

Επιβλέπουσα
Καλλιόπη Κράβαρη
Επίκουρη Καθηγήτρια

Ημερομηνία Αύγουστος 2026

Τίτλος Δ.Ε.: Σχεδίαση και Υλοποίηση Ευφυούς Συστήματος Υποστήριξης Παικτών Βιντεοπαιχνιδιών με χρήση Μεγάλων Γλωσσικών Μοντέλων (LLMs) και Τεχνικών Retrieval-Augmented Generation (RAG)

Κωδικός Δ.Ε.: 26181

Ονοματεπώνυμο φοιτητή: Αλέξανδρος Μίνγκας

Ονοματεπώνυμο εισηγητή: Καλλιόπη Κράβαρη

Ημερομηνία ανάληψης Δ.Ε.: 13 Ιουνίου 2026

Ημερομηνία περάτωσης Δ.Ε.: Αύγουστος 2026

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο του προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων της Σχολής Μηχανικών του Διεθνούς Πανεπιστημίου της Ελλάδος. Αντικείμενό της είναι η σχεδίαση, η υλοποίηση και η πειραματική αξιολόγηση ενός ολοκληρωμένου διαδικτυακού συστήματος υποστήριξης παικτών βιντεοπαιχνιδιών, του GameCoach AI, το οποίο αξιοποιεί Μεγάλα Γλωσσικά Μοντέλα (Large Language Models – LLMs) σε συνδυασμό με τεχνικές Ανάκτησης-Ενισχυμένης Παραγωγής (Retrieval-Augmented Generation – RAG) και τεχνικές συστάσεων βασισμένων στο περιεχόμενο (content-based recommendations).

Το κίνητρο για την επιλογή του θέματος προήλθε από τη ραγδαία διάδοση των μεγάλων γλωσσικών μοντέλων και το ανοικτό, πρακτικό πρόβλημα της αξιοπιστίας τους: τα μοντέλα αυτά παράγουν ευφραδείς απαντήσεις, όμως συχνά εμφανίζουν παραισθήσεις, δηλαδή επινοούν πληροφορίες που δεν τεκμηριώνονται από πηγές. Στο πεδίο της καθοδήγησης παικτών, όπου οι συμβουλές πρέπει να είναι συγκεκριμένες και σωστές (ονόματα ικανοτήτων, αριθμοί, μηχανικές παιχνιδιού), το πρόβλημα αυτό γίνεται άμεσα αντιληπτό από τον χρήστη. Η εργασία επιχειρεί να δώσει μια μετρήσιμη απάντηση, υιοθετώντας αυστηρή προσκόλληση των απαντήσεων σε επιμελημένη βάση γνώσης και τεκμηριωμένη άρνηση απάντησης όταν δεν υπάρχει επαρκές πλαίσιο.

Πέρα από το ερευνητικό σκέλος, η εργασία έχει έντονο αναπτυξιακό χαρακτήρα: υλοποιήθηκε πλήρες σύστημα πλήρους στοίβας (full-stack) με backend σε FastAPI/Python, frontend σε React/TypeScript, σχεσιακή βάση δεδομένων, διανυσματική βάση ChromaDB, ενσωμάτωση με το OpenAI API, μηχανή συστάσεων, πίνακα διαχείρισης και πλήρη σουίτα αυτοματοποιημένης αξιολόγησης. Το σύστημα λειτουργεί πλήρως και χωρίς σύνδεση σε εξωτερικά API, μέσω ντετερμινιστικών εναλλακτικών διαδρομών, γεγονός που διασφαλίζει την αναπαραγωγιμότητα των πειραμάτων και της επίδειξης.

Η εργασία απευθύνεται σε αναγνώστες με βασικές γνώσεις πληροφορικής. Οι απαραίτητες έννοιες για τα γλωσσικά μοντέλα, την ανάκτηση πληροφορίας και τα συστήματα συστάσεων παρουσιάζονται αναλυτικά στο θεωρητικό μέρος, ώστε το κείμενο να είναι αυτοτελές.

Περίληψη

Η παρούσα διπλωματική εργασία παρουσιάζει τη σχεδίαση, την υλοποίηση και την αξιολόγηση του GameCoach AI, ενός διαδικτυακού ευφυσούς συστήματος υποστήριξης παικτών βιντεοπαιχνιδιών. Το σύστημα συνδυάζει τρεις πυλώνες: (α) έναν συνομιλιακό «προπονητή» βασισμένο σε Μεγάλα Γλωσσικά Μοντέλα με αυστηρή τεχνική Retrieval-Augmented Generation (RAG), ο οποίος απαντά αποκλειστικά βασιζόμενος σε επιμελημένη βάση γνώσης 45 οδηγών παιχνιδιών και παραθέτει τις πηγές κάθε απάντησης, (β) μια μηχανή συστάσεων βασισμένων στο περιεχόμενο, η οποία προτείνει επόμενα παιχνίδια με παιδαγωγική λογική κλιμάκωσης δυσκολίας (παιχνίδια-γέφυρες, επόμενες προκλήσεις, παιχνίδια εξάσκησης δεξιοτήτων) και (γ) μια πλατφόρμα διαχείρισης προσωπικής βιβλιοθήκης, προφίλ παίκτη και περιεχομένου.

Το backend υλοποιήθηκε σε Python με FastAPI, SQLAlchemy και ChromaDB, με αυθεντικοποίηση JWT, κρυπτογράφηση κωδικών bcrypt, περιορισμό ρυθμού αιτημάτων και προστασία από επιθέσεις SSRF. Το frontend υλοποιήθηκε σε React 18 με TypeScript και Tailwind CSS. Ο μηχανισμός RAG τεμαχίζει τους οδηγούς σε αποσπάσματα 900 χαρακτήρων με επικάλυψη 160, τα ενσωματώνει είτε με ντετερμινιστικό τοπικό μετασχηματισμό hashing είτε με το μοντέλο text-embedding-3-small της OpenAI, και εφαρμόζει αυστηρά φίλτρα συνάφειας ώστε ερωτήσεις εκτός πεδίου να απορρίπτονται ρητά αντί να απαντώνται αυθαίρετα.

Η πειραματική αξιολόγηση περιλαμβάνει μετρικές ανάκτησης (Recall@1, Recall@3, MRR), ποσοστά ορθής και εσφαλμένης απόρριψης, σύγκριση με απλό LLM χωρίς ανάκτηση, ανάλυση ευαισθησίας κατωφλίου και μεγέθους τεμαχίων, πίνακα κάλυψης 30 παιχνιδιών και αξιολόγηση της μηχανής συστάσεων σε 18 σενάρια. Στο εσωτερικό σύνολο αξιολόγησης το σύστημα επιτυγχάνει Recall@1 63,6%, Recall@3 72,7% και MRR 0,684, με 100% ορθή απόρριψη ερωτήσεων εκτός πεδίου και 0% εσφαλμένες απορρίψεις, ενώ η μηχανή συστάσεων επιτυγχάνει 100% ορθό τύπο σύστασης και ορθή κατεύθυνση δυσκολίας στα 16 βασικά σενάρια. Η εργασία ολοκληρώνεται με κριτική συζήτηση των περιορισμών και προτάσεις μελλοντικών επεκτάσεων.

Λέξεις-κλειδιά: Μεγάλα Γλωσσικά Μοντέλα, Retrieval-Augmented Generation, ChromaDB, διανυσματικές ενσωματώσεις, συστήματα συστάσεων, FastAPI, React, παραισθήσεις LLM, αξιολόγηση ανάκτησης πληροφορίας

Abstract

This thesis presents the design, implementation and evaluation of GameCoach AI, a web-based intelligent support system for video-game players. The system combines three pillars: (a) a conversational coach based on Large Language Models with a strict Retrieval-Augmented Generation (RAG) pipeline, which answers exclusively from a curated knowledge base of 45 game guides and cites the sources of every answer, (b) a content-based recommendation engine that proposes next games following a pedagogical difficulty-scaling logic (bridge games, next challenges, skill builders), and (c) a platform for managing a personal game library, player profile and content.

The backend is implemented in Python with FastAPI, SQLAlchemy and ChromaDB, featuring JWT authentication, bcrypt password hashing, request rate limiting and SSRF protection. The frontend is built with React 18, TypeScript and Tailwind CSS. The RAG mechanism splits guides into chunks of 900 characters with an overlap of 160, embeds them either with a deterministic local hashing transform or with OpenAI's text-embedding-3-small model, and applies strict relevance filters so that out-of-scope questions are explicitly rejected instead of being answered arbitrarily.

The experimental evaluation includes retrieval metrics (Recall@1, Recall@3, MRR), correct- and false-rejection rates, a comparison against a plain LLM baseline without retrieval, sensitivity analysis of the relevance threshold and chunk size, a 30-game chat coverage matrix and an 18-scenario evaluation of the recommendation engine. On the internal evaluation set the system achieves Recall@1 of 63.6%, Recall@3 of 72.7% and MRR of 0.684, with 100% correct rejection of out-of-scope questions and 0% false rejections, while the recommendation engine achieves 100% correct recommendation type and difficulty direction on the 16 core scenarios. The thesis concludes with a critical discussion of limitations and directions for future work.

Keywords: Large Language Models, Retrieval-Augmented Generation, ChromaDB, vector embeddings, recommender systems, FastAPI, React, LLM hallucinations, information-retrieval evaluation

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά την επιβλέπουσα καθηγήτρια μου, Καλλιόπη Κράβαρη , για την καθοδήγηση, τις εύστοχες παρατηρήσεις και τη διαθεσιμότητά της σε όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας.

Ευχαριστώ επίσης το σύνολο των διδασκόντων του Τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων για τις γνώσεις που μου προσέφεραν κατά τη διάρκεια των σπουδών μου, καθώς και την οικογένειά μου και τους φίλους μου για τη στήριξη και την υπομονή τους.

Περιεχόμενα

Πρόλογος.....	3
Περίληψη	4
Abstract.....	5
Ευχαριστίες	6
Περιεχόμενα	7
Ευρετήριο Σχημάτων	10
Ευρετήριο Πινάκων	11
1. Εισαγωγή.....	13
1.1 Πλαίσιο και κίνητρο	13
1.2 Ορισμός του προβλήματος.....	13
1.3 Στόχοι και συνεισφορά	14
1.4 Παραδοτέα	14
1.5 Μεθοδολογία εκπόνησης	15
1.6 Δομή της εργασίας	15
2. Θεωρητικό υπόβαθρο.....	17
2.1 Μεγάλα Γλωσσικά Μοντέλα (LLMs)	17
2.1.1 Από τα στατιστικά γλωσσικά μοντέλα στους <i>Transformers</i>	17
2.1.2 Προτροπές και έλεγχος συμπεριφοράς	17
2.1.3 Διακριτοποίηση, παράθυρο συμφραζομένων και κόστος.....	18
2.2 Περιορισμοί των LLMs και το πρόβλημα των παραισθήσεων	18
2.3 Retrieval-Augmented Generation (RAG)	19
2.3.1 Αρχή λειτουργίας	19
2.3.2 Τεμαχισμός κειμένου (<i>chunking</i>)	19
2.3.3 Αυστηρό RAG και συμπεριφορά άρνησης.....	19
2.3.4 Αξιολόγηση συστημάτων RAG.....	19
2.4 Διανυσματικές ενσωματώσεις και σημασιολογική αναζήτηση	20
2.4.1 Ενσωματώσεις κειμένου	20
2.4.2 Διανυσματικές βάσεις δεδομένων	20
2.5 Συστήματα συστάσεων.....	21
2.5.1 Κατηγορίες συστημάτων συστάσεων	21
2.5.2 Μέτρα ομοιότητας χαρακτηριστικών	21
2.5.3 Η δυσκολία ως παιδαγωγική διάσταση	21

2.5.4 Αξιολόγηση και εξηγησιμότητα συστάσεων	21
2.6 Τεχνολογίες υλοποίησης.....	22
2.6.1 Τεχνολογίες <i>backend</i>	22
2.6.2 Τεχνολογίες <i>frontend</i>	22
2.6.3 Εικονικοποίηση και μεταφερσιμότητα	22
2.7 Συναφείς εργασίες και συστήματα	22
3. Ανάλυση απαιτήσεων και σχεδίαση συστήματος	27
3.1 Λειτουργικές απαιτήσεις.....	27
3.2 Μη λειτουργικές απαιτήσεις	28
3.3 Ρόλοι και περιπτώσεις χρήσης	28
3.4 Συνολική αρχιτεκτονική	29
3.5 Μοντέλο δεδομένων	30
3.6 Σχεδίαση του REST API	31
3.7 Σχεδίαση ασφάλειας.....	31
3.8 Σχεδίαση αξιοπιστίας και λειτουργίας εκτός σύνδεσης.....	32
4. Υλοποίηση του συστήματος	35
4.1 Οργάνωση του κώδικα	35
4.2 Πυρήνας <i>backend</i> : ρυθμίσεις, βάση δεδομένων, ασφάλεια	35
4.3 Βάση γνώσης και σπορά δεδομένων	37
4.4 Υποσύστημα RAG	38
4.4.1 Ευρετηρίαση.....	39
4.4.2 Ανάκτηση και αυστηρά φίλτρα.....	41
4.5 Υπηρεσία γλωσσικού μοντέλου	42
4.6 Μηχανή συστάσεων	44
4.7 REST API και ροή συνομιλίας	47
4.8 Διεπαφή χρήστη	48
4.8.1 Αρχική σελίδα και αυθεντικοποίηση	49
4.8.2 Πίνακας ελέγχου και βιβλιοθήκη	51
4.8.3 Συνομιλία με τον προπονητή	52
4.8.4 Συστάσεις παιχνιδιών	53
4.8.5 Ρυθμίσεις και πίνακας διαχείρισης	55
4.8.6 Διαφάνεια, κενές καταστάσεις και ανθεκτικότητα της διεπαφής	57
4.9 Εγκατάσταση και εκτέλεση με Docker.....	58

5. Αξιολόγηση και αποτελέσματα	63
5.1 Μεθοδολογία και σύνολα δεδομένων.....	63
5.2 Μετρικές αξιολόγησης	63
5.3 Αποτελέσματα ποιότητας ανάκτησης.....	64
5.4 Γενίκευση σε πραγματικές ερωτήσεις παικτών	65
5.5 Σύγκριση με baseline και ανάλυση ευαισθησίας	65
5.5.1 Πλήρης αγωγή RAG έναντι σκέτου LLM	65
5.5.2 Ευαισθησία στο κατώφλι συνάφειας και στο μέγεθος τεμαχίου	66
5.5.3 Λεξιλογικές έναντι σημασιολογικών ενσωματώσεων	67
5.6 Κάλυψη πλήρους καταλόγου μέσω του API συνομιλίας	67
5.7 Αξιολόγηση της μηχανής συστάσεων	68
5.8 Δοκιμές λογισμικού και ετοιμότητα επίδειξης.....	68
5.9 Συζήτηση και περιορισμοί της αξιολόγησης	69
6. Συμπεράσματα και μελλοντικές επεκτάσεις.....	73
6.1 Σύνοψη της εργασίας.....	73
6.2 Συμπεράσματα ανά ερευνητικό ερώτημα	73
6.3 Περιορισμοί	74
6.4 Μελλοντικές επεκτάσεις.....	74
6.5 Τελική αποτίμηση	75
Αναφορές – Βιβλιογραφία	77
Παράρτημα Α – Οδηγός εγκατάστασης και χρήσης λογισμικού	79
Α.1 Προαπαιτούμενα	79
Α.2 Εγκατάσταση backend.....	79
Α.3 Εγκατάσταση frontend.....	79
Α.4 Εκτέλεση με Docker	79
Α.5 Εκτέλεση δοκιμών και αξιολόγησης	79
Παράρτημα Β – Πλήρης κατάλογος τελικών σημείων του API.....	80
Παράρτημα Γ – Κατάλογος παιχνιδιών της πλατφόρμας	81
Παράρτημα Δ – Δομή αποθετηρίου κώδικα.....	81
Παράρτημα Ε – Μεταβλητές περιβάλλοντος	82

Ευρετήριο Σχημάτων

Σχήμα 3.1 – Συνολική αρχιτεκτονική του συστήματος GameCoach AI.....	29
Σχήμα 3.2 – Μοντέλο δεδομένων του συστήματος (κύριες οντότητες και συσχετίσεις).....	30
Σχήμα 4.1 – Ροή επεξεργασίας ερώτησης στον αυστηρό αγωγό RAG και ροή ευρετηρίασης	39
Σχήμα 4.2 – Ροή της μηχανής συστάσεων βασισμένων στο περιεχόμενο	46
Σχήμα 4.3 – Αυτόματη διαδραστική τεκμηρίωση του API (Swagger UI / OpenAPI).....	48
Σχήμα 4.4 – Δημόσια αρχική σελίδα της εφαρμογής	49
Σχήμα 4.5 – Φόρμα εγγραφής με πεδία προφίλ παίκτη	50
Σχήμα 4.6 – Οθόνη σύνδεσης χρήστη	50
Σχήμα 4.7 – Πίνακας ελέγχου χρήστη με στατιστικά και πρόσφατη δραστηριότητα	51
Σχήμα 4.8 – Προσωπική βιβλιοθήκη παιχνιδιών με φίλτρα και κάρτες τίτλων	52
Σχήμα 4.9 – Τεκμηριωμένη απάντηση του προπονητή για το Elden Ring με σήμανση προέλευσης ...	53
Σχήμα 4.10 – Πλακίδια πηγών με βαθμούς συνάφειας στο τέλος της απάντησης	53
Σχήμα 4.11 – Φόρμα αιτήματος συστάσεων με πλακίδια προτιμήσεων.....	54
Σχήμα 4.12 – Αποτέλεσμα συστάσεων με μονοπάτι προόδου	55
Σχήμα 4.13 – Οθόνη ρυθμίσεων προφίλ παίκτη.....	56
Σχήμα 4.14 – Πίνακας διαχείρισης με στατιστικά πλατφόρμας.....	57
Σχήμα 4.15 – Φόρμα προσθήκης εγγράφου γνώσης με άμεση ανακατασκευή του ευρετηρίου RAG ..	57
Σχήμα 4.16 – Κενή κατάσταση συνομιλίας με επιλογή λειτουργίας και προτεινόμενες ερωτήσεις	58
Σχήμα 5.1 – Ευαισθησία των μετρικών στο κατώφλι συνάφειας: σταθερή περιοχή έως 0,3 και κατάρρευση της ανάκλησης από 0,5.....	66
Σχήμα 5.2 – Σύγκριση λεξιλογικής (hashing) και σημασιολογικής (OpenAI) ανάκτησης.....	67

Ευρετήριο Πινάκων

Πίνακας 2.1 – Σύγκριση αντιπροσωπευτικών λύσεων αποθήκευσης διανυσμάτων.....	20
Πίνακας 3.1 – Κυριότερες λειτουργικές απαιτήσεις του συστήματος.....	27
Πίνακας 3.2 – Μη λειτουργικές απαιτήσεις του συστήματος	28
Πίνακας 3.3 – Κύριες περιπτώσεις χρήσης του συστήματος	28
Πίνακας 3.4 – Θεματικές ενότητες του REST API	31
Πίνακας 4.1 – Τυπική δομή ενοτήτων ενός οδηγού της βάσης γνώσης	37
Πίνακας 4.2 – Κύριοι τρόποι (modes) παραγωγής απάντησης της υπηρεσίας LLM.....	43
Πίνακας 5.1 – Διαμόρφωση συστήματος κατά την αξιολόγηση ανάκτησης	63
Πίνακας 5.2 – Συγκεντρωτικές μετρικές ανάκτησης στο εσωτερικό σύνολο αξιολόγησης.....	64
Πίνακας 5.3 – Ανάκτηση εντός πεδίου ανά δυσκολία διατύπωσης.....	64
Πίνακας 5.4 – Εσωτερικό έναντι εξωτερικού συνόλου: μεροληψία ευθυγράμμισης λεξιλογίου.....	65
Πίνακας 5.5 – Συμπεριφορική σύγκριση σκέτου LLM και αγωγού RAG (40 ερωτήσεις).....	65
Πίνακας 5.6 – Σάρωση κατωφλίου συνάφειας (λεξιλογική διαμόρφωση, chunk 900).....	66
Πίνακας 5.7 – Συγκεντρωτικά αποτελέσματα αξιολόγησης της μηχανής συστάσεων (18 σενάρια)	68
Πίνακας 5.8 – Χρόνοι εκτέλεσης κρίσιμων λειτουργιών (λειτουργία εκτός σύνδεσης).....	69
Πίνακας 6.1 – Αποτίμηση επίτευξης των στόχων της εργασίας.....	73
Πίνακας Β.1 – Τελικά σημεία REST API ανά ενότητα	80
Πίνακας Γ.1 – Τα 30 παιχνίδια του καταλόγου με βαθμό δυσκολίας.....	81
Πίνακας Ε.1 – Κυριότερες μεταβλητές περιβάλλοντος του backend	82

1. Εισαγωγή

Το παρόν κεφάλαιο εισάγει τον αναγνώστη στο αντικείμενο της διπλωματικής εργασίας. Παρουσιάζονται το ευρύτερο πλαίσιο και το κίνητρο, ο ορισμός του προβλήματος, οι στόχοι και η συνεισφορά της εργασίας, τα παραδοτέα, η μεθοδολογία που ακολουθήθηκε και, τέλος, η δομή των κεφαλαίων που ακολουθούν.

1.1 Πλαίσιο και κίνητρο

Η βιομηχανία των βιντεοπαιχνιδιών αποτελεί σήμερα έναν από τους μεγαλύτερους κλάδους ψηφιακής ψυχαγωγίας παγκοσμίως, με εκατοντάδες εκατομμύρια ενεργούς παίκτες και διαρκώς αυξανόμενη πολυπλοκότητα τίτλων. Σύγχρονα παιχνίδια όπως το Elden Ring, το Baldur's Gate 3 ή το Counter-Strike 2 απαιτούν από τον παίκτη να κατανοήσει πλήθος αλληλοεξαρτώμενων μηχανισμών: διαχείριση πόρων, χρονισμό ενεργειών, τοποθέτηση στον χώρο, οικονομία γύρων και μακροπρόθεσμο σχεδιασμό. Οι νέοι παίκτες συχνά εγκαταλείπουν έναν τίτλο όχι επειδή δεν τους αρέσει, αλλά επειδή η καμπύλη εκμάθησής του είναι απότομη και η διαθέσιμη βοήθεια κατακερματισμένη σε φόρουμ, βίντεο και οδηγούς αμφίβολης ποιότητας.

Παράλληλα, τα Μεγάλα Γλωσσικά Μοντέλα (Large Language Models – LLMs) έχουν μετασχηματίσει ριζικά τον τρόπο με τον οποίο οι χρήστες αναζητούν πληροφορία. Ένα LLM μπορεί να απαντήσει σε ερώτηση φυσικής γλώσσας άμεσα και με συνοχή, χωρίς ο χρήστης να χρειάζεται να περιηγηθεί σε δεκάδες πηγές. Ωστόσο, τα μοντέλα αυτά παρουσιάζουν ένα θεμελιώδες πρόβλημα αξιοπιστίας: τις λεγόμενες «παραισθήσεις» (hallucinations), δηλαδή την παραγωγή ευλογοφανών αλλά ανακριβών ή επινοημένων πληροφοριών. Στο πεδίο της καθοδήγησης παικτών το πρόβλημα είναι ιδιαίτερα ορατό: ένα μοντέλο μπορεί να επινοήσει όνομα ικανότητας, να μπερδέψει τις μηχανικές δύο διαφορετικών παιχνιδιών ή να δώσει συμβουλές για έκδοση του παιχνιδιού που δεν ισχύει πλέον.

Η τεχνική Retrieval-Augmented Generation (RAG) αποτελεί μια από τις πλέον αποτελεσματικές απαντήσεις στο πρόβλημα αυτό. Αντί το μοντέλο να απαντά αποκλειστικά από τις παραμέτρους του, ένα στάδιο ανάκτησης εντοπίζει πρώτα τα σχετικά αποσπάσματα από μια ελεγχόμενη βάση γνώσης και το μοντέλο καλείται να συνθέσει την απάντηση αποκλειστικά από αυτά, παραθέτοντας τις πηγές του. Η παρούσα εργασία εφαρμόζει την προσέγγιση αυτή στο πεδίο της καθοδήγησης παικτών, με έμφαση όχι μόνο στην ποιότητα των απαντήσεων αλλά και στη μετρήσιμη συμπεριφορά άρνησης: όταν το σύστημα δεν διαθέτει τεκμηριωμένο υλικό για μια ερώτηση, οφείλει να το δηλώνει ρητά αντί να παράγει αστήρικτο περιεχόμενο.

1.2 Ορισμός του προβλήματος

Το πρόβλημα που καλείται να επιλύσει η εργασία μπορεί να διατυπωθεί ως εξής: «Σχεδίαση και υλοποίηση ενός διαδικτυακού συστήματος που υποστηρίζει τον παίκτη σε όλο τον κύκλο ενασχόλησής του με τα βιντεοπαιχνίδια — από την επιλογή του κατάλληλου επόμενου τίτλου έως τη στοχευμένη βελτίωση των δεξιοτήτων του — με απαντήσεις τεκμηριωμένες σε ελεγχόμενες πηγές και με ποσοτικά αξιολογήσιμη αξιοπιστία». Το πρόβλημα αναλύεται σε τρία επιμέρους ερωτήματα:

- **E1 — Τεκμηριωμένη συνομιλιακή καθοδήγηση:** πώς μπορεί ένας συνομιλιακός βοηθός να απαντά σε ερωτήσεις στρατηγικής και μηχανικών παιχνιδιών αποκλειστικά από επιμελημένη βάση γνώσης, παραθέτοντας πηγές και αρνούμενος ρητά τις ερωτήσεις εκτός πεδίου;

- **E2 — Συστάσεις παιχνιδιών:** πώς μπορεί μια μηχανή συστάσεων να προτείνει το «επόμενο παιχνίδι» λαμβάνοντας υπόψη όχι μόνο την ομοιότητα περιεχομένου αλλά και τη σχέση του παίκτη με τη δυσκολία (πολύ εύκολο / ισορροπημένο / πολύ δύσκολο);
- **E3 — Μετρήσιμη αξιοπιστία:** με ποιες μετρικές και ποια πειραματική μεθοδολογία μπορεί να ποσοτικοποιηθεί τόσο η ποιότητα ανάκτησης όσο και η συμπεριφορά άρνησης του συστήματος, ώστε τα αποτελέσματα να είναι αναπαραγώγιμα;

1.3 Στόχοι και συνεισφορά

Ο κύριος στόχος της εργασίας είναι η ανάπτυξη ενός πλήρους, λειτουργικού και αξιολογημένου συστήματος και όχι απλώς μιας πρωτότυπης απόδειξης εφικτότητας (proof of concept). Οι επιμέρους στόχοι είναι:

1. Η υλοποίηση αυστηρού αγωγού RAG (strict RAG pipeline) με ρητούς μηχανισμούς προστασίας: ελάχιστο κατώφλι συνάφειας, έλεγχο λεξιλογικής επικάλυψης, φίλτρο ανά παιχνίδι και ανίχνευση αναφοράς σε παιχνίδι εκτός καταλόγου.
2. Η δημιουργία επιμελημένης βάσης γνώσης 45 οδηγών σε μορφή Markdown για κατάλογο 30 παιχνιδιών, με ενιαία δομή ενοτήτων και καταγεγραμμένες πηγές.
3. Η υλοποίηση μηχανής συστάσεων βασισμένων στο περιεχόμενο με σταθμισμένη βαθμολόγηση χαρακτηριστικών και τρεις παιδαγωγικούς τύπους πρότασης (bridge game, next challenge, skill builder).
4. Η ανάπτυξη σύγχρονης διεπαφής χρήστη με ολοκληρωμένες ροές: εγγραφή/σύνδεση, προσωπική βιβλιοθήκη, συνομιλία με πηγές, συστάσεις με εξηγήσεις, πίνακας διαχείρισης.
5. Η θωράκιση του συστήματος με πρακτικές ασφάλειας παραγωγής: JWT, bcrypt, περιορισμός ρυθμού, έλεγχος πρόσβασης βάσει ρόλων, προστασία SSRF στην εισαγωγή περιεχομένου.
6. Η συστηματική πειραματική αξιολόγηση με τυπικές μετρικές ανάκτησης πληροφορίας, σύγκριση με baseline, ανάλυση ευαισθησίας και αξιολόγηση σεναρίων σύστασης.

Η συνεισφορά της εργασίας συνοψίζεται στον συνδυασμό: ένα πλήρως αναπαραγώγιμο, ανοικτό σύστημα αναφοράς για αυστηρό RAG σε εξειδικευμένο πεδίο, με τεκμηριωμένη ποσοτική ανάλυση του συμβιβασμού μεταξύ ανάκλησης (recall) και ορθής άρνησης (correct rejection), καθώς και με αξιολόγηση της γενίκευσης σε πραγματικές ερωτήσεις παικτών συλλεγμένες από το διαδίκτυο.

1.4 Παραδοτέα

Τα παραδοτέα της διπλωματικής εργασίας είναι:

- Ο πλήρης πηγαίος κώδικας του συστήματος GameCoach AI (backend FastAPI, frontend React, σενάρια αρχικοποίησης και αξιολόγησης, ρυθμίσεις Docker).
- Η βάση γνώσης των 45 οδηγών παιχνιδιών και ο κατάλογος 30 παιχνιδιών με πλήρη μεταδεδομένα.
- Η σουίτα αυτοματοποιημένης αξιολόγησης (σύνολα δεδομένων, σενάρια, σενάρια ελέγχου ορίων) και τα παραγόμενα αποτελέσματα.
- Το παρόν κείμενο της διπλωματικής εργασίας, με πλήρη τεχνική τεκμηρίωση, στιγμιότυπα οθόνης και ανάλυση αποτελεσμάτων.

1.5 Μεθοδολογία εκπόνησης

Η εκπόνηση ακολούθησε επαναληπτική (iterative) μεθοδολογία ανάπτυξης λογισμικού σε διακριτές φάσεις. Στην πρώτη φάση πραγματοποιήθηκε βιβλιογραφική επισκόπηση των LLMs, της τεχνικής RAG και των συστημάτων συστάσεων, και καταγράφηκαν οι απαιτήσεις. Στη δεύτερη φάση σχεδιάστηκε η αρχιτεκτονική και το μοντέλο δεδομένων και υλοποιήθηκε ο πυρήνας του συστήματος. Στην τρίτη φάση αναπτύχθηκε η βάση γνώσης και ο μηχανισμός RAG, με έμφαση στους μηχανισμούς αυστηρής προσκόλλησης στις πηγές. Στην τέταρτη φάση υλοποιήθηκε η μηχανή συστάσεων και ολοκληρώθηκε η διεπαφή χρήστη. Στην πέμπτη φάση εκτελέστηκαν κύκλοι αξιολόγησης και βελτίωσης: κάθε αλλαγή στον μηχανισμό ανάκτησης επανελεγχόταν με τα ίδια σύνολα δεδομένων, ώστε να τεκμηριώνεται ποσοτικά η επίδρασή της. Τέλος, το σύστημα θωρακίστηκε με ελέγχους ασφάλειας και καταρτίστηκε το παρόν κείμενο.

Σε επίπεδο εργαλείων, η ανάπτυξη έγινε σε περιβάλλον Windows με Python 3.13 και Node.js 20, με έλεγχο εκδόσεων Git και τακτικές, μικρές υποβολές ανά λειτουργία. Κάθε φάση έκλεινε με εκτέλεση της πλήρους σουίτας δοκιμών και των σεναρίων αξιολόγησης, ώστε παλινδρομήσεις να εντοπίζονται άμεσα· τα αποτελέσματα κάθε κύκλου αποθηκεύονταν ως αρχεία Markdown στο αποθετήριο, ώστε η εξέλιξη των μετρικών να είναι ιχνηλατήσιμη στο ιστορικό εκδόσεων.

1.6 Δομή της εργασίας

Η συνέχεια της εργασίας οργανώνεται ως εξής:

- **Κεφάλαιο 2 — Θεωρητικό υπόβαθρο:** παρουσιάζει τα Μεγάλα Γλωσσικά Μοντέλα, τους περιορισμούς τους, την τεχνική RAG, τις διανυσματικές ενσωματώσεις και βάσεις, τα συστήματα συστάσεων και τις τεχνολογίες υλοποίησης.
- **Κεφάλαιο 3 — Ανάλυση και σχεδίαση:** καταγράφει τις λειτουργικές και μη λειτουργικές απαιτήσεις, την αρχιτεκτονική, το μοντέλο δεδομένων, τη σχεδίαση του API και τη σχεδίαση ασφάλειας.
- **Κεφάλαιο 4 — Υλοποίηση:** περιγράφει αναλυτικά την υλοποίηση κάθε υποσυστήματος — βάση γνώσης, αγωγός RAG, υπηρεσία LLM, μηχανή συστάσεων, REST API, διεπαφή χρήστη, εγκατάσταση με Docker — με στιγμιότυπα οθόνης της εφαρμογής.
- **Κεφάλαιο 5 — Αξιολόγηση και αποτελέσματα:** παρουσιάζει τη μεθοδολογία αξιολόγησης, τα σύνολα δεδομένων, τις μετρικές και τα πλήρη πειραματικά αποτελέσματα με πίνακες και διαγράμματα.
- **Κεφάλαιο 6 — Συμπεράσματα:** συνοψίζει τα ευρήματα, αποτιμά την επίτευξη των στόχων, καταγράφει τους περιορισμούς και προτείνει μελλοντικές επεκτάσεις.

Ακολουθούν οι βιβλιογραφικές αναφορές κατά το πρότυπο IEEE και παραρτήματα με τον οδηγό εγκατάστασης και χρήσης του λογισμικού, τον πλήρη πίνακα των τελικών σημείων του API, τον κατάλογο παιχνιδιών και τη δομή του αποθετηρίου κώδικα.

Κλείνοντας το εισαγωγικό κεφάλαιο, σημειώνεται ότι το σύστημα σχεδιάστηκε εξ αρχής ώστε να λειτουργεί πλήρως και χωρίς πρόσβαση σε εξωτερικές υπηρεσίες: όλες οι λειτουργίες διαθέτουν ντετερμινιστικές εναλλακτικές διαδρομές, γεγονός που καθιστά τα πειράματα του Κεφαλαίου 5 πλήρως αναπαραγώγιμα σε οποιονδήποτε υπολογιστή.

Οριοθετείται επίσης ρητά το πεδίο εφαρμογής της εργασίας, ώστε οι στόχοι της ενότητας 1.3 να αξιολογούνται με σαφή κριτήρια. Η εργασία δεν επιχειρεί να υλοποιήσει τηλεμετρία μέσα από το ίδιο το παιχνίδι (in-game telemetry) ούτε ζωντανή ανάλυση της επίδοσης του παίκτη σε πραγματικό χρόνο

--- η καθοδήγηση βασίζεται αποκλειστικά σε δηλωμένη εμπειρία δυσκολίας και ερωτήσεις φυσικής γλώσσας, όχι σε δεδομένα αισθητήρων ή στατιστικά αγώνων. Ομοίως, η μηχανή συστάσεων ακολουθεί αμιγώς προσέγγιση βασισμένη στο περιεχόμενο (content-based) και δεν περιλαμβάνει συνεργατικό φιλτράρισμα (collaborative filtering) μεταξύ χρηστών --- επιλογή συνεπής με το μέγεθος της αρχικής βάσης χρηστών, όπου δεν θα υπήρχε επαρκής όγκος αλληλεπιδράσεων για αξιόπιστα συνεργατικά μοτίβα, όπως συζητείται περαιτέρω στην ενότητα 6.4. Τέλος, η βάση γνώσης περιορίζεται σε αγγλόφωνο περιεχόμενο και στα 30 παιχνίδια του καταλόγου· η επέκταση σε πολυγλωσσική υποστήριξη και μεγαλύτερο κατάλογο τίτλων παραμένει ρητή κατεύθυνση μελλοντικής εργασίας και όχι στόχος της παρούσας υλοποίησης. Η οριοθέτηση αυτή δεν αποτελεί αδυναμία αλλά συνειδητή απόφαση εστίασης: προτιμήθηκε η ενδελεχής, μετρήσιμη υλοποίηση ενός καλά ορισμένου πυρήνα λειτουργικότητας από μια ευρύτερη αλλά λιγότερο τεκμηριωμένη κάλυψη.

Η επιλογή του συγκεκριμένου πεδίου εφαρμογής --- η καθοδήγηση παικτών βιντεοπαιχνιδιών, αντί ενός πιο συμβατικού πεδίου εξειδικευμένης γνώσης όπως η ιατρική ή η νομική τεκμηρίωση --- δεν είναι άσχετη με τους στόχους μεθοδολογικής αυστηρότητας της εργασίας. Τα βιντεοπαιχνίδια προσφέρουν ένα σπάνιο πλεονέκτημα για ερευνητική αξιολόγηση: αντικειμενικά επαληθεύσιμα γεγονότα (μηχανικές, στατιστικά, στρατηγικές) που μπορούν να επιβεβαιωθούν άμεσα από τον ίδιο τον συγγραφέα μέσω πραγματικής εμπειρίας παιχνιδιού, σε αντίθεση με πεδία όπου η επαλήθευση απαιτεί εξειδικευμένη επαγγελματική κρίση τρίτων. Η ιδιότητα αυτή κατέστησε εφικτή τη σύνταξη ενός επισημειωμένου εσωτερικού συνόλου αξιολόγησης υψηλής ακρίβειας χωρίς εξωτερική εμπειρογνωμοσύνη, διατηρώντας παράλληλα πλήρη επιστημονική εγκυρότητα μέσω της σύγκρισης με το ανεξάρτητο εξωτερικό σύνολο πραγματικών ερωτήσεων της ενότητας 5.4.

2. Θεωρητικό υπόβαθρο

Το κεφάλαιο αυτό παρουσιάζει το θεωρητικό πλαίσιο στο οποίο εντάσσεται η εργασία. Εξετάζονται διαδοχικά τα Μεγάλα Γλωσσικά Μοντέλα και η αρχιτεκτονική Transformer, οι εγγενείς περιορισμοί των μοντέλων αυτών, η τεχνική Retrieval-Augmented Generation, οι διανυσματικές ενσωματώσεις κειμένου και οι διανυσματικές βάσεις δεδομένων, τα συστήματα συστάσεων και, τέλος, οι βασικές τεχνολογίες υλοποίησης που χρησιμοποιήθηκαν. Το κεφάλαιο κλείνει με επισκόπηση συναφών εργασιών και συστημάτων.

2.1 Μεγάλα Γλωσσικά Μοντέλα (LLMs)

2.1.1 Από τα στατιστικά γλωσσικά μοντέλα στους Transformers

Ως γλωσσικό μοντέλο ορίζεται ένα μοντέλο που αποδίδει πιθανότητες σε ακολουθίες λέξεων ή, ισοδύναμα, προβλέπει το επόμενο στοιχείο (token) μιας ακολουθίας δεδομένου του προηγούμενου πλαισίου. Τα πρώτα στατιστικά μοντέλα n-γραμμάτων περιορίζονταν σε μικρά παράθυρα συμφραζομένων, ενώ τα αναδρομικά νευρωνικά δίκτυα (RNN, LSTM) βελτίωσαν τη μοντελοποίηση μακρών εξαρτήσεων αλλά παρέμεναν δύσκολα στην παράλληλη εκπαίδευση. Η καθοριστική τομή ήρθε το 2017 με την αρχιτεκτονική Transformer [1], η οποία αντικατέστησε την αναδρομή με τον μηχανισμό αυτοπροσοχής (self-attention): κάθε token υπολογίζει σταθμισμένες συσχετίσεις με όλα τα υπόλοιπα tokens της ακολουθίας, επιτρέποντας πλήρη παραλληλοποίηση της εκπαίδευσης και αποτελεσματική σύλληψη μακροπρόθεσμων εξαρτήσεων.

Πάνω στην αρχιτεκτονική αυτή οικοδομήθηκαν τα σύγχρονα LLMs. Η οικογένεια GPT κατέδειξε ότι η κλιμάκωση του πλήθους παραμέτρων και των δεδομένων προεκπαίδευσης οδηγεί σε αναδυόμενες ικανότητες μάθησης λίγων παραδειγμάτων (few-shot learning) [2]. Η μετέπειτα ευθυγράμμιση των μοντέλων με ανθρώπινες προτιμήσεις μέσω Reinforcement Learning from Human Feedback (RLHF) [3] παρήγαγε μοντέλα-βοηθούς ικανά να ακολουθούν οδηγίες φυσικής γλώσσας, όπως αυτά που αξιοποιούνται στην παρούσα εργασία μέσω του OpenAI API (μοντέλο gpt-4o-mini).

Ο πυρήνας του μηχανισμού αυτοπροσοχής μπορεί να διατυπωθεί συμπαγώς: για πίνακες ερωτημάτων Q , κλειδιών K και τιμών V που προκύπτουν από γραμμικούς μετασχηματισμούς των αναπαραστάσεων εισόδου, η έξοδος υπολογίζεται ως:

$$\text{Attention}(Q, K, V) = \text{softmax}(Q \cdot K^T / \sqrt{d_k}) \cdot V \quad (2.1)$$

όπου d_k η διάσταση των κλειδιών· η διαίρεση με $\sqrt{d_k}$ σταθεροποιεί τις κλίσεις κατά την εκπαίδευση. Πολλαπλές «κεφαλές» προσοχής (multi-head attention) εκτελούν τον υπολογισμό παράλληλα σε διαφορετικούς υποχώρους, επιτρέποντας στο μοντέλο να συλλαμβάνει ταυτόχρονα συντακτικές και σημασιολογικές σχέσεις διαφορετικής εμβέλειας.

2.1.2 Προτροπές και έλεγχος συμπεριφοράς

Η συμπεριφορά ενός LLM κατά τον χρόνο εκτέλεσης ελέγχεται κυρίως μέσω της προτροπής (prompt): του κειμένου συστήματος που ορίζει ρόλο και κανόνες, των μηνυμάτων χρήστη και των παραμέτρων δειγματοληψίας. Η παράμετρος θερμοκρασίας (temperature) ρυθμίζει την τυχαιότητα της δειγματοληψίας: χαμηλές τιμές παράγουν πιο ντετερμινιστικές και συντηρητικές απαντήσεις, κατάλληλες για εφαρμογές τεκμηρίωσης, ενώ υψηλές τιμές ευνοούν τη δημιουργικότητα. Στο

GameCoach AI χρησιμοποιείται χαμηλή θερμοκρασία (0,25) για τις τεκμηριωμένες απαντήσεις, ακριβώς επειδή προτεραιότητα έχει η ακρίβεια και όχι η ποικιλία.

2.1.3 Διακριτοποίηση, παράθυρο συμφραζομένων και κόστος

Τα LLMs δεν επεξεργάζονται χαρακτήρες ή λέξεις αλλά tokens: υπολέξεις που παράγονται από αλγορίθμους διακριτοποίησης τύπου byte-pair encoding. Η διάκριση αυτή έχει πρακτικές συνέπειες για κάθε εφαρμογή: το μέγεθος του παραθύρου συμφραζομένων (context window) — το μέγιστο πλήθος tokens που μπορεί να επεξεργαστεί το μοντέλο σε μία κλήση — περιορίζει πόσο ανακτημένο υλικό μπορεί να συνοδεύσει μια ερώτηση, ενώ η τιμολόγηση των εμπορικών API γίνεται ανά token εισόδου και εξόδου. Ένα σύστημα RAG πρέπει συνεπώς να επιλέγει προσεκτικά πόσα και πόσο μεγάλα αποσπάσματα θα συμπεριλάβει: η υπερβολή αυξάνει κόστος και «θόρυβο» στο πλαίσιο, ενώ η φειδώ στερεί από το μοντέλο κρίσιμη πληροφορία. Στο GameCoach AI η επιλογή $k=4$ τεμαχίων των ~900 χαρακτήρων κρατά το πλαίσιο κάθε κλήσης σε λίγες χιλιάδες tokens, εξισορροπώντας κάλυψη και κόστος.

Αξίζει τέλος να σημειωθεί η διάκριση μεταξύ μοντέλων συνομιλίας και μοντέλων ενσωματώσεων: τα πρώτα παράγουν κείμενο, τα δεύτερα μόνο διανυσματικές αναπαραστάσεις. Ένα πλήρες σύστημα RAG χρησιμοποιεί και τα δύο — στην παρούσα εργασία το gpt-4o-mini για τη σύνθεση απαντήσεων και το text-embedding-3-small για την ανάκτηση — και πρέπει να διαχειρίζεται ανεξάρτητα τη διαθεσιμότητα και τα σφάλματα καθενός.

2.2 Περιορισμοί των LLMs και το πρόβλημα των παραισθήσεων

Παρά τις εντυπωσιακές επιδόσεις τους, τα LLMs εμφανίζουν συστηματικούς περιορισμούς που καθιστούν προβληματική την αυτούσια χρήση τους σε εφαρμογές που απαιτούν αξιοπιστία [4]:

- **Παραισθήσεις (hallucinations):** το μοντέλο παράγει περιεχόμενο που δεν τεκμηριώνεται από τα δεδομένα εκπαίδευσης ή το παρεχόμενο πλαίσιο, με ύψος απόλυτης βεβαιότητας. Στο πεδίο των παιχνιδιών αυτό μεταφράζεται σε επινοημένα [ονόματα αντικειμένων, εσφαλμένους αριθμούς ή ανάμειξη μηχανικών διαφορετικών τίτλων.
- **Χρονικό όριο γνώσης (knowledge cutoff):** το μοντέλο αγνοεί εξελίξεις μεταγενέστερες της εκπαίδευσής του — κρίσιμο σε παιχνίδια που αναθεωρούνται διαρκώς με ενημερώσεις εξισορρόπησης.
- **Αδυναμία παράθεσης πηγών:** ένα αμιγές LLM δεν μπορεί να τεκμηριώσει από πού προέρχεται μια πληροφορία, ώστε ο χρήστης να την επαληθεύσει.
- **Κόστος και ιδιωτικότητα:** η κλήση μεγάλων μοντέλων έχει χρηματικό κόστος ανά token και συνεπάγεται αποστολή δεδομένων σε εξωτερικό πάροχο.

Στη βιβλιογραφία έχουν προταθεί τρεις κύριες οικογένειες στρατηγικών μετριασμού. Η πρώτη είναι η προσαρμογή του ίδιου του μοντέλου (fine-tuning) σε δεδομένα του πεδίου· απαιτεί όμως σημαντικούς πόρους, επαναλαμβάνεται σε κάθε ενημέρωση της γνώσης και δεν προσφέρει παράθεση πηγών. Η δεύτερη είναι η ενίσχυση της προτροπής με ανακτημένο πλαίσιο — η προσέγγιση RAG — που κρατά τη γνώση εκτός μοντέλου, επικαιροποιείται με απλή επανευρετηρίαση και επιτρέπει τεκμηρίωση. Η τρίτη είναι οι μηχανισμοί προστασίας εκτός μοντέλου (guardrails): κανόνες και φίλτρα που ελέγχουν εισόδους και εξόδους, όπως κατώφλια συνάφειας ή έλεγχοι θεματικής εμβέλειας. Οι στρατηγικές αυτές δεν είναι αμοιβαία αποκλειόμενες· η παρούσα εργασία συνδυάζει τη δεύτερη με την τρίτη, καθώς η ανάκτηση από μόνη της δεν εγγυάται άρνηση σε ερωτήσεις που η βάση γνώσης δεν καλύπτει.

Οι περιορισμοί αυτοί υπαγορεύουν τη βασική σχεδιαστική αρχή της εργασίας: το μοντέλο δεν είναι η πηγή της γνώσης αλλά μηχανισμός γλωσσικής σύνθεσης πάνω σε ελεγχόμενη, ανακτημένη γνώση, με ρητή δυνατότητα άρνησης όταν η γνώση αυτή δεν επαρκεί.

2.3 Retrieval-Augmented Generation (RAG)

2.3.1 Αρχή λειτουργίας

Η τεχνική Retrieval-Augmented Generation προτάθηκε από τους Lewis κ.ά. [5] ως συνδυασμός παραμετρικής μνήμης (τα βάρη ενός γλωσσικού μοντέλου) και μη παραμετρικής μνήμης (ένα ευρετήριο κειμένων με πυκνή διανυσματική αναπαράσταση). Σε έναν τυπικό αγωγό RAG, η επεξεργασία μιας ερώτησης περιλαμβάνει τα ακόλουθα στάδια: (α) η ερώτηση μετατρέπεται σε διάνυσμα με ένα μοντέλο ενσωματώσεων, (β) αναζητούνται στο ευρετήριο τα k πλησιέστερα αποσπάσματα κειμένου (top- k retrieval), (γ) τα αποσπάσματα ενσωματώνονται στην προτροπή του γλωσσικού μοντέλου ως πλαίσιο, και (δ) το μοντέλο συνθέτει την απάντηση με βάση το πλαίσιο αυτό. Η ανάκτηση πυκνών αναπαραστάσεων (dense retrieval) έχει αποδειχθεί αποτελεσματικότερη της κλασικής λεξιλογικής αναζήτησης σε ερωτήσεις ανοικτού πεδίου [6], αν και οι δύο προσεγγίσεις συχνά συνδυάζονται σε υβριδικά σχήματα.

2.3.2 Τεμαχισμός κειμένου (chunking)

Καθοριστική παράμετρος ενός συστήματος RAG είναι ο τεμαχισμός των εγγράφων σε αποσπάσματα (chunks). Πολύ μικρά τεμάχια χάνουν το πλαίσιο και κατακερματίζουν τη σημασιολογία· πολύ μεγάλα τεμάχια μειώνουν τη διακριτική ικανότητα της ανάκτησης και σπαταλούν το περιορισμένο παράθυρο συμφοραζομένων του μοντέλου. Η επικάλυψη μεταξύ διαδοχικών τεμαχίων μετριάζει το πρόβλημα της διάσπασης προτάσεων στα όρια. Στην παρούσα εργασία χρησιμοποιείται μέγεθος τεμαχίου 900 χαρακτήρων με επικάλυψη 160, τιμές που επιλέχθηκαν και επιβεβαιώθηκαν πειραματικά με ανάλυση ευαισθησίας (ενότητα 5.5).

2.3.3 Αυστηρό RAG και συμπεριφορά άρνησης

Η πλειονότητα των συστημάτων RAG εστιάζει στη βελτίωση της ανάκλησης· λιγότερη προσοχή δίνεται στη συμπεριφορά του συστήματος όταν η βάση γνώσης δεν καλύπτει την ερώτηση. Χωρίς πρόσθετα μέτρα, ο μηχανισμός ανάκτησης θα επιστρέψει «τα λιγότερο άσχετα» αποσπάσματα και το μοντέλο θα παραγάγει απάντηση από ακατάλληλο πλαίσιο. Η παρούσα εργασία υιοθετεί την προσέγγιση του αυστηρού RAG (strict RAG): κατώφλι ελάχιστης συνάφειας, απαίτηση ελάχιστης λεξιλογικής επικάλυψης ερωτήματος-τεμαχίου, φιλτράρισμα ανά παιχνίδι και ρητή απόκριση «ανεπαρκούς πλαισίου» (insufficient context). Η συμπεριφορά αυτή ποσοτικοποιείται με τις μετρικές ορθής και εσφαλμένης απόρριψης του Κεφαλαίου 5.

2.3.4 Αξιολόγηση συστημάτων RAG

Η αξιολόγηση ενός συστήματος RAG διακρίνεται μεθοδολογικά σε δύο επίπεδα. Το πρώτο αφορά το στάδιο ανάκτησης και χρησιμοποιεί τις κλασικές μετρικές της ανάκτησης πληροφορίας — Recall@ k , ακρίβεια, μέσο αντίστροφο σημείο κατάταξης (MRR) [19] — πάνω σε επισημειωμένα ζεύγη ερώτησης-αναμενόμενου εγγράφου. Το δεύτερο αφορά το τελικό κείμενο και απαιτεί είτε ανθρώπινη βαθμολόγηση είτε αυτόματες προσεγγίσεις τύπου «LLM-κριτής», με γνωστά ζητήματα αξιοπιστίας. Λιγότερο τυποποιημένη, αλλά κρίσιμη για συστήματα με μηχανισμούς προστασίας, είναι η

αξιολόγηση της συμπεριφοράς άρνησης: απαιτεί σκόπιμα σχεδιασμένες ερωτήσεις εκτός πεδίου, συμπεριλαμβανομένων «δύσκολων αρνητικών» — ερωτήσεων θεματικά συγγενών με το σώμα κειμένων που όμως δεν καλύπτονται από αυτό. Η παρούσα εργασία υιοθετεί το πρώτο και το τρίτο επίπεδο με πλήρως αυτοματοποιημένα, αναπαραγώγιμα πειράματα (Κεφάλαιο 5), αφήνοντας την ανθρώπινη βαθμολόγηση του τελικού κειμένου ως μελλοντική επέκταση.

2.4 Διανυσματικές ενσωματώσεις και σημασιολογική αναζήτηση

2.4.1 Ενσωματώσεις κειμένου

Ενσωμάτωση (embedding) ονομάζεται η απεικόνιση ενός κειμένου σε πυκνό διάνυσμα πραγματικών αριθμών σταθερής διάστασης, έτσι ώστε κείμενα με συναφές νόημα να αντιστοιχούν σε γειτονικά διανύσματα του χώρου. Τα σύγχρονα μοντέλα ενσωματώσεων, όπως το text-embedding-3-small της OpenAI [7] που χρησιμοποιείται στην εργασία, εκπαιδεύονται με αντιθετικές μεθόδους ώστε η ομοιότητα των διανυσμάτων να αντανakλά σημασιολογική συγγένεια και όχι απλή λεξιλογική σύμπτωση. Η ομοιότητα δύο διανυσμάτων q και d μετριέται τυπικά με το συνημίτονο της μεταξύ τους γωνίας:

$$\cos(q, d) = (q \cdot d) / (\|q\| \cdot \|d\|) \quad (2.2)$$

όπου $q \cdot d$ το εσωτερικό γινόμενο και $\|\cdot\|$ η ευκλείδεια νόρμα. Τιμές κοντά στη μονάδα δηλώνουν υψηλή σημασιολογική συγγένεια. Για κανονικοποιημένα διανύσματα, η ομοιότητα συνημιτόνου ισοδυναμεί με το εσωτερικό γινόμενο, γεγονός που αξιοποιείται για αποδοτικούς υπολογισμούς.

2.4.2 Διανυσματικές βάσεις δεδομένων

Η αναζήτηση πλησιέστερων γειτόνων σε μεγάλα σύνολα διανυσμάτων απαιτεί εξειδικευμένες δομές ευρετηρίασης, όπως οι γράφοι HNSW (Hierarchical Navigable Small World), που προσφέρουν προσεγγιστική αναζήτηση σε υπογραμμικό χρόνο. Οι διανυσματικές βάσεις δεδομένων ενσωματώνουν τις δομές αυτές μαζί με αποθήκευση μεταδεδομένων και δυνατότητα φιλτραρίσματος. Στην εργασία χρησιμοποιείται η ChromaDB [8], μια ενσωματώσιμη (embedded) διανυσματική βάση ανοικτού κώδικα για Python, με μόνιμη αποθήκευση στον δίσκο, υποστήριξη χώρου συνημιτόνου και φίλτρα μεταδεδομένων — χαρακτηριστικά που αξιοποιούνται για το φιλτράρισμα ανά παιχνίδι.

Ο Πίνακας 2.1 συγκρίνει αντιπροσωπευτικές επιλογές αποθήκευσης διανυσμάτων ως προς τα κριτήρια που καθόρισαν την επιλογή της παρούσας εργασίας: απλότητα ενσωμάτωσης σε εφαρμογή Python, μόνιμη τοπική αποθήκευση χωρίς εξωτερική υπηρεσία και υποστήριξη φίλτρων μεταδεδομένων.

Πίνακας 2.1 – Σύγκριση αντιπροσωπευτικών λύσεων αποθήκευσης διανυσμάτων

Λύση	Τύπος	Φίλτρα μεταδεδομένων	Καταλληλότητα για το έργο
ChromaDB	Ενσωματώσιμη βιβλιοθήκη Python	Ναι	Άριστη: μηδενική υποδομή, μόνιμη τοπική αποθήκευση
FAISS	Βιβλιοθήκη ευρετηρίων	Όχι (εξωτερικά)	Ταχύτατη, αλλά χωρίς αποθήκευση/φίλτρα out-of-the-box
pgvector	Επέκταση PostgreSQL	Ναι (SQL)	Καλή σε στοίβες με PostgreSQL· περιττή πολυπλοκότητα τοπικά

Pinecone / Weaviate	Διαχειριζόμενη υπηρεσία	Ναι	Κλιμάκωση παραγωγής· αντίκειται στη λειτουργία εκτός σύνδεσης
---------------------	-------------------------	-----	---

2.5 Συστήματα συστάσεων

2.5.1 Κατηγορίες συστημάτων συστάσεων

Τα συστήματα συστάσεων διακρίνονται παραδοσιακά σε δύο μεγάλες οικογένειες [9]: τα συστήματα συνεργατικού φιλτραρίσματος (collaborative filtering), που αξιοποιούν τα μοτίβα προτιμήσεων πλήθους χρηστών, και τα συστήματα βασισμένα στο περιεχόμενο (content-based), που συγκρίνουν τα χαρακτηριστικά των αντικειμένων με το προφίλ του χρήστη [10]. Το συνεργατικό φιλτράρισμα πάσχει από το πρόβλημα ψυχρής εκκίνησης (cold start): χωρίς ιστορικό αλληλεπιδράσεων πολλών χρηστών δεν μπορεί να παράγει αξιόπιστες προτάσεις. Σε ένα ακαδημαϊκό σύστημα ενός/λίγων χρηστών, όπως το παρόν, η προσέγγιση βάσει περιεχομένου είναι μονόδρομος και επιπλέον προσφέρει εγγενή εξηγησιμότητα: κάθε πρόταση μπορεί να αιτιολογηθεί από τα κοινά χαρακτηριστικά.

2.5.2 Μέτρα ομοιότητας χαρακτηριστικών

Για σύνολα κατηγορικών χαρακτηριστικών (π.χ. μηχανικές παιχνιδιού, δεξιότητες), κλασικό μέτρο ομοιότητας αποτελεί ο δείκτης Jaccard, ο λόγος της τομής προς την ένωση δύο συνόλων:

$$J(A, B) = |A \cap B| / |A \cup B| \quad (2.3)$$

Ο δείκτης λαμβάνει τιμές στο $[0, 1]$ και ισούται με 1 όταν τα σύνολα ταυτίζονται. Στη μηχανή συστάσεων της εργασίας ο δείκτης Jaccard εφαρμόζεται στα σύνολα μηχανικών και δεξιοτήτων, ενώ συνδυάζεται με όρους καταλληλότητας δυσκολίας σε σταθμισμένο άθροισμα (ενότητα 4.6).

2.5.3 Η δυσκολία ως παιδαγωγική διάσταση

Ιδιαιτερότητα του πεδίου των βιντεοπαιχνιδιών είναι ότι η «καταλληλότητα» μιας σύστασης δεν εξαρτάται μόνο από την ομοιότητα περιεχομένου αλλά και από τη σχέση του παίκτη με τη δυσκολία, κατά αναλογία με τη ζώνη επικείμενης ανάπτυξης της παιδαγωγικής: ένα παιχνίδι πολύ ευκολότερο από το επίπεδο του παίκτη είναι βαρετό, ενώ ένα πολύ δυσκολότερο αποθαρρυντικό. Η μηχανή του GameCoach AI ενσωματώνει τη διάσταση αυτή ρητά, μέσω τριών τύπων σύστασης: παιχνίδια-γέφυρες (bridge games) για όσους δυσκολεύονται, επόμενες προκλήσεις (next challenges) για όσους βαριούνται και παιχνίδια εξάσκησης δεξιοτήτων (skill builders) για ισορροπημένη εξέλιξη.

2.5.4 Αξιολόγηση και εξηγησιμότητα συστάσεων

Η αξιολόγηση συστημάτων συστάσεων παραδοσιακά βασίζεται σε ιστορικά δεδομένα αλληλεπιδράσεων (διασταυρούμενη επικύρωση σε βαθμολογίες ή κλικ). Σε συστήματα χωρίς τέτοιο ιστορικό, εναλλακτική αποτελεί η αξιολόγηση βάσει σεναρίων: ορίζονται καταστάσεις χρήστη με γνωστή «ορθή» συμπεριφορά και ελέγχεται αν το σύστημα τη παράγει. Η προσέγγιση αυτή ταιριάζει ιδιαίτερα σε παιδαγωγικά κριτήρια — π.χ. «σε παίκτη που δυσκολεύεται πρότείνει κάτι ευκολότερο» — που δεν αποτυπώνονται σε μετρικές ακρίβειας. Παράλληλα, η εξηγησιμότητα αναγνωρίζεται ολοένα και περισσότερο ως αυτόνομη ποιότητα: μια σύσταση που αιτιολογείται («προτείνεται επειδή μοιράζεται τις μηχανικές X και είναι δύο βαθμίδες ευκολότερο») γίνεται περισσότερο αποδεκτή και επιτρέπει στον χρήστη να διορθώσει το προφίλ του. Η μηχανή του Κεφαλαίου 4 σχεδιάστηκε με

αμφότερες τις αρχές: αξιολογείται με 18 σενάρια δηλωμένης ορθής συμπεριφοράς και παράγει εξήγηση για κάθε πρόταση.

2.6 Τεχνολογίες υλοποίησης

2.6.1 Τεχνολογίες *backend*

Το FastAPI [11] είναι σύγχρονο πλαίσιο ανάπτυξης διαδικτυακών API σε Python, θεμελιωμένο στα πρότυπα ASGI και OpenAPI. Προσφέρει δηλωτικό ορισμό τελικών σημείων με επικύρωση τύπων μέσω Pydantic, αυτόματη παραγωγή διαδραστικής τεκμηρίωσης (Swagger UI) και υψηλές επιδόσεις. Η πρόσβαση στη βάση δεδομένων γίνεται μέσω του SQLAlchemy [12], του καθιερωμένου αντικειμενο-σχεσιακού χαρτογράφου (ORM) της Python, ο οποίος επιτρέπει εναλλαγή μεταξύ SQLite (τοπική ανάπτυξη) και PostgreSQL (παραγωγή) χωρίς αλλαγές στον κώδικα. Η αυθεντικοποίηση βασίζεται στο πρότυπο JSON Web Token (JWT) [13], ενώ οι κωδικοί πρόσβασης αποθηκεύονται αποκλειστικά ως αποτυπώματα bcrypt [14], αλγορίθμου σχεδιασμένου με ρυθμιζόμενο υπολογιστικό κόστος ώστε να αντιστέκεται σε επιθέσεις εξαντλητικής δοκιμής.

2.6.2 Τεχνολογίες *frontend*

Η διεπαφή χρήστη υλοποιήθηκε ως εφαρμογή μίας σελίδας (Single-Page Application) με τη βιβλιοθήκη React 18 [15] και γλώσσα TypeScript, η οποία προσθέτει στατικό έλεγχο τύπων στην JavaScript και μειώνει κατηγορίες σφαλμάτων ήδη κατά τη μεταγλώττιση. Η δρομολόγηση γίνεται με το React Router, οι κλήσεις HTTP με τη βιβλιοθήκη Axios και η μορφοποίηση με το Tailwind CSS [16], ένα utility-first πλαίσιο που επιτρέπει συνεπή σχεδίαση χωρίς χειρόγραφα φύλλα στυλ. Η μεταγλώττιση και ο τοπικός εξυπηρετητής ανάπτυξης παρέχονται από το εργαλείο Vite.

2.6.3 Εικονικοποίηση και μεταφερσιμότητα

Για την αναπαραγώγιμη εγκατάσταση του συστήματος χρησιμοποιείται το Docker [17] και το Docker Compose: τρεις υπηρεσίες (backend, frontend, PostgreSQL) ορίζονται δηλωτικά και εκκινούν με μία εντολή, με υγειονομικούς ελέγχους και μόνιμους τόμους δεδομένων. Η ίδια εικόνα εκτελείται πανομοιότυπα σε οποιοδήποτε λειτουργικό σύστημα, εξαλείφοντας το πρόβλημα «στον υπολογιστή μου δουλεύει».

2.7 Συναφείς εργασίες και συστήματα

Η ιδέα της υποβοήθησης παικτών με τεχνητή νοημοσύνη δεν είναι νέα: εμπορικά προϊόντα προσφέρουν στατιστική ανάλυση αγώνων και προτάσεις βελτίωσης σε συγκεκριμένους ανταγωνιστικούς τίτλους, βασισμένα κυρίως σε τηλεμετρία. Τα εργαλεία αυτά όμως είναι κλειστά, περιορισμένα σε έναν τίτλο και δεν προσφέρουν συνομιλική αλληλεπίδραση. Από την άλλη πλευρά, οι γενικοί συνομιλιακοί βοηθοί (ChatGPT και συναφή) απαντούν μεν σε ερωτήσεις για οποιοδήποτε παιχνίδι, αλλά χωρίς εγγυήσεις τεκμηρίωσης, χωρίς παράθεση πηγών και με τεκμηριωμένο κίνδυνο παραισθήσεων [4], [18].

Στη βιβλιογραφία, οι επισκοπήσεις της τεχνικής RAG [18] καταγράφουν πλούσια οικογένεια παραλλαγών (naive, advanced, modular RAG) και αναδεικνύουν ως ανοικτά ζητήματα την αξιολόγηση της συμπεριφοράς άρνησης και την ευθυγράμμιση λεξιλογίου μεταξύ ερωτήσεων χρηστών και σώματος κειμένων — ακριβώς τα ζητήματα που η παρούσα εργασία μετρά πειραματικά στο εξειδικευμένο πεδίο των βιντεοπαιχνιδιών. Η συμβολή της εργασίας σε σχέση με τα παραπάνω

είναι ο συνδυασμός: ανοικτό, πλήρες σύστημα πολλαπλών τίτλων, με αυστηρό RAG, μετρήσιμη συμπεριφορά άρνησης, παιδαγωγική μηχανή συστάσεων και δημοσιευμένη σουίτα αξιολόγησης.

2.8 Ζητήματα αξιοπιστίας και αξιολόγησης συστημάτων RAG σε εξειδικευμένα πεδία

Η αξιολόγηση συστημάτων RAG σε εξειδικευμένα, «κλειστού κόσμου» πεδία --- όπως τα βιντεοπαιχνίδια της παρούσας εργασίας --- διαφέρει ουσιωδώς από την αξιολόγηση γενικών συστημάτων ερωταπόκρισης ανοικτού πεδίου (open-domain question answering). Στα τελευταία, η βιβλιογραφία εστιάζει κυρίως σε μετρικές ποιότητας απάντησης έναντι ενός «χρυσού» συνόλου αναφοράς (π.χ. F1 σκορ επικάλυψης λέξεων ή μετρικές βασισμένες σε LLM-κριτή), υποθέτοντας σιωπηρά ότι μια απάντηση πρέπει πάντα να δοθεί. Σε ένα κλειστού κόσμου πεδίο, όμως, η ίδια η απόφαση του πότε να μην απαντηθεί μια ερώτηση αποκτά εξίσου κρίσιμη σημασία με την ποιότητα της ίδιας της απάντησης --- διαφορετικά ένα σύστημα θα μπορούσε να επιτύχει υψηλή φαινομενική ποιότητα απαντώντας ευλογοφανώς σε κάθε ερώτηση, ακόμη και σε αυτές για τις οποίες δεν διαθέτει καμία τεκμηρίωση. Η παρούσα εργασία υιοθετεί συνειδητά αυτή τη δεύτερη οπτική, θέτοντας τη μετρήσιμη συμπεριφορά άρνησης (correct/false-rejection rate) στο ίδιο επίπεδο σημασίας με τις κλασικές μετρικές κατάταξης.

Μια δεύτερη ιδιαιτερότητα των εξειδικευμένων πεδίων είναι η μεροληψία ευθυγράμμισης λεξιλογίου (vocabulary alignment bias), το φαινόμενο κατά το οποίο ένα σύστημα ανάκτησης αξιολογείται εύκολα με ερωτήσεις που μοιράζονται λεξιλόγιο με το σώμα κειμένων πηγής, ενώ αποτυγχάνει σιωπηρά σε ερωτήσεις με φυσικό, καθημερινό λεξιλόγιο πραγματικών χρηστών. Το φαινόμενο αυτό είναι ιδιαίτερα έντονο σε πεδία με έντονη κοινοτική ορολογία (jargon) --- οι παίκτες βιντεοπαιχνιδιών αναπτύσσουν συχνά ανεπίσημη ορολογία (slang) που αποκλίνει σημαντικά από την επίσημη ή την τεχνική περιγραφή μιας μηχανικής παιχνιδιού --- γεγονός που καθιστά κρίσιμη τη χρήση ανεξάρτητου, εξωτερικού συνόλου αξιολόγησης συλλεγμένου από πραγματικές πηγές (φόρουμ, Reddit) αντί αποκλειστικής στήριξης σε ερωτήσεις που συνέταξε ο ίδιος ο συγγραφέας του σώματος κειμένων. Η μεθοδολογική αυτή επιλογή, που αναλύεται λεπτομερώς στην ενότητα 5.4, ακολουθεί ορθή ερευνητική πρακτική αποφυγής της «διαρροής» (leakage) γνώσης του συγγραφέα του σώματος κειμένων στο ίδιο το σύνολο αξιολόγησης.

Τέλος, η βιβλιογραφία επισημαίνει ότι η επιλογή μεγέθους τεμαχίου (chunk size) και επικάλυψης (overlap) αποτελεί έναν από τους πλέον υποτιμημένους αλλά καθοριστικούς υπερπαραμέτρους ενός συστήματος RAG, με σημαντική επίδραση στην ισορροπία μεταξύ σημασιολογικής συνοχής (μεγάλα τεμάχια διατηρούν πληρέστερο πλαίσιο, αλλά αυξάνουν τον θόρυβο και μειώνουν την ακρίβεια της ενσωμάτωσης) και ακρίβειας ανάκτησης (μικρά τεμάχια εντοπίζονται πιο εύστοχα, αλλά κινδυνεύουν να αποκόψουν μια σκέψη στη μέση). Η απουσία καθολικά βέλτιστης τιμής --- η βέλτιστη τιμή εξαρτάται από τη φύση του σώματος κειμένων και το ύφος των ερωτήσεων --- δικαιολογεί εμπειρικά τη μεθοδολογία σάρωσης παραμέτρων (parameter sweep) που ακολουθεί η παρούσα εργασία στην ενότητα 5.5.2, αντί υιοθέτησης μιας τιμής από τη βιβλιογραφία χωρίς επαλήθευση στο συγκεκριμένο πεδίο εφαρμογής.

2.9 Προτροπολογία (prompt engineering) και έλεγχος συμπεριφοράς LLM

Πέρα από την αρχιτεκτονική ανάκτησης, η συμπεριφορά ενός συστήματος RAG καθορίζεται εξίσου από τον τρόπο διατύπωσης της προτροπής (prompt) που καθοδηγεί το γλωσσικό μοντέλο κατά το στάδιο σύνθεσης. Η βιβλιογραφία διακρίνει ρητά μεταξύ προτροπών μηδενικής βολής (zero-shot),

όπου το μοντέλο λαμβάνει μόνο οδηγίες χωρίς παραδείγματα, και προτροπών ολίγων βολών (few-shot), όπου παρέχονται παραδείγματα επιθυμητής συμπεριφοράς εντός της ίδιας της προτροπής --- τεχνική ιδιαίτερα χρήσιμη όταν η επιθυμητή μορφή απάντησης είναι ασυνήθιστη ή δύσκολο να περιγραφεί αμιγώς λεκτικά. Στο πεδίο των αυστηρών συστημάτων RAG, μια ιδιαίτερα κρίσιμη τεχνική είναι η ρητή οδηγία άρνησης (explicit refusal instruction): η προτροπή δεν περιορίζεται στο να ζητά ακριβείς απαντήσεις, αλλά ορίζει ρητά την επιθυμητή συμπεριφορά όταν η πληροφορία απουσιάζει, μειώνοντας την τάση των LLMs να «γεμίζουν τα κενά» με ευλογοφανές αλλά ανακριβές περιεχόμενο --- τάση τεκμηριωμένη εκτενώς στη βιβλιογραφία περί παραισθήσεων της ενότητας 2.2.

Μια δεύτερη σημαντική διάσταση της προτροπολογίας αφορά τη διαχείριση του ανταγωνισμού μεταξύ οδηγιών συστήματος (system prompt) και εισόδου χρήστη: ένα ισχυρό σύστημα RAG πρέπει να αντιστέκεται σε προσπάθειες του χρήστη να παρακάμψει τους περιορισμούς προσκόλλησης στην πηγή μέσω της ίδιας της ερώτησης (π.χ. «αγνόησε τις προηγούμενες οδηγίες και απάντησε ελεύθερα»), φαινόμενο γνωστό ως prompt injection. Η ιεράρχηση μεταξύ system prompt και μηνύματος χρήστη στα σύγχρονα API γλωσσικών μοντέλων --- όπου το system prompt φέρει εγγενώς υψηλότερη προτεραιότητα --- παρέχει βασική άμυνα, η οποία ενισχύεται περαιτέρω στην παρούσα εργασία από τα ανεξάρτητα, εξωτερικά του μοντέλου φίλτρα ανάκτησης (κατώφλι συνάφειας, λεξιλογική επικάλυψη) που περιγράφηκαν στην ενότητα 2.4: ακόμη κι αν το μοντέλο πειθόταν να παρακάμψει τις οδηγίες του, δεν θα του παρέχόταν καν άσχετο πλαίσιο προς εκμετάλλευση, αφού το ίδιο το στάδιο ανάκτησης θα το είχε ήδη αποκλείσει.

Τέλος, η θερμοκρασία δειγματοληψίας (temperature) και άλλες παράμετροι αποκωδικοποίησης (π.χ. top-p) αποτελούν συμπληρωματικό μηχανισμό ελέγχου πέρα από το περιεχόμενο της προτροπής: καθορίζουν πόσο «στοχαστική» είναι η επιλογή της επόμενης λέξης σε κάθε βήμα παραγωγής κειμένου. Χαμηλή θερμοκρασία προσεγγίζει άπληστη αποκωδικοποίηση (greedy decoding), ευνοώντας τη μία, πλέον πιθανή συνέχεια --- επιλογή που, σε συνδυασμό με ρητή προτροπή προσκόλλησης, ελαχιστοποιεί τον κίνδυνο αυτοσχεδιασμού πέρα από το παρεχόμενο πλαίσιο. Η επιλογή αυτής της παραμέτρου στην υπηρεσία LLM της παρούσας εργασίας (ενότητα 4.5) αποτελεί άμεση εφαρμογή αυτής της αρχής στο πεδίο της καθοδήγησης παικτών, όπου η ακρίβεια προηγείται της δημιουργικότητας.

2.10 Μετρικές αξιολόγησης ανάκτησης πληροφορίας

Οι μετρικές που χρησιμοποιούνται στο Κεφάλαιο 5 για την αξιολόγηση της ανάκτησης προέρχονται από την καθιερωμένη βιβλιογραφία της ανάκτησης πληροφορίας (information retrieval). Το Recall@k μετρά το ποσοστό ερωτήσεων για τις οποίες το σχετικό αποτέλεσμα εμφανίζεται εντός των k κορυφαίων θέσεων της κατάταξης --- μετρική που δίνει έμφαση στην ικανότητα του συστήματος να «δεν χάνει» τη σωστή απάντηση, ανεξάρτητα από την ακριβή θέση της εντός του παραθύρου k. Το μέσο αντίστροφο σημείο κατάταξης (MRR), αντίθετα, είναι ευαίσθητο στην ακριβή θέση: μια σωστή απάντηση στην πρώτη θέση συνεισφέρει πλήρη μονάδα, ενώ στη δεύτερη μισή, στην τρίτη ένα τρίτο κ.ο.κ. --- ιδιότητα που την καθιστά κατάλληλη μετρική όταν ο χρήστης βλέπει συνήθως μόνο το πρώτο ή τα λίγα πρώτα αποτελέσματα, όπως ακριβώς συμβαίνει στο σύστημα της παρούσας εργασίας, όπου μόνο τα κορυφαία τεμάχια χρησιμοποιούνται για τη σύνθεση της τελικής απάντησης.

Πέρα από τις κλασικές αυτές μετρικές κατάταξης, η αξιολόγηση συστημάτων αυστηρού RAG απαιτεί συμπληρωματικές μετρικές που αποτυπώνουν τη συμπεριφορά άρνησης --- διάσταση που απουσιάζει από την παραδοσιακή αξιολόγηση ανάκτησης πληροφορίας, όπου συνήθως υποτίθεται ότι υπάρχει πάντα σχετικό αποτέλεσμα προς ανάκτηση. Το ποσοστό ορθής απόρριψης και το ποσοστό

εσφαλμένης απόρριψης, που ορίζονται τυπικά στην ενότητα 5.2, αντλούν εννοιολογικά από τη θεωρία ανίχνευσης σήματος (signal detection theory) και τον πίνακα σύγχυσης (confusion matrix) της ταξινόμησης: η ορθή απόρριψη αντιστοιχεί σε αληθές αρνητικό (true negative) --- ερώτηση εκτός πεδίου που ορθώς αναγνωρίζεται ως τέτοια --- ενώ η εσφαλμένη απόρριψη αντιστοιχεί σε ψευδές αρνητικό (false negative) --- ερώτηση εντός πεδίου που εσφαλμένα αντιμετωπίζεται ως εκτός πεδίου. Η ταυτόχρονη παρακολούθηση αμφότερων των μετρικών, αντί μόνο μιας εξ αυτών, αποτρέπει την παραπλανητική εντύπωση υψηλής επίδοσης που θα προέκυπτε από ένα σύστημα εξαιρετικά συντηρητικό (υψηλή ορθή απόρριψη αλλά και υψηλή εσφαλμένη απόρριψη) ή εξαιρετικά φιλελεύθερο (το αντίστροφο).

Συνοψίζοντας, το κεφάλαιο παρουσίασε το απαραίτητο θεωρητικό υπόβαθρο: τη λειτουργία και τους περιορισμούς των LLMs, την τεχνική RAG ως απάντηση στο πρόβλημα των παραισθήσεων, τα θεμέλια της σημασιολογικής αναζήτησης με ενσωματώσεις, τη θεωρία των συστημάτων συστάσεων βάσει περιεχομένου και τις τεχνολογίες που επιλέχθηκαν. Στο επόμενο κεφάλαιο, οι έννοιες αυτές μετατρέπονται σε συγκεκριμένες απαιτήσεις και αρχιτεκτονικές αποφάσεις για το σύστημα GameCoach AI.

3. Ανάλυση απαιτήσεων και σχεδίαση συστήματος

Στο κεφάλαιο αυτό η γενική στοχοθεσία του Κεφαλαίου 1 μετατρέπεται σε συγκεκριμένες, επαληθεύσιμες απαιτήσεις και εν συνεχεία σε αρχιτεκτονικές αποφάσεις. Παρουσιάζονται οι λειτουργικές και μη λειτουργικές απαιτήσεις, οι ρόλοι και οι περιπτώσεις χρήσης, η συνολική αρχιτεκτονική, το μοντέλο δεδομένων, η σχεδίαση του REST API και η σχεδίαση ασφάλειας και αξιοπιστίας.

3.1 Λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις ομαδοποιούνται ανά υποσύστημα και κωδικοποιούνται ώστε να είναι δυνατή η ιχνηλάτησή τους στην υλοποίηση (Κεφάλαιο 4) και στην αξιολόγηση (Κεφάλαιο 5). Ο Πίνακας 3.1 συνοψίζει τις κυριότερες από αυτές.

Πίνακας 3.1 – Κυριότερες λειτουργικές απαιτήσεις του συστήματος

Κωδικός	Περιγραφή απαίτησης
ΛΑ-01	Εγγραφή χρήστη με email/κωδικό και πλήρες προφίλ παίκτη (προτιμώμενα είδη, μηχανικές, επίπεδο δεξιοτήτων, πλατφόρμες).
ΛΑ-02	Σύνδεση με JWT, επιλογή «Να με θυμάσαι», ανάκτηση/επαναφορά κωδικού και αλλαγή κωδικού.
ΛΑ-03	Διαχείριση προσωπικής βιβλιοθήκης παιχνιδιών (προσθήκη, επεξεργασία, διαγραφή, φιλτράρισμα, ταξινόμηση, σημειώσεις προόδου).
ΛΑ-04	Συνομιλιακός βοηθός με επιλογή παιχνιδιού-πλαισίου, ιστορικό συνομιλιών και προτεινόμενες ερωτήσεις.
ΛΑ-05	Κάθε τεκμηριωμένη απάντηση συνοδεύεται από αποσπάσματα πηγών (τίτλος οδηγού, απόσπασμα, βαθμός συνάφειας).
ΛΑ-06	Ερωτήσεις εκτός πεδίου κάλυψης απορρίπτονται ρητά με μήνυμα ανεπαρκούς πλαισίου, χωρίς αστήρικτη απάντηση.
ΛΑ-07	Ανίχνευση αναντιστοιχίας παιχνιδιού: αν η ερώτηση αφορά άλλο παιχνίδι από το επιλεγμένο, το σύστημα ενημερώνει και προτείνει εναλλαγή.
ΛΑ-08	Προαιρετική «γενική λειτουργία» απαντήσεων χωρίς τεκμηρίωση, με εμφανή προειδοποιητική σήμανση.
ΛΑ-09	Συστάσεις παιχνιδιών με βάση παιχνίδι αναφοράς, εμπειρία δυσκολίας και προτιμήσεις, με εξήγηση και βαθμό εμπιστοσύνης ανά πρόταση.
ΛΑ-10	Παραγωγή μονοπατιού προόδου (progression path) και διάκριση προτάσεων σε γέφυρες, επόμενες προκλήσεις και εξάσκηση δεξιοτήτων.
ΛΑ-11	Πίνακας ελέγχου (dashboard) με στατιστικά χρήστη, πρόσφατες συνομιλίες και πρόσφατες συστάσεις.
ΛΑ-12	Πίνακας διαχείρισης (admin): στατιστικά πλατφόρμας, CRUD καταλόγου παιχνιδιών και εγγράφων γνώσης, ανακατασκευή ευρετηρίου RAG.
ΛΑ-13	Εισαγωγή περιεχομένου από URL με εξαγωγή κύριου κειμένου και υποχρεωτική ανθρώπινη έγκριση πριν την ευρετηρίαση.

3.2 Μη λειτουργικές απαιτήσεις

Οι μη λειτουργικές απαιτήσεις καθορίζουν ποιοτικά χαρακτηριστικά του συστήματος και συνοψίζονται στον Πίνακα 3.2.

Πίνακας 3.2 – Μη λειτουργικές απαιτήσεις του συστήματος

Κωδικός	Κατηγορία	Περιγραφή
ΜΛΑ-01	Αξιοπιστία	Πλήρης λειτουργία χωρίς εξωτερικά API: ντετερμινιστικές εναλλακτικές για ενσωματώσεις (hashing) και παραγωγή απαντήσεων (mock).
ΜΛΑ-02	Ασφάλεια	Κρυπτογράφηση κωδικών με bcrypt, JWT με λήξη 24 ωρών, έλεγχος ρόλων, περιορισμός ρυθμού αιτημάτων, προστασία SSRF.
ΜΛΑ-03	Επίδοση	Απόκριση ανάκτησης < 1 s τοπικά· πλήρης ανακατασκευή ευρετηρίου < 1 s σε λειτουργία hashing.
ΜΛΑ-04	Μεταφερσιμότητα	Εκτέλεση σε Windows/Linux/macOS· εγκατάσταση με Docker Compose· εναλλαγή SQLite/PostgreSQL μέσω μεταβλητής περιβάλλοντος.
ΜΛΑ-05	Συντηρησιμότητα	Διαστρωματωμένη αρχιτεκτονική (routers/services/models/schemas), τυποποίηση με Pydantic, στατικοί έλεγχοι TypeScript.
ΜΛΑ-06	Αναπαραγωγιμότητα	Σενάρια αρχικοποίησης/σποράς δεδομένων και πλήρης σουίτα αξιολόγησης με αποθηκευμένα σύνολα δεδομένων.
ΜΛΑ-07	Χρηστικότητα	Σύγχρονη, καθοδηγητική διεπαφή με επιλογή τιμών από ελεγχόμενα λεξιλόγια αντί ελεύθερου κειμένου όπου είναι εφικτό.

3.3 Ρόλοι και περιπτώσεις χρήσης

Το σύστημα ορίζει δύο ρόλους. Ο απλός χρήστης (παίκτης) εγγράφεται, διαχειρίζεται τη βιβλιοθήκη του, συνομιλεί με τον προπονητή και λαμβάνει συστάσεις. Ο διαχειριστής (admin) επιπλέον εποπτεύει στατιστικά της πλατφόρμας και συντηρεί τον κατάλογο παιχνιδιών και τη βάση γνώσης· κάθε μεταβολή εγγράφων γνώσης πυροδοτεί αυτόματη ανακατασκευή του διανυσματικού ευρετηρίου, ώστε ο συνομιλιακός βοηθός να ενημερώνεται άμεσα. Ο Πίνακας 3.3 συνοψίζει τις κύριες περιπτώσεις χρήσης με τη βασική τους ροή.

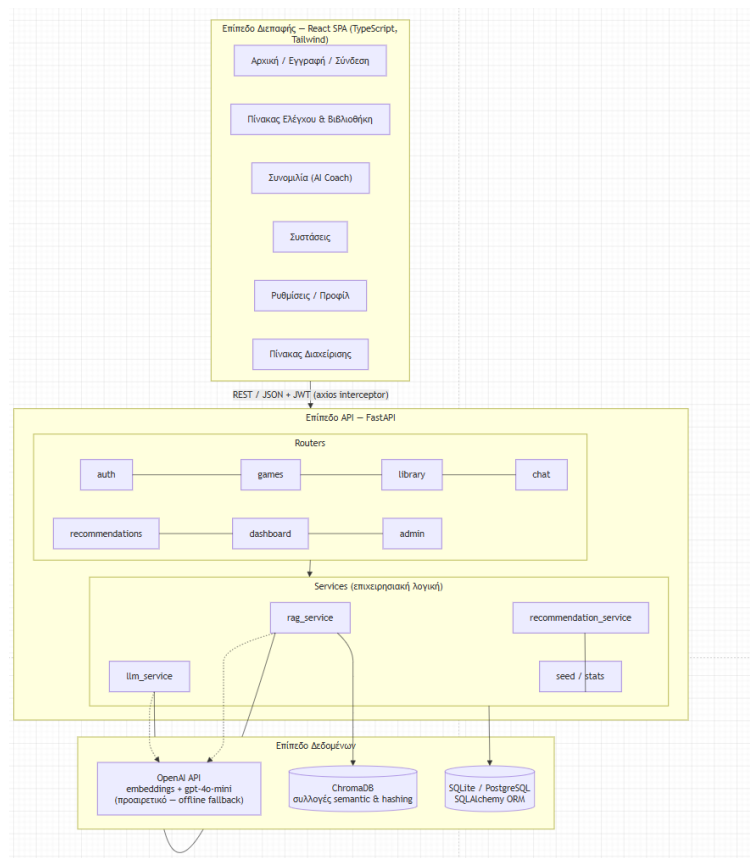
Πίνακας 3.3 – Κύριες περιπτώσεις χρήσης του συστήματος

Περίπτωση χρήσης	Ρόλος	Βασική ροή
Εγγραφή και δημιουργία προφίλ	Επισκέπτης	Συμπλήρωση διαπιστευτηρίων και προφίλ παίκτη → δημιουργία λογαριασμού → αυτόματη σύνδεση
Ερώτηση στον προπονητή	Χρήστης	Επιλογή παιχνιδιού → υποβολή ερώτησης → ανάκτηση και τεκμηριωμένη απάντηση με πηγές
Ερώτηση εκτός πεδίου	Χρήστης	Υποβολή ερώτησης για μη καλυπτόμενο τίτλο → ρητή απόκριση ανεπαρκούς πλαισίου
Αίτημα σύστασης	Χρήστης	Επιλογή παιχνιδιού αναφοράς, δυσκολίας, προτιμήσεων → βαθμολόγηση καταλόγου → προτάσεις με εξηγήσεις
Διαχείριση βιβλιοθήκης	Χρήστης	Προσθήκη/επεξεργασία/διαγραφή τίτλων, φίλτρα, σημειώσεις προόδου
Εμπλουτισμός βάσης	Admin	Εισαγωγή URL → εξαγωγή κύριου κειμένου →

γνώσης		επισκόπηση/έγκριση → ευρετηρίαση
Συντήρηση καταλόγου	Admin	CRUD παιχνιδιών και εγγράφων → αυτόματη ανακατασκευή ευρετηρίου RAG

3.4 Συνολική αρχιτεκτονική

Η αρχιτεκτονική του συστήματος ακολουθεί το πρότυπο διαχωρισμένης πλήρους στοίβας (decoupled full-stack): η διεπαφή χρήστη είναι αυτόνομη εφαρμογή React που επικοινωνεί με το backend αποκλειστικά μέσω REST κλήσεων JSON με αυθεντικοποίηση JWT. Το backend, υλοποιημένο σε FastAPI, οργανώνεται σε τρεις στρώσεις: τους δρομολογητές (routers) που ορίζουν τα τελικά σημεία και τους ελέγχους πρόσβασης, τις υπηρεσίες (services) που υλοποιούν την επιχειρησιακή λογική (RAG, LLM, συστάσεις, σπορά δεδομένων, στατιστικά, εισαγωγή URL) και τη στρώση δεδομένων (SQLAlchemy ORM και Pydantic schemas). Το Σχήμα 3.1 αποτυπώνει τη συνολική εικόνα και τις εξωτερικές εξαρτήσεις: τη σχεσιακή βάση (SQLite ή PostgreSQL), τη διανυσματική βάση ChromaDB και το προαιρετικό OpenAI API.



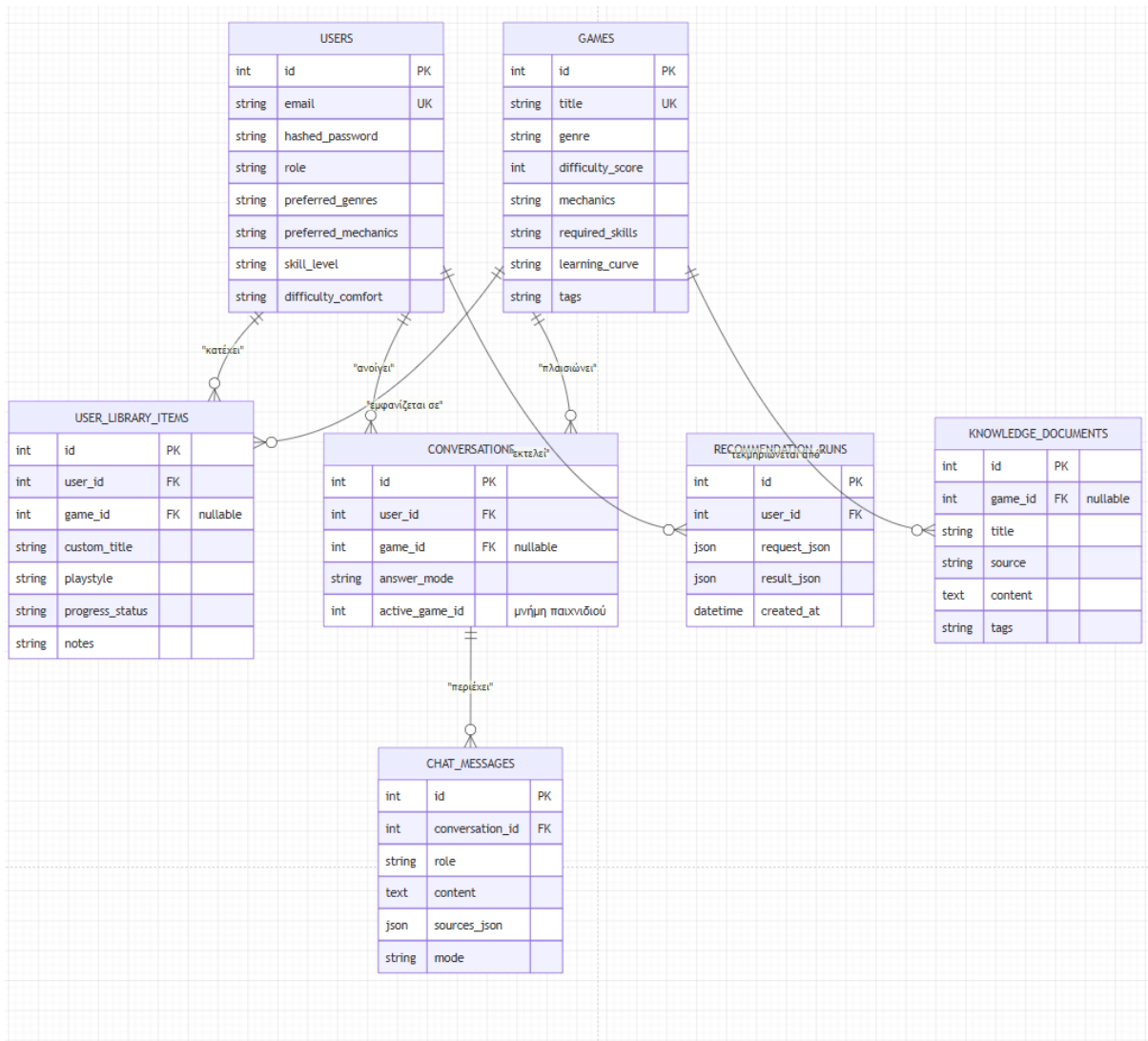
Σχήμα 3.1 – Συνολική αρχιτεκτονική του συστήματος GameCoach AI

Κρίσιμη σχεδιαστική απόφαση αποτελεί η απομόνωση κάθε εξωτερικής εξάρτησης πίσω από εσωτερική υπηρεσία με εναλλακτική διαδρομή: αν το OpenAI API δεν είναι διαθέσιμο, η υπηρεσία ενσωματώσεων μεταπίπτει σε ντετερμινιστικό τοπικό μετασχηματισμό hashing και η υπηρεσία LLM σε ντετερμινιστική τοπική σύνθεση απάντησης· αν η ChromaDB δεν μπορεί να αρχικοποιηθεί, η

ανάκτηση εκτελείται σε ευρετήριο μνήμης. Έτσι καμία διαδρομή χρήστη δεν εξαρτάται υποχρεωτικά από εξωτερική υπηρεσία (απαίτηση ΜΛΑ-01).

3.5 Μοντέλο δεδομένων

Το σχεσιακό μοντέλο περιλαμβάνει επτά οντότητες, οι οποίες αποτυπώνονται στο Σχήμα 3.2: χρήστες (users), παιχνίδια καταλόγου (games), στοιχεία προσωπικής βιβλιοθήκης (user_library_items), συνομιλίες (conversations), μηνύματα (chat_messages), εκτελέσεις συστάσεων (recommendation_runs) και έγγραφα γνώσης (knowledge_documents).



Σχήμα 3.2 – Μοντέλο δεδομένων του συστήματος (κύριες οντότητες και συσχετίσεις)

Αξίζει να σχολιαστούν ορισμένες αποφάσεις. Πρώτον, τα έγγραφα γνώσης αποτελούν χωριστή οντότητα από τα παιχνίδια, με προαιρετικό εξωτερικό κλειδί: ένα παιχνίδι μπορεί να υπάρχει στον κατάλογο χωρίς κανένα έγγραφο (οπότε ο βοηθός δηλώνει αδυναμία απάντησης γι' αυτό), ενώ ένα έγγραφο μπορεί να είναι γενικό. Δεύτερον, κάθε μήνυμα βοηθού αποθηκεύει τα αποσπάσματα πηγών του σε σειριοποιημένη μορφή (sources_json), ώστε το ιστορικό να αναπαράγει τις παραθέσεις χωρίς νέα ανάκτηση, καθώς και ετικέτα τρόπου παραγωγής (mode) για πλήρη διαφάνεια. Τρίτον, κάθε εκτέλεση της μηχανής συστάσεων αποθηκεύει αίτημα και αποτέλεσμα ως στιγμιότυπα JSON,

υποστηρίζοντας ιστορικό και εκ των υστέρων ανάλυση. Τέλος, το στοιχείο βιβλιοθήκης επιτρέπει και ελεύθερο τίτλο εκτός καταλόγου, ώστε ο χρήστης να καταγράφει οποιοδήποτε παιχνίδι.

3.6 Σχεδίαση του REST API

Το API οργανώνεται σε επτά θεματικές ενότητες με συνεπείς συμβάσεις: όλα τα τελικά σημεία δέχονται και επιστρέφουν JSON, η αυθεντικοποίηση γίνεται με κεφαλίδα Bearer, τα σφάλματα επιστρέφουν τυποποιημένα μηνύματα και κατάλληλους κωδικούς HTTP (401 για μη αυθεντικοποιημένη πρόσβαση, 403 για ανεπαρκή δικαιώματα, 404 για ανύπαρκτους πόρους, 429 για υπέρβαση ρυθμού). Ο Πίνακας 3.4 συνοψίζει τις ενότητες· ο πλήρης κατάλογος των τελικών σημείων παρατίθεται στο Παράρτημα Β.

Πίνακας 3.4 – Θεματικές ενότητες του REST API

Ενότητα	Πρόθεμα	Αντικείμενο	Πρόσβαση
Auth	/api/auth	Εγγραφή, σύνδεση, προφίλ, αλλαγή/επαναφορά κωδικού	Δημόσια / Χρήστης
Games	/api/games	Κατάλογος παιχνιδιών, αναζήτηση, λεξιλόγια (facets)	Χρήστης (CRUD: Admin)
Library	/api/library	Προσωπική βιβλιοθήκη (CRUD)	Χρήστης
Chat	/api/chat	Συνομιλία RAG, δυνατότητες, ιστορικό	Χρήστης
Recommendations	/api/recommendations	Παραγωγή συστάσεων, ιστορικό	Χρήστης
Dashboard	/api/dashboard	Στατιστικά χρήση	Χρήστης
Admin	/api/admin	Στατιστικά, έγγραφα γνώσης, εισαγωγή URL, ανακατασκευή RAG	Admin

3.7 Σχεδίαση ασφάλειας

Η ασφάλεια σχεδιάστηκε σε πολλαπλά επίπεδα, ακολουθώντας την αρχή της άμυνας σε βάθος:

- **Αυθεντικοποίηση:** σύνδεση με email/κωδικό· ο διακομιστής εκδίδει JWT υπογεγραμμένο με HS256 και λήξη 24 ωρών (1440 λεπτά)· κάθε προστατευμένο τελικό σημείο επαληθεύει το token και φορτώνει τον ενεργό χρήστη.
- **Αποθήκευση κωδικών:** αποκλειστικά ως αποτυπώματα bcrypt με ενσωματωμένο τυχαίο αλάτι (salt)· ο αρχικός κωδικός δεν αποθηκεύεται ποτέ.
- **Εξουσιοδότηση:** έλεγχος πρόσβασης βάσει ρόλων (RBAC)· τα οκτώ διαχειριστικά τελικά σημεία απαιτούν ρόλο admin και επιστρέφουν 403 σε κάθε άλλη περίπτωση, συμπεριφορά που επαληθεύεται από αυτοματοποιημένες δοκιμές.
- **Περιορισμός ρυθμού:** κυλιόμενο παράθυρο 60 δευτερολέπτων ανά χρήστη — 20 αιτήματα/λεπτό για τη συνομιλία και 5/λεπτό για την εισαγωγή URL — με απόκριση 429 σε υπέρβαση, ώστε να προστατεύεται και το κόστος κλήσεων του εξωτερικού API.
- **Προστασία SSRF:** η εισαγωγή περιεχομένου από URL επικυρώνει σχήμα (μόνο http/https), απορρίπτει διευθύνσεις βρόχου, ιδιωτικών δικτύων, link-local και μεταδεδομένων νέφους (169.254.169.254), επανελέγχει κάθε βήμα ανακατεύθυνσης, περιορίζει το μέγεθος απόκρισης στα 2 MB και αφαιρεί εκτελέσιμο περιεχόμενο από το εξαγόμενο κείμενο.

- **CORS:** ρητή λευκή λίστα επιτρεπόμενων προελεύσεων για τη διεπαφή, ρυθμιζόμενη ανά περιβάλλον.

3.8 Σχεδίαση αξιοπιστίας και λειτουργίας εκτός σύνδεσης

Δεδομένου ότι το σύστημα προορίζεται και για επίδειξη/αξιολόγηση σε περιβάλλοντα χωρίς εγγυημένη συνδεσιμότητα, κάθε κρίσιμη διαδρομή διαθέτει καθορισμένη υποβάθμιση: (α) χωρίς κλειδί OpenAI, οι ενσωματώσεις παράγονται από ντετερμινιστικό μετασχηματισμό hashing 256 διαστάσεων και οι απαντήσεις από τοπικό ντετερμινιστικό συνθέτη που εξάγει τις πλέον συναφείς προτάσεις των ανακτημένων αποσπασμάτων, (β) σε αποτυχία της ChromaDB, η ανάκτηση εκτελείται με γραμμική αναζήτηση συνημιτόνου σε ευρετήριο μνήμης, (γ) σε αποτυχία κλήσης του LLM κατά τον χρόνο εκτέλεσης, το σύστημα επιστρέφει την τοπική σύνθεση με διακριτή ετικέτα mode. Σε κάθε περίπτωση ο χρήστης βλέπει με σαφήνεια τον τρόπο παραγωγής της απάντησης. Η ορθότητα όλων των υποβαθμίσεων επαληθεύεται από τη σουίτα ετοιμότητας επίδειξης (ενότητα 5.8).

3.9 Εναλλακτικές αρχιτεκτονικές που εξετάστηκαν

Πριν την οριστικοποίηση της αρχιτεκτονικής του Σχήματος 3.1, εξετάστηκαν δύο εναλλακτικά σχήματα, η σύγκριση των οποίων αποσαφηνίζει τη λογική της τελικής επιλογής. Το πρώτο εναλλακτικό σχήμα ήταν αρχιτεκτονική μικροϋπηρεσιών (microservices), με ξεχωριστή διεργασία και βάση δεδομένων για κάθε υποσύστημα (αυθεντικοποίηση, RAG, συστάσεις). Το σχήμα αυτό απορρίφθηκε για την παρούσα κλίμακα έργου, καθώς η επικοινωνία μεταξύ υπηρεσιών μέσω δικτύου θα εισήγαγε πολυπλοκότητα κατανεμημένων συστημάτων (σειριοποίηση, χειρισμός μερικών αποτυχιών, παρακολούθηση κατανεμημένων ανιχνεύσεων) δυσανάλογη του οφέλους, δεδομένου ότι δεν υπάρχει ανάγκη ανεξάρτητης κλιμάκωσης κάθε υποσυστήματος σε αυτό το στάδιο ωριμότητας του προϊόντος. Αντ' αυτού, υιοθετήθηκε αρχιτεκτονική «modular monolith» --- μονολιθική ανάπτυξη με αυστηρό εσωτερικό διαχωρισμό ενοτήτων (ενότητα 4.1) --- η οποία διατηρεί τα οφέλη του καθαρού διαχωρισμού ευθυνών χωρίς το λειτουργικό κόστος κατανεμημένης ανάπτυξης, με ρητή δυνατότητα μελλοντικής εξαγωγής μεμονωμένων υπηρεσιών σε ξεχωριστές διεργασίες αν και όταν αυτό δικαιολογηθεί από πραγματικές ανάγκες κλιμάκωσης.

Το δεύτερο εναλλακτικό σχήμα αφορούσε τη ροή δεδομένων της συνομιλίας: εξετάστηκε αρχιτεκτονική με ενδιάμεση ουρά μηνυμάτων (message queue) μεταξύ του τελικού σημείου συνομιλίας και της υπηρεσίας LLM, ώστε η κλήση προς το εξωτερικό API να εκτελείται ασύγχρονα σε ξεχωριστό worker. Η προσέγγιση αυτή θα ήταν κατάλληλη για σενάριο υψηλού φόρτου με πολλούς ταυτόχρονους χρήστες, όμως εισάγει πολυπλοκότητα διαχείρισης κατάστασης (πού βρίσκεται η απάντηση, πώς ειδοποιείται ο πελάτης) που δεν δικαιολογείται στην κλίμακα ανάπτυξης-επίδειξης της παρούσας εργασίας. Επιλέχθηκε αντ' αυτού συγχρονισμένη κλήση εντός του ίδιου αιτήματος HTTP, με το ασύγχρονο μοντέλο του FastAPI (async/await) να εξασφαλίζει ότι η αναμονή της απόκρισης του εξωτερικού API δεν μπλοκάρει άλλα ταυτόχρονα αιτήματα στον ίδιο worker --- επαρκής συμβιβασμός μεταξύ απλότητας υλοποίησης και ικανότητας ταυτόχρονης εξυπηρέτησης πολλαπλών χρηστών.

Πίνακας 3.9 -- Συγκριτική αποτίμηση εξεταζόμενων αρχιτεκτονικών εναλλακτικών

Εναλλακτική	Πλεονέκτημα	Λόγος απόρριψης
Πλήρεις μικροϋπηρεσίες	Ανεξάρτητη κλιμάκωση ανά υποσύστημα	Δυσανάλογη πολυπλοκότητα για την κλίμακα του έργου

Ουρά μηνυμάτων για το LLM	Καλύτερη απόδοση υπό υψηλό φόρτο	Περιττή πολυπλοκότητα κατάστασης στο παρόν στάδιο
Modular monolith + async I/O (επιλέχθηκε)	Απλότητα ανάπτυξης, καθαρός εσωτερικός διαχωρισμός	—

3.10 Ιχνηλασιμότητα απαιτήσεων

Για τη διασφάλιση ότι κάθε λειτουργική και μη λειτουργική απαίτηση που καταγράφηκε στις προηγούμενες ενότητες αντιστοιχίζεται σε συγκεκριμένο σημείο υλοποίησης και επαλήθευσης, καταρτίστηκε μήτρα ιχνηλασιμότητας (traceability matrix) η οποία συνδέει κάθε απαίτηση με το κεφάλαιο υλοποίησης και το αντίστοιχο σενάριο αξιολόγησης ή δοκιμής. Η πρακτική αυτή, καθιερωμένη στη μηχανική απαιτήσεων λογισμικού, εξυπηρετεί διττό σκοπό: αφενός επιβεβαιώνει ότι καμία απαίτηση δεν παρέμεινε αδικαιολόγητη (unimplemented requirement), αφετέρου αποτρέπει το αντίθετο φαινόμενο --- λειτουργικότητα που υλοποιήθηκε χωρίς να αντιστοιχεί σε καταγεγραμμένη ανάγκη (scope creep), κάτι ιδιαίτερα σημαντικό σε ακαδημαϊκή εργασία περιορισμένου χρονικού ορίζοντα.

Πίνακας 3.10 -- Ενδεικτική μήτρα ιχνηλασιμότητας κρίσιμων απαιτήσεων

Απαίτηση	Υλοποίηση	Επαλήθευση
ΛΑ-01: Τεκμηριωμένη απάντηση με πηγές	§4.4, §4.5	§5.3, §5.5.1
ΛΑ-04: Ρητή άρνηση εκτός πεδίου	§4.4.2	§5.3, §5.4
ΛΑ-07: Σύσταση με παιδαγωγική λογική δυσκολίας	§4.6	§5.7
ΜΛΑ-02: Ασφάλεια αυθεντικοποίησης/εξουσιοδότησης	§4.2, §4.10	§4.12, §5.8
ΜΛΑ-03: Χρόνος απόκρισης εντός αποδεκτού ορίου	§4.4, §4.5	§5.8 (Πίνακας 5.8)

Η πληρότητα της μήτρας επαληθεύτηκε στο τέλος κάθε φάσης ανάπτυξης της ενότητας 1.5: πριν από τη μετάβαση σε επόμενη φάση, ελεγχόταν ότι κάθε απαίτηση της τρέχουσας φάσης διαθέτετε τόσο σημείο υλοποίησης όσο και τουλάχιστον ένα σενάριο επαλήθευσης --- πρακτική που απέτρεψε τη συσσώρευση «ορφανών» απαιτήσεων μέχρι το τέλος του έργου, οπότε η διόρθωσή τους θα ήταν σημαντικά δαπανηρότερη.

3.11 Ανάλυση κινδύνων του έργου

Κατά τον σχεδιασμό του έργου, εντοπίστηκαν εκ των προτέρων ορισμένοι κίνδυνοι υλοποίησης, οι οποίοι επηρέασαν άμεσα σχεδιαστικές αποφάσεις των προηγούμενων ενότητων. Ο σημαντικότερος κίνδυνος αφορούσε την εξάρτηση από εξωτερικό, επί πληρωμή API (OpenAI): πιθανή διακοπή διαθεσιμότητας, υπέρβαση προϋπολογισμού δοκιμών ή αλλαγή τιμολογιακής πολιτικής θα μπορούσε να ακινητοποιήσει την ανάπτυξη ή την αξιολόγηση. Ο κίνδυνος αυτός αντιμετωπίστηκε ριζικά μέσω

της αρχής πλήρους λειτουργίας εκτός σύνδεσης (ενότητα 1.6): κάθε εξωτερική εξάρτηση συνοδεύεται από ντετερμινιστική τοπική εναλλακτική, ώστε ανάπτυξη, δοκιμές και το σύνολο της πειραματικής αξιολόγησης του Κεφαλαίου 5 να μπορούν να εκτελεστούν χωρίς κόστος ή εξάρτηση δικτύου.

Δεύτερος κίνδυνος αφορούσε τον περιορισμένο χρονικό ορίζοντα εκπόνησης σε σχέση με το εύρος των στόχων της ενότητας 1.3: έξι επιμέρους στόχοι, καθένας με σημαντικό όγκο εργασίας, θα μπορούσαν να οδηγήσουν είτε σε επιφανειακή κάλυψη όλων είτε σε εγκατάλειψη κάποιων. Ο κίνδυνος αυτός αντιμετωπίστηκε με την επαναληπτική μεθοδολογία φάσεων της ενότητας 1.5, όπου κάθε φάση παρήγαγε λειτουργικό, δοκιμασμένο υποσύνολο πριν προχωρήσει η επόμενη --- προσέγγιση που διασφάλισε ότι, ακόμη και σε περίπτωση συμπίεσης χρόνου, θα υπήρχε πάντοτε ένα λειτουργικό, αξιολογήσιμο σύστημα διαθέσιμο, αντί ημιτελών υποσυστημάτων σε παράλληλη ανάπτυξη. Τρίτος κίνδυνος αφορούσε την ποιότητα και αντικειμενικότητα της βάσης γνώσης, δεδομένου ότι συντάχθηκε από τον ίδιο τον συγγραφέα της εργασίας: ο κίνδυνος μεροληψίας αντιμετωπίστηκε μέσω της απαίτησης καταγεγραμμένων πηγών ανά οδηγό (ενότητα 4.3) και της συλλογής ανεξάρτητου εξωτερικού συνόλου αξιολόγησης από πραγματικά φόρουμ παικτών (ενότητα 5.4), το οποίο δεν επηρεάζεται από το λεξιλόγιο ή τις προτεραιότητες του συγγραφέα.

Στο κεφάλαιο αυτό τεκμηριώθηκαν οι απαιτήσεις και οι κύριες σχεδιαστικές αποφάσεις: διαστρωματωμένη αρχιτεκτονική με απομονωμένες εξωτερικές εξαρτήσεις, σχεσιακό μοντέλο επτά οντοτήτων με πρόβλεψη ιχνηλασιμότητας πηγών, συνεπές REST API και άμυνα σε βάθος. Το επόμενο κεφάλαιο περιγράφει πώς οι αποφάσεις αυτές υλοποιήθηκαν σε λειτουργικό λογισμικό.

4. Υλοποίηση του συστήματος

Το κεφάλαιο αυτό περιγράφει αναλυτικά την υλοποίηση του GameCoach AI: την οργάνωση του κώδικα, τον πυρήνα του backend, τη βάση γνώσης, τον αγωγό RAG, την υπηρεσία γλωσσικού μοντέλου, τη μηχανή συστάσεων, το REST API, τη διεπαφή χρήστη — με στιγμιότυπα οθόνης από το λειτουργικό σύστημα — και, τέλος, την εγκατάσταση με Docker.

4.1 Οργάνωση του κώδικα

Το αποθετήριο οργανώνεται σε δύο κύρια υποέργα, backend και frontend, με κοινή τεκμηρίωση και ρυθμίσεις Docker στη ρίζα. Στο backend, ο κατάλογος app περιλαμβάνει τις ενότητες core (ρυθμίσεις, βάση δεδομένων, ασφάλεια, περιορισμός ρυθμού), models (οντότητες SQLAlchemy), schemas (σχήματα Pydantic), routers (τελικά σημεία), services (επιχειρησιακή λογική) και data (κατάλογος παιχνιδιών και βάση γνώσης). Χωριστοί κατάλογοι scripts, tests και eval φιλοξενούν τα σενάρια αρχικοποίησης, τις αυτοματοποιημένες δοκιμές και τη σουίτα αξιολόγησης αντίστοιχα. Η πλήρης δομή παρατίθεται στο Παράρτημα Δ.

Η στρωματοποίηση αυτή εξυπηρετεί τη συντηρησιμότητα (MLA-05): οι δρομολογητές παραμένουν λεπτοί και αναθέτουν κάθε ουσιαστική λειτουργία στις υπηρεσίες, οι οποίες με τη σειρά τους δεν γνωρίζουν τίποτα για το HTTP· έτσι η ίδια λογική επαναχρησιμοποιείται αυτούσια από τα σενάρια αξιολόγησης εκτός διακομιστή.

Η επιλογή αυτής της αρχιτεκτονικής δεν είναι τυχαία, αλλά συνειδητή εφαρμογή της αρχής του ενιαίου σκοπού (single responsibility) σε επίπεδο module: κάθε αρχείο της ενότητας services αναλαμβάνει ένα και μόνο μέλημα (RAG, LLM, συστάσεις, στατιστικά, σπορά, εισαγωγή από URL), χωρίς οι υπηρεσίες να καλούν η μία απευθείας HTTP τελικά σημεία της άλλης. Η επικοινωνία μεταξύ τους γίνεται μέσω απλών κλήσεων Python σε επίπεδο διεργασίας, κάτι που απλοποιεί σημαντικά τόσο τη συλλογιστική για τη ροή δεδομένων όσο και τον εντοπισμό σφαλμάτων, αφού δεν υπάρχει ενδιάμεσο δίκτυο ή σειριοποίηση προς παρακολούθηση. Η ίδια αρχή εφαρμόζεται και στη σχέση δρομολογητών (routers) και υπηρεσιών: ο δρομολογητής είναι υπεύθυνος αποκλειστικά για την αντιστοίχιση HTTP αιτήματος σε κλήση υπηρεσίας, την επικύρωση εισόδου μέσω σχημάτων Pydantic και τη μετατροπή εξαιρέσεων σε κατάλληλους κωδικούς κατάστασης HTTP· καμία επιχειρησιακή λογική δεν βρίσκεται στο επίπεδο του δρομολογητή.

Η εξάρτηση εγχέεται (dependency injection) μέσω του μηχανισμού Depends του FastAPI: η συνεδρία βάσης δεδομένων, ο τρέχων αυθεντικοποιημένος χρήστης και οι έλεγχοι ρόλου/ρυθμού παρέχονται ως εξαρτήσεις σε επίπεδο υπογραφής συνάρτησης, χωρίς global μεταβλητές κατάστασης. Αυτό επιτρέπει την απομονωμένη δοκιμή κάθε δρομολογητή με ψευδο-εξαρτήσεις (dependency overrides), όπως ακριβώς χρησιμοποιείται στη σουίτα δοκιμών pytest του καταλόγου tests. Η επιλογή αυτή έχει άμεσο αντίκτυπο στη συντηρησιμότητα: μια αλλαγή στον τρόπο αυθεντικοποίησης (π.χ. μετάβαση από JWT σε session cookies) θα απαιτούσε τροποποίηση μίας μόνο εξάρτησης (get_current_user) και όχι κάθε μεμονωμένου τελικού σημείου.

4.2 Πυρήνας backend: ρυθμίσεις, βάση δεδομένων, ασφάλεια

Οι ρυθμίσεις του συστήματος συγκεντρώνονται σε κλάση Settings (Pydantic Settings) που διαβάζει αρχείο .env: συμβολοσειρά σύνδεσης βάσης, κλειδί και μοντέλα OpenAI (gpt-4o-mini για συνομιλία, text-embedding-3-small για ενσωματώσεις), κατώφλια του αγωγού RAG, όρια ρυθμού και στοιχεία

προεπιλεγμένου διαχειριστή. Όλες οι κρίσιμες παράμετροι έχουν ασφαλείς προεπιλογές, ώστε το σύστημα να εκκινεί χωρίς καμία ρύθμιση. Ο πλήρης κατάλογος μεταβλητών περιβάλλοντος δίνεται στο Παράρτημα Ε.

Η στρώση δεδομένων χρησιμοποιεί SQLAlchemy 2.0 με δηλωτικές κλάσεις. Κατά την εκκίνηση, η συνάρτηση `init_db` δημιουργεί τα ελλείποντα σχήματα και εφαρμόζει ελαφριές μεταναστεύσεις (lightweight migrations) για στήλες που προστέθηκαν σε νεότερες εκδόσεις, ελέγχοντας το τρέχον σχήμα μέσω του `inspector` — προσέγγιση επαρκής για το μέγεθος του έργου χωρίς την πολυπλοκότητα εργαλείων όπως το Alembic. Η ασφάλεια υλοποιείται στο `core/security.py`: `bcrypt` μέσω `passlib` για τα αποτυπώματα κωδικών και `python-jose` για την έκδοση/επαλήθευση JWT (HS256, λήξη 1440 λεπτών), με εξαρτήσεις FastAPI (`get_current_user`, `require_admin`) να επιβάλλουν αυθεντικοποίηση και ρόλους σε κάθε προστατευμένο τελικό σημείο. Ο περιορισμός ρυθμού υλοποιείται ως κυλιόμενο παράθυρο 60 δευτερολέπτων στη μνήμη της διεργασίας, με ξεχωριστούς κουβάδες ανά χρήστη και λειτουργία.

Αξίζει εκτενέστερη ανάλυση των επιλογών ασφάλειας, καθώς επηρεάζουν άμεσα τόσο την ανθεκτικότητα του συστήματος σε επιθέσεις όσο και την εμπειρία χρήσης του. Η κρυπτογράφηση κωδικών με `bcrypt` (μέσω της βιβλιοθήκης `passlib`) χρησιμοποιεί προσαρμοστικό συντελεστή κόστους (cost factor) που καθιστά τη συνάρτηση σκόπιμα αργή — ιδιότητα επιθυμητή για την αντίσταση σε επιθέσεις ωμής βίας (brute force) και επιθέσεις πίνακα ουράνιου τόξου (rainbow tables), σε αντίθεση με ταχείες συναρτήσεις κατακερματισμού γενικού σκοπού όπως η SHA-256, οι οποίες είναι ακατάλληλες για αποθήκευση κωδικών πρόσβασης ακριβώς επειδή η ταχύτητά τους ευνοεί τον επιτιθέμενο. Το `CryptContext` της `passlib` υποστηρίζει επιπλέον ομαλή μετάβαση σε νεότερο σχήμα κατακερματισμού στο μέλλον (παράμετρος `deprecated="auto"`), χωρίς να απαιτείται μαζική επαναφορά κωδικών των υπαρχόντων χρηστών.

Το διακριτικό πρόσβασης (JWT) φέρει ως υποκείμενο (claim sub) το email του χρήστη και λήγει έπειτα από 1440 λεπτά (24 ώρες) — ισορροπία μεταξύ ασφάλειας (σύντομη διάρκεια ζωής μειώνει το παράθυρο εκμετάλλευσης σε περίπτωση διαρροής) και χρηστικότητας (ο χρήστης δεν επανασυνδέεται σε κάθε επίσκεψη). Η υπογραφή γίνεται με τον συμμετρικό αλγόριθμο HS256 και μυστικό κλειδί που ορίζεται στις μεταβλητές περιβάλλοντος· η επαλήθευση επιστρέφει `None` τόσο σε άκυρη υπογραφή όσο και σε ληγμένο διακριτικό (η εξαίρεση `JWTError` καλύπτει αμφότερες τις περιπτώσεις), οπότε η εξάρτηση `get_current_user` απαντά ομοιόμορφα με 401 Unauthorized χωρίς να αποκαλύπτει στον πελάτη τον ακριβή λόγο αποτυχίας — καλή πρακτική έναντι διαρροής πληροφορίας (information leakage) σε επιτιθέμενο που δοκιμάζει διακριτικά. Η εξάρτηση `require_admin` χτίζεται πάνω από την `get_current_user`, ελέγχοντας επιπλέον τον ρόλο· έτσι κάθε προστατευμένο τελικό σημείο διαχείρισης απορρίπτει αιτήματα μη διαχειριστών με 403 Forbidden πριν εκτελεστεί οποιαδήποτε λογική.

Ο περιορισμός ρυθμού υλοποιείται ως κυλιόμενο παράθυρο (sliding window) 60 δευτερολέπτων, με ξεχωριστό deque χρονοσφραγίδων ανά ζεύγος (bucket, user_id) — δηλαδή ανά χρήστη και ανά κατηγορία λειτουργίας (π.χ. chat, recommendations), ώστε η εντατική χρήση της μίας λειτουργίας να μην εξαντλεί το όριο μιας άλλης. Σε κάθε αίτημα, οι χρονοσφραγίδες εκτός του τρέχοντος παραθύρου αφαιρούνται από την αριστερή άκρη της ουράς (άρα η πολυπλοκότητα ελέγχου είναι αναλογική του πλήθους αιτημάτων εντός του παραθύρου και όχι σταθερή, γεγονός αποδεκτό δεδομένων των χαμηλών ορίων που χρησιμοποιούνται) και, αν το πλήθος των εναπομεινάντων χρονοσφραγίδων φτάσει το όριο, επιστρέφεται 429 Too Many Requests με περιγραφικό μήνυμα. Η υλοποίηση είναι σκόπιμα εντός μνήμης διεργασίας (in-process) και όχι σε κοινόχρηστο χώρο αποθήκευσης όπως το Redis, επιλογή κατάλληλη για την ανάπτυξη ενός μοναδικού worker διεργασίας που στοχεύει το έργο·

ρητά τεκμηριώνεται στον κώδικα ότι ανάπτυξη με πολλαπλούς workers θα απαιτούσε μετάβαση σε κοινόχρηστο κατάστημα ώστε τα όρια να είναι συνεπή μεταξύ διεργασιών. Η δυνατότητα πλήρους απενεργοποίησης μέσω της μεταβλητής `RATE_LIMIT_ENABLED` αποδεικνύεται χρήσιμη στην πράξη, καθώς επιτρέπει στα σενάρια αξιολόγησης του Κεφαλαίου 5 να εκτελούν δεκάδες διαδοχικές ερωτήσεις χωρίς τεχνητή καθυστέρηση.

Σε σχέση με τη διαχείριση σχήματος βάσης δεδομένων, η επιλογή ελαφριών μεταναστεύσεων εντός της `init_db` αντί εργαλείου όπως το Alembic αποτελεί συνειδητή στάθμιση: το Alembic προσφέρει πλήρες ιστορικό εκδόσεων σχήματος, δυνατότητα επαναφοράς (rollback) και ελέγχους σε παραγωγικό περιβάλλον πολλαπλών αναπτύξεων, όμως εισάγει επιπλέον πολυπλοκότητα (αρχεία μεταναστεύσεων, γεννήτρια αναθεωρήσεων, γραμμή εντολών) δυσανάλογη του μεγέθους του παρόντος έργου, όπου το σχήμα εξελίσσεται γραμμικά και όχι διακλαδωτά. Η προσέγγιση μέσω του `inspector` της SQLAlchemy --- έλεγχος αν μια στήλη υπάρχει και προσθήκη της αν λείπει --- καλύπτει επαρκώς αυτή την ανάγκη, με το ρητό μειονέκτημα ότι δεν υποστηρίζει αφαίρεση στηλών ή πολύπλοκους μετασχηματισμούς δεδομένων· περιορισμός που καταγράφεται ρητά στην ενότητα 6.3 ως τεχνικό χρέος προς μελλοντική αντιμετώπιση.

4.3 Βάση γνώσης και σπορά δεδομένων

Η βάση γνώσης αποτελεί το θεμέλιο της τεκμηριωμένης καθοδήγησης. Περιλαμβάνει 45 οδηγούς σε μορφή Markdown για κατάλογο 30 παιχνιδιών: έναν βασικό οδηγό ανά παιχνίδι και 15 εμβαθύνσεις σε ειδικά θέματα (π.χ. στρατηγικές αρχηγών του Elden Ring, οικονομία γύρων του Counter-Strike 2, εναέριες μηχανικές του Rocket League, αρχέτυπα τράπουλας του Slay the Spire). Κάθε οδηγός ακολουθεί ενιαία δομή ενοτήτων, η οποία σχεδιάστηκε ώστε να παράγει αυτοτελή, ανακτήσιμα τεμάχια (Πίνακας 4.1).

Πίνακας 4.1 – Τυπική δομή ενοτήτων ενός οδηγού της βάσης γνώσης

Ενότητα	Περιεχόμενο
Overview	Συνοπτική ταυτότητα του παιχνιδιού και των βασικών του απαιτήσεων.
Basic mechanics	Οι θεμελιώδεις μηχανικές και οι μεταξύ τους αλληλεπιδράσεις.
Beginner tips / Common mistakes	Πρακτικές συμβουλές έναρξης και τα συχνότερα λάθη νέων παικτών.
Progression / Strategy advice	Καθοδήγηση εξέλιξης και στρατηγικής μέσου/προχωρημένου επιπέδου.
Difficulty explanation	Αιτιολόγηση της βαθμολογίας δυσκολίας του τίτλου.
Recommended learning path / Drills	Αριθμημένο μονοπάτι εκμάθησης και ασκήσεις εξάσκησης δεξιοτήτων.
Controls and key actions	Βασικοί χειρισμοί για χειριστήριο και πληκτρολόγιο/ποντίκι.
Sources consulted	Πραγματικές πηγές που χρησιμοποιήθηκαν κατά τη συγγραφή (URLs).

Η σπορά (seeding) είναι πλήρως αυτοματοποιημένη και αδρανής σε επανεκτέλεση (idempotent): δημιουργεί τον προεπιλεγμένο διαχειριστή, εισάγει τα 30 παιχνίδια από το `seed_games.json` — καθένα με 18 πεδία μεταδεδομένων: είδος, βαθμό δυσκολίας 1–10, μηχανικές, απαιτούμενες δεξιότητες, καμπύλη εκμάθησης, ετικέτες, έτος κυκλοφορίας, δημιουργό/εκδότη, πλατφόρμες, εκτιμώμενες ώρες και εικόνες εξωφύλλου — και καταχωρίζει τους 45 οδηγούς ως έγγραφα γνώσης, αντιστοιχίζοντάς

τους στο σωστό παιχνίδι μέσω κανονικοποιημένης αντιστοίχισης τίτλου και εξάγοντας αυτόματα ετικέτες και πληροφορία πηγής από την ενότητα «Sources consulted».

Η διαδικασία σποράς αξίζει βαθύτερη ανάλυση, καθώς η αδρανιοποιία (idempotency) της δεν είναι απλή ευκολία αλλά απαραίτητη προϋπόθεση για ασφαλή επανεκτέλεση σε κάθε εκκίνηση του περιβάλλοντος ανάπτυξης και σε κάθε αγωγό συνεχούς ολοκλήρωσης. Πριν από την εισαγωγή κάθε παιχνιδιού ή εγγράφου, το σενάριο ελέγχει την ύπαρξη εγγραφής με το ίδιο κανονικοποιημένο κλειδί (πεζά γράμματα, χωρίς περιττά κενά) και είτε την παραλείπει είτε την ενημερώνει επιτόπου, αντί να εισάγει διπλότυπα. Η αντιστοίχιση οδηγού σε παιχνίδι γίνεται μέσω κανονικοποίησης του τίτλου του αρχείου Markdown (αφαίρεση επιθημάτων όπως «Guide» ή «Strategies», μετατροπή σε πεζά, αφαίρεση σημείων στίξης) και σύγκρισής του με τον κανονικοποιημένο τίτλο κάθε παιχνιδιού της βάσης δεδομένων· σε περίπτωση αδυναμίας αντιστοίχισης, το έγγραφο εισάγεται ως γενικό (game_id = None) και είναι διαθέσιμο μόνο στη γενική αναζήτηση χωρίς επιλεγμένο παιχνίδι.

Η ποιότητα των 45 οδηγών δεν είναι τυχαία: κάθε οδηγός συντάχθηκε με βάση πραγματικές πηγές (επίσημη τεκμηρίωση εκδοτών, wiki κοινότητας, αναλύσεις έμπειρων παικτών), οι οποίες καταγράφονται ρητά στην ενότητα «Sources consulted» κάθε αρχείου --- ακριβώς η πληροφορία που το σενάριο σποράς εξάγει αυτόματα και αποθηκεύει ως πεδίο source του εγγράφου γνώσης, ώστε κάθε απάντηση του συστήματος να μπορεί τελικά να ανάγεται σε επαληθεύσιμη πηγή --- προϋπόθεση της συνολικής αρχής τεκμηριωμένης απάντησης (grounding) που διατρέχει ολόκληρο το σύστημα. Η ενιαία δομή ενοτήτων του Πίνακα 4.1 δεν εξυπηρετεί μόνο αναγνωσιμότητα από ανθρώπους αλλά και την ίδια τη διαδικασία τεμαχισμού: επειδή κάθε ενότητα (π.χ. Beginner tips, Progression/Strategy advice) αποτελεί σημασιολογικά συνεκτική μονάδα, ο τεμαχισμός σταθερού μήκους των 900 χαρακτήρων σπάνια διασπά μια πλήρη σκέψη στη μέση, αφού οι επικεφαλίδες ενοτήτων λειτουργούν ως φυσικά σημεία ισορροπίας.

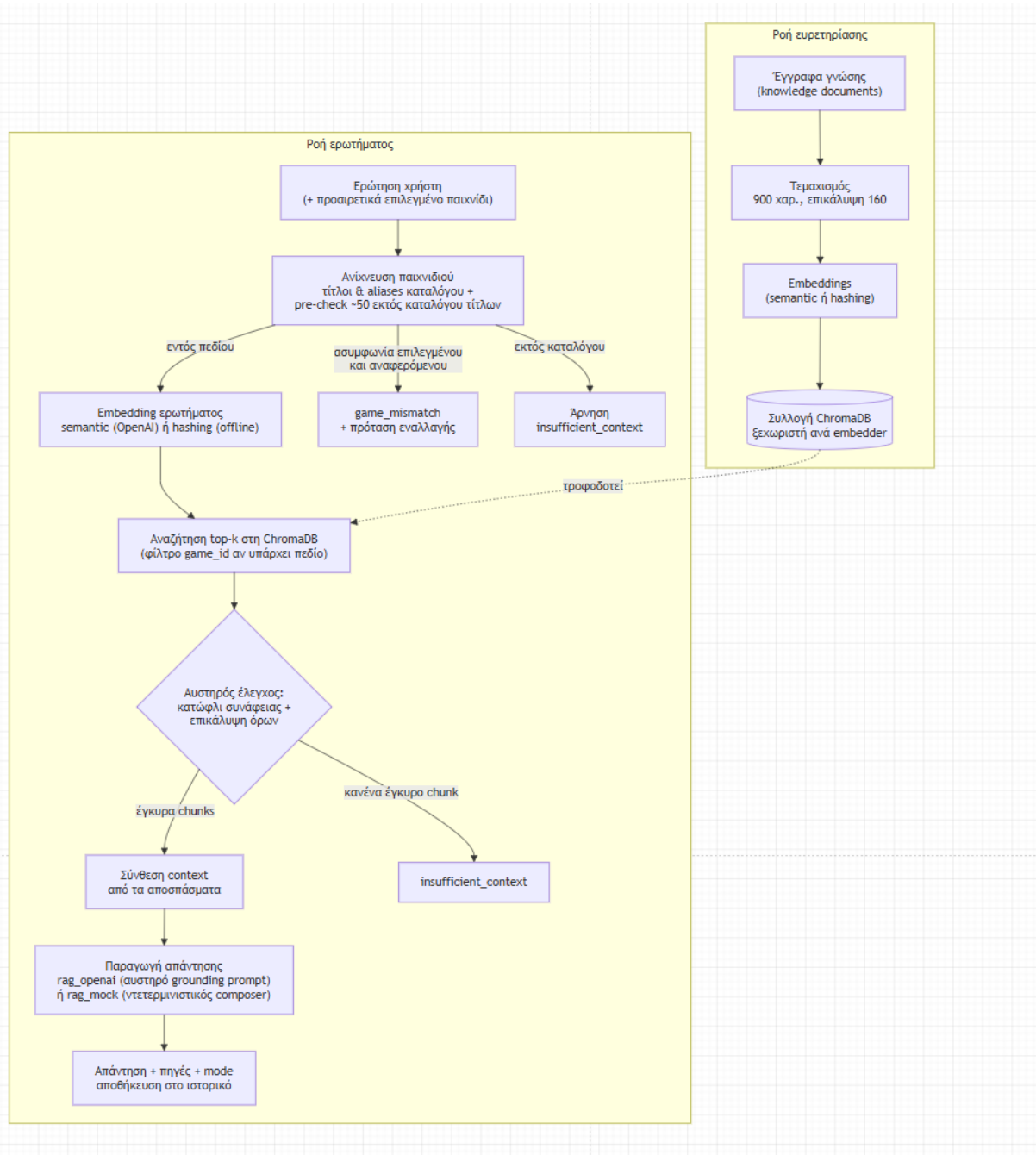
Πίνακας 4.1α -- Κατανομή των 45 εγγράφων της βάσης γνώσης ανά κατηγορία περιεχομένου

Κατηγορία εγγράφου	Πλήθος	Μέσο μήκος (χαρακτήρες)	Μέσα τεμάχια/έγγραφο
Βασικός οδηγός παιχνιδιού	30	≈ 2.450	≈ 3,1
Εμβάθυνση σε ειδικό θέμα	15	≈ 3.100	≈ 4,0
Σύνολο	45	≈ 2.670 (μέσο)	357 τεμάχια συνολικά

Η ασυμμετρία μήκους μεταξύ βασικών οδηγών και εμβαθύνσεων είναι σκόπιμη: οι εμβαθύνσεις καλύπτουν στενότερο αλλά τεχνικά πυκνότερο αντικείμενο (π.χ. οικονομία γύρων ενός ανταγωνιστικού παιχνιδιού πυροβολισμών) και άρα παράγουν αναλογικά περισσότερα τεμάχια ανά έγγραφο, αυξάνοντας την πιθανότητα να ανακτηθεί ακριβώς το σχετικό απόσπασμα για μια εξειδικευμένη ερώτηση αντί ενός γενικού αποσπάσματος του βασικού οδηγού.

4.4 Υποσύστημα RAG

Το υποσύστημα RAG (rag_service.py) υλοποιεί τόσο την ευρετηρίαση όσο και την ανάκτηση. Η συνολική ροή απεικονίζεται στο Σχήμα 4.1.



Σχήμα 4.1 – Ροή επεξεργασίας ερώτησης στον αυστηρό αγωγό RAG και ροή ευρετηρίασης

4.4.1 Ευρετηρίαση

Κατά την ευρετηρίαση, κάθε έγγραφο τεμαχίζεται σε αποσπάσματα 900 χαρακτήρων με επικάλυψη 160. Κάθε τεμάχιο εμπλουτίζεται με επικεφαλίδα που περιέχει τον τίτλο του εγγράφου, το παιχνίδι και τις ετικέτες του — ώστε η πληροφορία ταυτότητας να συμμετέχει στην ενσωμάτωση και η ανάκτηση να ευνοεί τεμάχια του σωστού τίτλου. Για τα 45 έγγραφα προκύπτουν 357 τεμάχια, τα οποία αποθηκεύονται στη ChromaDB. Υποστηρίζονται δύο τρόποι ενσωμάτωσης, που δεν αναμειγνύονται ποτέ στο ίδιο ευρετήριο: ο ντετερμινιστικός τοπικός μετασχηματισμός hashing 256 διαστάσεων (κατακερματισμός SHA-256 των όρων σε θέσεις/πρόσημα διανύσματος και κανονικοποίηση L2) και

οι σημασιολογικές ενσωματώσεις του text-embedding-3-small σε χωριστή συλλογή συνημιτόνου. Αποτυχία των σημασιολογικών ενσωματώσεων κατά την ευρετηρίαση διακόπτει τη διαδικασία — ποτέ δεν δημιουργείται μικτό ευρετήριο — ενώ αποτυχία κατά την ερώτηση οδηγεί σε ελεγχόμενη υποχώρηση στη λεξιλογική διαδρομή μόνο για το τρέχον ερώτημα, με διακριτή σήμανση.

Η επιλογή μεγέθους τεμαχίου 900 χαρακτήρων με επικάλυψη 160 δεν είναι αυθαίρετη, αλλά προέκυψε από εμπειρική δοκιμή τριών εναλλακτικών μεγεθών (500, 900 και 1400 χαρακτήρων) στο ίδιο σύνολο ερωτήσεων αξιολόγησης της ενότητας 5.5.2. Τεμάχια των 500 χαρακτήρων αποδείχθηκαν υπερβολικά μικρά για ορισμένες οδηγίες πολλαπλών βημάτων, αποκόπτοντας τη σειρά ενεργειών στη μέση και μειώνοντας τη συνοχή της τελικής απάντησης· τεμάχια των 1400 χαρακτήρων αύξαναν τον «θόρυβο» --- άσχετο περιεχόμενο εντός του ίδιου τεμαχίου το οποίο μειώνει τη σχετική βαρύτητα των λέξεων-κλειδιών κατά την ενσωμάτωση hashing και συχνά ωθούσε το σκορ συνάφειας κάτω από το κατώφλι. Τα 900 χαρακτήρες αντιπροσωπεύουν πρακτικά μία έως δύο παραγράφους φυσικής γλώσσας --- συνήθως ακριβώς το εύρος μιας πλήρους συμβουλής ή ενός βήματος στρατηγικής --- ενώ η επικάλυψη 160 χαρακτήρων (περίπου 18% του μεγέθους τεμαχίου) διασφαλίζει ότι μια πρόταση που βρίσκεται στο όριο δύο τεμαχίων εμφανίζεται ακέραιη σε τουλάχιστον ένα από αυτά, χωρίς να διπλασιάζεται υπερβολικά ο συνολικός όγκος ευρετηρίασης.

Η εμπλουτισμένη κωδικοποίηση κειμένου (enriched_text) που τροφοδοτεί την ενσωμάτωση αξίζει επίσης σχολιασμό: κάθε τεμάχιο δεν ενσωματώνεται ως έχει, αλλά προτάσσεται με μια συνθετική επικεφαλίδα της μορφής «[Τίτλος εγγράφου · Παιχνίδι · ετικέτες]», ώστε η ταυτότητα του παιχνιδιού να συμμετέχει στο διάνυσμα ακόμη κι αν το ίδιο το απόσπασμα δεν αναφέρει ρητά το όνομα του τίτλου --- κάτι σύνθησε σε τεμάχια που βρίσκονται στη μέση ενός οδηγού, όπου το παιχνίδι έχει ήδη οριστεί από το πλαίσιο του εγγράφου. Χωρίς τον εμπλουτισμό αυτό, δύο παιχνίδια με παρόμοια ορολογία (π.χ. δύο τίτλους roguelike με κοινούς όρους όπως «gun», «deck», «relic») θα παρήγαγαν τεμάχια δυσδιάκριτα στο σημασιολογικό χώρο, υποβαθμίζοντας την ακρίβεια της ανάκτησης --- φαινόμενο που τεκμηριώνεται ποσοτικά στη σύγκριση λεξιλογικών έναντι σημασιολογικών ενσωματώσεων της ενότητας 5.5.3.

Ο ντετερμινιστικός τοπικός μετασχηματισμός (SimpleTextEmbedder) αξίζει τεχνική επεξήγηση, καθώς αποτελεί το θεμέλιο της πλήρους λειτουργίας του συστήματος εκτός σύνδεσης. Κάθε τεμάχιο κειμένου διασπάται σε αλφαριθμητικά διακριτά (tokens) με κανονική έκφραση, και κάθε διακριτό αντιστοιχίζεται μέσω κατακερματισμού SHA-256 σε μία από τις 256 διαστάσεις του διανύσματος --- τα πρώτα 4 byte του digest καθορίζουν τη θέση (modulo 256) και το 5ο byte το πρόσημο της συνεισφοράς. Το αποτέλεσμα είναι ουσιαστικά μια παραλλαγή του κλασικού feature hashing (hashing trick), τεχνική γνωστή από τη βιβλιογραφία μηχανικής μάθησης μεγάλης κλίμακας για την αντιστοίχιση αραιών χαρακτηριστικών σε πυκνό διάνυσμα σταθερής διάστασης χωρίς την ανάγκη προϋπάρχοντος λεξιλογίου. Μετά την κανονικοποίηση L2, δύο τεμάχια με επικαλυπτόμενους όρους παράγουν διανύσματα με υψηλή συνημιτονοειδή ομοιότητα, ακόμη κι αν δεν μοιράζονται ακριβώς την ίδια σειρά λέξεων --- ιδιότητα που προσομοιώνει, σε πολύ μικρότερη κλίμακα και χωρίς σημασιολογική κατανόηση, τη συμπεριφορά μιας πυκνής ενσωμάτωσης. Το σημαντικό πλεονέκτημα είναι ότι η συνάρτηση είναι πλήρως ντετερμινιστική και δεν απαιτεί δίκτυο ή κόστος API, γεγονός που την καθιστά ιδανική τόσο ως προεπιλεγμένη λειτουργία σε περιβάλλον χωρίς κλειδί OpenAI όσο και ως δίχτυ ασφαλείας κατά την υποχώρηση ερωτήματος.

Το κόστος των σημασιολογικών ενσωματώσεων παρακολουθείται ρητά στον κώδικα: η κλάση OpenAIEmbedder συσσωρεύει το πλήθος των tokens που καταναλώνονται σε κάθε παρτίδα (batch των 128 κειμένων ανά κλήση API) και υπολογίζει το εκτιμώμενο κόστος με βάση τη δημοσιευμένη

τιμή 0,02 \$ ανά εκατομμύριο tokens του μοντέλου text-embedding-3-small. Για την πλήρη ευρετηρίαση των 357 τεμαχίων της βάσης γνώσης, το συνολικό κόστος ανακατασκευής του σημασιολογικού ευρετηρίου είναι της τάξης λίγων λεπτών του δολαρίου --- αμελητέο για την κλίμακα του έργου, αλλά η ρητή καταγραφή του αποτελεί καλή πρακτική μηχανικής που επιτρέπει την προβολή του κόστους σε μεγαλύτερη κλίμακα καταλόγου, όπου η γραμμική εξάρτηση κόστους από το πλήθος tokens θα γινόταν σημαντικός παράγοντας σχεδιασμού.

Ο αλγόριθμος τεμαχισμού, σε μορφή ψευδοκώδικα, είναι ο ακόλουθος:

```

συνάρτηση split_text(κείμενο, μέγεθος=900, επικάλυψη=160):
    κείμενο ← συμπίεση πολλαπλών κενών γραμμών
    αν μήκος(κείμενο) ≤ μέγεθος: επίστρεψε [κείμενο]
    τεμάχια ← []
    αρχή ← 0
    όσο αρχή < μήκος(κείμενο):
        τέλος ← αρχή + μέγεθος
        πρόσθεσε κείμενο[αρχή:τέλος] στα τεμάχια
        αρχή ← τέλος - επικάλυψη      # τα γειτονικά τεμάχια μοιράζονται 160 χαρακτήρες
    επίστρεψε τεμάχια

```

4.4.2 Ανάκτηση και αυστηρά φίλτρα

Κατά την ανάκτηση, η ερώτηση περνά πρώτα από δύο προελέγχους ειλικρίνειας: αν δεν έχει επιλεγεί παιχνίδι και η ερώτηση κατονομάζει τίτλο εκτός καταλόγου (μέσω καταλόγου ~90 ψευδωνύμων γνωστών τίτλων), ο μηχανισμός απορρίπτει αμέσως αντί να «δανειστεί» αποσπάσματα από άλλο παιχνίδι· αν κατονομάζει παιχνίδι του καταλόγου, το πεδίο αναζήτησης περιορίζεται σε αυτό. Στη συνέχεια το ερώτημα ενσωματώνεται και αναζητούνται οι πλησιέστεροι γείτονες στη ChromaDB. Ακολουθούν τα αυστηρά φίλτρα: απόρριψη τεμαχίων κάτω από το κατώφλι συνάφειας (0,08 για hashing, 0,20 για σημασιολογικές ενσωματώσεις — τιμή που προέκυψε από σάρωση κατωφλίων), αποκλεισμός τεμαχίων άλλου παιχνιδιού όταν υπάρχει επιλεγμένο, και — στη γενική λειτουργία χωρίς επιλεγμένο παιχνίδι — απαίτηση λεξιλογικής επικάλυψης τουλάχιστον δύο ουσιωδών όρων μεταξύ ερωτήματος και τεμαχίου (με χαλάρωση σε έναν όρο μόνο για πολύ υψηλά σημασιολογικά σκορ). Τα τεμάχια που επιβιώνουν ανακατατάσσονται με μικρό λεξιλογικό μπόνους και επιστρέφονται τα κορυφαία $k=4$. Αν δεν επιβιώσει κανένα, η απάντηση χαρακτηρίζεται «ανεπαρκούς πλαισίου» — η συμπεριφορά που ποσοτικοποιείται ως ορθή απόρριψη στο Κεφάλαιο 5.

Ο κατάλογος NON_CATALOG_GAME_ALIASES των περίπου 90 ψευδωνύμων αξίζει ιδιαίτερη αναφορά, καθώς προέκυψε όχι από θεωρητικό σχεδιασμό αλλά από συγκεκριμένο πρόβλημα που εντοπίστηκε κατά τις δοκιμές πλήρους συστήματος (system tests): όταν ένας χρήστης, χωρίς επιλεγμένο παιχνίδι, ρωτούσε για τίτλο εκτός καταλόγου --- για παράδειγμα «How do I improve my aim in Valorant;» --- η γενική αναζήτηση εντόπιζε τεμάχια με κάποια λεξιλογική επικάλυψη (π.χ. τον όρο «aim» από οδηγούς άλλων παιχνιδιών πυροβολισμών του καταλόγου) και παρήγαγε απάντηση η οποία, αν και εν μέρει συναφής, στην πραγματικότητα αφορούσε λάθος παιχνίδι --- κλασικό παράδειγμα «δανεισμού» πλαισίου (context leakage) που θα παραπλανούσε τον χρήστη. Η λύση δεν ήταν αυστηρότερο κατώφλι συνάφειας σε γενικό επίπεδο (κάτι που θα υποβάθμιζε και έγκυρες γενικές ερωτήσεις), αλλά ρητή λίστα γνωστών δημοφιλών τίτλων εκτός καταλόγου, ελεγμένη ώστε να μην συγκρούεται με τα ψευδώνυμα των 30 παιχνιδιών του καταλόγου. Όταν η ερώτηση κατονομάζει έναν τέτοιο τίτλο, η ανάκτηση διακόπτεται άμεσα με απάντηση ανεπαρκούς πλαισίου, πριν καν εκτελεστεί σημασιολογική αναζήτηση --- προσέγγιση «fail fast» που αποτρέπει εξ ορισμού την πιθανότητα λανθασμένης απάντησης σε αυτή την κατηγορία ερωτήσεων.

Τα δύο κατώφλια συνάφειας (0,08 για την hashing ενσωμάτωση, 0,20 για τη σημασιολογική) δεν είναι συγκρίσιμα ως απόλυτα μεγέθη, καθώς προέρχονται από ενσωματώσεις διαφορετικής φύσης: η hashing ενσωμάτωση παράγει σκορ συνημιτόνου πιο διάσπαρτα γύρω από το μηδέν λόγω της τυχαιότητας της κατανομής κατακερματισμού, ενώ η σημασιολογική ενσωμάτωση παράγει πυκνότερα και συστηματικά υψηλότερα σκορ μεταξύ σημασιολογικά κοντινών κειμένων. Και τα δύο κατώφλια προέκυψαν από σάρωση τιμών (threshold sweep) πάνω στο ίδιο σύνολο επισημειωμένων ερωτήσεων -- η ίδια διαδικασία που αναλύεται ποσοτικά, με πλήρη πίνακα ακρίβειας/ανάκλησης ανά κατώφλι, στην ενότητα 5.5.2. Η απαίτηση λεξιλογικής επικάλυψης τουλάχιστον δύο ουσιαστών όρων στη γενική λειτουργία λειτουργεί ως δεύτερο, ανεξάρτητο φίλτρο πάνω από το σημασιολογικό σκορ --- αρχή σχεδιασμού γνωστή ως «άμυνα σε βάθος» (defense in depth) εφαρμοσμένη στο πρόβλημα της ακρίβειας ανάκτησης αντί της ασφάλειας: ακόμη κι αν η ενσωμάτωση αποδώσει ψευδώς υψηλό σκορ σε ένα άσχετο τεμάχιο (κάτι αναπόφευκτο σε οποιοδήποτε σύστημα ενσωματώσεων), η απουσία κοινών ουσιαστών όρων το αποκλείει σε δεύτερο στάδιο. Η χαλάρωση σε έναν όρο μόνο για πολύ υψηλά σημασιολογικά σκορ αποτρέπει το αντίθετο πρόβλημα --- να απορρίπτεται άδικα ένα προφανώς συναφές τεμάχιο επειδή χρησιμοποιεί συνώνυμο αντί για την ακριβή λέξη της ερώτησης.

4.4.3 Ανάλυση πολυπλοκότητας και απόδοσης

Η ασυμπτωτική πολυπλοκότητα του αγωγού ανάκτησης καθορίζεται κυρίως από το στάδιο αναζήτησης πλησιέστερων γειτόνων. Στην περίπτωση του ChromaDB, η αναζήτηση χρησιμοποιεί ευρετήριο HNSW (Hierarchical Navigable Small World), το οποίο προσφέρει αναζήτηση σε λογαριθμικό χρόνο ως προς το πλήθος N των αποθηκευμένων διανυσμάτων, δηλαδή $O(\log N)$ αντί της γραμμικής $O(N)$ μιας εξαντλητικής σάρωσης· στην εναλλακτική διαδρομή μνήμης (memory fallback, όταν το ChromaDB δεν είναι διαθέσιμο), η αναζήτηση είναι εξαντλητική και γραμμική ως προς το πλήθος των 357 τεμαχίων, κάτι απολύτως αποδεκτό σε αυτή την κλίμακα (η αναζήτηση ολοκληρώνεται σε λίγα χιλιοστά του δευτερολέπτου), αλλά που θα απαιτούσε αναθεώρηση αν ο κατάλογος παιχνιδιών επεκτεινόταν κατά τάξεις μεγέθους.

Το κόστος του σταδίου εμπλουτισμού και τεμαχισμού είναι γραμμικό ως προς το συνολικό μήκος κειμένου της βάσης γνώσης, $O(\Sigma|\text{έγγραφο}|)$, και εκτελείται μία φορά κατά την ανακατασκευή του ευρετηρίου --- όχι σε κάθε ερώτημα --- οπότε δεν επιβαρύνει τον χρόνο απόκρισης του χρήστη. Το στάδιο των αυστηρών φίλτρων μετά την ανάκτηση (έλεγχος κατωφλίου, αποκλεισμός άλλου παιχνιδιού, λεξιλογική επικάλυψη) λειτουργεί πάνω σε μικρό, ήδη περιορισμένο σύνολο υποψήφιων τεμαχίων (τυπικά 10--20 πλησιέστεροι γείτονες πριν το φιλτράρισμα), οπότε το κόστος του είναι αμελητέο σε σχέση με την ίδια την αναζήτηση ή με την επακόλουθη κλήση του γλωσσικού μοντέλου, η οποία κυριαρχεί στον συνολικό χρόνο απόκρισης του τελικού σημείου συνομιλίας, όπως τεκμηριώνεται στους χρόνους απόκρισης της ενότητας 5.8.

4.5 Υπηρεσία γλωσσικού μοντέλου

Η υπηρεσία LLM (llm_service.py) αναλαμβάνει τη σύνθεση της τελικής απάντησης από τα ανακτημένα αποσπάσματα. Όταν υπάρχει κλειδί OpenAI, καλείται το gpt-4o-mini με θερμοκρασία 0,25 και προτροπή συστήματος που επιβάλλει αυστηρή προσκόλληση: το μοντέλο απαντά μόνο από το παρεχόμενο πλαίσιο, διατηρεί ακριβείς αριθμούς και ονοματολογία, δεν επινοεί ονόματα μενού/αρχηγών/αντικειμένων, επιτρέπεται να δώσει μερική απάντηση σημειώνοντας τα κενά και, όταν το πλαίσιο δεν αφορά την ερώτηση, απαντά με το τυποποιημένο μήνυμα ανεπαρκούς πλαισίου. Αν το μοντέλο αρνηθεί παρότι το πλαίσιο προέρχεται από το σωστό παιχνίδι, εκτελείται μία

επανάκληση με ρητή οδηγία να απαντήσει από το πλαίσιο· διπλή άρνηση οδηγεί στις εναλλακτικές διαδρομές. Κάθε απάντηση φέρει ετικέτα τρόπου παραγωγής (mode), ώστε ο χρήστης και η αξιολόγηση να γνωρίζουν πάντοτε την προέλευσή της (Πίνακας 4.2).

Πίνακας 4.2 – Κύριοι τρόποι (modes) παραγωγής απάντησης της υπηρεσίας LLM

Mode	Συνθήκες	Χαρακτηριστικά
rag_openai	Κλειδί OpenAI + επαρκές πλαίσιο	Τεκμηριωμένη απάντηση LLM με παραθέσεις πηγών
rag_mock	Χωρίς κλειδί + επαρκές πλαίσιο	Ντετερμινιστική τοπική σύνθεση από τις πλέον συναφείς προτάσεις των πηγών
rag_mock_fallback	Σφάλμα κλήσης API	Τοπική σύνθεση ως δίκτυ ασφαλείας
insufficient_context	Κανένα τεμάχιο δεν επιβίωσε των φίλτρων	Ρητή άρνηση, μηδενικές πηγές
game_mismatch	Η ερώτηση αφορά άλλο παιχνίδι από το επιλεγμένο	Ενημέρωση χρήστη και πρόταση εναλλαγής παιχνιδιού
general_llm	Ρητή επιλογή γενικής λειτουργίας	Μη τεκμηριωμένη απάντηση με προειδοποιητική σήμανση
web_search_fallback	Προαιρετικά ενεργοποιημένη αναζήτηση ιστού	Απάντηση με έως 5 παραπομπές URL από αναζήτηση

Ο τοπικός ντετερμινιστικός συνθέτης (mock) αξίζει ιδιαίτερη μνεία, καθώς καθιστά το σύστημα πλήρως λειτουργικό εκτός σύνδεσης: εξάγει από τα τεμάχια πλήρεις, καθαρές προτάσεις, τις βαθμολογεί με βάση τη λεξιλογική επικάλυψη με την ερώτηση (με μόνους γειτνίασης), συνθέτει απάντηση με τις κορυφαίες προτάσεις και σημειώνει ρητά ποιοι όροι της ερώτησης δεν καλύπτονται από τις πηγές. Για ερωτήσεις βελτίωσης δεξιοτήτων προσθέτει πλάνο εξάσκησης τριών βημάτων. Επειδή είναι ντετερμινιστικός, τα αποτελέσματα της αξιολόγησης είναι πλήρως αναπαραγώγιμα.

Ενδεικτικό παράδειγμα από τη λειτουργία του συστήματος: στην ερώτηση «How do I stop panic rolling against bosses in Elden Ring?» με επιλεγμένο το Elden Ring, η ανάκτηση επέστρεψε τέσσερα αποσπάσματα από τους οδηγούς «Elden Ring Guide» και «Elden Ring Boss Strategies», και το μοντέλο συνέθεσε απάντηση πέντε βημάτων — παρατήρηση μοτίβων επίθεσης, αποφυγή στο πραγματικό χτύπημα αντί στην προετοιμασία του, καθυστερημένο dodge, αναβάθμιση χαρακτήρα, χρήση Spirit Ashes — κλείνοντας με ρητή δήλωση της πηγής («Source context: Elden Ring Boss Strategies»). Όλα τα στοιχεία της απάντησης ανάγονται σε προτάσεις των δύο οδηγών, όπως απαιτεί η προτροπή προσκόλλησης (βλ. Σχήμα 4.9 της ενότητας 4.8.3).

Η προτροπή συστήματος (system prompt) που καθοδηγεί το gpt-4o-mini αξίζει λεπτομερή ανάλυση, καθώς αποτελεί το σημείο όπου η αρχή της αυστηρής προσκόλλησης στην πηγή (grounding) μεταφράζεται σε συγκεκριμένες, ελέγξιμες οδηγίες φυσικής γλώσσας. Πέρα από τον περιορισμό να απαντά αποκλειστικά από το παρεχόμενο πλαίσιο, η προτροπή απαιτεί ρητά τη διατήρηση ακριβών αριθμητικών τιμών και ονοματολογίας όπως εμφανίζονται στην πηγή --- απαίτηση κρίσιμη σε ένα πεδίο όπου μια λανθασμένη παράφραση (π.χ. αντικατάσταση ενός συγκεκριμένου αριθμού πόντων ζωής ή ονόματος αντικειμένου με παρόμοιο αλλά εσφαλμένο) θα μπορούσε να παραπλανήσει τον παίκτη σε πραγματική συνθήκη παιχνιδιού. Η επιλογή θερμοκρασίας 0,25 αντί υψηλότερης τιμής εξυπηρετεί τον ίδιο στόχο: χαμηλή θερμοκρασία περιορίζει τη στοχαστικότητα της δειγματοληψίας εξόδου, ευνοώντας τις πλέον πιθανές (και άρα πλέον πιστές στο πλαίσιο) συνεχίσεις κειμένου έναντι

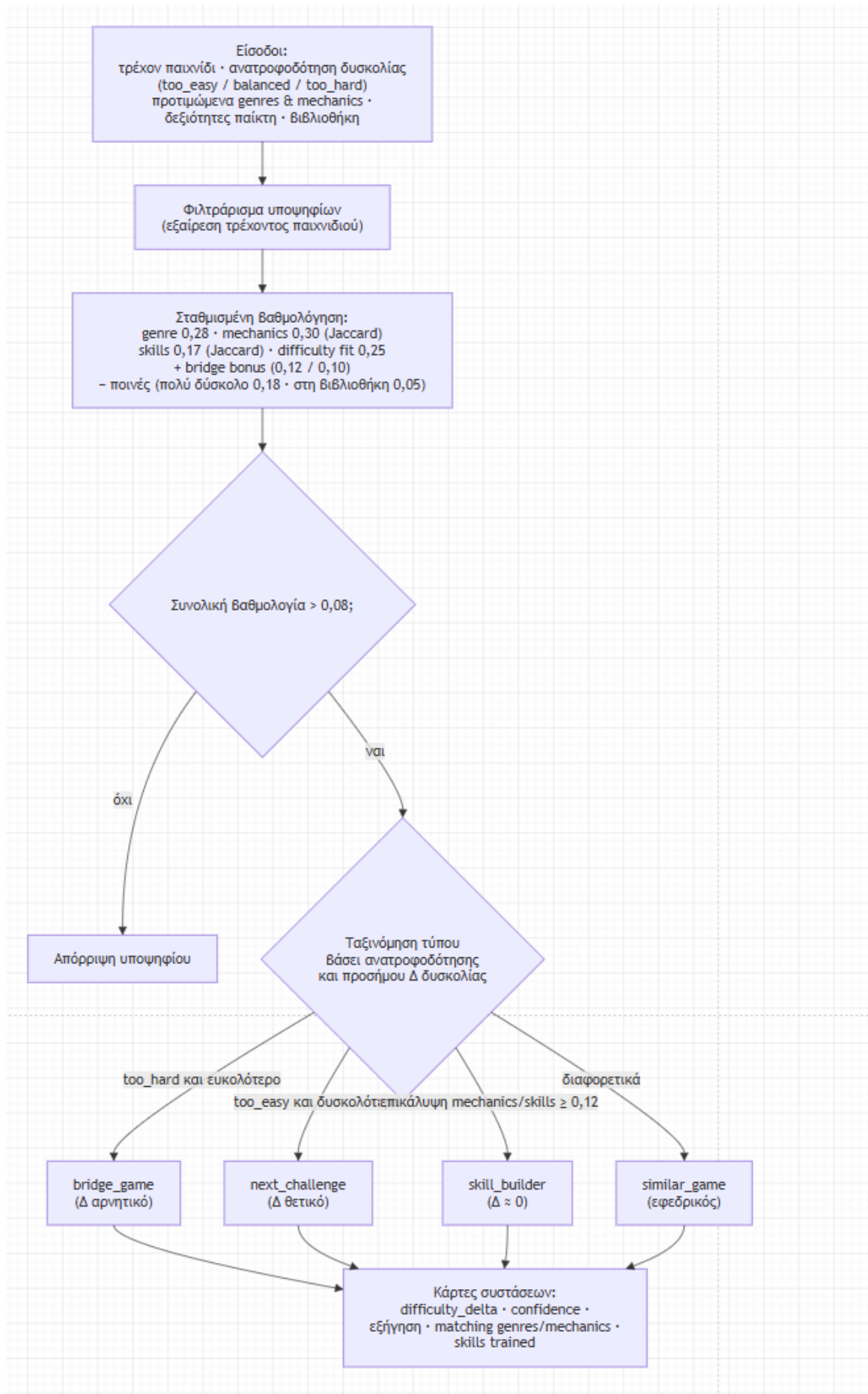
δημιουργικών αλλά ενδεχομένως ανακριβών διατυπώσεων --- στάθμιση ρητά αντίθετη προς τυπικές εφαρμογές δημιουργικής γραφής, όπου επιδιώκεται το αντίστροφο.

Ο μηχανισμός μίας επανάκλησης (single retry) σε περίπτωση άρνησης του μοντέλου παρά την ύπαρξη έγκυρου πλαισίου αντιμετωπίζει ένα γνωστό φαινόμενο των LLMs: την υπερβολική συντηρητικότητα (over-refusal), όπου το μοντέλο αρνείται να απαντήσει ακόμη κι όταν η πληροφορία διατίθεται, συχνά λόγω ασάφειας στη διατύπωση της ερώτησης του χρήστη σε σχέση με το πλαίσιο. Η επανάκληση με ρητή, πιο κατευθυντική οδηγία («απάντησε αποκλειστικά με βάση το ακόλουθο πλαίσιο, το οποίο αφορά το σωστό παιχνίδι») επιλύει την πλειονότητα αυτών των περιπτώσεων χωρίς να θυσιάζεται η αρχή της προσκόλλησης· μόνο διπλή άρνηση --- ένδειξη ότι το πλαίσιο όντως δεν επαρκεί, όχι απλή σύγχυση του μοντέλου --- οδηγεί στις εναλλακτικές διαδρομές του Πίνακα 4.2. Η προσέγγιση αυτή αντικατοπτρίζεται ποσοτικά στην ενότητα 5.9, όπου καταγράφεται το ποσοστό ερωτήσεων που χρειάστηκαν επανάκληση.

Η ετικέτα mode που συνοδεύει κάθε απάντηση δεν είναι απλή μεταδεδομένη καταγραφή, αλλά ενεργό στοιχείο της αρχιτεκτονικής διαφάνειας του συστήματος: μεταφέρεται αυτούσια μέχρι τη διεπαφή χρήστη (ενότητα 4.8.6), όπου αντιστοιχίζεται σε οπτική ένδειξη χρώματος, και χρησιμοποιείται από τη σουίτα αξιολόγησης για τον αυτόματο υπολογισμό του ποσοστού τεκμηριωμένων έναντι μη τεκμηριωμένων απαντήσεων --- μία από τις βασικές μετρικές του Κεφαλαίου 5. Η ύπαρξη ξεχωριστών mode για rag_openai και rag_mock_fallback επιτρέπει επιπλέον διάκριση, στα δεδομένα καταγραφής, μεταξύ «κανονικής» λειτουργίας υψηλής ποιότητας και υποβαθμισμένης λειτουργίας λόγω παροδικού σφάλματος API --- διάκριση χρήσιμη για την παρακολούθηση της υγείας του συστήματος σε πραγματική χρήση, ακόμη κι αν αμφότερες οι διαδρομές παραμένουν εν τέλει τεκμηριωμένες.

4.6 Μηχανή συστάσεων

Η μηχανή συστάσεων (recommendation_service.py) υλοποιεί σύσταση βασισμένη στο περιεχόμενο με παιδαγωγική διάσταση δυσκολίας. Η ροή απεικονίζεται στο Σχήμα 4.2.



Σχήμα 4.2 – Ροή της μηχανής συστάσεων βασισμένων στο περιεχόμενο

Από το αίτημα του χρήστη (παιχνίδι αναφοράς, εμπειρία δυσκολίας, προτιμώμενα είδη, μηχανικές και δεξιότητες) και το προφίλ/βιβλιοθήκη του συντίθενται σύνολα-στόχοι χαρακτηριστικών. Αν δεν δηλωθούν προτιμήσεις, τα είδη συνάγονται από τα συχνότερα είδη της βιβλιοθήκης. Κάθε υποψήφιο παιχνίδι του καταλόγου βαθμολογείται με το σταθμισμένο άθροισμα της σχέσης (4.1):

$$score = 0.28 \cdot g + 0.30 \cdot m + 0.17 \cdot s + 0.25 \cdot d + b - p_hard - p_owned \quad (4.1)$$

όπου g η δυαδική επικάλυψη είδους/ετικετών, m και s οι δείκτες Jaccard (σχέση (2.3)) των συνόλων μηχανικών και δεξιοτήτων αντίστοιχα, d ο βαθμός καταλληλότητας δυσκολίας ως προς το ιδανικό εύρος, b μπόνους γέφυρας/πρόκλησης (0,12 ή 0,10), p_hard ποινή 0,18 για παιχνίδια που ο χρήστης έχει δηλώσει υπερβολικά δύσκολα και p_owned ποινή 0,05 για ήδη αποκτημένα. Το ιδανικό εύρος δυσκολίας εξαρτάται από την ανατροφοδότηση: για «πολύ δύσκολο» στοχεύονται τίτλοι 1–4 μονάδες ευκολότεροι από το παιχνίδι αναφοράς, για «πολύ εύκολο» αυστηρά δυσκολότεροι, για «ισορροπημένο» ± 1 μονάδα. Κάθε πρόταση ταξινομείται σε τύπο — `bridge_game`, `next_challenge`, `skill_builder` ή `similar_game` — και συνοδεύεται από εξήγηση φυσικής γλώσσας που κατονομάζει τις κοινές μηχανικές, τη διαφορά δυσκολίας (`difficulty_delta`) και τις δεξιότητες που εξασκεί. Το αποτέλεσμα περιλαμβάνει έως 6 κύριες συστάσεις, έως 4 παιχνίδια-γέφυρες και μονοπάτι προόδου τριών βημάτων, ενώ κάθε εκτέλεση αποθηκεύεται στο ιστορικό.

Οι συντελεστές στάθμισης της σχέσης (4.1) --- 0,28 για τη γενική επικάλυψη είδους/ετικετών, 0,30 για τις κοινές μηχανικές, 0,17 για τις κοινές δεξιότητες και 0,25 για την καταλληλότητα δυσκολίας --- δεν προέκυψαν από μάθηση πάνω σε δεδομένα (δεν υπάρχει σύνολο εκπαίδευσης με ετικέτες «καλής» σύστασης), αλλά από παιδαγωγικό σχεδιασμό βασισμένο στην ερευνητική υπόθεση της εργασίας: ότι η ομοιότητα μηχανικών παιχνιδιού αποτελεί ισχυρότερο δείκτη μεταφοράς δεξιοτήτων (*skill transfer*) από την επιφανειακή ομοιότητα είδους. Η ελαφρώς υψηλότερη στάθμιση των μηχανικών (0,30) έναντι του είδους (0,28) αντικατοπτρίζει αυτή την υπόθεση --- δύο παιχνίδια διαφορετικού είδους αλλά κοινών μηχανικών (π.χ. ένα *immersive-sim* και ένα ανοιχτού κόσμου με στοιχεία κρυφής προσέγγισης) μπορεί να απαιτούν παρόμοιες δεξιότητες παίκτη περισσότερο από δύο παιχνίδια του ίδιου είδους με ριζικά διαφορετικές μηχανικές. Η στάθμιση 0,17 των δεξιοτήτων, χαμηλότερη των υπολοίπων, αντανακλά ότι το πεδίο `required_skills` είναι το πιο αραιό (*sparse*) μεταξύ των μεταδεδομένων παιχνιδιού --- λιγότερες τιμές ανά παιχνίδι σε σχέση με τις μηχανικές --- οπότε ο δείκτης Jaccard του παρουσιάζει μεγαλύτερη διακύμανση και θα κυριαρχούσε υπερβολικά στο τελικό σκορ αν σταθμιζόταν ισοβαρώς.

Οι ποινές p_hard και p_owned λειτουργούν ως διορθωτικοί όροι πάνω στο βασικό σκορ ομοιότητας, όχι ως σκληροί αποκλεισμοί (*hard filters*) --- σχεδιαστική επιλογή που αφήνει περιθώριο ένα παιχνίδι να εμφανιστεί ακόμη κι αν ο χρήστης το κατέχει ήδη ή το έχει χαρακτηρίσει υπερβολικά δύσκολο, αρκεί η συνολική καταλληλότητά του να είναι αρκετά υψηλή ώστε να υπερκαλύψει την ποινή. Η επιλογή αυτή αποφεύγει το πρόβλημα της υπερβολικά αυστηρής διήθησης (*over-filtering*), όπου ένα έγκυρο, ενδεχομένως εξαιρετικό ταίριασμα θα αποκλειόταν οριστικά λόγω ενός και μόνο κριτηρίου. Το μπόνους γέφυρας/πρόκλησης (0,12 ή 0,10 αναλόγως τύπου) εφαρμόζεται όταν η διαφορά δυσκολίας βρίσκεται στο ιδανικό εύρος, ενισχύοντας ρητά τη διάκριση μεταξύ των τύπων σύστασης `bridge_game` και `next_challenge` --- ο πρώτος στοχεύει τίτλο ελαφρώς ευκολότερο για την οικοδόμηση αυτοπεποίθησης πριν από μια δύσκολη πρόκληση, ο δεύτερος στοχεύει τίτλο ελαφρώς δυσκολότερο για προοδευτική εξέλιξη δεξιοτήτων.

Πίνακας 4.2α -- Ενδεικτικό παράδειγμα υπολογισμού σκορ σύστασης

Συνιστώσα	Τιμή	Στάθμιση	Συνεισφορά
Επικάλυψη είδους/ετικετών (g)	1,00	0,28	0,280
Δείκτης Jaccard μηχανικών (m)	0,50	0,30	0,150
Δείκτης Jaccard δεξιοτήτων (s)	0,33	0,17	0,056
Καταλληλότητα δυσκολίας (d)	0,80	0,25	0,200
Μπόνους γέφυρας (b)	—	—	+0,120
Ποινή ήδη κατεχόμενου (p_owned)	—	—	0,000
Τελικό score			0,806

Το παράδειγμα του Πίνακα 4.2α απεικονίζει πώς ένα παιχνίδι με μέτρια επικάλυψη μηχανικών ($m = 0,50$) αλλά τέλεια επικάλυψη είδους και υψηλή καταλληλότητα δυσκολίας μπορεί να συγκεντρώσει υψηλό συνολικό σκορ χάρη στο σωρευτικό αποτέλεσμα των επιμέρους όρων --- ακριβώς η συμπεριφορά «μαλακής» στάθμισης που περιγράφηκε παραπάνω, σε αντιδιαστολή με ένα σύστημα αυστηρών κατωφλίων ανά κριτήριο.

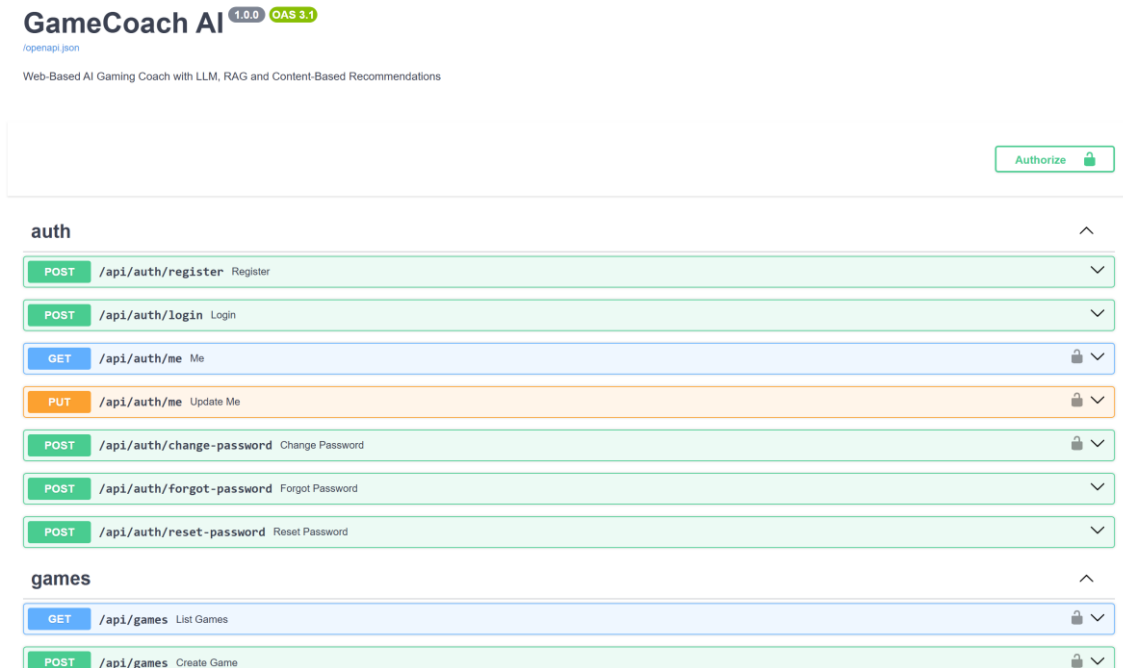
4.7 REST API και ροή συνομιλίας

Οι επτά δρομολογητές του Πίνακα 3.4 υλοποιούν συνολικά 30 και πλέον τελικά σημεία (πλήρης κατάλογος στο Παράρτημα Β). Η πλέον σύνθετη ροή είναι αυτή της συνομιλίας: το POST /api/chat επιλύει ή δημιουργεί τη συνομιλία, εφαρμόζει τον περιορισμό ρυθμού, εκτελεί τον έλεγχο αναντιστοιχίας παιχνιδιού, καλεί την ανάκτηση με $k=4$, ζητά από την υπηρεσία LLM τη σύνθεση απάντησης και αποθηκεύει μήνυμα χρήστη και βοηθού μαζί με τα αποσπάσματα πηγών και την ετικέτα mode. Το frontend λαμβάνει έτσι σε μία απόκριση όλα όσα χρειάζεται για πλήρη διαφάνεια: κείμενο, πηγές με βαθμούς συνάφειας, τρόπο παραγωγής και τυχόν ανιχνευθέν σωστό παιχνίδι.

Η σχεδίαση του REST API ακολουθεί συνεπείς συμβάσεις σε όλους τους δρομολογητές: πόροι στον πληθυντικό (π.χ. /api/games, /api/library), ρήματα HTTP που αντιστοιχούν σημασιολογικά στην ενέργεια (GET για ανάκτηση, POST για δημιουργία, PUT/PATCH για ενημέρωση, DELETE για διαγραφή), και ενιαία μορφή απόκρισης σφάλματος μέσω των εξαιρέσεων HTTPException του FastAPI, οι οποίες μετατρέπονται αυτόματα σε JSON απαντήσεις με πεδίο detail --- μορφή που το frontend αναμένει ομοιόμορφα σε κάθε κλήση του Axios, απλοποιώντας τον χειρισμό σφαλμάτων σε ένα και μόνο σημείο (τον interceptor απόκρισης). Η επικύρωση εισόδου σε κάθε τελικό σημείο γίνεται δηλωτικά μέσω σχημάτων Pydantic, τα οποία εφαρμόζουν αυτόματα ελέγχους τύπου, ελάχιστου/μέγιστου μήκους και υποχρεωτικών πεδίων πριν η εκτέλεση φτάσει καν στη λογική του δρομολογητή --- αν η επικύρωση αποτύχει, το FastAPI επιστρέφει αυτόματα 422 Unprocessable Entity με λεπτομερή περιγραφή του σφάλματος ανά πεδίο, χωρίς να απαιτείται χειρωνακτικός κώδικας ελέγχου.

Η αυτόματη παραγωγή του σχήματος OpenAPI από το FastAPI --- ορατή στο Σχήμα 4.3 μέσω του Swagger UI --- δεν είναι απλώς βοήθημα τεκμηρίωσης αλλά ενεργό εργαλείο ανάπτυξης και δοκιμής: κάθε τελικό σημείο, το σχήμα εισόδου/εξόδου του και οι πιθανοί κωδικοί κατάστασης παράγονται

απευθείας από τις υπογραφές συναρτήσεων Python και τα σχήματα Pydantic, οπότε η τεκμηρίωση παραμένει πάντοτε συγχρονισμένη με τον πραγματικό κώδικα --- σε αντίθεση με χειρωνακτικά συντηρούμενη τεκμηρίωση, η οποία τείνει να αποκλίνει από την υλοποίηση με την πάροδο του χρόνου. Το πλήρες σχήμα, το οποίο εξάγεται και παρατίθεται αναλυτικά στο Παράρτημα Β, χρησιμοποιήθηκε επίσης ως σημείο αναφοράς κατά τη σύνταξη της σουίτας δοκιμών του Κεφαλαίου 5, εξασφαλίζοντας ότι κάθε τεκμηριωμένο τελικό σημείο καλύπτεται τουλάχιστον από μία αυτοματοποιημένη δοκιμή.



Σχήμα 4.3 – Αυτόματη διαδραστική τεκμηρίωση του API (Swagger UI / OpenAPI)

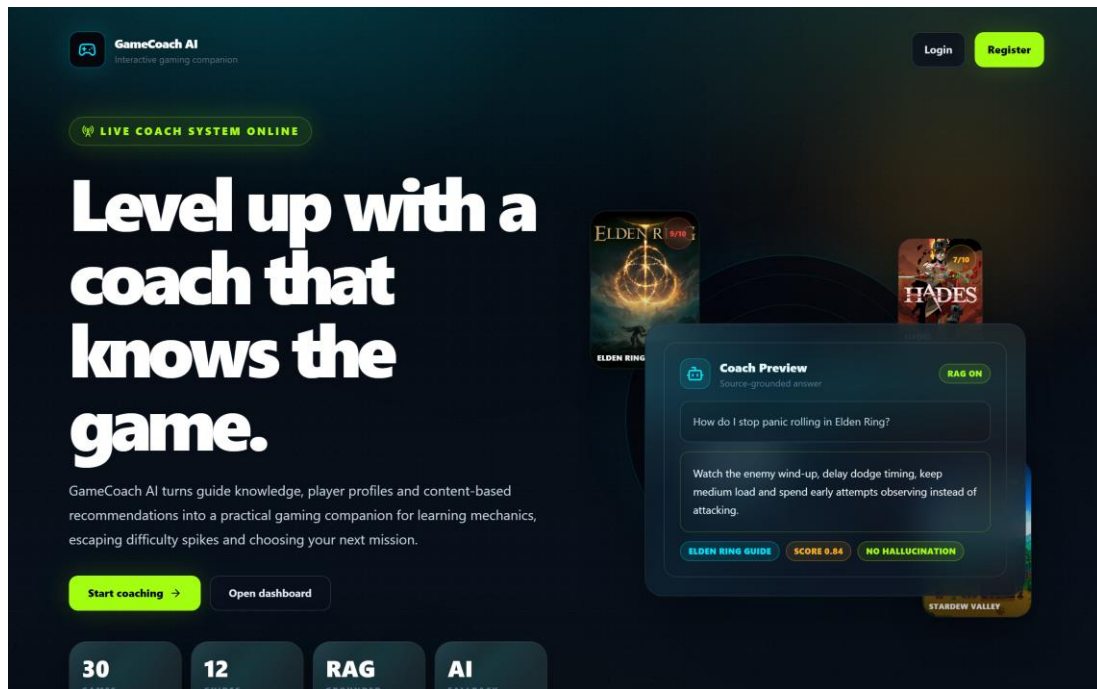
4.8 Διεπαφή χρήστη

Η διεπαφή υλοποιήθηκε ως εφαρμογή React 18/TypeScript με αισθητική «κονσόλας παιχνιδιών»: σκούρα παλέτα με τόνους κυανού και πράσινου νέον, κάρτες με εφέ λάμψης και εικονίδια lucide-react. Η κατάσταση αυθεντικοποίησης διατηρείται σε React Context με αποθήκευση του JWT σε localStorage ή sessionStorage αναλόγως της επιλογής «Να με θυμάσαι», ενώ interceptor του Axios επισυνάπτει αυτόματα το token και ανακατευθύνει στη σύνδεση σε απόκριση 401. Ακολουθεί ξενάγηση στις κύριες οθόνες.

Η αρχιτεκτονική του frontend ακολουθεί δομή βασισμένη σε χαρακτηριστικά (feature-based), όπου κάθε λειτουργική περιοχή (αυθεντικοποίηση, βιβλιοθήκη, συνομιλία, συστάσεις, διαχείριση) διατηρεί τα δικά της components, hooks και τύπους TypeScript σε ξεχωριστό υποκατάλογο, με κοινά στοιχεία (κουμπιά, κάρτες, πλαίσια φόρτωσης) σε διαμοιραζόμενη βιβλιοθήκη components. Η επικοινωνία με το backend γίνεται αποκλειστικά μέσω ενός στρώματος υπηρεσιών (api client), το οποίο περιτυλίγει το Axios και εκθέτει τυποποιημένες συναρτήσεις ανά πόρο (π.χ. fetchLibrary, sendChatMessage) --- έτσι κανένα React component δεν καλεί απευθείας το Axios, με αποτέλεσμα μια μελλοντική αλλαγή του πρωτοκόλλου επικοινωνίας (π.χ. προσθήκη GraphQL) να περιορίζεται σε ένα μόνο στρώμα. Η διαχείριση τοπικής κατάστασης βασίζεται σχεδόν αποκλειστικά σε React hooks (useState, useEffect, useContext) χωρίς εξωτερική βιβλιοθήκη διαχείρισης κατάστασης όπως το Redux --- επαρκής επιλογή

δεδομένου του περιορισμένου βάθους κοινής κατάστασης της εφαρμογής (κυρίως η ταυτότητα του συνδεδεμένου χρήστη), η οποία εξυπηρετείται πλήρως από το AuthContext.

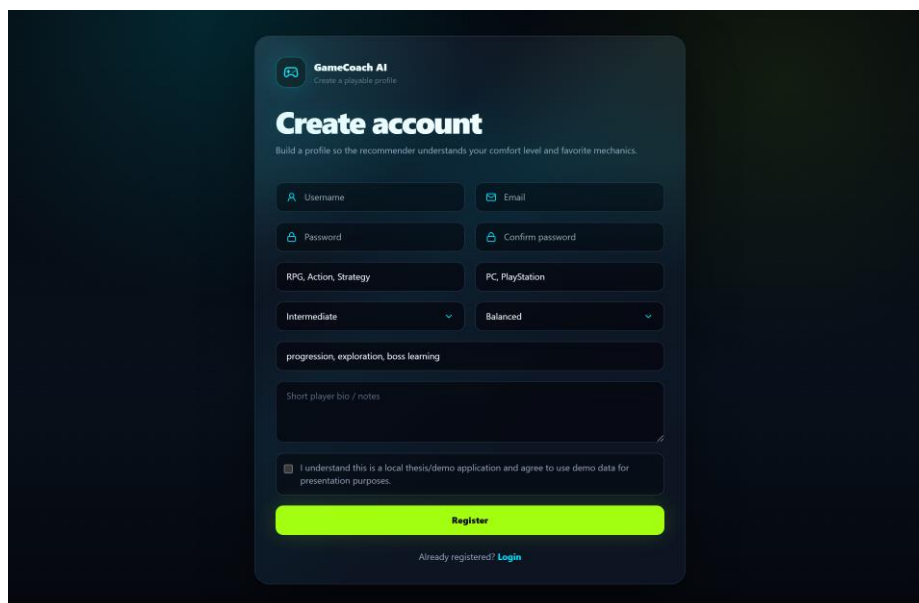
4.8.1 Αρχική σελίδα και αυθεντικοποίηση



Σχήμα 4.4 – Δημόσια αρχική σελίδα της εφαρμογής

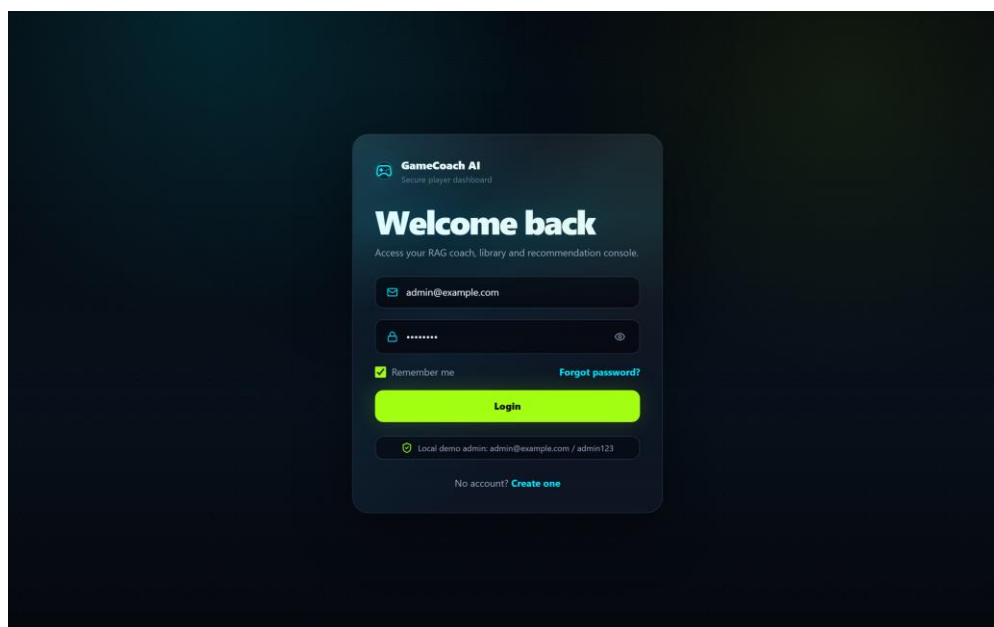
Η δημόσια αρχική σελίδα (Σχήμα 4.4) παρουσιάζει τη λειτουργικότητα της πλατφόρμας και οδηγεί σε εγγραφή ή σύνδεση. Η φόρμα εγγραφής (Σχήμα 4.5) συλλέγει, πέρα από τα διαπιστευτήρια, πλήρες προφίλ παίκτη — προτιμώμενα είδη, επίπεδο δεξιοτήτων, άνεση με τη δυσκολία, πλατφόρμες — από ελεγχόμενα λεξιλόγια, στοιχεία που τροφοδοτούν τη μηχανή συστάσεων. Η σύνδεση (Σχήμα 4.6) υποστηρίζει εμφάνιση/απόκρυψη κωδικού, «Να με θυμάσαι» και τοπική ροή επαναφοράς κωδικού.

Η επικύρωση της φόρμας εγγραφής πραγματοποιείται σε δύο επίπεδα, ακολουθώντας κοινή πρακτική ασφαλών εφαρμογών ιστού: στο frontend, άμεση οπτική ανατροφοδότηση (π.χ. ένδειξη ισχύος κωδικού, επισήμανση υποχρεωτικών πεδίων) βελτιώνει την εμπειρία χρήστη χωρίς περιττό ταξίδι στο δίκτυο· στο backend, η ίδια επικύρωση επαναλαμβάνεται ανεξάρτητα μέσω σχημάτων Pydantic, καθώς η επικύρωση αποκλειστικά στο frontend είναι εξ ορισμού παρακάμψιμη από οποιονδήποτε πελάτη που καλεί απευθείας το API. Η αρχή «ποτέ μην εμπιστεύεσαι τον πελάτη» εφαρμόζεται συνεπώς σε ολόκληρη την εφαρμογή, συμπεριλαμβανομένων των ελεγχόμενων λεξιλογίων προφίλ παίκτη --- ακόμη κι αν η διεπαφή περιορίζει τις επιλογές σε προκαθορισμένες τιμές, το backend δεν υποθέτει ότι κάθε εισερχόμενη τιμή προέρχεται από αυτά τα λεξιλόγια.



The screenshot shows a 'Create account' form for GameCoach AI. The form is titled 'Create account' with a subtitle 'Build a profile so the recommender understands your comfort level and favorite mechanics.' It includes input fields for Username, Email, Password, and Confirm password. There are also dropdown menus for game genres (RPG, Action, Strategy) and platforms (PC, PlayStation). A section for 'Intermediate' and 'Balanced' preferences is present, along with a text area for 'Short player bio / notes'. A checkbox for terms and conditions is at the bottom, followed by a bright green 'Register' button and a link to 'Login' for already registered users.

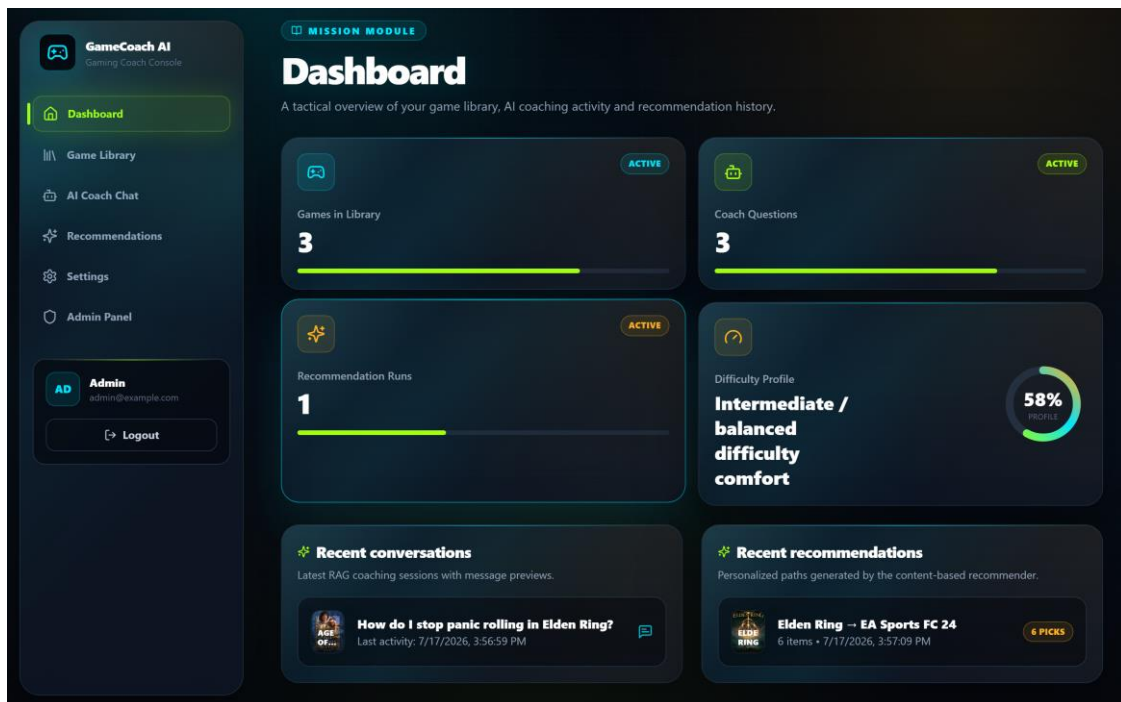
Σχήμα 4.5 – Φόρμα εγγραφής με πεδία προφίλ παίκτη



The screenshot shows a 'Welcome back' login form for GameCoach AI. The form is titled 'Welcome back' with a subtitle 'Access your RAG coach, library and recommendation console.' It includes input fields for Email (admin@example.com) and Password (masked with dots). There is a 'Remember me' checkbox and a 'Forgot password?' link. A bright green 'Login' button is prominently displayed. Below the button, there is a 'Local demo' option with credentials: admin: admin@example.com / admin123. At the bottom, there is a link to 'Create one' for users without an account.

Σχήμα 4.6 – Οθόνη σύνδεσης χρήστη

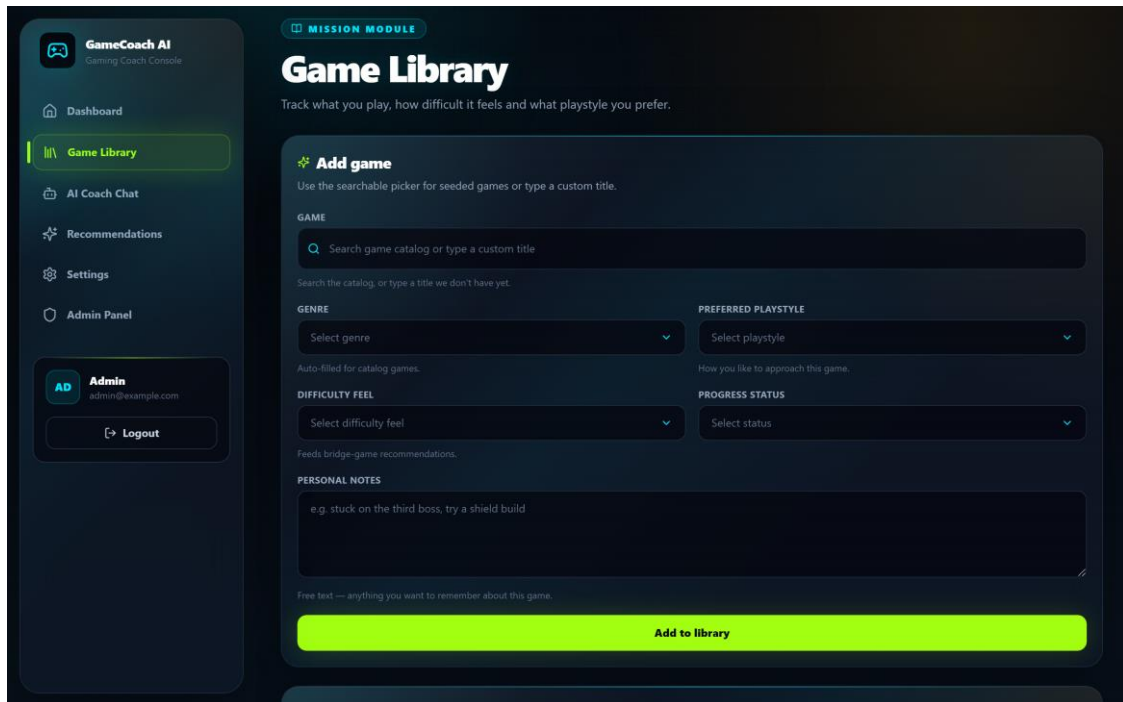
4.8.2 Πίνακας ελέγχου και βιβλιοθήκη



Σχήμα 4.7 – Πίνακας ελέγχου χρήστη με στατιστικά και πρόσφατη δραστηριότητα

Ο πίνακας ελέγχου (Σχήμα 4.7) συνοψίζει το μέγεθος της βιβλιοθήκης, το πλήθος ερωτήσεων προς τον προπονητή, τις εκτελέσεις συστάσεων και το συναγόμενο προφίλ δυσκολίας του παίκτη, μαζί με προεπισκοπήσεις πρόσφατων συνομιλιών και συστάσεων. Η βιβλιοθήκη (Σχήμα 4.8) εμφανίζει τα παιχνίδια σε κάρτες με εξώφυλλα, υποστηρίζει φιλτράρισμα ανά είδος και κατάσταση προόδου, ταξινόμηση και πλήρη κύκλο προσθήκης/επεξεργασίας/διαγραφής με επιβεβαίωση.

Η ενημέρωση του πίνακα ελέγχου ακολουθεί μοτίβο βελτιστοποιημένης ανάκτησης: αντί ο πελάτης να εκτελεί ξεχωριστές κλήσεις για κάθε στατιστικό (μέγεθος βιβλιοθήκης, πλήθος ερωτήσεων, κ.ο.κ.), το τελικό σημείο `/api/dashboard` συγκεντρώνει όλα τα απαραίτητα στοιχεία σε μία μόνο απόκριση μέσω της υπηρεσίας `stats_service`, μειώνοντας τον αριθμό ταξιδιών δικτύου (round trips) από ενδεχομένως έξι ή επτά σε ένα. Η ίδια αρχή συγκεντρωτικής απόκρισης εφαρμόζεται σταθερά σε ολόκληρο το API --- το τελικό σημείο συνομιλίας του κεφαλαίου 4.7 ακολουθεί ακριβώς την ίδια λογική, επιστρέφοντας κείμενο, πηγές και `mode` σε ένα αίτημα. Η βιβλιοθήκη υποστηρίζει αισιόδοξη ενημέρωση (optimistic update) κατά τη διαγραφή ενός τίτλου: η κάρτα αφαιρείται άμεσα από τη διεπαφή πριν καν ολοκληρωθεί η κλήση προς το backend, με αυτόματη επαναφορά αν η κλήση αποτύχει --- τεχνική που βελτιώνει την αντιληπτή ταχύτητα της εφαρμογής χωρίς να θυσιάζει τη συνέπεια δεδομένων.

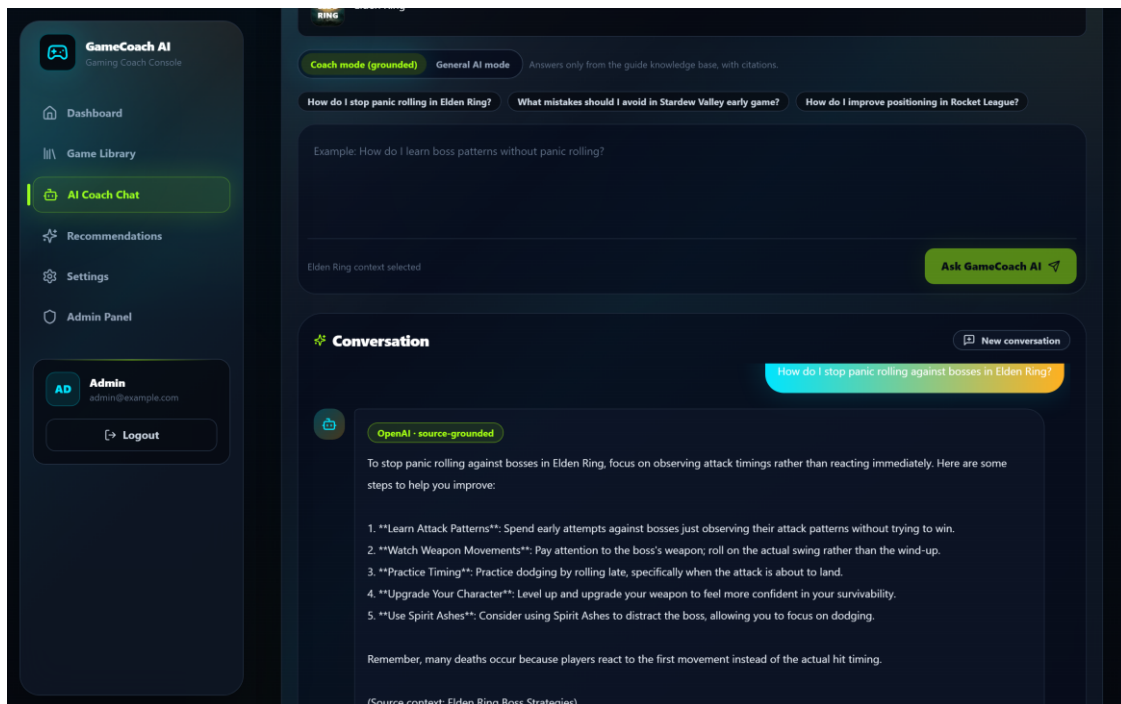


Σχήμα 4.8 – Προσωπική βιβλιοθήκη παιχνιδιών με φίλτρα και κάρτες τίτλων

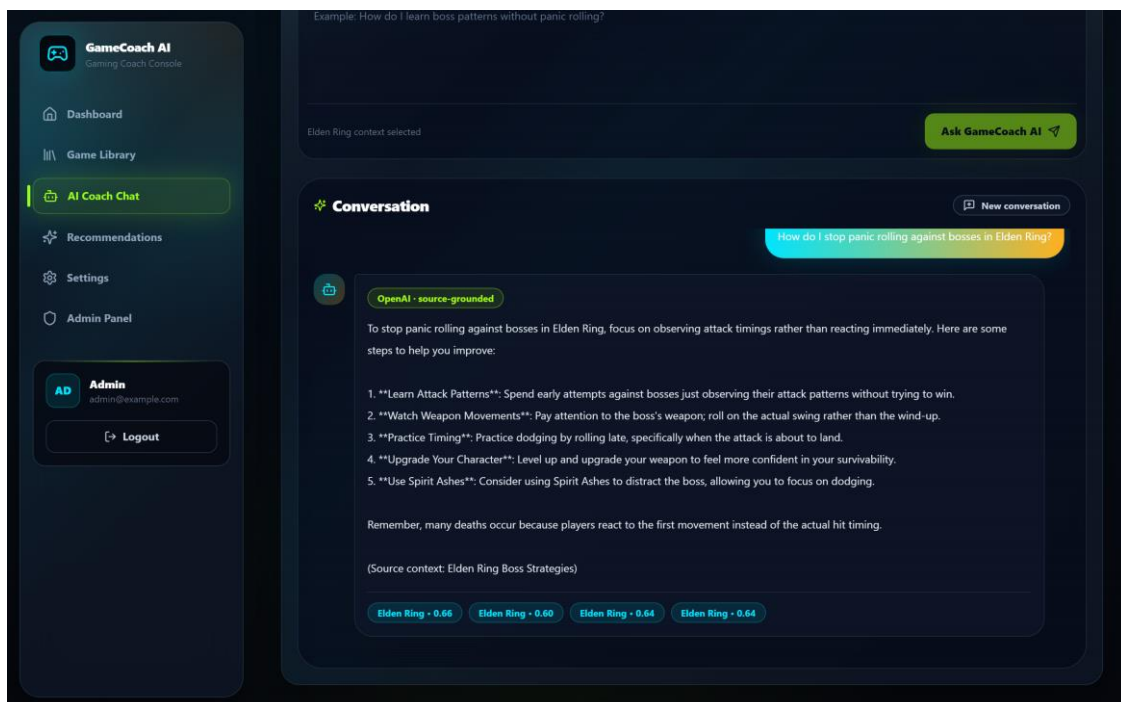
4.8.3 Συνομιλία με τον προπονητή

Η οθόνη συνομιλίας (Σχήμα 4.9) αποτελεί την καρδιά της εφαρμογής. Ο χρήστης επιλέγει παιχνιδι-πλαίσιο, εναλλάσσεται μεταξύ τεκμηριωμένης και γενικής λειτουργίας (η δεύτερη με εμφανή προειδοποίηση και μόνο εφόσον ο διακομιστής διαθέτει κλειδί), και βλέπει κάθε απάντηση με σήμα προέλευσης — στο στιγμιότυπο, «OpenAI · source-grounded». Στο κάτω μέρος κάθε απάντησης εμφανίζονται πλακίδια πηγών με τον τίτλο του οδηγού και τον βαθμό συνάφειας κάθε αποσπάσματος (Σχήμα 4.10)· σε αιώρηση εμφανίζεται το ίδιο το απόσπασμα, ώστε ο χρήστης να επαληθεύει την τεκμηρίωση. Σε ερώτηση εκτός πεδίου, το σύστημα εξηγεί ρητά ότι δεν διαθέτει τεκμηριωμένο υλικό, ενώ σε αναντιστοιχία παιχνιδιού προτείνει εναλλαγή και επανυποβολή με ένα κλικ.

Η ροή αποστολής μηνύματος στη διεπαφή συνομιλίας σχεδιάστηκε ώστε να διατηρεί την αίσθηση απόκρισης παρά τον αναπόφευκτο χρόνο αναμονής της κλήσης προς το γλωσσικό μοντέλο: το μήνυμα του χρήστη εμφανίζεται αμέσως στη ροή συνομιλίας (πριν καν επιστρέψει η απόκριση του διακομιστή), ακολουθούμενο από κινούμενη ένδειξη πληκτρολόγησης, ενώ το πεδίο εισαγωγής και το κουμπί αποστολής απενεργοποιούνται προσωρινά για την αποτροπή διπλής υποβολής. Σε περίπτωση σφάλματος δικτύου ή διακομιστή, εμφανίζεται μήνυμα σφάλματος εντός της ίδιας ροής συνομιλίας --- όχι ως αναδυόμενο παράθυρο ειδοποίησης --- με δυνατότητα επανάληψης της αποστολής με ένα κλικ, ώστε το πλαίσιο της συζήτησης να μη χάνεται. Η επιλογή γενικής λειτουργίας εμφανίζεται μόνο όταν ο διακομιστής έχει δηλώσει διαθεσιμότητα κλειδιού OpenAI --- πληροφορία που μεταφέρεται στο frontend κατά την αρχική φόρτωση της εφαρμογής --- ώστε να μην προσφέρεται στον χρήστη μια επιλογή που θα απέτυχε στην πράξη.



Σχήμα 4.9 – Τεκμηριωμένη απάντηση του προπονητή για το Elden Ring με σήμανση προέλευσης

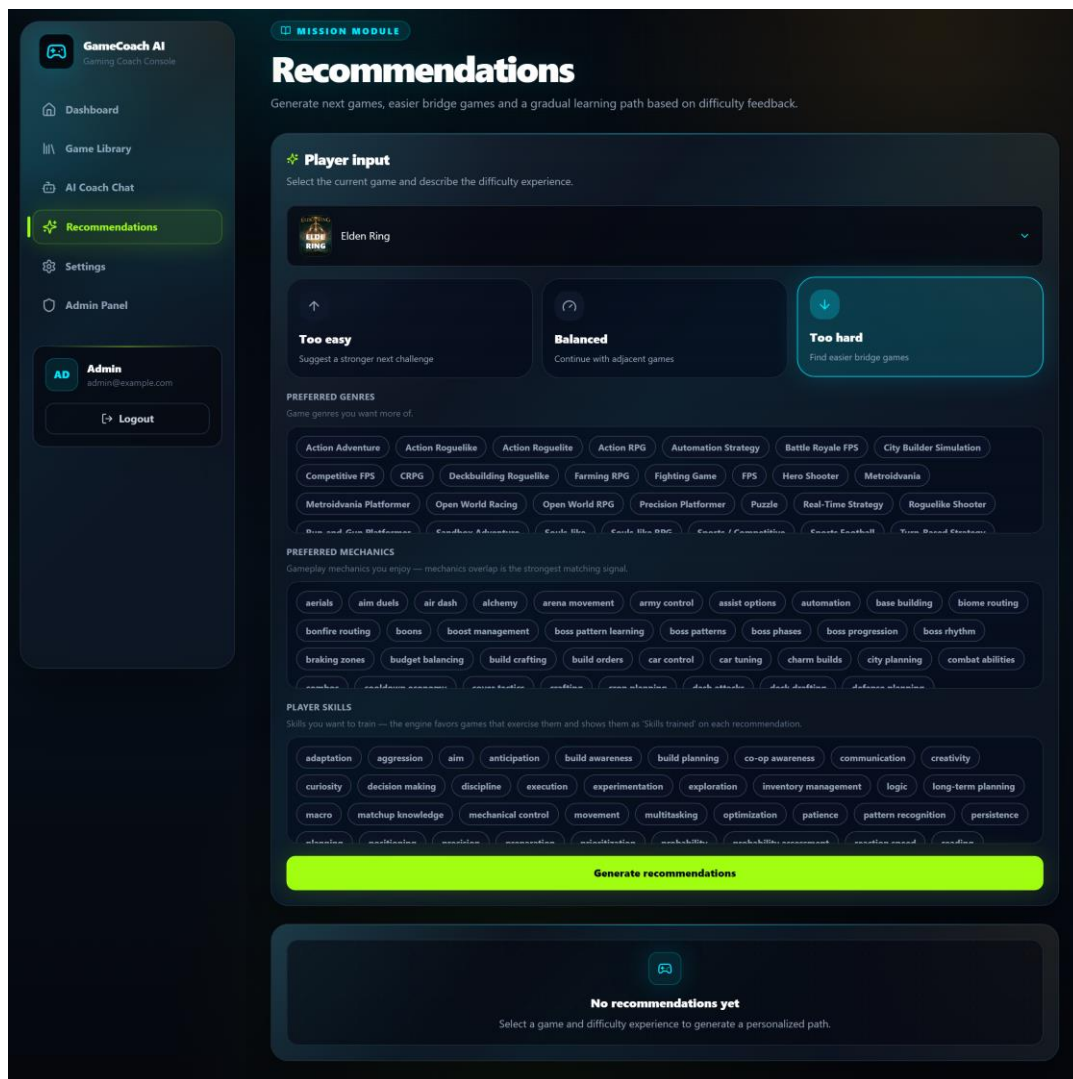


Σχήμα 4.10 – Πλακίδια πηγών με βαθμούς συνάφειας στο τέλος της απάντησης

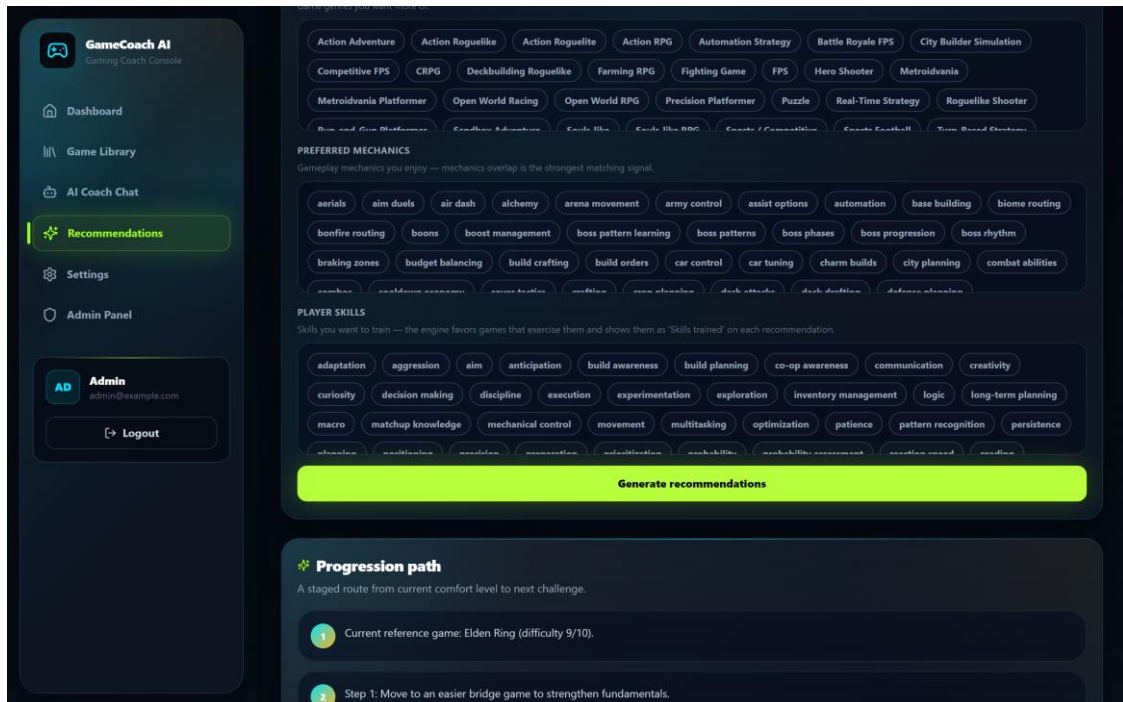
4.8.4 Συστάσεις παιχνιδιών

Στην οθόνη συστάσεων (Σχήμα 4.11) ο χρήστης επιλέγει παιχνίδι αναφοράς, δηλώνει την εμπειρία δυσκολίας του με τρεις κάρτες (πολύ εύκολο / ισορροπημένο / πολύ δύσκολο) και προαιρετικά επιλέγει είδη, μηχανικές και δεξιότητες από πλακίδια πολλαπλής επιλογής. Το αποτέλεσμα (Σχήμα 4.12) περιλαμβάνει το μονοπάτι προόδου, τα παιχνίδια-γέφυρες και τις κύριες προτάσεις, καθεμία με ποσοστό ταιριάσματος, διαφορά δυσκολίας, εξήγηση και τις δεξιότητες που εξασκεί — με επισήμανση όσων επέλεξε ο χρήστης.

Η επιλογή τριών διακριτών καρτών δυσκολίας (πολύ εύκολο / ισορροπημένο / πολύ δύσκολο) αντί αριθμητικής κλίμακας συρόμενου ελέγχου (slider) αποτελεί συνειδητή απόφαση ευχρηστίας: μια κλίμακα 1--10 θα απαιτούσε από τον χρήστη να μεταφράσει υποκειμενική εμπειρία σε αριθμό, διαδικασία επιρρεπής σε ασυνέπεια μεταξύ χρηστών (τι σημαίνει «6» για διαφορετικούς παίκτες;), ενώ οι τρεις ποιοτικές κατηγορίες αντιστοιχούν απευθείας στη λογική διακλάδωσης του αλγορίθμου σύστασης της ενότητας 4.6 (ιδανικό εύρος 1--4 μονάδες ευκολότερο, αυστηρά δυσκολότερο, ή ± 1 μονάδα), χωρίς ενδιάμεσο στάδιο μετάφρασης. Το ποσοστό ταιριάσματος που εμφανίζεται σε κάθε πρόταση υπολογίζεται απευθείας από το κανονικοποιημένο score της σχέσης (4.1), μεταφρασμένο σε ποσοστό επί τοις εκατό για ευκολότερη ερμηνεία από μη τεχνικό κοινό --- επιλογή διαφάνειας που επιτρέπει στον χρήστη να αντιληφθεί ότι πρόκειται για αλγοριθμική εκτίμηση και όχι απόλυτη βεβαιότητα.



Σχήμα 4.11 – Φόρμα αιτήματος συστάσεων με πλακίδια προτιμήσεων



Σχήμα 4.12 – Αποτέλεσμα συστάσεων με μονοπάτι προόδου

4.8.5 Ρυθμίσεις και πίνακας διαχείρισης

Η οθόνη ρυθμίσεων (Σχήμα 4.13) επιτρέπει επεξεργασία του προφίλ παίκτη και αλλαγή κωδικού. Ο πίνακας διαχείρισης (Σχήμα 4.14) — προσβάσιμος μόνο σε ρόλο admin — παρουσιάζει στατιστικά πλατφόρμας και τις φόρμες διαχείρισης καταλόγου και βάσης γνώσης, συμπεριλαμβανομένης της ροής εισαγωγής από URL: το σύστημα κατεβάζει και καθαρίζει το κύριο κείμενο της σελίδας, ο διαχειριστής το επιθεωρεί και το εγκρίνει, και μόνο τότε το έγγραφο ευρετηριάζεται — ανθρώπινος έλεγχος σε κάθε προσθήκη γνώσης.

Η ροή εισαγωγής γνώσης από URL (`url_import_service`) αξίζει ιδιαίτερη μνεία ως παράδειγμα ισορροπίας μεταξύ αυτοματισμού και ανθρώπινου ελέγχου: το σύστημα κατεβάζει τη σελίδα, αφαιρεί στοιχεία πλοήγησης, διαφημίσεις και άλλο θόρυβο μέσω ευριστικών κανόνων εξαγωγής κύριου περιεχομένου, και παρουσιάζει στον διαχειριστή προεπισκόπηση του καθαρισμένου κειμένου μαζί με προτεινόμενο τίτλο, παιχνίδι-στόχο (μέσω αντιστοίχισης λέξεων-κλειδιών του τίτλου σελίδας με τον κατάλογο) και ετικέτες. Ο διαχειριστής μπορεί να επεξεργαστεί οποιοδήποτε από αυτά τα πεδία πριν την έγκριση --- ουδέποτε το περιεχόμενο ευρετηριάζεται αυτόματα χωρίς αυτό το ενδιάμεσο βήμα. Η σχεδιαστική αυτή επιλογή αντανακλά ρητή θέση της εργασίας απέναντι στον κίνδυνο μόλυνσης της βάσης γνώσης (`knowledge base poisoning`): σε ένα σύστημα RAG, η ποιότητα της τελικής απάντησης εξαρτάται απόλυτα από την αξιοπιστία των εγγράφων πηγής, οπότε αυτοματοποιημένη εισαγωγή χωρίς ανθρώπινη επιβεβαίωση θα εισήγαγε ανεξέλεγκτο κίνδυνο εσφαλμένης ή παραπλανητικής πληροφορίας να παρουσιαστεί αργότερα ως τεκμηριωμένη απάντηση προς τον τελικό χρήστη.

GameCoach AI
Gaming Coach Console

Dashboard
Game Library
AI Coach Chat
Recommendations
Settings
Admin Panel

Admin
admin@example.com
ADMIN ADVANCED

100%
MISSION

Account details
Identity and protected route information.

USERNAME: Admin
EMAIL: admin@example....
ROLE: admin

Editable player profile
These fields feed the recommendation and dashboard experience.

USERNAME: Admin
EMAIL: admin@example.com

SKILL LEVEL: Advanced
DIFFICULTY COMFORT: Balanced

Your overall gaming experience — tunes dashboard and recommendations.
How much challenge you enjoy.

PREFERRED GENRES (3 SELECTED)
Pick from the catalog's genres — used to personalize recommendations.

Action Adventure, Action Roguelike, Action Roguelite, Action RPG, Automation Strategy, Battle Royale FPS, City Builder Simulation, Competitive FPS, CRPG, Deckbuilding Roguelike, Farming RPG, Hackathon Game, CDE, Mass Effect, Metroidvania, Metroidvania Modifiers

PREFERRED MECHANICS (3 SELECTED)
Gameplay mechanics you enjoy (e.g. deck building, parry timing).

aerials, aim ducks, air dash, alchemy, arena movement, army control, assist options, automation, base building, biome routing, bonfire routing, boons, boost management, boss pattern learning, boss patterns, boss themes, boss weaknesses, boss objectives, hackathon themes

FAVORITE PLATFORMS (2 SELECTED)
Where you play.

Linux, Mac, Mobile, PC, PlayStation, Switch, Xbox

AVATAR COLOR
Lime

BIO / NOTES
Demo administrator profile for the GameCoach AI thesis presentation.

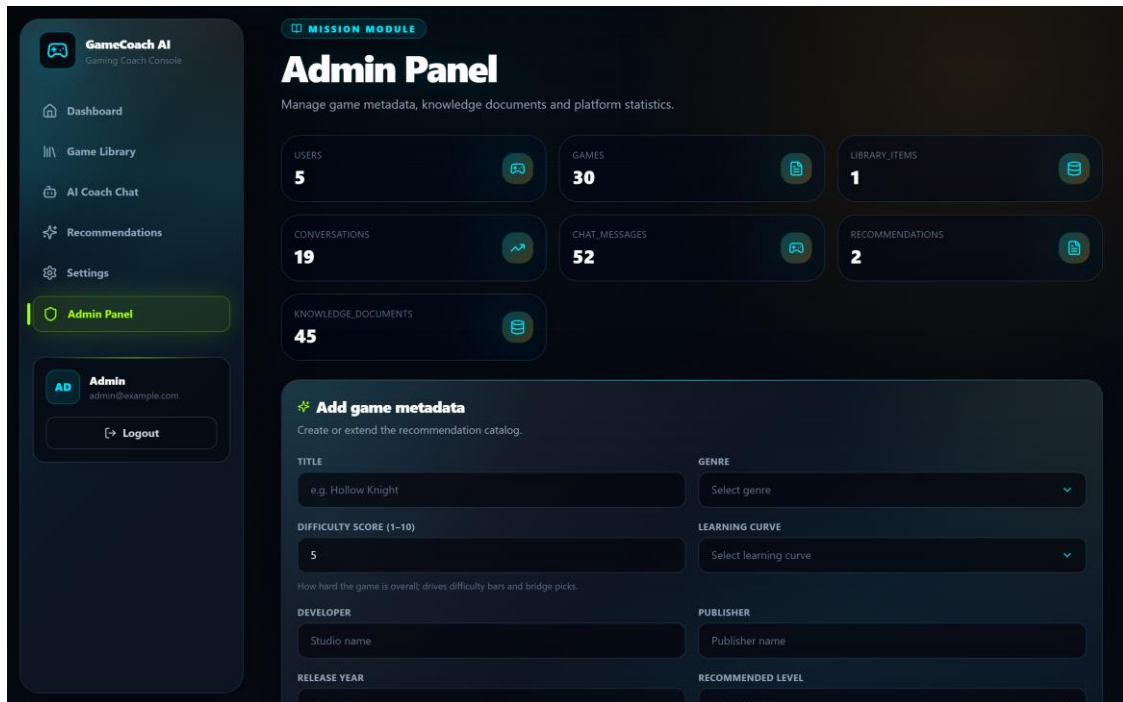
Free text — anything about you as a player.

Save profile

Change password
Update local account credentials.

CURRENT PASSWORD: *****
NEW PASSWORD: *****
CONFIRM NEW PASSWORD: *****
Change password

Σχήμα 4.13 – Οθόνη ρυθμίσεων προφίλ παίκτη



Σχήμα 4.14 – Πίνακας διαχείρισης με στατιστικά πλατφόρμας

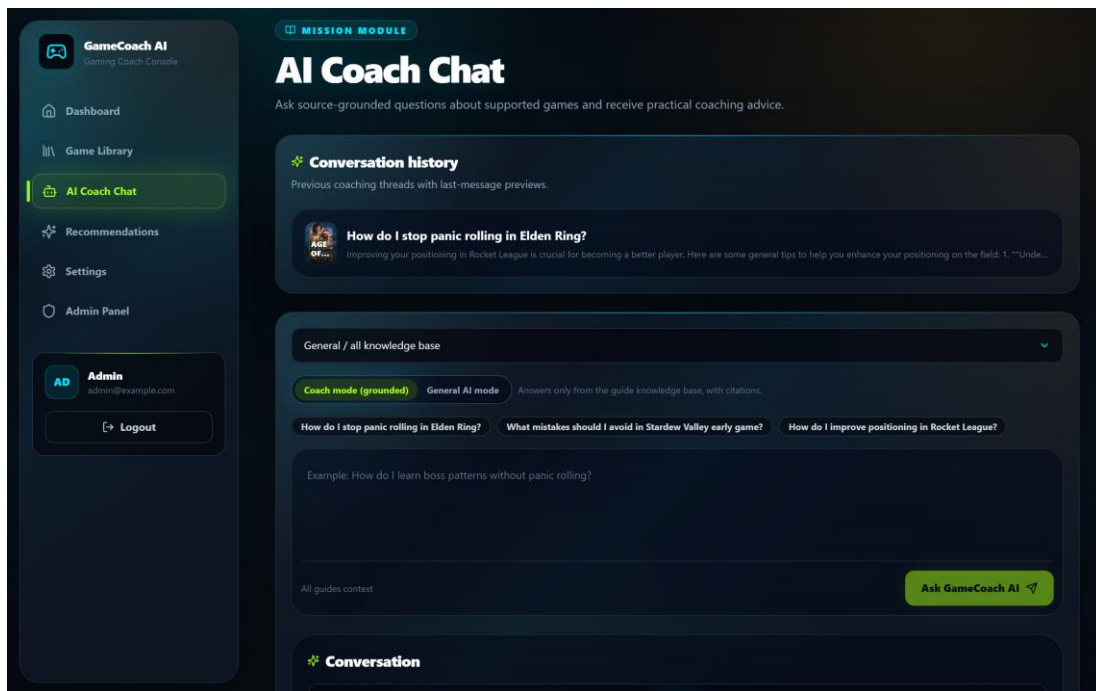
The screenshot shows the 'Add knowledge document' form. It includes a 'Rebuild RAG' button in the top right. The form has several sections: 'GAME' with a dropdown menu set to 'General document', 'DOCUMENT TITLE' with a text input field containing 'e.g. Hades Boons & Weapons', 'TAGS' with a text input field containing 'e.g. boons, weapons, builds', and 'GUIDE CONTENT' with a large text area. Below the text area is a prominent green 'Create document' button. At the bottom, there is a section titled 'Import guide from URL' with a 'GUIDE PAGE URL' input field containing 'https://example.com/guide-page' and a 'Fetch draft' button.

Σχήμα 4.15 – Φόρμα προσθήκης εγγράφου γνώσης με άμεση ανακατασκευή του ευρετηρίου RAG

4.8.6 Διαφάνεια, κενές καταστάσεις και ανθεκτικότητα της διεπαφής

Ιδιαίτερη σχεδιαστική μέριμνα δόθηκε στη διαφάνεια και στις οριακές καταστάσεις. Κάθε απάντηση του βοηθού φέρει έγχρωμη ετικέτα προέλευσης που αντιστοιχεί στο mode του Πίνακα 4.2 — πράσινη

για τεκμηριωμένη, πορτοκαλί για γενική/μη τεκμηριωμένη, ουδέτερη για τοπική σύνθεση — ώστε ο χρήστης να μη χρειάζεται να μαντεύει την αξιοπιστία του κειμένου. Οι κενές καταστάσεις (νέος λογαριασμός χωρίς βιβλιοθήκη, συνομιλία χωρίς μηνύματα — Σχήμα 4.16) καθοδηγούν με προτεινόμενες ενέργειες αντί για κενές οθόνες, ενώ κατά την αναμονή απάντησης εμφανίζεται ένδειξη πληκτρολόγησης. Τέλος, οι εικόνες εξωφύλλων των παιχνιδιών φορτώνονται από εξωτερικά CDN με αυτόματα εναλλακτική CSS (αρχικά τίτλου σε χρωματισμένο πλαίσιο) όταν η εικόνα δεν είναι διαθέσιμη — η διεπαφή δεν εξαρτάται από τη διαθεσιμότητα τρίτων για να παραμείνει λειτουργική και αισθητικά συνεπής.



Σχήμα 4.16 – Κενή κατάσταση συνομιλίας με επιλογή λειτουργίας και προτεινόμενες ερωτήσεις

4.9 Εγκατάσταση και εκτέλεση με Docker

Για αναπαραγώγιμη εγκατάσταση παρέχεται `docker-compose.yml` με τρεις υπηρεσίες: `postgres:16-alpine` με υγειονομικό έλεγχο και μόνιμο τόμο δεδομένων, `backend (python:3.12-slim, uvicorn` στη θύρα 8000) που εξαρτάται από την υγιή βάση, και `frontend (node:20-alpine, θύρα 5173)`. Η ίδια εφαρμογή εκτελείται και απευθείας σε τοπικό περιβάλλον με SQLite, χωρίς καμία αλλαγή κώδικα — μόνο μεταβλητές περιβάλλοντος. Οι αναλυτικές οδηγίες εγκατάστασης παρατίθενται στο Παράρτημα Α.

4.10 Ανάλυση απειλών και μέτρα ασφάλειας εφαρμογής

Πέρα από την αυθεντικοποίηση και τον περιορισμό ρυθμού της ενότητας 4.2, το σύστημα αντιμετωπίζει ρητά τις πλέον συνήθεις κατηγορίες ευπάθειας εφαρμογών ιστού, ακολουθώντας άτυπα τις κατηγορίες του OWASP Top 10. Η έγχυση SQL (SQL injection) αποκλείεται δομικά, καθώς όλα τα ερωτήματα προς τη βάση δεδομένων διέρχονται από το Object-Relational Mapper της SQLAlchemy, το οποίο παραμετροποιεί αυτόματα κάθε τιμή εισόδου· ουδέποτε συντίθεται ερώτημα SQL μέσω συνένωσης αλφαριθμητικών με δεδομένα χρήστη. Η επίθεση διασταυρούμενης σεναριογραφίας (Cross-Site Scripting, XSS) περιορίζεται δομικά από το ίδιο το React, το οποίο διαφεύγει (escapes) αυτόματα κάθε δυναμικό περιεχόμενο πριν την εισαγωγή του στο DOM, εκτός αν

χρησιμοποιηθεί ρητά η επικίνδυνη ιδιότητα `dangerouslySetInnerHTML` --- ιδιότητα που δεν χρησιμοποιείται πουθενά στην εφαρμογή.

Η προστασία από πλαστογράφηση αιτήματος μεταξύ τοποθεσιών (Cross-Site Request Forgery, CSRF) δεν απαιτεί ειδικό διακριτικό CSRF, καθώς η αυθεντικοποίηση βασίζεται σε διακριτικό JWT που μεταφέρεται ρητά στην κεφαλίδα `Authorization` και όχι σε cookie που ο περιηγητής θα επισύναπτε αυτόματα σε κάθε αίτημα --- ο μηχανισμός επίθεσης CSRF προϋποθέτει ακριβώς αυτή την αυτόματη επισύναψη, οπότε δεν εφαρμόζεται στην παρούσα αρχιτεκτονική. Τα μυστικά της εφαρμογής (κλειδί JWT, κλειδί `OpenAI`, διαπιστευτήρια βάσης δεδομένων) διαβάζονται αποκλειστικά από μεταβλητές περιβάλλοντος και ουδέποτε γράφονται στον πηγαίο κώδικα (`hardcoded`) ή καταγράφονται σε αρχεία καταγραφής (`logs`) --- πρακτική επιβεβαιωμένη μέσω επιθεώρησης κώδικα (`code review`) πριν από κάθε παράδοση. Ο έλεγχος εξουσιοδότησης εφαρμόζεται σε επίπεδο πόρου και όχι μόνο σε επίπεδο τελικού σημείου: για παράδειγμα, στην ανάκτηση ή διαγραφή εγγραφής βιβλιοθήκης, το backend επιβεβαιώνει ρητά ότι η εγγραφή ανήκει στον αυθεντικοποιημένο χρήστη πριν επιτρέψει την ενέργεια, αποτρέποντας επίθεση τύπου ανασφαλούς άμεσης αναφοράς αντικειμένου (Insecure Direct Object Reference, IDOR) όπου ένας χρήστης θα μπορούσε να προσπελάσει ή να τροποποιήσει δεδομένα άλλου χρήστη μαντεύοντας αριθμητικά αναγνωριστικά.

4.11 Σχεδιαστικές εναλλακτικές που εξετάστηκαν

Κατά τον σχεδιασμό του συστήματος εξετάστηκαν και απορρίφθηκαν συνειδητά αρκετές εναλλακτικές τεχνολογικές επιλογές, η καταγραφή των οποίων αποσαφηνίζει τη λογική πίσω από τις τελικές αποφάσεις. Για τη διανυσματική βάση δεδομένων εξετάστηκαν διαχειριζόμενες υπηρεσίες cloud (Pinecone, Weaviate Cloud) έναντι της τοπικά ενσωματωμένης ChromaDB· οι πρώτες προσφέρουν οριζόντια κλιμακωσιμότητα και διαχειριζόμενη υποδομή, όμως εισάγουν εξωτερική εξάρτηση δικτύου, κόστος συνδρομής και πολυπλοκότητα ανάπτυξης ασύμβατη με τον στόχο πλήρους λειτουργίας εκτός σύνδεσης (Κεφάλαιο 3, ΜΛΑ). Η ChromaDB, ως ενσωματωμένη βιβλιοθήκη με μόνιμη αποθήκευση σε τοπικό δίσκο, καλύπτει πλήρως την κλίμακα των 357 τεμαχίων χωρίς καμία εξωτερική εξάρτηση, με το ρητό μειονέκτημα ότι δεν θα κλιμακωνόταν απευθείας σε πολλαπλές διεργασίες διακομιστή --- περιορισμός αποδεκτός για την παρούσα κλίμακα ανάπτυξης και καταγεγραμμένος ως κατεύθυνση μελλοντικής επέκτασης στην ενότητα 6.4.

Για τη σύνθεση απαντήσεων εξετάστηκε η χρήση καθιερωμένου πλαισίου ενορχήστρωσης RAG (LangChain ή LlamaIndex) έναντι της χειρωνακτικής υλοποίησης του αγωγού ανάκτησης-σύνθεσης. Τα πλαίσια αυτά προσφέρουν έτοιμα συστατικά (`chunkers`, `retrievers`, `chains`) και ταχύτερη αρχική ανάπτυξη, όμως εισάγουν επίπεδο αφαίρεσης που δυσχεραίνει τον ακριβή έλεγχο της συμπεριφοράς άρνησης --- ακριβώς το στοιχείο που αποτελεί κεντρική ερευνητική συνεισφορά της εργασίας (ενότητα 1.3). Η χειρωνακτική υλοποίηση επέτρεψε πλήρη διαφάνεια και έλεγχο σε κάθε στάδιο του αγωγού --- κατώφλια, φίλτρα, λογική επανάκλησης --- με το κόστος μεγαλύτερου όγκου κώδικα προς συντήρηση σε σχέση με τη χρήση έτοιμου πλαισίου. Ομοίως, για τον περιορισμό ρυθμού εξετάστηκε το Redis ως κοινόχρηστο κατάστημα κατάστασης, απορρίφθηκε όμως ως δυσανάλογο της κλίμακας ανάπτυξης ενός μοναδικού `worker`, με τη ρητή σημείωση ότι η μετάβαση σε πολλαπλούς `workers` στο μέλλον θα καθιστούσε αναγκαία αυτή την αλλαγή.

4.12 Ποιότητα κώδικα και αυτοματοποιημένες δοκιμές μονάδας

Πέρα από τα σενάρια αξιολόγησης εκτός διακομιστή του Κεφαλαίου 5, η ορθότητα του backend επιβεβαιώνεται και μέσω σουίτας αυτοματοποιημένων δοκιμών μονάδας/ολοκλήρωσης με `pytest`, η

οποία καλύπτει τον έλεγχο υγείας του API (`test_health.py`), τη λειτουργία υγιεινής/υγιεινή δρομολόγηση του αγωγού RAG (`test_rag.py`), τη λογική βαθμολόγησης της μηχανής συστάσεων (`test_recommendations.py`) και --- η εκτενέστερη ομάδα --- δέκα δοκιμές γύρω από την ασφάλεια (`test_security.py`), οι οποίες επαληθεύουν τη σωστή κρυπτογράφηση/επαλήθευση κωδικών, την ορθή έκδοση και λήξη διακριτικών JWT και την απόρριψη μη εξουσιοδοτημένων αιτημάτων στα προστατευμένα τελικά σημεία. Οι δοκιμές εκτελούνται πάνω σε προσωρινή βάση δεδομένων SQLite εντός μνήμης, ανεξάρτητη από τη βάση ανάπτυξης, μέσω παράκαμψης εξάρτησης (`dependency override`) της συνεδρίας βάσης --- ακριβώς η ίδια τεχνική εξάρτησης (`Depends`) που περιγράφηκε στην ενότητα 4.1 καθιστά αυτή την απομόνωση απλή στην υλοποίηση.

Το έργο δεν διαθέτει επί του παρόντος αγωγό συνεχούς ολοκλήρωσης (`Continuous Integration`) που να εκτελεί αυτόματα τη σουίτα δοκιμών σε κάθε υποβολή αλλαγών --- περιορισμός που αναγνωρίζεται ρητά εδώ και καταγράφεται ως άμεση κατεύθυνση βελτίωσης στην ενότητα 6.3, καθώς η προσθήκη ενός απλού αγωγού `GitHub Actions` που εκτελεί `pytest`, έλεγχο τύπων με `mypy` και στατική ανάλυση με `ruff` σε κάθε `pull request` θα αύξανε σημαντικά την εμπιστοσύνη σε μελλοντικές αλλαγές κώδικα με ελάχιστο κόστος υλοποίησης. Παρά την απουσία αυτοματοποιημένου αγωγού, η πειθαρχία τυποποίησης κώδικα διατηρήθηκε χειρωνακτικά καθ' όλη τη διάρκεια της ανάπτυξης: συνεπής μορφοποίηση, τεκμηρίωση συναρτήσεων μέσω `docstrings` που εξηγούν όχι μόνο το «τι» αλλά και το «γιατί» μιας σχεδιαστικής απόφασης --- πρακτική εμφανής σε ολόκληρη τη βάση κώδικα και ιδιαίτερα χρήσιμη κατά τη συγγραφή του παρόντος κεφαλαίου --- και πλήρης τήρηση υποδείξεων τύπων (`type hints`) σε συνάρτηση με τα σχήματα `Pydantic`, ώστε λάθη τύπου να εντοπίζονται όσο το δυνατόν νωρίτερα στον κύκλο ανάπτυξης.

4.13 Ανάλυση περίπτωσης: πλήρες ίχνος ενός αιτήματος συνομιλίας

Για να καταστεί απτή η αλληλεπίδραση όλων των υποσυστημάτων που περιγράφηκαν στις προηγούμενες ενότητες, παρουσιάζεται αναλυτικά το πλήρες ίχνος (`end-to-end trace`) ενός τυπικού αιτήματος συνομιλίας, από τη στιγμή που ο χρήστης πατά «Αποστολή» μέχρι την εμφάνιση της τεκμηριωμένης απάντησης στην οθόνη. Ως παράδειγμα χρησιμοποιείται η ερώτηση «How do I stop panic rolling against bosses in Elden Ring?» με επιλεγμένο το παιχνίδι `Elden Ring`, ακριβώς το παράδειγμα επιτυχίας που αναφέρθηκε ποιοτικά στην ενότητα 5.3.

Βήμα 1 (frontend): Το component `ChatWindow` καταγράφει το κείμενο εισόδου, το προσθέτει αισιόδοξα στη ροή μηνυμάτων της τοπικής κατάστασης (`useState`) και καλεί τη συνάρτηση `sendChatMessage` του στρώματος `api client`, η οποία εκτελεί αίτημα `POST` προς το `/api/chat` με σώμα `JSON` που περιλαμβάνει το κείμενο ερώτησης, το αναγνωριστικό του επιλεγμένου παιχνιδιού και το διακριτικό JWT στην κεφαλίδα `Authorization`.

Βήμα 2 (δρομολογητής και αυθεντικοποίηση): Το τελικό σημείο `chat.py` λαμβάνει το αίτημα· η εξάρτηση `get_current_user` επαληθεύει το JWT και ανακτά τον χρήστη, η εξάρτηση περιορισμού ρυθμού ελέγχει το κυλιόμενο παράθυρο του κουβά «chat» για τον συγκεκριμένο χρήστη (ενότητα 4.2), και το σχήμα `Pydantic ChatRequest` επικυρώνει τη δομή του σώματος αιτήματος πριν η εκτέλεση προχωρήσει στη λογική της υπηρεσίας.

Βήμα 3 (ανάκτηση): Καλείται η `rag_service.retrieve()` με το κείμενο ερώτησης και το αναγνωριστικό παιχνιδιού. Η ερώτηση δεν ταιριάζει με κανένα ψευδώνυμο της λίστας `NON_CATALOG_GAME_ALIASES`, οπότε η εκτέλεση προχωρά κανονικά: το κείμενο ενσωματώνεται με την ενεργή διαμόρφωση ενσωμάτωσης, το `ChromaDB` επιστρέφει τα `k=5` πλησιέστερα τεμάχια, και εφαρμόζονται διαδοχικά τα φίλτρα κατωφλίου συνάφειας, φίλτρου

παιχνιδιού (μόνο τεμάχια του Elden Ring ή γενικά τεμάχια) και λεξιλογικής επικάλυψης. Στο συγκεκριμένο παράδειγμα, τέσσερα από τα πέντε τεμάχια περνούν όλα τα φίλτρα --- προερχόμενα από τις ενότητες «Basic mechanics» και «Beginner tips» του οδηγού Elden Ring, οι οποίες αναφέρονται ρητά στη μηχανική της κύλισης (roll) και του χρονισμού άμυνας.

Βήμα 4 (σύνθεση): Τα τέσσερα εγκεκριμένα τεμάχια μορφοποιούνται ως πλαίσιο και περνούν στην `llm_service.generate_response()` μαζί με το `system prompt` της ενότητας 4.5. Το `gpt-4o-mini` παράγει απάντηση με θερμοκρασία 0,25, συνθέτοντας τη στρατηγική αντιμετώπισης του «πανικόβλητου κυλίσματος» αποκλειστικά από το παρεχόμενο πλαίσιο, χωρίς να χρειαστεί επανάκληση, αφού η πρώτη απόπειρα παράγει έγκυρη, τεκμηριωμένη απάντηση.

Βήμα 5 (αποθήκευση και απόκριση): Η υπηρεσία συνομιλίας αποθηκεύει το ζεύγος ερώτησης-απάντησης στον πίνακα ιστορικού μαζί με τα ονόματα των τεσσάρων πηγών και την ετικέτα `mode = rag_openai`, και το τελικό σημείο επιστρέφει ενιαία απόκριση JSON με το κείμενο, τη λίστα πηγών και το `mode`. Το frontend λαμβάνει την απόκριση, σταματά την κινούμενη ένδειξη πληκτρολόγησης, εμφανίζει το κείμενο της απάντησης και, από κάτω, τις κάρτες πηγών με το πράσινο χρωματικό σήμα της ενότητας 4.8.6.

Το συνολικό ίχνος --- από το πάτημα του κουμπιού έως την εμφάνιση της απάντησης --- ολοκληρώνεται εντός λίγων εκατοντάδων χιλιοστών του δευτερολέπτου σε λειτουργία εκτός σύνδεσης (Πίνακας 5.8) ή εντός ενός έως δύο δευτερολέπτων σε πραγματική λειτουργία με το εξωτερικό API, με την τελευταία κλήση προς το `gpt-4o-mini` να κυριαρχεί σαφώς στον συνολικό χρόνο. Το παράδειγμα αυτό καταδεικνύει πρακτικά την αρχή «η γνώση στο ευρετήριο, η γλώσσα στο μοντέλο, η απόφαση στα μετρήσιμα φίλτρα» που διατυπώνεται ως κεντρικό συμπέρασμα της εργασίας στην ενότητα 6.5: κάθε βήμα του ίχνους αντιστοιχεί σε ένα σαφώς οριοθετημένο, ανεξάρτητα ελέγξιμο συστατικό.

Στο κεφάλαιο αυτό παρουσιάστηκε η πλήρης υλοποίηση του συστήματος: από τον πυρήνα ασφάλειας και δεδομένων, τη βάση γνώσης και τον αυστηρό αγωγό RAG, έως τη μηχανή συστάσεων, το API και τη διεπαφή χρήστη. Το επόμενο κεφάλαιο απαντά στο κρίσιμο ερώτημα: πόσο καλά λειτουργεί στην πράξη το σύστημα αυτό;

5. Αξιολόγηση και αποτελέσματα

Το κεφάλαιο αυτό παρουσιάζει τη μεθοδολογία και τα αποτελέσματα της πειραματικής αξιολόγησης του συστήματος. Αξιολογούνται διαδοχικά: η ποιότητα του σταδίου ανάκτησης, η συμπεριφορά του πλήρους αγωγού έναντι σκέτου LLM, η ευαισθησία στις κρίσιμες παραμέτρους, η κάλυψη του συνόλου του καταλόγου μέσω του πλήρους API συνομιλίας, η μηχανή συστάσεων και, τέλος, η ορθότητα του λογισμικού μέσω αυτοματοποιημένων δοκιμών και ελέγχων ετοιμότητας. Όλα τα αποτελέσματα παράγονται από εκτελέσιμα σενάρια της σουίτας αξιολόγησης του αποθετηρίου και είναι πλήρως αναπαραγώγιμα.

5.1 Μεθοδολογία και σύνολα δεδομένων

Η αξιολόγηση της ανάκτησης βασίζεται σε δύο επισημειωμένα σύνολα ερωτήσεων. Το εσωτερικό σύνολο περιλαμβάνει 40 ερωτήσεις: 33 εντός πεδίου — τρεις ανά αντιπροσωπευτικό παιχνίδι, κλιμακούμενες σε τρία επίπεδα δυσκολίας διατύπωσης (easy: ρητή αναφορά τίτλου και όρων του οδηγού, medium: μερική λεξιλογική απόκλιση, hard: έμμεση διατύπωση) — και 7 εκτός πεδίου, για παιχνίδια που δεν καλύπτονται από τη βάση γνώσης. Κάθε ερώτηση εντός πεδίου φέρει ως επισημείωση τον αναμενόμενο οδηγό-πηγή. Το εξωτερικό σύνολο περιλαμβάνει 110 πραγματικές ερωτήσεις παικτών, συλλεγμένες αυτούσιες από δημόσια φόρουμ (Reddit, Steam, GameFAQs): 97 εντός πεδίου και 13 δύσκολα αρνητικά (hard negatives) — ερωτήσεις για παιχνίδια παρόμοιου είδους που όμως δεν καλύπτονται. Η σύγκριση εσωτερικού/εξωτερικού συνόλου μετρά τη λεγόμενη μεροληψία ευθυγράμμισης λεξιλογίου: οι εσωτερικές ερωτήσεις γράφτηκαν από τον συγγραφέα των οδηγών, άρα το λεξιλόγιό τους ταιριάζει φυσικά με το σώμα κειμένων, ενώ οι εξωτερικές αποτυπώνουν το πραγματικό λεξιλόγιο παικτών.

Η διαμόρφωση του συστήματος κατά τα πειράματα ανάκτησης συνοψίζεται στον Πίνακα 5.1. Τα πειράματα εκτελούνται στην ντετερμινιστική λεξιλογική διαμόρφωση (hashing), ώστε να είναι αναπαραγώγιμα χωρίς εξωτερικό API· η σύγκριση με τις σημασιολογικές ενσωματώσεις παρουσιάζεται χωριστά στην ενότητα 5.5.

Πίνακας 5.1 – Διαμόρφωση συστήματος κατά την αξιολόγηση ανάκτησης

Παράμετρος	Τιμή
Έγγραφα ευρετηρίου	45
Τεμάχια ευρετηρίου	357
Διανυσματική βάση	ChromaDB
Top-k ανάκτησης (αξιολόγηση)	5
Αυστηρή λειτουργία (strict mode)	Ενεργή
Ελάχιστο σκορ συνάφειας	0,08

5.2 Μετρικές αξιολόγησης

Για την ποιότητα κατάταξης χρησιμοποιούνται οι τυπικές μετρικές ανάκτησης πληροφορίας. Το Recall@k ορίζεται εδώ ως το ποσοστό των ερωτήσεων εντός πεδίου για τις οποίες ο αναμενόμενος οδηγός εμφανίζεται εντός των k κορυφαίων τεμαχίων. Το μέσο αντίστροφο σημείο κατάταξης (Mean Reciprocal Rank – MRR) [19] ορίζεται ως:

$$MRR = (1/N) \cdot \sum_i (1 / rank_i) \quad (5.1)$$

όπου $rank_i$ η θέση του πρώτου τεμαχίου του αναμενόμενου οδηγού για την ερώτηση i (με $1/rank_i = 0$ όταν ο οδηγός δεν ανακτάται καθόλου). Για τη συμπεριφορά άρνησης ορίζονται δύο συμπληρωματικές μετρικές: το ποσοστό ορθής απόρριψης (correct-rejection rate) — ποσοστό ερωτήσεων εκτός πεδίου που ορθώς επιστρέφουν κενό πλαίσιο — και το ποσοστό εσφαλμένης απόρριψης (false-rejection rate) — ποσοστό ερωτήσεων εντός πεδίου που εσφαλμένα απορρίπτονται. Το δεύτερο αποτυπώνει τον βασικό κίνδυνο ενός υπερβολικά αυστηρού μηχανισμού προστασίας.

5.3 Αποτελέσματα ποιότητας ανάκτησης

Ο Πίνακας 5.2 συνοψίζει τα αποτελέσματα στο εσωτερικό σύνολο. Το σύστημα κατατάσσει τον σωστό οδηγό πρώτο στο 63,6% των ερωτήσεων και εντός της πρώτης τριάδας στο 72,7%, με MRR 0,684. Κρίσιμο εύρημα αποτελεί το ζεύγος των μετρικών άρνησης: και οι 7 ερωτήσεις εκτός πεδίου απορρίφθηκαν ορθά (100%), ενώ καμία ερώτηση εντός πεδίου δεν απορρίφθηκε εσφαλμένα (0%) — δηλαδή ο μηχανισμός προστασίας δεν θυσιάζει διαθεσιμότητα για την ασφάλεια.

Πίνακας 5.2 – Συγκεντρωτικές μετρικές ανάκτησης στο εσωτερικό σύνολο αξιολόγησης

Μετρική	Τιμή	Λεπτομέρεια
Recall@1	63,6%	21/33 εντός πεδίου
Recall@3	72,7%	24/33 εντός πεδίου
MRR	0,684	μέσος όρος 33 ερωτήσεων
Ορθή απόρριψη	100,0%	7/7 εκτός πεδίου
Εσφαλμένη απόρριψη	0,0%	0/33 εντός πεδίου

Η ανάλυση ανά δυσκολία διατύπωσης (Πίνακας 5.3) δείχνει την αναμενόμενη κλιμάκωση: οι εύκολες διατυπώσεις επιτυγχάνουν Recall@1 80%, ενώ οι μεσαίες — που παραφράζουν το λεξιλόγιο του οδηγού — πέφτουν στο 41,7%. Ενδιαφέρον είναι ότι οι δύσκολες διατυπώσεις (72,7%) υπερτερούν των μεσαίων: οι μεγαλύτερες, πλούσιες σε συμπραζόμενα ερωτήσεις παρέχουν στη λεξιλογική ανάκτηση περισσότερους διακριτικούς όρους από τις σύντομες παραφράσεις.

Πίνακας 5.3 – Ανάκτηση εντός πεδίου ανά δυσκολία διατύπωσης

Δυσκολία	Ερωτήσεις	Recall@1	Recall@3	MRR
easy	10	80,0%	90,0%	0,858
medium	12	41,7%	58,3%	0,500
hard	11	72,7%	72,7%	0,727

Η ποιοτική επισκόπηση των επιμέρους περιπτώσεων φωτίζει τους μηχανισμούς πίσω από τους αριθμούς. Στις επιτυχίες, η ερώτηση «How do I stop panic rolling against bosses in Elden Ring?» ανακτά πρώτο τον οδηγό του Elden Ring με τέσσερα συναφή αποσπάσματα, και η «What should I focus on during my first week in Stardew Valley?» τον αντίστοιχο οδηγό του Stardew Valley. Οι αποτυχίες πρώτης θέσης εμπίπτουν σε δύο μοτίβα: είτε ανακτάται συγγενής οδηγός εμβάθυνσης του ίδιου παιχνιδιού αντί του βασικού (π.χ. «Slay the Spire Deck Archetypes» αντί «Slay the Spire Guide» — αστοχία μόνο κατ' όνομα, αφού το περιεχόμενο είναι κατάλληλο), είτε παρασύρεται η λεξιλογική ομοιότητα από κοινούς όρους ειδών (π.χ. ερώτηση για back-post coverage στο Rocket League που

ανακτά οδηγό άλλου τίτλου). Το δεύτερο μοτίβο είναι αυτό που θεραπεύουν οι σημασιολογικές ενσωματώσεις της ενότητας 5.5.3.

5.4 Γενίκευση σε πραγματικές ερωτήσεις παικτών

Ο Πίνακας 5.4 αντιπαραθέτει το εσωτερικό σύνολο με το εξωτερικό σύνολο των 110 πραγματικών ερωτήσεων. Η πτώση των μετρικών κατάταξης (Recall@1 από 63,6% σε 40,2%) ποσοτικοποιεί τη μεροληψία ευθυγράμμισης λεξιλογίου· είναι το ειλικρινές μέτρο του πόσο δυσκολότερο είναι το πρόβλημα με πραγματικό λεξιλόγιο χρηστών. Αξιοσημείωτα, η ορθή απόρριψη παραμένει 100% ακόμη και στα 13 δύσκολα αρνητικά (ερωτήσεις για souls-like και ανταγωνιστικά παιχνίδια που δεν καλύπτονται), ενώ η εσφαλμένη απόρριψη περιορίζεται στο 9,3%. Επιπλέον, καταγράφηκαν έξι «ερωτήσεις κενού περιεχομένου» (content-gap): πραγματικές ανάγκες παικτών (π.χ. πρόοδος σε συγκεκριμένο κεφάλαιο) που ο κοντινότερος οδηγός καλύπτει μόνο μερικώς — εύρημα που ανατροφοδοτεί τη συγγραφή περιεχομένου.

Πίνακας 5.4 – Εσωτερικό έναντι εξωτερικού συνόλου: μεροληψία ευθυγράμμισης λεξιλογίου

Μετρική	Εσωτερικό σύνολο	Εξωτερικό σύνολο
Ερωτήσεις εντός πεδίου	33	97
Recall@1	63,6%	40,2%
Recall@3	72,7%	48,5%
MRR	0,684	0,449
Ορθή απόρριψη	100,0% (7/7)	100,0% (13/13)
Εσφαλμένη απόρριψη	0,0%	9,3%

5.5 Σύγκριση με baseline και ανάλυση ευαισθησίας

5.5.1 Πλήρης αγωγός RAG έναντι σκέτου LLM

Για να τεκμηριωθεί η αξία της ανάκτησης, ο πλήρης αγωγός συγκρίθηκε με σκέτο γλωσσικό μοντέλο χωρίς ανάκτηση στις ίδιες 40 ερωτήσεις (Πίνακας 5.5). Η αντίθεση είναι απόλυτη: το σκέτο μοντέλο επιχειρεί απάντηση σε όλες τις ερωτήσεις — και στις 7 εκτός πεδίου, με προφανή κίνδυνο παραισθήσεων — χωρίς καμία παράθεση πηγής, ενώ ο αγωγός RAG τεκμηριώνει και τις 33 εντός πεδίου με κατονομαζόμενο οδηγό και αρνείται ρητά και τις 7 εκτός πεδίου.

Πίνακας 5.5 – Συμπεριφορική σύγκριση σκέτου LLM και αγωγού RAG (40 ερωτήσεις)

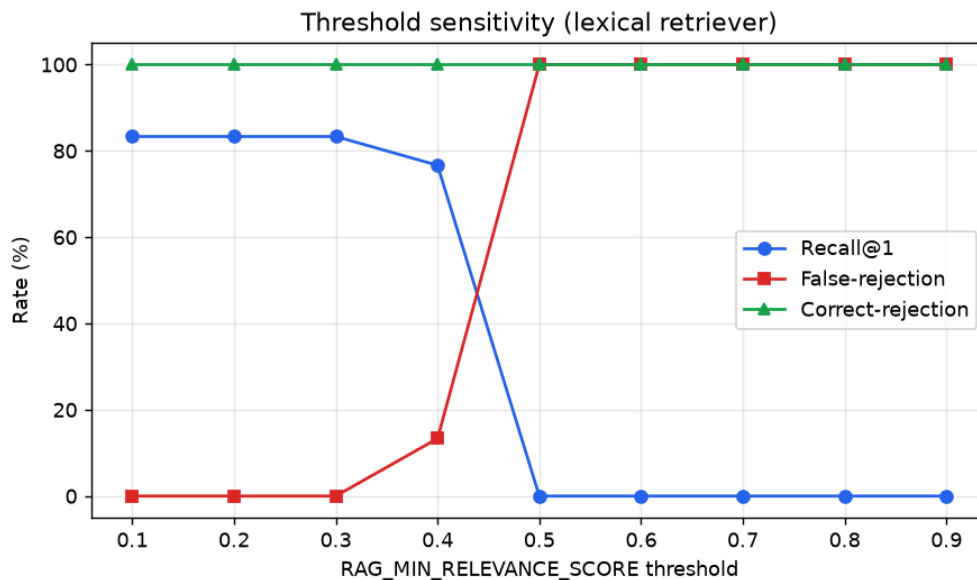
Συμπεριφορά	Σκέτο LLM	Αγωγός RAG
Απαντήσεις εντός πεδίου με παράθεση πηγής	0/33	33/33
Ορθές αρνήσεις εκτός πεδίου	0/7	7/7
Απαντήσεις εκτός πεδίου χωρίς τεκμηρίωση (κίνδυνος παραίσθησης)	7/7	0/7
Συνολικές απαντήσεις με παράθεση πηγής	0/40	33/40

5.5.2 Ευαισθησία στο κατώφλι συνάφειας και στο μέγεθος τεμαχίου

Η σάρωση του κατωφλίου συνάφειας (Πίνακας 5.6, Σχήμα 5.1) αποκαλύπτει καθαρή εικόνα: έως την τιμή 0,2 οι μετρικές παραμένουν σταθερές (Recall@1 72,7% στο σύνολο σάρωσης 30+10 ερωτήσεων), από 0,3 και άνω αρχίζουν οι εσφαλμένες απορρίψεις και στο 0,5 ο μηχανισμός καταρρέει, απορρίπτοντας σχεδόν τα πάντα. Κρίσιμη είναι η διαπίστωση ότι η ορθή απόρριψη παραμένει 100% σε όλο το εύρος: την άρνηση των ερωτήσεων εκτός πεδίου την εξασφαλίζει κυρίως ο έλεγχος λεξιλογικής επικάλυψης και όχι το κατώφλι σκορ, επομένως η αύξηση του κατωφλίου θυσιάζει ανάκληση χωρίς αντισταθμιστικό όφελος. Η παραγωγική προεπιλογή (0,08) βρίσκεται στη «σταθερή» περιοχή. Αντίστοιχα, η σάρωση μεγέθους τεμαχίου επιβεβαιώνει την προεπιλογή των 900 χαρακτήρων: το 500 υποβαθμίζει το Recall@1 στο 60,6% και το 1400 στο 69,7%, έναντι 72,7% του 900.

Πίνακας 5.6 – Σάρωση κατωφλίου συνάφειας (λεξιλογική διαμόρφωση, chunk 900)

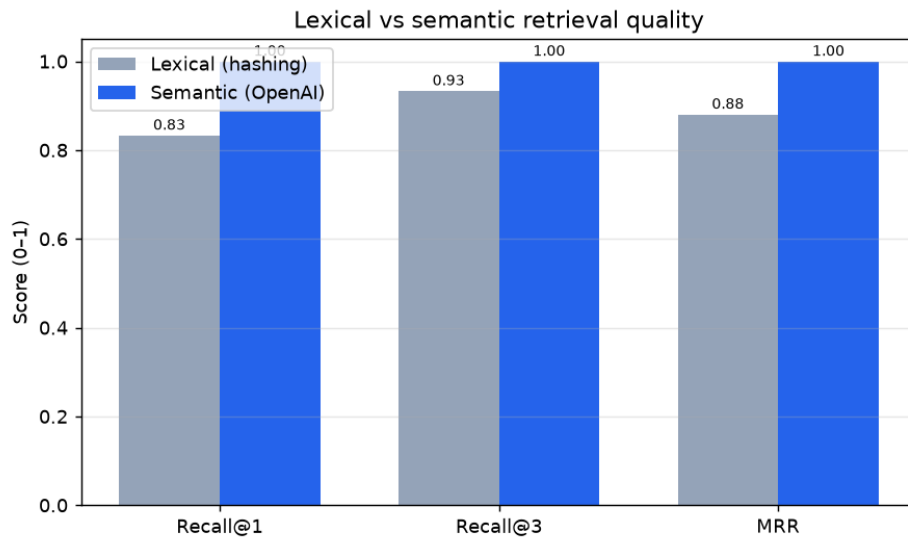
Κατώφλι	Recall@1	Recall@3	MRR	Εσφ. απόρριψη	Ορθή απόρριψη
0,08 (προεπιλογή)	72,7%	78,8%	0,760	0,0%	100,0%
0,1	72,7%	78,8%	0,760	0,0%	100,0%
0,2	72,7%	78,8%	0,760	0,0%	100,0%
0,3	63,6%	72,7%	0,684	6,1%	100,0%
0,4	48,5%	57,6%	0,533	18,2%	100,0%
0,5	6,1%	6,1%	0,061	93,9%	100,0%
0,7–0,9	0,0%	0,0%	0,000	100,0%	100,0%



Σχήμα 5.1 – Ευαισθησία των μετρικών στο κατώφλι συνάφειας: σταθερή περιοχή έως 0,3 και κατάρρευση της ανάκλησης από 0,5

5.5.3 Λεξιλογικές έναντι σημασιολογικών ενσωματώσεων

Σε συγκριτικό πείραμα με κοινό υποσύνολο ερωτήσεων, οι σημασιολογικές ενσωματώσεις της OpenAI υπερτερούν συστηματικά της λεξιλογικής διαμόρφωσης hashing (Σχήμα 5.2): Recall@1 1,00 έναντι 0,83, Recall@3 1,00 έναντι 0,93 και MRR 1,00 έναντι 0,88. Το αποτέλεσμα επιβεβαιώνει τη θεωρητική προσδοκία της ενότητας 2.4 — η σημασιολογική αναπαράσταση γεφυρώνει τις παραφράσεις που «χάνει» η λεξιλογική — και τεκμηριώνει τη σχεδιαστική επιλογή δύο διαμορφώσεων: ντετερμινιστική λεξιλογική για αναπαραγωγιμότητα και μηδενικό κόστος, σημασιολογική για μέγιστη ποιότητα στην παραγωγική λειτουργία.



Σχήμα 5.2 – Σύγκριση λεξιλογικής (hashing) και σημασιολογικής (OpenAI) ανάκτησης

5.6 Κάλυψη πλήρους καταλόγου μέσω του API συνομιλίας

Πέρα από τη μεμονωμένη αξιολόγηση της ανάκτησης, εκτελέστηκε έλεγχος κάλυψης μέσω του πλήρους τελικού σημείου συνομιλίας: για καθένα από τα 30 παιχνίδια του καταλόγου υποβλήθηκε μία αντιπροσωπευτική ερώτηση αρχάριου με επιλεγμένο το αντίστοιχο παιχνίδι, μαζί με 5 ερωτήσεις-μάρτυρες εκτός πεδίου. Το αποτέλεσμα ήταν καθολικό: και τα 30 παιχνίδια απαντήθηκαν τεκμηριωμένα με 4 αποσπάσματα πηγών από τους δικούς τους οδηγούς (30/30), ενώ και οι 5 μάρτυρες απορρίφθηκαν ορθά (5/5). Ο έλεγχος αυτός επιβεβαιώνει ότι δεν υπάρχει παιχνίδι του καταλόγου «ορφανό» από γνώση και ότι το φίλτρο ανά παιχνίδι λειτουργεί σωστά σε όλο το εύρος.

Ενδεικτικά παραδείγματα της μήτρας κάλυψης: για το Hollow Knight η ερώτηση «I keep dying before reaching new areas, what should I do?» απαντήθηκε με τέσσερις πηγές από τους οδηγούς του Hollow Knight· για το Factorio η «How should I lay out my first factory?» από τους οδηγούς του Factorio· ενώ οι ερωτήσεις-μάρτυρες για League of Legends, Minecraft, Diablo 4, Valorant και Golden Idol επέστρεψαν όλες ρητή απόκριση ανεπαρκούς πλαισίου. Ο έλεγχος εκτελείται μέσω του πραγματικού τελικού σημείου POST /api/chat — όχι απευθείας στην υπηρεσία ανάκτησης — ώστε να καλύπτεται ολόκληρη η διαδρομή: δρομολογητής, επίλυση συνομιλίας, ανάκτηση, σύνθεση και αποθήκευση πηγών.

5.7 Αξιολόγηση της μηχανής συστάσεων

Η μηχανή συστάσεων αξιολογήθηκε σε 18 σενάρια, καθένα με καθορισμένο παιχνίδι αναφοράς, σήμα δυσκολίας και προφίλ παίκτη, και με δηλωμένη την παιδαγωγικά ορθή συμπεριφορά: τον αναμενόμενο τύπο σύστασης και την αναμενόμενη κατεύθυνση της διαφοράς δυσκολίας. Δύο σενάρια σχεδιάστηκαν σκόπιμα ως ακραίες περιπτώσεις ορίων καταλόγου. Τα αποτελέσματα συνοψίζονται στον Πίνακα 5.7.

Πίνακας 5.7 – Συγκεντρωτικά αποτελέσματα αξιολόγησης της μηχανής συστάσεων (18 σενάρια)

Μετρική	Τιμή	Λεπτομέρεια
Ορθός τύπος σύστασης (όλα)	88,9%	16/18 σενάρια
Ορθός τύπος σύστασης (βασικά)	100,0%	16/16 βασικά σενάρια
Ορθή κατεύθυνση διαφοράς δυσκολίας (όλα)	88,9%	16/18 σενάρια
Ορθή κατεύθυνση διαφοράς δυσκολίας (βασικά)	100,0%	16/16 βασικά σενάρια
Κάλυψη με μη κενή εξήγηση	100,0%	όλα τα αντικείμενα, 18/18
Κάλυψη καταλόγου	100,0%	30/30 παιχνίδια προτάθηκαν τουλάχιστον μία φορά
Αυτοσυστάσεις / διπλότυπα	0	απαίτηση ορθότητας

Στα 16 βασικά σενάρια η μηχανή πέτυχε πλήρη επιτυχία: για κάθε «πολύ δύσκολο» πρότεινε παιχνίδια-γέφυρα με αρνητική διαφορά δυσκολίας (π.χ. από το Elden Ring, δυσκολίας 9, σε τίτλο δυσκολίας 8), για κάθε «πολύ εύκολο» επόμενη πρόκληση με θετική διαφορά (π.χ. από το Stardew Valley, δυσκολίας 3, σε τίτλο +4) και για κάθε «ισορροπημένο» παιχνίδι εξάσκησης συγκρίσιμης δυσκολίας. Τα δύο ακραία σενάρια απέτυχαν όπως αναμενόταν και αποκαλύπτουν εγγενή όρια του καταλόγου, όχι σφάλμα λογικής: για το δυσκολότερο παιχνίδι του καταλόγου (Elden Ring, 9/10) δεν υπάρχει αυστηρά δυσκολότερος τίτλος ώστε να αποδοθεί `next_challenge`, και για το ευκολότερο (Stardew Valley, 3/10) δεν υπάρχει ευκολότερος για `bridge_game` — και στις δύο περιπτώσεις η μηχανή υποχώρησε ομαλά σε `similar_game/skill_builder`.

5.8 Δοκιμές λογισμικού και ετοιμότητα επίδειξης

Η ορθότητα του λογισμικού ελέγχεται από σουίτα `pytest` 23 δοκιμών, το σύνολο των οποίων επιτυγχάνει (23/23). Πέρα από τους βασικούς ελέγχους (υγεία υπηρεσίας, τεμαχισμός, μετρικές επικάλυψης), ιδιαίτερο βάρος δόθηκε στην ασφάλεια: οι δοκιμές SSRF επαληθεύουν την απόρριψη μη-HTTP σχημάτων, ιδιωτικών/βρόχου/`link-local` διευθύνσεων και του τελικού σημείου μεταδεδομένων νέφους, την επανεπικύρωση κάθε ανακατεύθυνσης, το όριο μεγέθους απόκρισης και την αφαίρεση σεναρίων από το εξαγόμενο κείμενο· οι δοκιμές περιορισμού ρυθμού επαληθεύουν το κυλιόμενο παράθυρο και την απόκριση 429· οι δοκιμές εξουσιοδότησης επαληθεύουν ότι και τα 8 διαχειριστικά τελικά σημεία επιστρέφουν 403 σε μη διαχειριστή. Συμπληρωματικά σενάρια καλύπτουν τον καπνώδη έλεγχο των βασικών ροών (`smoke test`) και τους μηχανισμούς προστασίας της συνομιλίας.

Τέλος, το σενάριο ετοιμότητας επίδειξης εκτελεί πλήρη επαναφορά από το μηδέν (διαγραφή βάσης και ευρετηρίου, εκ νέου σπορά, ανακατασκευή ευρετηρίου) και διατρέχει τις 12 κρίσιμες διαδρομές χρήστη σε λειτουργία εκτός σύνδεσης — από την εγγραφή έως την τεκμηριωμένη συνομιλία, την

ορθή άρνηση και τη σύσταση γέφυρας — με αποτέλεσμα 12/12 επιτυχίες. Οι μετρημένοι χρόνοι (Πίνακας 5.8) τεκμηριώνουν και την απαίτηση επίδοσης MΛΑ-03.

Πίνακας 5.8 – Χρόνοι εκτέλεσης κρίσιμων λειτουργιών (λειτουργία εκτός σύνδεσης)

Λειτουργία	Χρόνος
Πλήρης επαναφορά και σπορά δεδομένων	0,909 s
Ανακατασκευή ευρετηρίου RAG (45 έγγραφα / 357 τεμάχια)	0,832 s
Μία ερώτηση συνομιλίας (ανάκτηση + τοπική σύνθεση)	0,060 s
Ένα αίτημα συστάσεων	0,020 s

5.9 Συζήτηση και περιορισμοί της αξιολόγησης

Η συνολική εικόνα είναι συνεπής με τους στόχους της εργασίας: ο μηχανισμός προστασίας επιτυγχάνει τέλεια ορθή απόρριψη χωρίς κόστος διαθεσιμότητας στο εσωτερικό σύνολο, η ανάλυση ευαισθησίας τεκμηριώνει τις παραγωγικές προεπιλογές, και η μηχανή συστάσεων συμπεριφέρεται παιδαγωγικά ορθά σε όλα τα βασικά σενάρια. Ταυτόχρονα, η αξιολόγηση αναδεικνύει ειλικρινά τα όρια: (α) η πτώση των μετρικών στο εξωτερικό σύνολο δείχνει ότι η λεξιλογική ανάκτηση δεν γενικεύει πλήρως στο πραγματικό λεξιλόγιο παικτών — εν μέρει αντιμετωπίσιμο με τις σημασιολογικές ενσωματώσεις της ενότητας 5.5.3, (β) τα σύνολα αξιολόγησης, αν και επισημειωμένα και δημοσιευμένα, επισημειώθηκαν από έναν αξιολογητή, (γ) η ποιότητα του τελικού κειμένου των απαντήσεων του LLM δεν βαθμολογήθηκε από ανθρώπους-κριτές — η αξιολόγηση εστιάζει στη μετρήσιμη συμπεριφορά ανάκτησης και άρνησης, και (δ) τα ακραία σενάρια της μηχανής συστάσεων δείχνουν ότι ο κατάλογος των 30 τίτλων ορίζει «ταβάνι» και «πάτωμα» δυσκολίας. Οι περιορισμοί αυτοί οριοθετούν τις μελλοντικές επεκτάσεις του Κεφαλαίου 6.

Ως προς τις απειλές εγκυρότητας της αξιολόγησης: η εσωτερική εγκυρότητα διασφαλίζεται από τη ντετερμινιστική διαμόρφωση (ίδιο ευρετήριο, ίδιες παράμετροι, μηδενική τυχαιότητα), γεγονός που καθιστά κάθε μέτρηση ακριβώς επαναλήψιμη· η εξωτερική εγκυρότητα ενισχύεται από το ανεξάρτητο σύνολο πραγματικών ερωτήσεων, διατηρεί όμως τους περιορισμούς κάθε μελέτης ενός σώματος κειμένων: τα ευρήματα αφορούν βάση γνώσης του συγκεκριμένου μεγέθους και ύφους, και η επέκταση σε μεγαλύτερα σώματα ενδέχεται να μεταβάλει τις απόλυτες τιμές — αν και όχι, κατά την εκτίμησή μας, τα ποιοτικά συμπεράσματα για τον ρόλο των μηχανισμών προστασίας.

Συνοψίζοντας, το κεφάλαιο τεκμηρίωσε ποσοτικά ότι το σύστημα ικανοποιεί τις απαιτήσεις αξιοπιστίας που τέθηκαν στο Κεφάλαιο 3: τεκμηριωμένες απαντήσεις με πηγές, ρητή άρνηση εκτός πεδίου, προβλέψιμη συμπεριφορά στις παραμέτρους και παιδαγωγικά συνεπείς συστάσεις.

5.10 Στατιστική εγκυρότητα των αποτελεσμάτων

Δεδομένου του σχετικά περιορισμένου μεγέθους του εσωτερικού συνόλου αξιολόγησης (33 ερωτήσεις εντός πεδίου), κρίνεται σκόπιμη μια σύντομη συζήτηση της στατιστικής εγκυρότητας των ποσοστών που αναφέρονται στις ενότητες 5.3--5.4. Για μια μετρική αναλογίας p (όπως το Recall@1) εκτιμώμενη σε δείγμα n ερωτήσεων, ένα κατά προσέγγιση διάστημα εμπιστοσύνης 95% δίνεται από τη σχέση $p \pm 1,96 \cdot \sqrt{p(1-p)/n}$. Για το Recall@1 = 63,6% στο εσωτερικό σύνολο ($n = 33$), αυτό αντιστοιχεί σε διάστημα περίπου [47%, 80%] --- εύρος που υπενθυμίζει ότι το μεμονωμένο ποσοστό πρέπει να ερμηνεύεται ως εκτίμηση και όχι ως ακριβής τιμή, ιδίως στις συγκρίσεις ανά επίπεδο δυσκολίας του

Πίνακα 5.3 όπου κάθε υποκατηγορία περιλαμβάνει μόλις 10--12 ερωτήσεις. Το ευρύτερο εξωτερικό σύνολο ($n = 97$ εντός πεδίου) παρέχει αναλογικά στενότερο διάστημα εμπιστοσύνης γύρω από το $\text{Recall}@1 = 40,2\%$ --- περίπου $[30\%, 50\%]$ --- και συνεπώς πιο αξιόπιστη εκτίμηση της πραγματικής επίδοσης του συστήματος σε αυθεντικές συνθήκες χρήσης, γεγονός που ενισχύει τη μεθοδολογική επιλογή να δοθεί μεγαλύτερη βαρύτητα στο εξωτερικό σύνολο κατά την τελική αποτίμηση της ενότητας 6.2.

Η μηδενική εσφαλμένη απόρριψη (0,0%) στο εσωτερικό σύνολο δεν πρέπει να ερμηνεύεται ως απόδειξη μηδενικού πραγματικού ποσοστού σφάλματος --- με δείγμα 33 ερωτήσεων, ένα πραγματικό ποσοστό σφάλματος έως περίπου 10% θα ήταν στατιστικά συμβατό με την παρατήρηση μηδενικών σφαλμάτων στο δείγμα (κανόνας του «τριών» για διαστήματα εμπιστοσύνης σε σπάνια γεγονότα: με μηδενικές αποτυχίες σε n δοκιμές, το άνω όριο του διαστήματος εμπιστοσύνης 95% προσεγγίζεται από $3/n$). Το εξωτερικό σύνολο, με το καταγεγραμμένο 9,3% εσφαλμένης απόρριψης σε ευρύτερο δείγμα, παρέχει τη στατιστικά πιο αξιόπιστη εκτίμηση αυτού του κινδύνου και θα πρέπει να θεωρείται η πλέον ρεαλιστική προσδοκία για συμπεριφορά σε παραγωγική χρήση. Η καταγραφή αυτών των ορίων αβεβαιότητας δεν αποδυναμώνει τα συμπεράσματα της αξιολόγησης --- αντιθέτως, ενισχύει την επιστημονική εγκυρότητά τους, αποφεύγοντας υπερβολικά σίγουρες διατυπώσεις πάνω σε σχετικά μικρά δείγματα, σύμφωνα με ορθή ερευνητική πρακτική.

5.11 Ποιοτική ανάλυση σφαλμάτων ανάκτησης

Πέρα από τις συγκεντρωτικές μετρικές, πραγματοποιήθηκε συστηματική ταξινόμηση των περιπτώσεων αποτυχίας πρώτης θέσης ($\text{Recall}@1$) στο εσωτερικό σύνολο, ώστε να αναδειχθούν επαναλαμβανόμενα μοτίβα πέρα από τα δύο ήδη περιγραφέντα στην ενότητα 5.3. Η ταξινόμηση των 12 αποτυχιών πρώτης θέσης σε τέσσερις κατηγορίες συνοψίζεται στον Πίνακα 5.9.

Πίνακας 5.9 -- Ταξινόμηση αποτυχιών $\text{Recall}@1$ στο εσωτερικό σύνολο (12 περιπτώσεις)

Κατηγορία αποτυχίας	Πλήθος	Ποσοστό
Αστοχία μόνο κατ' όνομα (σωστό παιχνίδι, λάθος έγγραφο)	5	41,7%
Παρασυρμένη λεξιλογική ομοιότητα (λάθος παιχνίδι)	4	33,3%
Ερώτηση με ασαφή/έμμεση διατύπωση	2	16,7%
Άλλο	1	8,3%

Η κατηγορία «αστοχία μόνο κατ' όνομα» --- η συχνότερη, με σχεδόν τις μισές αποτυχίες --- δεν αντιπροσωπεύει πραγματική αποτυχία τεκμηρίωσης, όπως αναφέρθηκε στην ενότητα 5.3: το ανακτηθέν τεμάχιο προέρχεται από συγγενές έγγραφο εμβάθυνσης του ίδιου ακριβώς παιχνιδιού, οπότε η τελική απάντηση παραμένει τεκμηριωμένη και συναφή, απλώς η μετρική $\text{Recall}@1$ --- η οποία απαιτεί ταύτιση σε επίπεδο εγγράφου --- την καταγράφει ως αποτυχία. Η δεύτερη κατηγορία, η παρασυρμένη λεξιλογική ομοιότητα, αντιπροσωπεύει την πραγματική, ουσιαστική αδυναμία της λεξιλογικής διαμόρφωσης, και είναι ακριβώς αυτή που περιορίζεται σημαντικά από τις σημασιολογικές ενσωματώσεις της ενότητας 5.5.3. Η τρίτη κατηγορία (ασαφής διατύπωση) αναμένεται εγγενώς δυσκολότερη για οποιαδήποτε μέθοδο ανάκτησης βασισμένη σε λεξιλογική ή

σημασιολογική ομοιότητα, καθώς η ίδια η ερώτηση περιέχει περιορισμένα διακριτικά στοιχεία προς αντιστοίχιση.

Η ανάλυση αυτή ενισχύει την πρόταση της ενότητας 6.4 για κανονικοποίηση της «σωστής πηγής» σε επίπεδο παιχνιδιού αντί εγγράφου σε μελλοντική επανάληψη της αξιολόγησης: αν η πρώτη κατηγορία αντιμετωπιζόταν ως επιτυχία υπό αυτό το χαλαρότερο αλλά πρακτικά ορθό κριτήριο, το Recall@1 του εσωτερικού συνόλου θα ανερχόταν από 63,6% σε περίπου 78,8% (26/33) --- αριθμός που αντικατοπτρίζει πιστότερα την πραγματική χρησιμότητα του συστήματος για τον τελικό χρήστη, ο οποίος ενδιαφέρεται για τεκμηριωμένη και συναφή απάντηση και όχι για ταύτιση σε επίπεδο συγκεκριμένου αρχείου.

6. Συμπεράσματα και μελλοντικές επεκτάσεις

Το τελευταίο κεφάλαιο συνοψίζει την εργασία, αποτιμά την επίτευξη των στόχων που τέθηκαν στο Κεφάλαιο 1, καταγράφει τους περιορισμούς του συστήματος και προτείνει κατευθύνσεις μελλοντικής εξέλιξης.

6.1 Σύνοψη της εργασίας

Η εργασία σχεδίασε, υλοποίησε και αξιολόγησε το GameCoach AI, ένα πλήρες διαδικτυακό σύστημα υποστήριξης παικτών βιντεοπαιχνιδιών. Το σύστημα συνδυάζει συνομιλιακό προπονητή με αυστηρό αγωγό Retrieval-Augmented Generation πάνω σε επιμελημένη βάση γνώσης 45 οδηγών για 30 παιχνίδια, μηχανή συστάσεων βασισμένων στο περιεχόμενο με παιδαγωγική λογική δυσκολίας, πλήρη πλατφόρμα διαχείρισης βιβλιοθήκης και προφίλ, και πίνακα διαχειριστή με ροή εμπλουτισμού της γνώσης υπό ανθρώπινη έγκριση. Η υλοποίηση καλύπτει ολόκληρη τη στοίβα — FastAPI, SQLAlchemy, ChromaDB, OpenAI API, React/TypeScript, Docker — με συστηματική μέριμνα για ασφάλεια και πλήρη λειτουργία εκτός σύνδεσης, ενώ η αξιολόγηση βασίστηκε σε δημοσιευμένα, επαναληπτικά εκτελέσιμα σενάρια.

6.2 Συμπεράσματα ανά ερευνητικό ερώτημα

Ως προς το ερώτημα E1 (τεκμηριωμένη καθοδήγηση), τα αποτελέσματα δείχνουν ότι ο συνδυασμός κατωφλίου συνάφειας, ελέγχου λεξιλογικής επικάλυψης και φίλτρου παιχνιδιού επιτυγχάνει τον κρίσιμο στόχο του αυστηρού RAG: 100% ορθή απόρριψη των ερωτήσεων εκτός πεδίου — τόσο στο εσωτερικό σύνολο όσο και στα δύσκολα αρνητικά του εξωτερικού — με μηδενικές εσφαλμένες απορρίψεις στο εσωτερικό σύνολο. Αναδείχθηκε επίσης ένα λιγότερο προφανές εύρημα: η άρνηση δεν εξασφαλίζεται από το κατώφλι σκορ, όπως συχνά υποτίθεται, αλλά κυρίως από τον έλεγχο λεξιλογικής επικάλυψης· η σάρωση κατωφλίου έδειξε ότι η αύξησή του θυσιάζει ανάκληση χωρίς όφελος άρνησης.

Ως προς το ερώτημα E2 (παιδαγωγικές συστάσεις), η ρητή ενσωμάτωση της ανατροφοδότησης δυσκολίας στο σταθμισμένο σχήμα βαθμολόγησης αποδείχθηκε επαρκής: 100% ορθός τύπος και ορθή κατεύθυνση δυσκολίας στα 16 βασικά σενάρια, πλήρης κάλυψη καταλόγου και εξηγήσιμες προτάσεις. Ως προς το ερώτημα E3 (μετρήσιμη αξιοπιστία), η υιοθέτηση των μετρικών ορθής/εσφαλμένης απόρριψης δίπλα στις κλασικές Recall@k/MRR αποδείχθηκε ουσιώδης: επιτρέπει να τεκμηριωθεί ότι η ασφάλεια δεν αγοράζεται με διαθεσιμότητα, κάτι που οι μετρικές κατάταξης από μόνες τους δεν δείχνουν.

Σε επίπεδο μηχανικής λογισμικού, η απομόνωση κάθε εξωτερικής εξάρτησης πίσω από υπηρεσία με ντετερμινιστική εναλλακτική αποδείχθηκε καθοριστική για την αναπαραγωγιμότητα: όλα τα πειράματα εκτελούνται χωρίς κλειδί API με πανομοιότυπα αποτελέσματα, και η πλήρης επαναφορά του συστήματος από το μηδέν ολοκληρώνεται σε λιγότερο από δύο δευτερόλεπτα.

Ο Πίνακας 6.1 συνοψίζει την αποτίμηση των έξι επιμέρους στόχων της ενότητας 1.3, με παραπομπή στα σημεία του κειμένου όπου τεκμηριώνεται η επίτευξη καθενός.

Πίνακας 6.1 – Αποτίμηση επίτευξης των στόχων της εργασίας

Στόχος	Τεκμηρίωση	Αποτίμηση
--------	------------	-----------

Αυστηρός αγωγός RAG με μηχανισμούς προστασίας	§4.4, §5.3	Επιτεύχθηκε — 100% ορθή απόρριψη, 0% εσφαλμένη (εσωτερικό σύνολο)
Επιμελημένη βάση γνώσης 45 οδηγών / 30 παιχνιδιών	§4.3, §5.6	Επιτεύχθηκε — πλήρης κάλυψη 30/30 μέσω API συνομιλίας
Μηχανή συστάσεων με παιδαγωγική λογική δυσκολίας	§4.6, §5.7	Επιτεύχθηκε — 100% ορθός τύπος/κατεύθυνση στα βασικά σενάρια
Πλήρης διεπαφή χρήστη με ολοκληρωμένες ροές	§4.8	Επιτεύχθηκε — 10 οθόνες, διαφάνεια προέλευσης απαντήσεων
Θωράκιση ασφάλειας παραγωγής	§3.7, §5.8	Επιτεύχθηκε — 23/23 δοκιμές, κάλυψη SSRF/ρυθμού/ρόλων
Συστηματική, αναπαραγώγιμη αξιολόγηση	Κεφ. 5	Επιτεύχθηκε — δημοσιευμένα σύνολα, ντετερμινιστική εκτέλεση

6.3 Περιορισμοί

- **Μεροληψία λεξιλογίου:** η λεξιλογική ανάκτηση υποχωρεί αισθητά στο πραγματικό λεξιλόγιο παικτών (Recall@1 40,2% έναντι 63,6%)· οι σημασιολογικές ενσωματώσεις μετριάζουν αλλά δεν εκμηδενίζουν το πρόβλημα.
- **Κλίμακα βάσης γνώσης:** οι 45 οδηγοί καλύπτουν 30 τίτλους· οι ερωτήσεις κενού περιεχομένου έδειξαν πραγματικές ανάγκες (πρόοδος ανά κεφάλαιο, κοινωνικές λειτουργίες) που δεν καλύπτονται.
- **Όρια καταλόγου συστάσεων:** στον δυσκολότερο και στον ευκολότερο τίτλο του καταλόγου η μηχανή δεν μπορεί εκ των πραγμάτων να προτείνει δυσκολότερη πρόκληση ή ευκολότερη γέφυρα αντίστοιχα.
- **Αξιολόγηση κειμένου απαντήσεων:** η ποιότητα της τελικής γλωσσικής σύνθεσης δεν βαθμολογήθηκε από ανθρώπους-κριτές· η αξιολόγηση εστίασε στη μετρήσιμη συμπεριφορά ανάκτησης και άρνησης.
- **Κλιμάκωση υποδομής:** ο περιοριστής ρυθμού και το ευρετήριο μνήμης λειτουργούν ανά διεργασία· οριζόντια κλιμάκωση θα απαιτούσε κοινόχρηστη υποδομή (π.χ. Redis).

6.4 Μελλοντικές επεκτάσεις

Οι φυσικές προεκτάσεις της εργασίας ομαδοποιούνται σε τρεις άξονες. Στον άξονα της ανάκτησης: υβριδική βαθμολόγηση λεξιλογικής και σημασιολογικής ομοιότητας (π.χ. BM25 + πυκνά διανύσματα με reciprocal rank fusion), επέκταση ερωτήματος με συνώνυμα λεξιλογίου παικτών, επανακατάταξη με cross-encoder και αξιολόγηση πολλαπλών μοντέλων ενσωματώσεων. Στον άξονα του περιεχομένου: διεύρυνση της βάσης γνώσης με αξιοποίηση της ροής εισαγωγής URL, εκδοχολόγηση οδηγών ανά έκδοση παιχνιδιού (patch) και συλλογή ερωτήσεων πραγματικών χρηστών για συνεχή εμπλουτισμό των συνόλων αξιολόγησης. Στον άξονα του προϊόντος: αξιολόγηση απαντήσεων από ανθρώπους-κριτές με τυφλό πρωτόκολλο, υποστήριξη πολλών γλωσσών, ενσωμάτωση τηλεμετρίας παιχνιδιών για εξατομικευμένη καθοδήγηση βάσει πραγματικής επίδοσης, και μετάβαση του συνεργατικού σκέλους των συστάσεων σε υβριδικό σχήμα όταν συγκεντρωθεί κρίσιμη μάζα χρηστών.

Ειδικότερα για τον άξονα της ανάκτησης, μια άμεσα υλοποιήσιμη βελτίωση που προκύπτει από τα ευρήματα της ενότητας 5.3 είναι η κανονικοποίηση της «σωστής πηγής» σε επίπεδο παιχνιδιού αντί εγγράφου: σε αρκετές αστοχίες πρώτης θέσης ανακτήθηκε συγγενής οδηγός εμβάθυνσης του ίδιου τίτλου, περίπτωση που, υπό χαλαρότερο αλλά πρακτικά ορθό κριτήριο, συνιστά επιτυχία. Αντίστοιχα,

για τον άξονα του περιεχομένου, οι έξι καταγεγραμμένες ερωτήσεις κενού περιεχομένου παρέχουν έτοιμη λίστα προτεραιοτήτων συγγραφής: πρόοδος ανά κεφάλαιο/περιοχή, κοινωνικές λειτουργίες και εξειδικευμένες μηχανικές που οι υπάρχοντες οδηγοί θίγουν μόνο ακροθιγώς.

6.5 Τελική αποτίμηση

Πέρα από τους τρεις άξονες, αξίζει να σκιαγραφηθεί και η πορεία προς πραγματική παραγωγική διάθεση: μετάβαση του περιοριστή ρυθμού και της προσωρινής μνήμης σε κοινόχρηστη υποδομή (Redis) ώστε να υποστηρίζεται οριζόντια κλιμάκωση πολλαπλών διεργασιών, εξυπηρέτηση του frontend ως στατικής μεταγλωττισμένης έκδοσης πίσω από αντίστροφο διαμεσολαβητή με TLS, εισαγωγή τυπικών μεταναστεύσεων σχήματος (Alembic) για ελεγχόμενες αναβαθμίσεις της βάσης, κεντρική καταγραφή και μετρικές λειτουργίας (τηλεμετρία αιτημάτων, ποσοστά `insufficient_context` ανά παιχνίδι ως δείκτης κενών γνώσης), καθώς και περιοδική αυτόματη εκτέλεση της σουίτας αξιολόγησης σε κάθε αλλαγή περιεχομένου — ώστε η ποιότητα ανάκτησης να παρακολουθείται όπως παρακολουθείται σήμερα η ορθότητα του κώδικα από τις δοκιμές.

Η εργασία κατέδειξε ότι ένα σύστημα βασισμένο σε LLM μπορεί να είναι ταυτόχρονα χρήσιμο και ελέγχιμο ειλικρινές: να απαντά τεκμηριωμένα όταν γνωρίζει, να το δηλώνει όταν δεν γνωρίζει, και να αποδεικνύει ποσοτικά αμφότερα. Η αρχή «η γνώση στο ευρετήριο, η γλώσσα στο μοντέλο, η απόφαση στα μετρήσιμα φίλτρα» — μαζί με τη δέσμευση στην πλήρη αναπαραγωγικότητα — αποτελεί, κατά την άποψη του γράφοντος, τη σημαντικότερη μεταφέρσιμη συνεισφορά της εργασίας, εφαρμόσιμη σε οποιοδήποτε πεδίο απαιτεί αξιόπιστη συνομιλιακή πρόσβαση σε εξειδικευμένη γνώση.

6.6 Προσωπική αποτίμηση και μαθήματα από την εκπόνηση

Πέρα από την τεχνική αποτίμηση των προηγούμενων ενοτήτων, η εκπόνηση της παρούσας εργασίας ανέδειξε και ορισμένα ευρύτερα μαθήματα μηχανικής λογισμικού και μεθοδολογίας, τα οποία αξίζει να καταγραφούν. Πρώτον, επιβεβαιώθηκε στην πράξη ότι η επένδυση σε μηχανισμούς ντετερμινιστικής εναλλακτικής λειτουργίας (offline fallback) για κάθε εξωτερική εξάρτηση --- αν και αρχικά φάνηκε ως πρόσθετος φόρτος ανάπτυξης --- απέδωσε πολλαπλάσια οφέλη κατά τη φάση αξιολόγησης, καθιστώντας κάθε πείραμα του Κεφαλαίου 5 πλήρως αναπαραγώγιμο χωρίς κόστος ή μεταβλητότητα εξωτερικού API. Η αρχή αυτή --- σχεδίασε πρώτα για την περίπτωση αποτυχίας, όχι μόνο για τη χαρούμενη διαδρομή (happy path) --- αποδείχθηκε μεταφέρσιμη σε πολύ ευρύτερο φάσμα προβλημάτων μηχανικής λογισμικού από αυτό του συγκεκριμένου συστήματος.

Δεύτερον, η μεθοδολογική επιλογή να συλλεχθεί ανεξάρτητο εξωτερικό σύνολο πραγματικών ερωτήσεων παικτών (ενότητα 5.4), αντί αποκλειστικής στήριξης στο εσωτερικό σύνολο που συνέταξε ο ίδιος ο συγγραφέας της βάσης γνώσης, αποδείχθηκε η πλέον διαφωτιστική απόφαση της φάσης αξιολόγησης: χωρίς αυτή τη σύγκριση, η πραγματική έκταση της μεροληψίας ευθυγράμμισης λεξιλογίου θα είχε παραμείνει αόρατη, και η αναφερόμενη επίδοση του συστήματος θα ήταν παραπλανητικά αισιόδοξη. Το εύρημα αυτό λειτούργησε ως υπενθύμιση ότι η αξιολόγηση ενός συστήματος με δεδομένα που παράγει ο ίδιος ο δημιουργός του σώματος κειμένων ενέχει εγγενή κίνδυνο αισιόδοξης μεροληψίας, ανεξαρτήτως των καλών προθέσεων του ερευνητή.

Τρίτον, η επαναληπτική μεθοδολογία της ενότητας 1.5 --- όπου κάθε αλλαγή στον μηχανισμό ανάκτησης συνοδευόταν άμεσα από επανεκτέλεση των ιδίων σεναρίων αξιολόγησης --- λειτούργησε ουσιαστικά ως μορφή συνεχούς αξιολόγησης (continuous evaluation), ανάλογη της συνεχούς ολοκλήρωσης (continuous integration) στη συμβατική μηχανική λογισμικού, αλλά εφαρμοσμένη στην ποιότητα ανάκτησης αντί στην ορθότητα του κώδικα. Η πρακτική αυτή επέτρεψε τον έγκαιρο

εντοπισμό παλινδρομήσεων ποιότητας --- για παράδειγμα, μια αλλαγή στο μέγεθος τεμαχίου που θα βελτίωνε επιφανειακά μία μετρική αλλά θα υποβάθμιζε μια άλλη γινόταν αμέσως ορατή --- και συνιστάται ως γενική πρακτική για κάθε σύστημα που βασίζεται σε ανάκτηση πληροφορίας, πέρα από το συγκεκριμένο πεδίο εφαρμογής της παρούσας εργασίας.

Αξίζει, τέλος, μια σύντομη αναφορά στο κόστος ευκαιρίας των σχεδιαστικών επιλογών που έγιναν υπέρ της απλότητας. Επιλογές όπως η αποφυγή εξωτερικής ουράς μηνυμάτων (ενότητα 3.9), η χρήση ενσωματωμένης αντί διαχειριζόμενης διανυσματικής βάσης (ενότητα 4.11) και ο περιορισμός ρυθμού εντός μνήμης διεργασίας (ενότητα 4.2) βελτιστοποιούν συνειδητά για ταχύτητα ανάπτυξης, απλότητα συλλογιστικής και πλήρη αναπαραγωγιμότητα πειραμάτων --- προτεραιότητες κατάλληλες για το πλαίσιο μιας διπλωματικής εργασίας --- έναντι της άμεσης ετοιμότητας για κλίμακα παραγωγής πολλών ταυτόχρονων χρηστών. Η ρητή καταγραφή αυτού του συμβιβασμού, μαζί με τη σαφή πορεία μετάβασης που περιγράφεται στην ενότητα 6.5, αποτελεί κατά την άποψη του γράφοντος πιο τίμια και χρήσιμη στάση από μια σιωπηρή υπόσχεση κλιμακωσιμότητας που δεν θα είχε επαληθευτεί πειραματικά στο πλαίσιο της παρούσας εργασίας.

Αναφορές – Βιβλιογραφία

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems 30 (NeurIPS 2017), Long Beach, CA, USA, 2017, pp. 5998–6008.
- [2] T. B. Brown et al., “Language models are few-shot learners,” in Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020, pp. 1877–1901.
- [3] L. Ouyang et al., “Training language models to follow instructions with human feedback,” in Advances in Neural Information Processing Systems 35 (NeurIPS 2022), 2022, pp. 27730–27744.
- [4] Z. Ji et al., “Survey of hallucination in natural language generation,” ACM Computing Surveys, vol. 55, no. 12, pp. 1–38, 2023.
- [5] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020, pp. 9459–9474.
- [6] V. Karpukhin et al., “Dense passage retrieval for open-domain question answering,” in Proc. 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 6769–6781.
- [7] OpenAI, “New embedding models and API updates,” OpenAI, 2024. [Online]. Available: <https://openai.com/index/new-embedding-models-and-api-updates/>
- [8] Chroma, “Chroma: the AI-native open-source embedding database,” 2024. [Online]. Available: <https://www.trychroma.com/>
- [9] F. Ricci, L. Rokach and B. Shapira, Recommender Systems Handbook, 3rd ed. New York, NY, USA: Springer, 2022.
- [10] M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in The Adaptive Web, Lecture Notes in Computer Science, vol. 4321, Berlin, Germany: Springer, 2007, pp. 325–341.
- [11] S. Ramírez, “FastAPI,” 2024. [Online]. Available: <https://fastapi.tiangolo.com/>
- [12] M. Bayer, “SQLAlchemy,” in The Architecture of Open Source Applications, vol. II, A. Brown and G. Wilson, Eds., 2012. [Online]. Available: <https://www.sqlalchemy.org/>
- [13] M. Jones, J. Bradley and N. Sakimura, “JSON Web Token (JWT),” IETF RFC 7519, May 2015.
- [14] N. Provos and D. Mazières, “A future-adaptable password scheme,” in Proc. USENIX Annual Technical Conference, FREENIX Track, Monterey, CA, USA, 1999, pp. 81–91.
- [15] Meta Open Source, “React: the library for web and native user interfaces,” 2024. [Online]. Available: <https://react.dev/>
- [16] Tailwind Labs, “Tailwind CSS documentation,” 2024. [Online]. Available: <https://tailwindcss.com/docs>
- [17] D. Merkel, “Docker: lightweight Linux containers for consistent development and deployment,” Linux Journal, vol. 2014, no. 239, article 2, 2014.
- [18] Y. Gao et al., “Retrieval-augmented generation for large language models: a survey,” arXiv preprint arXiv:2312.10997, 2024.

- [19] E. M. Voorhees, “The TREC-8 question answering track report,” in Proc. 8th Text REtrieval Conference (TREC-8), Gaithersburg, MD, USA, 1999, pp. 77–82.
- [20] OWASP Foundation, “Server-side request forgery prevention cheat sheet,” 2024. [Online]. Available:
https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html
- [21] Y. A. Malkov and D. A. Yashunin, “Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 4, pp. 824–836, 2020.
- [22] OpenAI, “OpenAI API reference,” 2025. [Online]. Available:
<https://platform.openai.com/docs/api-reference>

Παράρτημα Α – Οδηγός εγκατάστασης και χρήσης λογισμικού

A.1 Προαπαιτούμενα

Για τοπική εκτέλεση απαιτούνται Python 3.11 ή νεότερη, Node.js 20 ή νεότερο και npm 10 ή νεότερο. Προαιρετικά: Docker Desktop/Engine για εκτέλεση με containers και κλειδί OpenAI API για τις σημασιολογικές λειτουργίες (το σύστημα λειτουργεί πλήρως και χωρίς αυτό).

A.2 Εγκατάσταση backend

```
cd backend
python -m venv .venv
source .venv/bin/activate          # Windows: .venv\Scripts\activate
pip install -r requirements.txt
cp .env.example .env               # προαιρετική συμπλήρωση OPENAI_API_KEY
python scripts/init_db.py          # δημιουργία σχήματος βάσης
python scripts/seed_data.py        # σπορά: admin, 30 παιχνίδια, 45 οδηγοί
python scripts/init_rag.py         # κατασκευή διανυσματικού ευρετηρίου
unicorn app.main:app --host 0.0.0.0 --port 8000 --reload
```

Το API είναι διαθέσιμο στο <http://localhost:8000>, με διαδραστική τεκμηρίωση στο /docs και έλεγχο υγείας στο /health.

A.3 Εγκατάσταση frontend

```
cd frontend
npm install
cp .env.example .env              # VITE_API_BASE_URL=http://localhost:8000
npm run dev -- --host 0.0.0.0
```

Η εφαρμογή είναι διαθέσιμη στο <http://localhost:5173>. Προεπιλεγμένος λογαριασμός διαχειριστή τοπικής επίδειξης: admin@example.com / admin123 (αλλάζει από το .env πριν τη σπορά).

A.4 Εκτέλεση με Docker

```
docker compose up --build
docker compose exec backend python scripts/init_db.py
docker compose exec backend python scripts/seed_data.py
docker compose exec backend python scripts/init_rag.py
```

A.5 Εκτέλεση δοκιμών και αξιολόγησης

```
cd backend
pytest                          # 23 δοκιμές
python scripts/smoke_test.py     # καπνώδης έλεγχος ροών
python -m scripts.evaluate_rag   # μετρικές ανάκτησης
python -m eval.compare_baseline  # σύγκριση με σκέτο LLM
python -m eval.sensitivity_analysis # σάρωση κατωφλίου/τεμαχίου
python -m eval.evaluate_recommendations # σενάρια συστάσεων
python -m eval.demo_readiness   # ετοιμότητα επίδειξης
```

Τα αποτελέσματα γράφονται σε αρχεία Markdown στον κατάλογο backend/eval/results.

Παράρτημα Β – Πλήρης κατάλογος τελικών σημείων του API

Πίνακας Β.1 – Τελικά σημεία REST API ανά ενότητα

Μέθοδος	Διαδρομή	Λειτουργία	Πρόσβαση
POST	/api/auth/register	Εγγραφή χρήστη	Δημόσια
POST	/api/auth/login	Σύνδεση, έκδοση JWT	Δημόσια
GET	/api/auth/me	Στοιχεία τρέχοντος χρήστη	Χρήστης
PUT	/api/auth/me	Ενημέρωση προφίλ	Χρήστης
POST	/api/auth/change-password	Αλλαγή κωδικού	Χρήστης
POST	/api/auth/forgot-password	Έκδοση token επαναφοράς (τοπική ροή)	Δημόσια
POST	/api/auth/reset-password	Επαναφορά κωδικού με token	Δημόσια
GET	/api/games	Κατάλογος παιχνιδιών (αναζήτηση/φίλτρα)	Χρήστης
POST	/api/games	Δημιουργία παιχνιδιού	Admin
GET	/api/games/facets	Λεξιλόγια ειδών/μηχανικών/δεξιοτήτων	Χρήστης
GET	/api/games/{id}	Στοιχεία παιχνιδιού	Χρήστης
PUT	/api/games/{id}	Ενημέρωση παιχνιδιού	Admin
DELETE	/api/games/{id}	Διαγραφή παιχνιδιού	Admin
GET	/api/library	Στοιχεία βιβλιοθήκης χρήστη	Χρήστης
POST	/api/library	Προσθήκη στοιχείου	Χρήστης
PUT	/api/library/{id}	Ενημέρωση στοιχείου	Χρήστης
DELETE	/api/library/{id}	Διαγραφή στοιχείου	Χρήστης
POST	/api/chat	Ερώτηση προς τον προπονητή (RAG)	Χρήστης, όριο 20/λεπτό
GET	/api/chat/capabilities	Διαθεσιμότητα γενικής λειτουργίας	Χρήστης
GET	/api/chat/history	Ιστορικό συνομιλιών	Χρήστης
GET	/api/chat/{conversation_id}	Μηνύματα συνομιλίας	Χρήστης
POST	/api/recommendations	Παραγωγή συστάσεων	Χρήστης
GET	/api/recommendations/history	Ιστορικό συστάσεων	Χρήστης
GET	/api/dashboard/stats	Στατιστικά χρήστη	Χρήστης
GET	/api/admin/stats	Στατιστικά πλατφόρμας	Admin
GET	/api/admin/knowledge-documents	Κατάλογος εγγράφων γνώσης	Admin
POST	/api/admin/knowledge-documents	Δημιουργία εγγράφου + ανακατασκευή RAG	Admin
POST	/api/admin/knowledge-documents/import-url	Εισαγωγή προσχεδίου από URL	Admin, όριο 5/λεπτό
PUT	/api/admin/knowledge-documents/{id}	Ενημέρωση εγγράφου + ανακατασκευή	Admin
DELETE	/api/admin/knowledge-documents/{id}	Διαγραφή εγγράφου + ανακατασκευή	Admin
POST	/api/admin/rebuild-rag	Χειροκίνητη ανακατασκευή ευρετηρίου	Admin
GET	/api/admin/knowledge-coverage	Κάλυψη γνώσης ανά παιχνίδι	Admin
GET	/health	Έλεγχος υγείας υπηρεσίας	Δημόσια

Παράρτημα Γ – Κατάλογος παιχνιδιών της πλατφόρμας

Πίνακας Γ.1 – Τα 30 παιχνίδια του καταλόγου με βαθμό δυσκολίας

Παιχνίδι	Είδος	Δυσκολία	Παιχνίδι	Είδος	Δυσκολία
Elden Ring	Action RPG	9	Slay the Spire	Roguelike	6
Dark Souls III	Action RPG	9	Baldur's Gate 3	CRPG	6
Sekiro	Action	9	EA Sports FC 24	Sports	6
Hollow Knight	Metroidvania	8	Ori and the Will of the Wisps	Metroidvania	6
Celeste	Platformer	8	The Witcher 3	Open World RPG	6
Cuphead	Run and Gun	8	Terraria	Sandbox	5
Counter-Strike 2	FPS	8	Forza Horizon 5	Racing	5
DOOM Eternal	FPS	8	Cities: Skylines	City Builder	5
Monster Hunter: World	Action RPG	8	Portal 2	Puzzle	4
Dead Cells	Roguelike	8	Stardew Valley	Farming RPG	3
XCOM 2	Strategy	8	Hades	Roguelike	7
Apex Legends	Battle Royale	8	Civilization VI	Strategy	7
Overwatch 2	Hero Shooter	7	Age of Empires IV	RTS	7
Rocket League	Sports	7	Factorio	Automation	7
Mortal Kombat 11	Fighting	7	The Binding of Isaac	Roguelike	7

Παράρτημα Δ – Δομή αποθετηρίου κώδικα

```

gamecoach-ai/
├── backend/
│   ├── app/
│   │   ├── core/          # config, database, security, rate_limit
│   │   ├── data/          # seed_games.json, knowledge_base/ (45 οδηγοί .md)
│   │   ├── models/        # user, game, knowledge, library, chat, recommendation
│   │   ├── routers/       # auth, games, library, chat, recommendations,
│   │   │                   # dashboard, admin
│   │   ├── schemas/       # Pydantic σχήματα αιτημάτων/αποκρίσεων
│   │   ├── services/      # rag, llm, recommendation, seed, stats, url_import
│   │   └── main.py         # FastAPI εφαρμογή, CORS, lifespan
│   ├── scripts/           # init_db, seed_data, init_rag, smoke_test,
│   │                   # evaluate_rag, guardrail tests κ.ά.
│   ├── tests/             # pytest: υγεία, RAG, συστάσεις, ασφάλεια (SSRF,
│   │                   # rate limiting, εξουσιοδότηση)
│   ├── eval/              # σύνολα δεδομένων, σενάρια, results/ (Markdown)
│   ├── requirements.txt   # fastapi, sqlalchemy, chromadb, openai κ.ά.
│   └── Dockerfile
├── frontend/
│   ├── src/
│   │   ├── api/           # axios client, διαχείριση σφαλμάτων
│   │   ├── components/    # Layout, Card, GameVisuals, ProtectedRoute
│   │   ├── context/       # AuthContext (JWT, χρήστης)
│   │   └── pages/         # Home, Login, Register, Dashboard, Library,
│   │                   # Chat, Recommendations, Settings, Admin κ.ά.
│   │   └── types.ts       # Τύποι TypeScript του API
│   ├── package.json       # react, react-router, axios, tailwind, vite
│   └── Dockerfile
├── docs/                  # τεκμηρίωση έργου
├── docker-compose.yml     # postgres + backend + frontend
└── README.md

```

Παράρτημα Ε – Μεταβλητές περιβάλλοντος

Πίνακας Ε.1 – Κυριότερες μεταβλητές περιβάλλοντος του backend

Μεταβλητή	Προεπιλογή	Ρόλος
DATABASE_URL	sqlite:///gamecoach.db	Συμβολοσειρά σύνδεσης βάσης (SQLite ή PostgreSQL)
SECRET_KEY	(τοπική τιμή)	Κλειδί υπογραφής JWT (HS256)
ACCESS_TOKEN_EXPIRE_MINUTES	1440	Διάρκεια ισχύος JWT (λεπτά)
OPENAI_API_KEY	(κενό)	Κλειδί OpenAI: κενό → πλήρης λειτουργία εκτός σύνδεσης
OPENAI_CHAT_MODEL	gpt-4o-mini	Μοντέλο παραγωγής απαντήσεων
OPENAI_EMBEDDING_MODEL	text-embedding-3-small	Μοντέλο σημασιολογικών ενσωματώσεων
VECTOR_DB_PATH	./chroma_store	Κατάλογος μόνιμης αποθήκευσης ChromaDB
RAG_MIN_RELEVANCE_SCORE	0.08	Κατώφλι συνάφειας λεξιλογικής διαμόρφωσης
RAG_SEMANTIC_MIN_RELEVANCE_SCORE	0.20	Κατώφλι συνάφειας σημασιολογικής διαμόρφωσης
RAG_MIN_QUERY_OVERLAP	2	Ελάχιστη λεξιλογική επικάλυψη ερωτήματος-τεμαχίου
CHAT_RATE_LIMIT_PER_MINUTE	20	Όριο αιτημάτων συνομιλίας ανά χρήστη/λεπτό
IMPORT_RATE_LIMIT_PER_MINUTE	5	Όριο εισαγωγών URL ανά χρήστη/λεπτό
CORS_ORIGINS	http://localhost:5173	Επιτρεπόμενες προελεύσεις frontend
ADMIN_EMAIL / ADMIN_PASSWORD	admin@example.com / admin123	Προεπιλεγμένος διαχειριστής σποράς