

הערכה עצמית – פרויקט Chatbot Ollama 3997 🛠️

מזהה קבוצה:

30af9fe9-f034-4e99-972f-319b8171fb1d

תעודות זהות:

206591596

205965437

קישור לריפון:

<https://github.com/Almog16/3997-chatbot-ollama>

הציון העצמי שלי: 96 / 100

♦ תיעוד פרויקט (PRD + ארכיטקטורה) – 20/20

- נכתב PRD מפורט הכולל מטרות, KPIs, תרחישי שימוש, דרישות פונקציונליות ולא-פונקציונליות.
- התיעוד כולל סיכונים ופתרונות, משתמשים שונים, ואבני דרך ברורות.
- תיעוד הארכיטקטורה מקצועי, פירוט של הזרימה בין רכיבים, והסבר על תפקיד כל מודול.

♦ README ותיעוד קוד – 15/15

- README מקיף ומדויק עם הוראות הפעלה צעד-אחר-צעד, פקודות Makefile, בדיקות, ניפוי שגיאות וצילומי מסך.
- הקוד מלווה ב-docstrings וב-type hints, וכולל התייחסות להפרדת שכבות ולתחזוקה עתידית.
- שילוב של קבצי ADR ותיקיית docs / יוצר חוויית תיעוד שלמה.

♦ מבנה פרויקט ואיכות קוד – 15/15

- מבנה מודולרי ונקי, חלוקה נכונה ל-tests/, docs/, client/, ./src/.
- עקביות בשמות משתנים ופונקציות.
- Makefile ורשימת build מקצועית ונוחה. שימוש ב-Ruff בשביל לשמור על איכות קוד גבוהה לפי סטנדרטים.

♦ קונפיגורציה ואבטחה – 10/10

- שימוש תקין בקבצי `env` ובמשתני סביבה. כל הגדרות ה-`ports`, המודלים וה-`dependencies` מנוהלות דרך `Makefile` ו-`pyproject.toml`.
- אין מפתחות API בקוד.
- `gitignore` מעודכן.

♦ בדיקות ואיכות (14/15) – Testing & QA

- כיסוי בדיקות מצוין: כ-95% לפי הדוח.
- בדיקות מקיפות לצד שרת באמצעות `unit tests` והרצה נוחה ע"י `make test`. בנוסף בדיקות E2E לצד לקוח באמצעות `Playwright`.
- טיפול מלא במקרי קצה, הודעות שגיאה ברורות ולוגים מצוינים.
- בדיקות נוגעות גם ל-`agent mode` ול-`fallback scenarios`.

♦ מחקר וניתוח – 12/15

- תיעוד הבחירות הארכיטקטוניות והמניעים מאחוריהן.
- ניתוח ביצועים ו-`trade-offs` בין מודלים (כמו `gemma3` לעומת `qwen3`).
- חסר ניתוח אמפירי (`benchmarking` כמותי) או טבלאות תוצאות מניסויים.

♦ ממשק משתמש והרחבה (10/10) – UI/UX & Extensibility

- ממשק רספונסיבי, נקי וברור.
- תמיכה ב-`multi-chat` וב-`Agent Mode` עם הצגת סטטוס וכלים.
- הוכחה ברורה ליכולת הרחבה באמצעות מדריך הוספת כלים (`adding-tools`).

🗨️ נימוק כללי

הפרויקט נבנה מתוך הקפדה לעבור על כל סעיפי דף ההנחיות להערכה העצמית, תוך השקעה של כ-3–4 ימי פיתוח שכללו אפיון, תיעוד, פיתוח, בדיקות ותיעוד ויזואלי. השקענו מאמץ רב במטרה להגיש עבודה ברמה גבוהה באמת.

במהלך העבודה למדנו כיצד לנצל בצורה מושכלת את **כלי ה-AI** כדי לתמוך בכל שלבי הפיתוח, מהאפיון והעיצוב הארכיטקטוני ועד לבדיקות, תיעוד ושיפור חוויית המשתמש. השתמשנו בכלי הבינה המלאכותית כשותף תהליכי שאפשר לנו **לשפר באופן איטרטיבי** כל רכיב במערכת, לזהות נקודות חולשה בזמן אמת, וללטש את התוצאה הסופית עד שהגיעה לסטנדרט גבוה במיוחד של איכות ודיוק.

הפרויקט מדגים הבנה מלאה של תכנון מערכות מבוססות **Agentic AI**, שילוב נכון של **LangGraph** לניהול זרימות מורכבות, ויישום עקבי של עקרונות **אבטחה, נגישות והרחבה**.

מעבר לדרישות הרשמיות של המשימה, החלטנו להרחיב את היקף הפיתוח ולתמוך גם ב־**Agent Mode** הכולל שימוש ב־**tools (כלים חיצוניים)**. החלטה זו נועדה להדגים את יכולת ההרחבה של המערכת ואת הפוטנציאל שלה לתמוך בסוכנים חכמים בעלי יכולת אינטראקציה עם העולם החיצון, גם אם לא נדרשנו לכך במפורש. ההוספה הזו חיזקה את היכולות של הפרויקט, והפכה אותו לדוגמה מעשית למערכת מודולרית, גמישה ועתירת יכולות.

נקודות לשיפור: ניתן להעמיק את ניתוח הנתונים ולהוסיף השוואות ביצועים בין מודלים.

הצהרת יושר אקדמי (Academic Integrity Declaration)

אני מצהיר/ה בזאת ש:

ההערכה העצמית שלי היא כנה ואמיתית

בדקתי את העבודה מול כל הקריטריונים לפני קביעת הציון

אני מודעת/ת שציון עצמי גבוה יוביל לבדיקה דקדקנית יותר

אני מקבל/ת את העובדה שהציון הסופי עשוי להיות שונה מהציון העצמי

העבודה היא פרי עבודתי/נו (של הקבוצה) ואני/ו אחראים לכל תוכנה

10.11.25

Assignment 1: LLM Agent Orchestration

Grade Report

Student ID:	38993
Team:	30af9fe9-f034-4e99-972f-319b8171fb1d
Repository:	https://github.com/Almog16/3997-chatbot-ollama
Assessment Date:	December 01, 2025

FINAL GRADE ✓	71%
STATUS	GOOD

Your Performance

Good effort. You demonstrated solid foundation in software engineering. There are areas where you can strengthen your work to achieve higher marks.

✓ Strengths

- ✓ Exceptional research component (HAS JUPYTER NOTEBOOK with plots - analysis.ipynb)
- ✓ Perfect security (has .env.example and .gitignore)
- ✓ Strong architecture documentation (3 ADRs + architecture.md with mermaid diagrams)
- ✓ FastAPI + LangGraph implementation
- ✓ Proper git branching strategy (4 feature branches with PR workflow)
- ✓ Good testing coverage (26 tests with 72 assertions)
- ✓ Excellent modularity (71 lines/file average)
- ✓ 6 professional screenshots
- ✓ Comprehensive prompt documentation

■ Areas Needing Attention

- CRITICAL: Add cost analysis documentation (0/10 points lost)
- Improve PRD completeness (currently 33%, missing problem statement, success metrics)
- Increase docstring coverage (Python 73%, JS 38%)
- Add CI/CD pipeline
- Add test coverage configuration
- Add style guide documentation
- Add pre-commit hooks
- Improve commit message quality (some generic messages)

Continue improving. Focus on the areas identified above to elevate your work to the next level.

Assessed: December 01, 2025

This grade reflects your overall software engineering practices