

Chatbot Ollama 3997 – פרויקט עצמאי

מצהה קבוצה:

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

תעודות זהות:

206591596

205965437

קישור לריפורן:

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

הציון העצמי שלי: 96 / 100

♦ **тиיעוד פרויקט (PRD + ארכיטקטורה) – 20/20**

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

♦ **15/15 ותיעוד קוד – README**

- README מקייף ומודיק עם הוראות הפעלה צעד-אחר-צעד, פקודות Makefile, בדיקות, ניפוי, שגיאות וצלומי מסך.
- הקוד מלאה ב-`hints`, `type` ו-`docstrings`, וכלל התייחסות להפרדת שכבות ותחזוקה עתידית.
- שילוב של קבצי `ADR` ותיקיית `docs`/ יוצר חוותית תיעוד שלמה.

♦ **מבנה פרויקט ואיכות קוד – 15/15**

- מבנה מודולרי ונקי, חלוקה נconaה ל-`./src/`, `client/`, `docs/`, `tests/`-`ds`.
- עקביות בשמות משתנים ופונקציות.
- `Makefile` ורוטינת `build` מקצועית ונוחה. שימוש ב-`Ruff` בשבייל לשמזור על איכות קוד גבוהה לפני סטנדרטים.

♦ **קונפיגורציה ובטחה – 10/10**

- שימוש תקין בקבצי `.env` ובמשתני סביבה. כל הגדרות `ports`, המודלים וה-dependencies מנווהות דרך `Makefile` ו-`pyproject.toml`.

- אין מפתחות API בקוד.
- `gitignore` מעודכן.

♦ בדיקות ואיכות – (14/15)

- CISCO בדיקות מצוין: כ-95% לפי הדוח.
- בדיקות מקיפות לצד שרת באמצעות `unit tests` והרצת נוחה ע"י `make test`. בנוסף בדיקות E2E לצד廉價 באמצעות `Playwright`.
- טיפול מלא במקרים קצה, הודיעות שגיאה ברורות ולוגים מצוינים.
- בדיקות נוגעות גם ל-`agent mode` ו-`fallback scenarios`.

♦ מחקר וניתוח – 12/15

- תיעוד הבחירה הארכיטקטונית והמנועים מאחוריהן.
- ניתוח ביצועים | trade-offs בין מודלים (כמו `gemma3` לעומת `qwen3`).
- חסר ניתוח אמפירי (benchmarking כמוות) או טבלאות תוצאות מניסויים.

♦ ממשק משתמש ורחבה (10/10)

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

♦ נימוק כללי

הפרויקט נבנה מ厚厚 הקפדה לעبور על כל סעיף דף ההנחיות להערכת העצמית, תוך השקעה של כ-3–4 ימי פיתוח شاملן אפיון, תיעוד, פיתוח, בדיקות ותיעוד ויזואלי. השקעה ממוץ רב במטרה להציג עבודה ברמה גבוהה באממת.

במהלך העבודה למדנו כיצד לנצל بصورة מושכלת את **כלי ה-IA** כדי לתמוך בכל שלבי הפיתוח, מהאפיון והעיצוב הארכיטקטוני ועד לבדיקות, תיעוד ושיפור חווית המשתמש. השתמשנו בכל היבינה המלאכותית כשותף תהילתי שאפשר לנו לשפר באופן איטרטיבי כל רכיב המערכת, לזרות נקודות חולשה בזמן אמת, וללטש את התוצאה הסופית עד שהגיעה לסטנדרט גובה במיוחד של איכות ודיוק.

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

מעבר לדרישות הרשמיות של המשימה, החלטו להרחיב את היקף הפיתוח ולתמוך גם ב-**Agent Mode** הcoil שימוש ב-**tools** (כליים חיצוניים). החלטה זו נעדה להציג את יכולת ההרחבה של המערכת ואת הפוטנציאלי שלה לתמוך בסוכנים חכמים בעלי יכולת אינטראקטיבית עם העולם החיצון, גם אם לא נדרש לכך במפורש. ההוספה הזאת חיזקה את היכולות של הפרויקט, והפכה אותו לדוגמה מעשית לממשק מודולרי, גמיש ועתירת יכולות.

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

הצהרת יושר אקדמי (Academic Integrity Declaration)

אני מצהיר/ה בזאת ש:

הערכתה העצמית שלי היא כנה ואמיתית

בדקתי את העבודה מול כל הקритריונים לפני קביעת הציון

אני מודע/ת שציון עצמי גבוה יוביל לבדיקה דקדקנית יותר

אני מקבל/ת את העבודה שהציון הסופי עשוי להיות שונה מהציון עצמי

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

10.11.25

Assignment 1: LLM Agent Orchestration

Grade Report

Student ID:	38982
Team:	30af9fe9-f034-4e99-972f-319b8171fb1d
Repository:	https://github.com/Almog16/3997-chatbot-ollama
Assessment Date:	December 01, 2025



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
- ✓ Good testing coverage (26 tests with 72 assertions)
- ✓ 6 professional screenshots
- ✓ Modular codebase (71 lines/file average)
- ✓ 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

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