

## ollama rag - מטלה מספר 5 – הערכה עצמית

### אפשרות מספר 3

זהה קבוצה:

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

תעודות זהות:

206591596

205965437

קישור לリフォן:

<https://github.com/hilibr1/3997-ollama-rag>

הציון העצמי שלי: 96 / 100

### • **тиיעוד הפרויקט (PRD + ארכיטקטורה) – 20/20**

- PRD מקייף (docs/PRD.md): מסמך דרישות מוצר מפורט המכוסה את כל הניסויים (1-4), יעדים, מדדים ומטרות פדגוגיות
- תייעוד ארכיטקטורה מקייף (docs/architecture\_experiments\_1\_2.md): תייעוד טכני מעמיק עם דיאגרמות, החלטות עיצוב ותיאור ריכבים
- מדריכי ניסוי ספציפיים: מדריכים מפורטים נפרדים לניסוי 3 (אינטרגציה ענן) וניסוי 4 (שליפה קוונטוטואלית ודרוג מחדש) (docs/configuration.md)
- תייעוד קונפיגורציה (docs/configuration.md)

### • **14/15 ותייעוד קוד – README**

- README מקייף: מבנה הטוב עם סעיפים ברורים להתקנה, שימוש ופתרון בעיות
- הוראות שלב-אחר-שלב: הוראות התקינה מפורטות
- דוגמאות שימוש: מספר דוגמאות המציגות כיצד להריץ ניסויים שונים
- תייעוד קוד: docstrings עקובים לכל הפונקציות והמחלקות
- תייעוד נוסף עבור הניסויים תחת docs

### • **מבנה הפרויקט ואיכות הקוד – 14/15**

- עיצוב מודולרי: רכיבי RAG מופרדים בצורה נקייה
- הפרדת אחריות: לכל מודול אחריות יחידה ומוגדרת היטב
- שמות עקובים: עיקבה אחר 8 PEP utilities
- עקרון DRY, פונקציונליות משותפת הוצאה ל-
- חלק קטן מהקוד האינטגרטיבי יכול להיות מפורק לתת-פונקציות כדי לשפר קראות, כדוגמת קוד הניסויים עצמו, אבל בחרנו להשאיר אותו פחות מודולרי על מנת לשפר את הקראות שלו (מאחר ומדובר בניסוי שרצה כוסף).

## ◆ קונפיגורציה אבטחה – 9/10

- utzoreת YAML עם דרישות משתני סביבה ותמייה בארגומנטים של CLI
- ל התוצאה הרגישה (מפתחות URLs, API) משתמש במשתני סביבה. שימוש ב .env.example ו gitignore.
- 

## ◆ בדיקות ואיכות – 14/15

- סוייתת בדיקות מקיפה: 97 בדיקות ייחודו המכסות את כל רכיבי ה-RAG הליבטיים
- כיסוי גובה עבור מודול ה-RAG הליבטי
- אסטרטגיית בדיקות ממוקדת: מדידת כיסוי מתמקדת ב-rag/src/ (ליבת ה-RAG), למעט סקריפטי אינטגרציה
- בדיקות מקרי קצה: בדיקות מכוסות תנאי שגיאה, קלטים ריקים, קבצים חסרים, פרמטרים לא חוקיים
- שימוש נכון ב-mocks: לבוד ייחדות תחת בדיקה
- ארגון בדיקות: מבנה הבדיקות משקף את מבנה המקור

## ◆ מחקר וניתוח תוצאות – 15/15

- תוצאות ניסויים: 15 קבצי תוצאות ב-results/experiments ו results/comparison/compare\_\*.md
- סיכון: results/SUMMARY.md המרכז את כל הממצאים
- ויזואלייזציות: תיירות results/visualizations
- ניסויים שיטתיים: 15 יציאות ניסוי מלאות על פני כל 4 סוג הניסויים
  - וריאציות פרמטרים מרובות:
- ניסוי 2: בדיקת גודל LLM, מודלי embedding, פרמטרי chunking
- ניסוי 3: השוואת מקומי מול ענן (מצבים A, B, C עם AI OpenAI)
- ניסוי 4: בדיקת שליפה קוונטקטואלית ודירוג חדש
- תוצאות מפורטות: ניסוי כולל מדדים ב-JSON + ניתוח Markdown
  - דוחות השוואת 4 מסמכים המנתחים tradeoffs
  - מדדים במעקב: זמן אינדוקס, latency שאילתת, שימוש בזיכרון, דיק תשובה, עלויות טוקנים

## ◆ חוויה משתמש והרחבה – 10/10

- LI אינטואיטיבי: משק שורת פקודה ברור עם ברירות מחדל הגינויות
- ารגומנטים גמישים: תמייה בדריסט כל פרמטר תצורה דרך CLI
- logging ההודעות שגיאה בחרופות: הודעות שגיאת מהלך אינדוקס, שליפה ויצירה
- ארქיטקטורה מודולרית: קל להרחבה עם רכיבים חדשים
- עיצוב מוכן פלאגינים: רכיבי chunker, retriever, reranker RAG ניתנים להחלפה

## **סיכום כללי**

הפרויקט מציג רמה גבוהה בכל היבטי הפיתוח: תיעוד, ארכיטקטורה, איכות קוד, בדיקות, ניסויים אמפיריים וניתוח נתונים. הוא משלב בין הנדסת תוכנה מקצועית לבין מחקר אקדמי מבוסס-נתונים.

1. **שילוב בין עומק הנדסי לעומק מחקרי** – המערכת בנוייה בצורה מודולרית, נקייה ומורחבת, תוך ביצוע ניסויים מתוכננים היטב וניתוחים סטטיסטיים שימושיים הבנוי עומקה של ההשפעות של כל פרמטר.
2. **הקפדה על תיעוד** – ה-PRD, מסמך הארכיטקטורה וה-README מספקים תמונה מלאה וברורה של כל מרכיבי המערכת. כל החלטה מוסברת, וכל רכיב מתועד בrama שמצוירה מסמכי מוצר ותוכן בתעשייתו.
3. **איכות קוד גבוהה ועקבית** – המבנה המודולרי, שמירה על סטנדרטים אחידים, והפרדה חזקה בין שכבות הופכים את המערכת לקללה לתחזוקה ולהרחבה – אפילו חשוב בפרויקטים אמיתיים.
4. **בדיקות תוכנה בrama מקצועית** – מספר גבוהה של בדיקות, כיסוי גבוהה על החלקים הקritisטיים ושימוש נכון במקומות ממצביים על בגרות הנדסית והבנה של חשיבות בדיקות במערכות מובוזרות ורבות-רכיבים.
5. **חויה למשתמש והרחבה עתידית** – CLI נוח, יכולת להחליף מודלים בקלות, קונפיגורציה גמישה ורכיבים הניתנים להחלפה.

במהלך התרגיל למדנו כיצד להשתמש בוגן, להריץ ניסויים, ולהשווות בין שיטות שונות, כגון:

- בניית יישום RAG בסיסי מלא
- ביצוע ניסויים שיטתיים עם שינוי פרמטרים
- השוואת פרישה מקומית מול ענן
- יישום והערכת אסטרטגיות RAG מתקדמות (שליפה קונטקטואלית, דירוג מחדש)
- ניתוח תוצאות עם טבלאות והשוואות
- הסקת מסקנות עצמאיות המבוססות על ראיות אמפיריות

### **הצהרת יושר אקדמי (Academic Integrity Declaration)**

אני מצהיר/ה בזאת ש:

הערכתה העצמית שלי היא כנה ואמיתית

בדקתי את העבודה מול כל הקритריון לפני קביעת הציון

אני מודע/ת שציון עצמי גבוה יוביל לבדיקה דקדקנית יותר

אני מקבל/ת את העבודה שהצין הסופי עשוי להיות שונה מהציון העצמי

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

