

מטלת בית Jeen AI Solution

חלק הראשון- חשיבה עסקית ופרומפטים

1. בחר חברה מוכרת בשוק הישראלי
2. תאר שלושה יישומי AI שיכולים לסייע לחברה שבחרת, על אחד מתוך היישומים האלו להיות בוט AI
3. עליך ליצור מצגת עם השקפים הבאים:
 - 3.1. הסבר בקצרה על שנים מתוך שלושת יישומי ה-AI שאתה מציע לחברה
 - 3.2. הסבר בהרחבה על היישום השלישי (הבוט) ותאר כיצד הוא משרת את החברה
 - 3.3. בחירת מודל LLM-
 - 3.3.1. ערוך השוואה בין מודלי LLM של **Open AI** היכולים להיות פוטנציאליים ליישום זה
 - 3.3.2. ציין איזה מודל LLM הוא המודל האידיאלי מבחינתך ליישומים מסוג זה והסבר מדוע
 - 3.4. יצירת הבוט-
 - 3.4.1. עבור יישום הבוט AI צור פרומפט למודל **Gemini 2.0 Flash**
 - 3.4.2. הצג את הפרומפט המלא שכתבת
 - 3.5. הרץ את הפרומפט ב- Google ai studio והצג דוגמה לשני שיחות משתמש (צילומי מסך)
 - 3.6. שקף סיכום עם פסקה קצרה המציגה את התהליך מחקר שעשית (עד 200 מילים)
 4. דרישות הגשה: חלק זה יכתב בשפה העברית וישלח כמצגת

חלק שני- כתיבת מודול פייתון

1. פתח מודול ב Python-שמממש תהליך (Retrieval-Augmented Generation) RAG פשוט, הכולל שני סקריפטים נפרדים:
 - **index_documents.py:**
 - קלט: קובץ PDF או DOCX.
 - חילוץ טקסט נקי מהקובץ.
 - חלוקת הטקסט למקטעים (Chunks) באמצעות אחת משלוש אסטרטגיות:
 - גודל קבוע עם חפיפה (Fixed-size with overlap)
 - חלוקה לפי משפטים (Sentence-based splitting)
 - חלוקה לפי פסקאות (Paragraph-based splitting)
 - יצירת Embedding לכל מקטע בעזרת Google Gemini API
 - שמירת המקטעים יחד עם הוקטורים במסד PostgreSQL
 - **search_documents.py:**
 - קלט: שאילתה טקסטואלית מהמשתמש.
 - יצירת Embedding לשאילתה דרך Gemini API
 - השוואת וקטור השאילתה לכל הוקטורים במסד הנתונים באמצעות cosine similarity.
 - החזרת חמשת המקטעים הדומים ביותר לשאילתה.
2. השתמש ב- Gemini API להפקת Embeddings
3. הקפד על מבנה מסד נתונים PostgreSQL עם העמודות:
 - id מזהה ייחודי
 - chunk_text טקסט המקטע
 - Embedding וקטור ההטבעה

- Filename שם הקובץ המקורי
- split_strategy שיטת החלוקה שנבחרה
- created_at תאריך ההוספה (רשות)

4. אבטחת מידע:

- אל תשמור מפתחות API או פרטי חיבור בקוד.
- השתמש בקובץ .env עם משתנים מתאימים (GEMINI_API_KEY, POSTGRES_URL)
- ודא שקובץ .env נמצא ב-.gitignore

5. דרישות הגשה:

- קישור ל-GitHub עם שני הסקריפטים index_documents.py ו-search_documents.py
- קובץ README.md עם הסברים ברורים לגבי התקנה, הפעלה ודוגמאות שימוש.
- קוד נקי, מתועד ומודולרי.
- אין צורך בממשק גרפי, CLI בלבד מספק.