🕸️ מבצע "רשת הכתבים"

1. רקע מבצעי

במהלך הלחימה בעזה ולאחר אירועי השביעי באוקטובר, אותרו במנהרות ובמתקנים תת-קרקעיים מאגרי מידע רחבי היקף הכוללים תיעוד מצולם של מכתבים, מסמכים והתכתבויות פנימיות. החומרים כוללים תצלומים של תכתובות בין גורמים שונים וכן מסמכים מודפסים וכתבי יד.

המידע הגולמי אינו ממוין ואינו ניתן לחיפוש. כדי להפיק ממנו ערך מודיעיני נדרש תהליך טכנולוגי שיאפשר חילוץ טקסט מתמונות, ניקוי ועיבוד התוכן, חישוב מדדים בסיסיים, ואינדוקס מרכזי לצורך חיפוש ושליפה.

---

2. המשימה

מערכת "רשת הכתבים" מוקמת כ־Pipeline מבוסס אירועים (Event-Driven), מודולרי ומבוזר. עליכם לממש מערכת יציבה ונקייה ארכיטקטונית שתדע:

- לקלוט מאגר תמונות מתיקייה לוקאלית בפרויקט
- לחלץ מהן טקסט
- להעשיר את המידע במטא־דאטה בסיסי מתוך קובץ התמונה
- להעביר את הנתונים בין שלבים באמצעות Kafka
- לשמור קבצים בינאריים ב־MongoDB
- ליצור אינדקס חיפוש מרכזי ב־Elasticsearch
- לאפשר שליפה דרך API ודשבורד

---

3. מבט ארכיטקטוני

המערכת בנויה כשירותים עצמאיים המתקשרים ביניהם באמצעות Kafka. כל שירות אחראי על שלב אחד מוגדר בזרימת הנתונים. הנתונים עוברים מ־Ingestion דרך ניקוי וניתוח ועד אינדוקס וחיפוש.

---

4. דרישות מימוש – רכיבי המערכת

📥 Ingestion Service

עליכם לממש שירות קליטה האחראי על:

- קריאת קבצי תמונות מתיקייה לוקאלית בפרויקט
- יצירת מזהה ייחודי (image_id) לכל קובץ

- חילוץ מטא־דאטה מהתמונה הכולל לפחות: file_size_bytes (גודל הקובץ בבייטים), width (רוחב בפיקסלים), height (גובה בפיקסלים), file_format (פורמט הקובץ)
- ביצוע OCR וחילוץ טקסט גולמי
- שליחת הקובץ הבינארי בלבד ל־MongoDB Loader באמצעות REST
- פרסום אירוע RAW ל־Kafka הכולל image_id, raw_text ו־metadata

שירות זה מהווה את נקודת הכניסה ל־Pipeline.

🗄️ MongoDB Loader Service

עליכם לממש שירות REST ייעודי האחראי על:

- קבלת קובץ בינארי בבקשת POST
- שמירת הקובץ ב־MongoDB באמצעות GridFS
- עבודה מופרדת לחלוטין מ־Kafka ומהלוגיקה של עיבוד טקסט

שירות זה שומר קבצים בלבד ואינו מנהל מטא־דאטה או טקסט.

🧼 Clean Service

עליכם לממש שירות צרכן (Consumer) האחראי על:

- האזנה ל־Topic RAW
- ניקוי סימני פיסוק ותווים מיוחדים מיותרים (כגון .,!?"'() וכדומה), הסרת Stop Words בסיסיות, והמרת כל הטקסט ל־lowercase
- פרסום אירוע CLEAN ל־Kafka

אין לבצע חישובים אנליטיים בשלב זה.

📊 Analytics Service

עליכם לממש שירות צרכן נוסף האחראי על:

- האזנה ל־Topic CLEAN
- חישוב מדדים בסיסיים על הטקסט
- פרסום אירוע ANALYTICS ל־Kafka

השירות עוסק בניתוח בלבד ואינו משנה את הטקסט.

🔎 Elastic Indexer

עליכם לממש שירות צרכן רב־נושאי האחראי על:

- האזנה ל־RAW, CLEAN ו־ANALYTICS
- איחוד הנתונים למסמך יחיד לפי image_id
- ביצוע Upsert למסמך באינדקס Elasticsearch

השירות יוצר אינדקס חיפוש אחיד ומרכזי.

🌐 Query API

עליכם לממש שירות REST האחראי על:

- קבלת בקשות חיפוש
- ביצוע שאילתות ל־Elasticsearch
- החזרת תוצאות בצורה מובנית

השירות אינו מבצע עיבוד נתונים.

📈 Streamlit Dashboard

עליכם לממש ממשק תצוגה האחראי על:

- קריאה ל־Query API
- הצגת תוצאות בצורה גרפית וברורה
- חקירה אינטראקטיבית של הנתונים

---

5. טכנולוגיות לשימוש

- Python 3.10+
- FastAPI – למימוש שירותי REST
- Apache Kafka – מתווך הודעות בין השירותים
- confluent-kafka – קליינט Kafka עבור Python
- MongoDB + GridFS – אחסון קבצים בינאריים
- pymongo – חיבור ל־MongoDB
- Elasticsearch – מנוע אינדוקס וחיפוש
- elasticsearch (Python client) – חיבור ל־Elasticsearch
- Streamlit – ממשק תצוגה
- OCR: יש להשתמש ב־pytesseract (עטיפה ל־Tesseract)

---

6. דרישות כלליות

- אין לבצע Hard-Coding של כתובות, פורטים או סיסמאות.
- כל תצורה חייבת להגיע ממשתני סביבה (ENV).

- יש לעבוד עם Git בצורה מסודרת ועם קומיטים נקיים.
- יש לשמור על שמות קבצים ומחלקות לפי Naming Convention עקבי.
- אין להשתמש ב־print(); יש לממש Logging מסודר.
- יש להקפיד על הפרדת אחריות בין רכיבים.
- הקוד חייב להיות מודולרי, קריא וניתן להרחבה.

המטרה היא לא רק לגרום למערכת לעבוד — אלא לבנות אותה נכון.

אתם מקבלים אחריות מלאה על הארכיטקטורה, על איכות הקוד ועל עמידה בעקרונות מקצועיים.

בהצלחה במימוש מבצע "רשת הכתבים".