**TrendSpotter**

**מטרת הפרויקט:**  
זיהוי בזמן אמת מגמות טרנדיות בנושאים שונים, מיפוי לפי אזור גאוגרפי, ומתן מענה לשאלות כמו:

* "מה המסעדה הכי טרנדית בתל אביב?"
* "מה הטרנד הכי חזק בדרום קוריאה?"

שלב ראשון: מקורות מידע מטוויטר בלבד.  
שלבים הבאים: הרחבה לאינסטגרם ופייסבוק.

**שלבים מפורטים:**

שלב 1 – הגדרת תחום ותכולה ראשונית

* מקור נתונים בשלב זה - טוויטר (Twitter API v2).
* סוג המידע שנאסוף - ציוצים פופולריים לפי האשטגים, מילות מפתח, אזור גאוגרפי, ושפה.
* שפת תכנות וטכנולוגיות –
  + - Backend: Python (Flask/FastAPI)
    - ספריות עיקריות - tweep או requests), pandas, nltk / spaCy, openai.
    - DB: SQLite/PostgreSQL לאחסון נתונים והיסטוריה.
    - לשלבים מתקדמים - React

שלב 2 – הכנת סביבת פיתוח

שלב 3 – חיבור ל-API של טוויטר

* הרשמה ל-Twitter Developer Portal
* פתיחת פרויקט חדש וקבלת API Keys.
* חיבור באמצעות tweepy:
  + - כתיבת מודול שמבצע חיפוש ציוצים לפי:
    - מילות מפתח (keywords)
    - האשטגים (#)
    - מיקום גאוגרפי (geo search)
    - מגבלות זמן (לדוגמה – 24 שעות אחרונות)

שלב 4 – עיבוד ראשוני של טקסט (NLP)

* ניקוי הטקסט:
  + - הסרת קישורים, אזכורים (@user), האשטגים (#hashtag), סמיילים.
* הפחתת רעשים:
  + - המרה לאותיות קטנות, הסרת סימני פיסוק, מילות עצירה (stop words).
* טוקניזציה:
  + - פיצול המילים ליחידות בסיסיות.
* למטרות זיהוי טרנדים:
  + - ספירת הופעות של מילים וביטויים נפוצים.

שלב 5 – זיהוי מגמות

* ספירת הופעות:
  + - כמה פעמים הופיעה כל מילה/ביטוי בפרק זמן נתון.
* השוואה לתקופות קודמות (Spike Detection):
  + - אם מילה הופיעה פי 2–3 יותר מהרגיל → מועמדת להיות טרנד.
* סינון:
  + - הסרת מילים כלליות/ריקות ("היום", "חדש").
* שמירה:
  + - הכנסת המידע למסד הנתונים עם חותמת זמן.

שלב 6 – בניית סוכן ה-AI

מטרות הסוכן:

* איסוף נתונים אוטומטי:
  + - כל X דקות להריץ שאילתות API.
* עיבוד:
  + - הרצת תהליך NLP לניקוי והפקת מגמות.
* מענה שאלות:
  + - קבלת שאלה חופשית מהמשתמש (לדוגמה: "מה הכי טרנדי בניו יורק?").
    - שליפת הנתונים הרלוונטיים מה-DB.
    - בניית תשובה ידידותית למשתמש.

שלב 7 – בניית מסד נתונים

* טבלאות מוצעות:
  + - trendsמילה/ביטוי, מיקום, ספירה, אחוז עלייה, תאריך.
    - Tweetsטקסט מלא, מזהה ציוץ, תאריך, מיקום.
* שמירת היסטוריה:
  + - לצורך השוואה וניתוח לטווח ארוך.

שלב 8 – נקודת כניסה (app.py)

* כתיבת API עם Flask/FastAPI:
  + - /trends?location=tel\_aviv – מחזיר טרנדים למיקום נתון.
    - /ask?query="מה המסעדה הכי טרנדית בתל אביב" – מחזיר תשובה מעובדת ע"י AI.
* חיבור ה-AI (OpenAI API):
  + - שילוב GPT להסבר/סיכום של הנתונים הגולמיים.

שלב 9 – ממשק משתמש בסיסי

* גרסה ראשונית (CLI או ווב פשוט):
  + - תיבת חיפוש לשאלה חופשית.
    - הצגת רשימת טרנדים רלוונטיים עם מדד פופולריות.
* שלבים מתקדמים:
  + - Dashboard גרפי עם גרפים, מפה אינטראקטיבית.

שלב 10 – הרחבות עתידיות

* מקורות נוספים: Instagram Graph API, Facebook Graph API.

Machine Learning לזיהוי טרנדים:

* + - לא רק ספירה, אלא ניתוח הקשרים בין נושאים.
* חיזוי מגמות:
  + - שימוש ב-ML לניבוי אילו טרנדים צפויים לצמוח.

המפתח:

AIzaSyDzzFDO-hUlWywCo9JteP4YOuLWSEfD-z0

למנגו אטלס -

mongodb+srv://avivtamari:VZgOmJJyxnc5Rhzh@trendspotter.rcvrxee.mongodb.net/