PRD: מנתח טרנדים של סרטים/סדרות (MVP - גרסת TMDB)

תאריך: 18 באוקטובר 2025 מחבר: [שם המפתח/ת] גרסה: 1.1

1. חזון ומטרות

חזון: לפתח מערכת בקאנד מודרנית, מבוססת-אירועים (Event-Driven) ומדרגית, המחשבת ציון טרנד סופי (Final Trend Score) לסרט או סדרה, על ידי שילוב אוטומטי של אירועים פנימיים (צפיות באפליקציה) ונתונים חיצוניים (ציוני קהל מ-TMDB).

מטרה: להדגים שליטה מלאה בארכיטקטורת מיקרוסרביסים, כולל אינטגרציה עם API צד שלישי, באמצעות Docker, Kubernetes ותקשורת אסינכרונית (RabbitMQ).

2. מפרט טכנולוגי (Tech Stack)

| רכיב | טכנולוגיה | תפקיד |
| --- | --- | --- |
| Backend API | FastAPI (Python) | קבלת אירועים (צפיות) והצגת תוצאות סופיות. |
| Persistence | PostgreSQL + SQLAlchemy (ORM) | שמירת נתוני הסרטים/סדרות והציונים המחושבים. |
| Message Broker | RabbitMQ | ניהול תורים והעברת אירועים בין השירותים. |
| Workers | Python (Pika/Celery) | Internal Worker ו-External Worker לעיבוד אסינכרוני. |
| API חיצוני | TMDB API (The Movie Database) | מקור לציון הקהל החיצוני (vote\_average). |
| Frontend UI (MVP) | React / Vue.js (בסיסי) | ממשק פונקציונלי לצפייה בתוצאות ושליחת טריגרים. |
| Containerization | Docker Compose | הפעלת סביבת הפיתוח המקומית (כל 6 הרכיבים). |
| Deployment | Kubernetes (Minikube) | פריסת המערכת המלאה. |

ייצוא אל Sheets

3. מודל נתונים ואינטגרציה עם TMDB

3.1. אינטגרציה עם TMDB (התפקיד של External Worker)

ה-External Worker יבצע קריאת API אמיתית ל-TMDB. לצורך הפרויקט, נשתמש ב-*TMDB ID* כמזהה המוצר.

| קריאה ל-TMDB API (דוגמה) | נתונים רלוונטיים |
| --- | --- |
| GET /3/movie/{tmdb\_id} | title, release\_date, vote\_average (הציון החיצוני). |

ייצוא אל Sheets

3.2. מודל הנתונים (PostgreSQL Schema)

| שדה | סוג | תפקיד |
| --- | --- | --- |
| id | INT (PK) | מזהה הסרט/סדרה. |
| tmdb\_id | INT (Unique) | המזהה של הסרט ב-TMDB (מפתח לאינטגרציה). |
| title | VARCHAR | שם הסרט/סדרה. |
| internal\_views\_count | INT | מונה צפיות פנימיות בתוך האפליקציה שלך (מעודכן ע"י Internal Worker). |
| external\_score | FLOAT | הציון שחזר מ-TMDB (vote\_average, בין 0.0 ל-10.0). |
| final\_trend\_score | FLOAT | הציון המשולב הסופי (לדוגמה: ממוצע משוקלל של צפיות פנימיות וציון חיצוני). |
| last\_updated\_at | TIMESTAMP | מתי הציון עודכן לאחרונה. |

ייצוא אל Sheets

4. דרישות פונקציונליות (Features & Requirements)

א. שירות API ראשי (FastAPI)

| # | פיצ'ר | דרישה |
| --- | --- | --- |
| R1.1 | רישום צפייה (Event Source) | POST /api/movies/view עם {tmdb\_id: int}. שולח אירוע ל-MQ. |
| R1.2 | טריגר לעדכון חיצוני | POST /api/movies/trigger\_external\_update/{tmdb\_id}. שולח אירוע ל-MQ. |
| R1.3 | קבלת ציון סופי | GET /api/movies/{tmdb\_id}/trend. מחזיר את הציון המשולב. |
| R1.4 | רשימת סרטים | GET /api/movies. מחזיר רשימה של כל הסרטים המנוהלים. |
| R1.5 | רישום סרט חדש | POST /api/movies/register עם {tmdb\_id: int}. מוסיף סרט חדש ל-DB. |

ייצוא אל Sheets

ב. שירותי Workers (Internal & External)

| # | Worker | פעולה |
| --- | --- | --- |
| R2.1 | Internal Worker | מאזין לתור VIEW\_EVENT. מעדכן את internal\_views\_count ב-Postgres. |
| R2.2 | External Worker | מאזין לתור EXTERNAL\_SCORE\_UPDATE. מבצע קריאה ל-TMDB API. מעדכן את external\_score ואת הציון המשוקלל final\_trend\_score ב-Postgres. |

ייצוא אל Sheets

ג. דרישות ממשק משתמש בסיסי (Frontend UI)

| # | דרישה פונקציונלית | תיאור |
| --- | --- | --- |
| F1.1 | לוח סרטים (Dashboard) | הצגת טבלה המציגה את כל שדות הסרט: שם, מונה צפיות, ציון TMDB, וציון טרנד סופי. |
| F1.2 | טריגר צפייה | כפתור "הוסף צפייה" ליד כל סרט. שולח בקשת POST ל-R1.1. |
| F1.3 | טריגר חיצוני | כפתור "עדכן ציון TMDB". שולח בקשת POST ל-R1.2. |
| F1.4 | רענון נתונים | מנגנון רענון ידני (כפתור) לטעינת הנתונים העדכניים מה-API (R1.4). |

ייצוא אל Sheets

5. תוכנית הפריסה והבדיקות (DevOps & QA)

| תחום | דרישה |
| --- | --- |
| Docker Compose | קובץ יחיד המפעיל את כל 6 הרכיבים (כולל TMDB Worker שמשתמש במפתח API אמיתי). |
| Kubernetes Deployment | קובצי YAML נפרדים לכל אחד מה-Workers וה-FastAPI. |
| בדיקות | Unit Tests ללוגיקה העסקית ולפונקציות הגישה ל-DB/TMDB. |
| Onboarding | קובץ README.md ברור שמסביר את הוראות ההתקנה, ההפעלה והשימוש. |