

בפרויקט. אנחנו הופכים את ה-YUV.AI Trends לגרסה רזה (MVP) המבוססת על טכנולוגיות JavaScript מקצה לקצה, מותאמת ל-Vercel, תוך שמירה על העיצוב המינימליסטי ודגש על CSS נקי ללא ספריות חיצוניות (בלי Tailwind או Bootstrap).

האתר של יובל : [GitHub - hoodini/yuv-ai-trends: YUV.AI Developers AI Trends - Beautiful Gen AI & ML news aggregator with Apple-inspired design](https://github.com/hoodini/yuv-ai-trends)

הסבר על סקילס : <https://www.youtube.com/watch?v=d4dPB8bJVMl>

## מסמך דרישות: אגרגטור חדשות AI (גרסה רזה)

### 1. תיאור המוצר

אפליקציית ווב המרכזת טרנדים שבועיים מעולם ה-AI/ML. המערכת תציג רשימה של פרויקטים פופולריים מ-GitHub ו-Hugging Face, תסכם אותם בעזרת LLM, ותציג אותם בממשק נקי בהשראת Apple.

### 2. ארכיטקטורה טכנולוגית

- **Framework:** Next.js (ל-Vercel-מותאם).
- **Frontend:** React (Functional Components).
- **Backend:** Next.js API Routes (Serverless Functions).
- **Styling:** Vanilla CSS (CSS Modules) - ללא ספריות עיצוב.
- **Storage:** LocalStorage (לבד לשמירת מפתחות).

### 3. דרישות פונקציונליות (MVP)

#### א. שרת (Backend / API Routes)

- **Endpoint לאיסוף נתונים:** פונקציה שתמשוך נתונים מ-GitHub API (פרויקטים עם הכי הרבה כוכבים ב-24 השעות האחרונות תחת תגיות "AI" או "Machine Learning").
- **Endpoint לסיכום:** פונקציה המקבלת טקסט ושולחת אותו ל-API של OpenAI/Anthropic/Groq כדי לקבל סיכום של עד 3 שורות.
- **Caching:** מימוש זיכרון מטמון בסיסי בשרת (למשל ל-5 דקות) כדי לא לחרוג ממרחב הבקשות של ה-APIs.

#### ב. ממשק משתמש (Frontend)

- **דף הבית:** תצוגת "Feed" של כרטיסים (Cards). כל כרטיס כולל:

- שם הפרויקט/מאמר.
- תיאור מקורי קצר.
- כפתור "Summarize" (או סיכום אוטומטי).
- **מסך הגדרות (Modal או דף נפרד):** מקום להזנת מפתח API אישי (שיישמר ב-LocalStorage) כדי שהמשתמש יוכל לבצע סיכומים על חשבון.

## 4. דרישות עיצוב (Apple-inspired CSS)

מכיוון שאסור להשתמש בספריות, נתמקד ב:

- אפשר להשתמש ב כל skill שתבחרו מהבנק של Cloud Code tu או לחילופין ברשימה הבאה (לא עקרוני)
- טיפוגרפיה: שימוש בפונטים מסוג Sans-serif
- צבעים: רקע לבן נקי או אפור בהיר מאוד (#F5F5F7), טקסט שחור/אפור כהה.
- אפקטים: פינות מעוגלות (`border-radius: 12px`), צללים עדינים מאוד (`box-shadow`).

## 5. שלבי הביצוע

1. **שלב א':** הקמת פרויקט Next.js והגדרת ה-API Route הראשון שמושך נתונים מ-GitHub.
2. **שלב ב':** יצירת רכיב React לכרטיס (NewsCard) ונתיבת ה-CSS הבסיסי ב-`styles/globals.css`.
3. **שלב ג':** חיבור לממשק סיכום (AI) המשתמש במפתח ה-API שנמצא בדפדפן.
4. **שלב ד':** פריסה (Deployment) ל-Vercel בלחיצת כפתור.

## 6. הערות

- אין מניעה להשתמש ב AI , ו skill. הנחת המוצא שיש לכם Copilot ומדל openAI מעצם היותכם עם כתובת ac, או שימוש בכל מודל אחר Groq (חינמי), claud code, .....
- vercel כלי ענן נוח להעביר פרויקטים היישר מ GitHub
- react /next.js ילמדו עד שבוע לפני הגשת התרגיל .
- בסיכומי של תרגיל אתם אמורים להבין כל שורת קוד .
- שם הפרויקט שלכם ב GitHub יהיה Ex2
- ב README.MD הסברים , API וכתובת האתר .

