# פרויקט סיום – טכנולוגיות אינטרנט ופיתוח פול סטאק

**DerbyUp – ניחושי כדורגל לארגונים | מגישים - ניר דהן , ליאב צרפתי | מרצה – אוהד אסולין | הגשה – אוגוסט 2026**

## מסמך סקייל בסיסי

### הקדמה

בעולם של היום, באותו האופן שהעלאת מוצר לאוויר מתבצעת בזמן שיא, כך גם מוצרים יכולים להגיע ל-scale גבוה תוך רגע. בעקבות כך נדרש בעינינו מכל מוצר להיערך לאותו הרגע – בו המוצר יכול "להתפוצץ" ולצבור כמות משתמשים גבוהה בזמן קצר. ההיערכות תבוא לידי ביטוי בזיהוי הרגעים בהם צפוי עומס על המערכת, והכנת צעדים "נצורים" למתן מענה לאותן הבעיות.

במוצר הקיים איתו יצאנו לשוק, ביום בו נפתחה ליגת העובדים לצ'ק פוינט, נוספו לנו כמעט 500 משתמשים – שינוי גדול ומהיר בסקייל. הגדילה עבדה בצורה חלקה, כתוצאה מהכנה מיטבית שבאה לידי ביטוי בהגדלת הקיבולות של השירותים החיצוניים, בניית caching, פריסת השרתים במספר נקודות בעולם ועוד.

במסמך זה נפרט את הצעדים שנבצע במידה והסקייל שלנו יגדל.

## מה יקרה עם עשרות עד מאות משתמשים פעילים

### פרופיל העומס הצפוי

| מאפיין | ערך צפוי |
|---|---|
| ארגון ממוצע | 100–500 עובדים |
| שיעור השתתפות | 20–40% |
| משתמשים פעילים בליגה | 30–200 |
| תבנית העומס | פרצי שיא, לא זרימה אחידה |

### הפרץ - התובנה המרכזית

השימוש **אינו מפוזר**. הוא מתרכז בשני חלונות:

| חלון | מה קורה | עומס |
|---|---|---|
| כשעה לפני מחזור | כולם מנחשים | שיא כתיבה |
| מיד אחרי עיבוד (בסיום משחק) | כולם בודקים את הטבלה | שיא קריאה |
| שאר הזמן | פעילות מינימלית | נמוך עד אפסי |

**המשמעות:** ליגה של 200 עובדים יכולה לייצר 200 כתיבות ב-10 דקות ואז "שקט" יחסי. התכנון מתייחס לשיא, לא לממוצע.

## שאילתות למסד הנתונים שעלולות להיות כבדות

### שאילתות "כבדות" בפוטנציאל

| # | שאילתה | למה כבדה | מענה בתכנון |
|---|---|---|---|
| 1 | דירוג הליגה | אגרגציה מחושבת לכל חבר | אינדקס חלקי + pagination + מסלול ל-view materialized |
| 2 | לידרבורד האתר | מיון על כל המשתמשים | זול - עמודת cache + אינדקס ממוין |
| 3 | היסטוריית ניחושים | גדלה ללא הגבלה | אינדקס + pagination |
| 4 | משחקים קרובים | סינון על תחרות וטווח זמן | אינדקס משולב |
| 5 | עיבוד ניחוש | עדכון המוני | עיבוד באצווה |
| 6 | התראות שלא נקראו | סינון על עמודה דלילה | אינדקס חלקי |
| 7 | הטבלה החיה | ניחושים תלויים של כל החברים | חסומה מלמעלה - רק משחקים חיים, רק pending, רק שוק המנצחת |

### טבלת דירוג - השאילתה היקרה ביותר במערכת

זו הנקודה הכבדה ביותר, כתוצאה ישירה מהחלטה ארכיטקטונית: לליגה אין עמודת ניקוד מאוחסנת, והדירוג מחושב בזמן קריאה מסכום הנקודות של כל חבר על משחקי התחרות של הליגה, מרגע הצטרפותו.

```
select lm.user_id, p.display_name,
       coalesce(pred.pts,0) + coalesce(puz.pts,0) as league_points
from league_members lm
join profiles p on p.id = lm.user_id
left join lateral (
  select sum(pr.points_earned) as pts
  from predictions pr
  join questions q on q.id = pr.question_id
  join games    g on g.id = q.game_id
  where pr.user_id = lm.user_id and pr.status = 'correct'
    and g.competition_id = $1 and pr.predicted_at >= lm.joined_at
) pred on true
left join lateral (...) puz on true
where lm.league_id = $2
order by league_points desc;
```

**העלות:** subquery לכל משתתף - **O(members × predictions_per_member)**. לליגה של 200 משתתפים עם 100 ניחושים לכל אחד - כ-20,000 שורות. מהתייעצות עם סוכני הקידוד בהם השתמשנו, ראינו שעלות זו זניחה.

**מה מוריד את העלות:**

| מנגנון | תרומה |
|---|---|
| **idx_pred_user_status** -אינדקס חלקי על **status='correct'** | סורק רק ניחושים נכונים - כשליש מהנפח |
| **idx_games_comp_kickoff** | סינון התחרות בלי סריקת **games** |
| **games** מוגבלת לעונה הנוכחית בלבד | ~2,000 שורות, לא היסטוריה מצטברת |

### לידרבורד האתר - "זול"

בניגוד לדירוג הליגה, הלוח הכללי אינו מחושב:

```
select display_name, avatar_url, total_points
from profiles order by total_points desc limit 20 offset $1;
```

**profiles.total_points** הוא cache שמתעדכן בעיבוד הניחוש, ועם **idx_profiles_leaderboard** (אינדקס ממוין) ה-DB קורא 20 שורות ישירות מהאינדקס -בלי מיון ובלי אגרגציה.

## אינדקסים

**קריטיים -** בלעדיהם כל שאילתה היא סריקה מלאה.

| אינדקס | סוג | השאילתה | הרווח |
|---|---|---|---|
| idx_pred_user_status | חלקי (status='correct') | טבלת דירוג | סורק רק ניחושים נכונים |
| idx_pred_user_time | B-tree משולב | היסטוריה | סריקה ממוקדת |
| idx_pred_question_status | B-tree משולב | עיבוד | איתור מהיר של הממתינים |
| idx_games_comp_kickoff | B-tree משולב | משחקי התחרות | סינון יעיל בדירוג |
| idx_games_status_kickoff | B-tree משולב | משחקים קרובים + שומר הסנכרון החי | סינון יעיל; משרת גם את ה-cron |
| idx_notif_user_unread | חלקי | התראות | אינדקס קטן - רק לא-נקראות |
| idx_players_trgm | trigram GIN | autocomplete | חיפוש %text% יעיל |
| idx_profiles_site_admin | חלקי (is_site_admin) | רשימת מנהלי האתר | קומץ שורות במקום סריקת כל הפרופילים |
| idx_advisor_insights_game | B-tree משולב | ניתוח למשחק לפי hash | פגיעת cache בקריאה אחת |
| idx_advisor_pick_date | B-tree משולב | הבחירות של היום | קריאת הדאשבורד |
| idx_advisor_messages_thread | B-tree משולב | שחזור שיחה | ישן-לחדש בלי מיון |

# הימנעות מטעינת מידע מיותר

## טעינה לפי הצורך

| מה נטען | איך נטען |
|---|---|
| היסטוריית ניחושים | 20 לעמוד |
| דירוג ליגה | 20 לעמוד |
| לידרבורד האתר | 20 לעמוד, מוגבל ל-100 בשרת |
| משחקים | לפי יום בלוח השנה, שאילתה אחת לכל הניחושים בעמוד |
| שאלות | רק של המשחק הנוכחי |
| התראות | 20 אחרונות |
| נתוני שחקנים ליועץ | נשלפים רק כשהשאלה באמת דורשת אותם |

## cache

| שכבה | מנגנון |
|---|---|
| נתוני משחקים | נשמרים ב-DB, לא נמשכים מה-API בכל בקשה |
| עמודים סטטיים | Vercel cache |
| נתוני משתמש | revalidatePath() - רענון ממוקד, לא גורף |
| ניתוחי יועץ | advisor_insights - ניתוח אחד למשחק משרת את כל מי שפותח אותו |
| תשובות הספק ליועץ | api_cache ב-DB, תפוגה 12ש׳ - שורד גם החלפת פונקציית serverless |
| העצה היומית (יועץ AI) | מחושבת בלילה, הדאשבורד ודף הנחיתה קוראים טבלה, לא מודל |

# Pagination

**קיים - בכל רשימה שיכולה לגדול.**

```
Select ... from bets where user_id = $1
order by placed_at desc limit 20 offset $2;
```

| רשימה | גודל עמוד |
|---|---|
| היסטוריית ניחושים | 20 |
| טבלת דירוג | 20 |
| משחקים | 20 |
| התראות | 20 |

### מגבלה ידועה: OFFSET

OFFSET דורש מה-DB לדלג על N שורות פיזית. בעמוד 500 זה מתחיל להיות יקר.

למה זה מקובל כאן: אף רשימה במוצר לא צפויה להגיע לעומקים כאלה - ליגה של 200 חברים היא 10 עמודים. מה נעשה אם כן: מעבר ל-keyset pagination (where placed_at < $cursor), שעלותו קבועה ולא תלויה בעומק.

# מגבלה אמיתית - מכסת API-Football

בפרויקט זה אנו משתמשים בAPI בו אנחנו משתמשים למוצר אותו שיווקנו בפועל ועליו פירטנו בחלקים קודמים במסמך. במוצר המקורי, השתמשנו בחבילה של 75,000 קריאות ליום כאשר בממוצע נעשה שימוש בכ-15,000 קריאות במוצר שמנה מעל 2,000 משתמשים. כרגע אנחנו לא רואים את הוספת פרויקט זה לסך קריאות ה- APIשלו, אך יבוצע מעקב על היקף הקריאות ובמידה ויידרש תבוצע הגדלה של החבילה.

## מגבלה נוספת – LLM API tokens

השימוש במודל שפה יכול להשתנות, המנעד גדול והמחיר הסופי תלוי בכמות ובאיכות. בשביל לקבל תוצאות יותר מהימנות ואיכותיות, נדרש שימוש במודל מסחרי מוכר, אך במידה ורוצים להוזיל עלויות ניתן להשתמש במודל חינמי. מניתוח פעולות בסיסיות ביועץ, קיבלנו את הנתונים הבאים:

| פעולה | טוקנים | עלות |
|---|---|---|
| **ניתוח ראשון למשחק** | כ-2,100 | 0.0013$ |
| **שאלת המשך** | כ-2,700 | 0.0012$ |
| **חסימה במסווג** | כ-540 | 0.00006$ |
| **חסימה בשכבה 1** | 0 | 0$ |

בפרויקט שלנו העדפנו את האמינות ואיכות התשובה על פני המחיר, ולכן בחרנו להשתמש בשלב זה במודל מסחרי מוכר – Gemini 3.5 Flash.

## סנכרון חי של משחקים

סנכרון שרץ כל דקה נשמע "יקר", אך הוא לא – ראשית הוא רץ רק בזמנים שיש משחקים פעילים, ויתרה מכך הוא גם קורא באותה הבקשה לכל המשחקים המשוחקים באותו הזמן, וגם מעדכן את בסיס הנתונים בזמן אמת, כך שאין חשיבות לכמות המשתמשים המחוברים.

# הפרדה בין צד לקוח לצד שרת

| רץ בשרת | רץ בלקוח |
|---|---|
| שליפת נתונים | אינטראקציה בטפסים |
| כל הלוגיקה העסקית | ולידציה ויזואלית |
| כל הולידציות המחייבות | מצב UI מקומי |
| כל בדיקות ההרשאות | החלפת ערכת נושא |
| מפתחות API | תזמון הרענון בזמן משחק חי |

# מגבלות המוצר הנוכחי

| # | מגבלה | ההשלכה | ביטוי בפרויקט |
|---|---|---|---|
| 1 | העיבוד רץ סדרתית - מספר משחקים מסתיימים במקביל | ה-cron מתארך | ניחושים רבים ממתינים |
| 2 | תלות בספק חיצוני יחיד | אין נתונים אם הוא נופל | בעת תקלה ב-API הנתונים לא יעודכנו |
| 3 | Supabase free tier | MB500, מגבלת חיבורים | משתמשים רבים במקביל ימצו את המכסה |
| 4 | Vercel free tier | timeout של 10 שניות ל-function | עיבוד ניחוש המוני |
| 5 | יחסים זמניים למשחקים רחוקים | היחס המוצג עשוי להשתנות עד הפתיחה | הובהר בממשק; הניקוד לפי נקודת הניחוש |
| 6 | pg_cron מדייק עד דקה | שער מופיע תוך כ-60ש' ולא כ-15ש' | מקובל - זו תצוגה, לא ניקוד |
| 7 | עלות יקרה של מודל שפה | מספר קריאות מוגבל ליום | מכסה יומית למשתמש + cache משותף |

# שיפורים לגרסה עתידית

## טווח קצר - עד כ-10,000 משתמשים

| שיפור | מה פותר |
|---|---|
| Supabase pro + connection pooling | מגבלה 3 |
| Redis לקאשינג של טבלאות דירוג | עומס קריאות לשרתים |
| Realtime Supabase או SSE לטבלה חיה | מבטל את ה-polling של 30 השניות |
| תור עבודה לעיבוד במקום cron | מגבלה 1 |

| שיפור | מה פותר |
|---|---|
| מקור נתונים גיבוי | מגבלה 2 |
| הגדלת השימוש במודל שפה ומעבר למודל טוב יותר (3.7) | שיפור איכות התשובות |
| ניקוי api_cache מתוזמן | טבלה שגדלה בלי מחיקת רשומות שפג תוקפן |
| CDN לתמונות | עומס קריאות לשרתים |

## טווח ארוך - מעל 10,000 משתמשים

| שיפור | מה פותר |
|---|---|
| פיצול (partitioning) של **predictions** לפי תאריך | טבלה שגדלה ללא הגבלה |
| Read replicas | הפרדת קריאה מכתיבה |
| מחיקת עונות ישנות | שמירה על טבלאות חמות קטנות |