פרויקט סיום – טכנולוגיות אינטרנט ופיתוח פול סטאק

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

תכנון טכני מפורט

הקדמה

כחלק מתהליך בניית הפרויקט, ביצענו תכנון טכני מפורט אשר אמור להקל על בניית האתר. כחלק מההכנה לבניית האתר תכננו את מבנה הפרויקט, הקומפוננטות, ה-API, בסיס הנתונים ועוד.

מבנה התיקיות

```
derbyup/
├── app/
│   ├── (auth)/
│   │   ├── login/page.tsx
│   │   └── signup/page.tsx
│   ├── (app)/
│   │   ├── layout.tsx
│   │   ├── dashboard/page.tsx
│   │   ├── games/
│   │   │   ├── page.tsx
│   │   │   └── [id]/page.tsx
│   │   ├── leagues/
│   │   │   ├── page.tsx
│   │   │   ├── new/page.tsx
│   │   │   └── [id]/
│   │   │       ├── page.tsx
│   │   │       └── admin/page.tsx
│   │   ├── advisor/page.tsx
│   │   ├── join/page.tsx  ·  join/[code]/page.tsx
│   │   ├── leaderboard/page.tsx
│   │   ├── predictions/page.tsx
│   │   ├── challenge/page.tsx
│   │   ├── notifications/page.tsx
│   │   ├── profile/page.tsx
│   │   └── loading.tsx  ·  error.tsx
│   ├── (admin)/
│   │   ├── layout.tsx
│   │   └── admin/
│   │       ├── page.tsx
│   │       ├── users/page.tsx  ·  users/[id]/page.tsx
│   │       ├── games/page.tsx
│   │       └── leagues/page.tsx
│   ├── api/cron/
│   │   ├── sync-fixtures/route.ts
│   │   ├── settle/route.ts
│   │   ├── sync-live/route.ts
│   │   └── advisor-pick/route.ts
│   ├── layout.tsx
│   ├── page.tsx
│   └── globals.css
│
├── components/
│   ├── ui/
│   ├── layout/
│   ├── games/
│   ├── leagues/
│   ├── advisor/
│   ├── landing/
│   ├── admin/
│   ├── site-admin/
│   ├── auth/
```

```
│   ├── challenge/
│   └── shared/
│
├── lib/
│   ├── domain/
│   │   ├── scoring.ts
│   │   ├── prediction-rules.ts
│   │   ├── settlement.ts
│   │   ├── exact-score.ts
│   │   ├── live-projection.ts
│   │   ├── standings.ts
│   │   ├── achievements.ts
│   │   ├── puzzle.ts
│   │   └── types.ts
│   ├── advisor/
│   │   ├── provider.ts
│   │   ├── prompts.ts
│   │   ├── guard.ts
│   │   ├── classifier.ts
│   │   ├── schema.ts
│   │   ├── context.ts
│   │   ├── football.ts
│   │   ├── daily-pick.ts
│   │   └── pricing.ts  ·  db.ts  ·  types.ts
│   ├── actions/
│   │   ├── auth.ts · leagues.ts · predictions.ts · challenge.ts
│   │   ├── admin.ts · site-admin.ts · advisor.ts
│   │   └── notifications.ts · profile.ts · types.ts
│   ├── cron/
│   │   ├── sync-fixtures.ts · settle.ts · sync-live.ts
│   │   └── advisor-pick.ts · auth.ts
│   ├── live/league.ts
│   ├── landing/fixtures.ts
│   ├── supabase/
│   │   └── client.ts · server.ts · admin.ts
│   ├── football-api/
│   │   └── client.ts · mapping.ts · types.ts
│   ├── validation/
│   ├── i18n/
│   ├── achievements/award.ts
│   ├── challenge/club-crests.ts
│   ├── theme-colors.ts
│   └── format.ts  ·  utils.ts
│
├── middleware.ts
├── supabase/migrations/
├── scripts/
├── tests/
│   ├── unit/
│   ├── components/
│   ├── integration/
│   └── e2e/
├── types/database.ts
└── docs/
```

מבנה הקומפוננטות המרכזיות

קומפוננטה	סוג	תפקיד	Props עיקריים
GameRow	Server	שורת משחק; בוריאנט חי - תוצאה ודקה	homeTeam, awayTeam, kickoffAt, isFeatured, live
LiveScore	Server	תוצאה, דקה ונקודה פועמת	scoreHome, scoreAway, minute
LiveRefresher	Client	רענון העמוד כל 30ש׳ בזמן משחק	intervalSeconds
QuestionCard	Client	שאלה, אפשרויות, ניחוש וביטול	outcomes, bonusPct, existing, locked, provisional, homeTeam, awayTeam
ScoreDrumPicker	Client	בורר התוצאה המדויקת	homeTeam, awayTeam, value, isDraw
Fixture	Server	הצגת מפגש אחידה - הבית תמיד באותו צד	home, away, homeLogo, awayLogo
LeaderboardTable	Server	טבלת דירוג - משרתת את שני הלוחות	rows, currentUserId, showCorrectCount, liveDeltas
PrizeList	Server	פרסי הליגה לפי מקום	prizes, note
InviteCodeBox	Client	הצגה והעתקה של הקוד	code
LeagueGames	Server	משחקי הליגה הזמינים לניחוש	games, predictedByGame
PuzzleBoard	Client	לוח האתגר, כולל האוטוקומפליט	clubA, clubB, initialAttempts
ResultModal	Client	תוצאת האתגר	isCorrect, pointsEarned, answers
AdvisorPanel	Client	היועץ - ניתוח, מכסה וצ׳אט	gameId
InsightCard	Server	כרטיס דעת היועץ	insight
SettleGameDialog	Client	רישום תוצאה ידני	gameId, homeTeam, awayTeam
UserActions	Client	מינוי מנהל ומחיקת משתמש	userId, displayName, isAdmin, isSelf
Pagination	Server	ניווט עמודים	page, hasNext, baseUrl
EmptyState	Server	מצב ריק עם קריאה לפעולה	icon, title, body, action

מבנה בסיס הנתונים

סכימה מלאה

```
-- ═══ 1. Profiles ═══
create table profiles (
  id                uuid primary key references auth.users(id) on delete cascade,
  username          varchar(30) unique not null,
  display_name      varchar(60),
  avatar_url        text,
  total_points      numeric(10,2) not null default 0,
  total_predictions integer not null default 0,
  total_correct     integer not null default 0,
  is_site_admin     boolean not null default false,
  created_at        timestamptz not null default now(),
  updated_at        timestamptz not null default now()
);

-- ═══ 2. Competitions ═══  (id = API identifier)
create table competitions (
  id        integer primary key,
  name      varchar(80) not null,
  country   varchar(60) not null,
  logo_url  text,
  season    integer not null,
  is_active boolean not null default true
);

-- ═══ 3. Games ═══
create table games (
  id              uuid primary key default gen_random_uuid(),
  fixture_id      integer unique not null,
  competition_id  integer not null references competitions(id),
  home_team       varchar(80) not null,
```

```
  away_team       varchar(80) not null,
  home_logo       text,
  away_logo       text,
  kickoff_at      timestamptz not null,
  status          varchar(20) not null default 'scheduled'
                  check (status in ('scheduled','live','finished','postponed','cancelled')),
  score_home      smallint,
  score_away      smallint,
  minute          smallint
                  check (minute is null or (minute >= 0 and minute <= 130)),
  live_updated_at timestamptz,
  settled_at      timestamptz,
  updated_at      timestamptz not null default now()
);

-- ═══ 4. Prediction Questions ═══
create table questions (
  id               uuid primary key default gen_random_uuid(),
  game_id          uuid not null references games(id) on delete cascade,
  type             varchar(20) not null
                   check (type in ('match_result','over_under_2_5','btts')),
  outcomes         jsonb not null,   -- [{key,label,odds}]
  odds_provisional boolean not null default true,
  correct_outcome  varchar(20),
  resolved_at      timestamptz,
  created_at       timestamptz not null default now(),
  unique (game_id, type)
);

-- ═══ 5. Predictions ═══
create table predictions (
  id               uuid primary key default gen_random_uuid(),
  user_id          uuid not null references profiles(id) on delete cascade,
  question_id      uuid not null references questions(id) on delete cascade,
  selected_outcome varchar(20) not null,
  odds             numeric(6,2) not null check (odds >= 1),
  odds_provisional boolean not null default false,
  exact_score      varchar(3)
                   check (exact_score is null or exact_score ~ '^[0-9]-[0-9]$'),
  bonus_pct        smallint not null default 0
                   check (bonus_pct between 0 and 100),
  points_earned    numeric(10,2),
  status           varchar(16) not null default 'pending'
                   check (status in ('pending','correct','incorrect','void','cancelled')),
  predicted_at     timestamptz not null default now(),
  settled_at       timestamptz,
  cancelled_at     timestamptz,
  unique (user_id, question_id)
);

-- ═══ 6. Leagues ═══
create table leagues (
  id                 uuid primary key default gen_random_uuid(),
  name               varchar(60) not null,
  description        text,
  creator_id         uuid references profiles(id) on delete cascade,  -- NEW: nullable
  competition_id     integer not null references competitions(id),
  invite_code        varchar(8) unique not null,
  is_public          boolean not null default false,
  prizes             jsonb,        -- [{place:1, prize:'Concert ticket'}, ...]
  prize_note         text,
  featured_game_id   uuid references games(id) on delete set null,
  featured_bonus_pct smallint not null default 0
                     check (featured_bonus_pct between 0 and 100),
  status             varchar(16) not null default 'active'
                     check (status in ('active','archived')),
  created_at         timestamptz not null default now(),
  constraint leagues_creator_matches_visibility
```

```
    check ((is_public and creator_id is null)
        or (not is_public and creator_id is not null))
);

-- ═══ 7. League Members ═══
create table league_members (
  id        uuid primary key default gen_random_uuid(),
  league_id uuid not null references leagues(id) on delete cascade,
  user_id   uuid not null references profiles(id) on delete cascade,
  joined_at timestamptz not null default now(),
  unique (league_id, user_id)
);

-- ═══ 8. Notifications ═══
create table notifications (
  id         uuid primary key default gen_random_uuid(),
  user_id    uuid not null references profiles(id) on delete cascade,
  type       varchar(24) not null
             check (type in ('prediction_settled','league_joined','achievement')),
  title      varchar(120) not null,
  body       text,
  link_url   text,
  read_at    timestamptz,
  created_at timestamptz not null default now()
);

-- ═══ 9. Achievements ═══
create table user_achievements (
  id              uuid primary key default gen_random_uuid(),
  user_id         uuid not null references profiles(id) on delete cascade,
  achievement_key varchar(40) not null,
  earned_at       timestamptz not null default now(),
  unique (user_id, achievement_key)
);

-- ═══ 10. Daily Challenge ═══
create table daily_puzzles (
  id            uuid primary key default gen_random_uuid(),
  play_date     date unique not null,
  club_a        varchar(80) not null,
  club_b        varchar(80) not null,
  valid_answers jsonb not null,
  created_at    timestamptz not null default now()
);

-- ═══ 11. Challenge Attempts ═══
create table puzzle_attempts (
  id             uuid primary key default gen_random_uuid(),
  user_id        uuid not null references profiles(id) on delete cascade,
  puzzle_id      uuid not null references daily_puzzles(id) on delete cascade,
  answer         varchar(80) not null,
  is_correct     boolean not null,
  attempt_number smallint not null check (attempt_number between 1 and 3),
  points_earned  numeric(10,2) not null default 0,
  created_at     timestamptz not null default now(),
  unique (user_id, puzzle_id, attempt_number)
);

-- ═══ 12. Players Dataset - For Challenge ═══
create table bridge_players (
  id              uuid primary key default gen_random_uuid(),
  name            varchar(80) not null,
  normalized_name varchar(80) not null unique
);

-- ═══ 13. Advisor Insights ═══  NEW
-- Shared cache of match analyses. Keyed by the match AND a hash of the inputs,
```

```
-- so a changed price produces a new row instead of serving stale advice.
create table advisor_insights (
  id           uuid primary key default gen_random_uuid(),
  game_id      uuid not null references games(id) on delete cascade,
  context_hash varchar(64) not null,
  payload      jsonb not null,
  model        varchar(60) not null,
  created_at   timestamptz not null default now(),
  unique (game_id, context_hash)
);

-- ═══ 14. Advisor Daily Pick ═══  NEW
-- One analysed match per competition per day, chosen by cron.
create table advisor_daily_pick (
  id             uuid primary key default gen_random_uuid(),
  pick_date      date not null,
  competition_id integer not null references competitions(id) on delete cascade,
  game_id        uuid not null references games(id) on delete cascade,
  payload        jsonb not null,
  created_at     timestamptz not null default now(),
  unique (pick_date, competition_id)
);

-- ═══ 15. Advisor Conversations ═══  NEW
-- One thread per person per match.
create table advisor_conversations (
  id         uuid primary key default gen_random_uuid(),
  user_id    uuid not null references profiles(id) on delete cascade,
  game_id    uuid not null references games(id) on delete cascade,
  created_at timestamptz not null default now(),
  updated_at timestamptz not null default now(),
  unique (user_id, game_id)
);

-- ═══ 16. Advisor Messages ═══  NEW
create table advisor_messages (
  id              uuid primary key default gen_random_uuid(),
  conversation_id uuid not null references advisor_conversations(id) on delete cascade,
  role            varchar(10) not null check (role in ('user','assistant')),
  content         text not null,
  blocked         boolean not null default false,  -- answered by a guard, not the model
  created_at      timestamptz not null default now()
);

-- ═══ 17. Advisor Usage ═══  NEW
-- Daily quota. The composite key is what makes the consume upsert atomic.
create table advisor_usage (
  user_id        uuid not null references profiles(id) on delete cascade,
  usage_date     date not null,
  question_count smallint not null default 0,
  primary key (user_id, usage_date)
);

-- ═══ 18. API Cache ═══  NEW
-- API-Football responses, TTL 12h. No RLS policy at all: invisible to users,
-- touched only by the service role and the dedicated functions.
create table api_cache (
  cache_key  varchar(200) primary key,
  payload    jsonb not null,
  fetched_at timestamptz not null default now()
);
```

פעולות CRUD מרכזיות

ישות	CREATE	READ	UPDATE	DELETE
פרופיל	טריגר בהרשמה	/profile, /dashboard	updateProfile ; מינוי מנהל אתר דרך admin_set_site_admin	admin_delete_user - מ-auth.users, הכול ב-cascade
ליגה	createLeague	/leagues, /leagues/[id]	setFeaturedGame, updatePrizes	ארכוב, לא מחיקה
חברות	joinLeague	טבלת דירוג	אין מה לעדכן	עזיבת ליגה
ניחוש	makePrediction (כולל exactScore)	/predictions, /games/[id]	עיבוד (status, points_earned)	ביטול עד 10 דק' לפני פתיחה
משחק	sync cron	/games	סנכרון יומי · סנכרון חי (תוצאה ודקה) · עיבוד · admin_settle_game	-
שאלה	sync cron	/games/[id]	עיבוד (correct_outcome)	cascade עם המשחק
התראה	בעיבוד / הצטרפות	/notifications	markNotificationRead	ניקוי ישנות
ניסיון באתגר	submitPuzzleAnswer	/challenge	-	-
ניתוח יועץ	advisor_publish_insight - הפותח הראשון	כל מי שפותח את המשחק	אין - מחיר שזז יוצר שורה חדשה	cascade עם המשחק
שיחת יועץ	השאלה הראשונה למשחק	הפאנל - לבעלים בלבד	אין policy - תמליל אינו נערך	cascade ממחיקת חשבון

תיאור ה-API

Server Actions

```
// lib/actions/leagues.ts
createLeague(input: { name, description?, competitionId })
  → { ok: true, leagueId, inviteCode } | { ok: false, error }

joinLeague(input: { inviteCode })
  → { ok: true, leagueId } | { ok: false, error }

updatePrizes(input: { leagueId, prizes, note? })

// lib/actions/predictions.ts
makePrediction(input: { questionId, outcome, exactScore? })
  → { ok: true, predictionId, potentialPoints } | { ok: false, error, code }

cancelPrediction(input: { predictionId })
  → { ok: true } | { ok: false, error, code }

// lib/actions/admin.ts
setFeaturedGame(input: { leagueId, gameId, bonusPct })
settleGameManually(input: { gameId, scoreHome, scoreAway })

// lib/actions/challenge.ts
submitPuzzleAnswer(input: { puzzleId, answer })
  → { ok: true, isCorrect, pointsEarned, attemptsLeft } | { ok: false, error }

// lib/actions/site-admin.ts
settleGameSiteWide(input: { gameId, scoreHome, scoreAway })
setSiteAdmin(input: { userId, value })
deleteUser(input: { userId })

// lib/actions/advisor.ts
getGameInsight(gameId)
  → { insight, cached, remaining } | { error }
askAdvisor(input: { gameId, question })
  → { answer, refused, remaining } | { error }
getAdvisorThread(gameId)    → ChatTurn[]
advisorQuotaRemaining()     → number
```

Route Handlers (cron)

Endpoint	תזמון	מתזמן	פעולה
POST api/cron/sync-fixtures/	0 4 * * *	Cron Vercel	משחקי העונה + יחסים ל-14 יום + בניית שאלות
POST api/cron/settle/	*/10 * * * *	pg_net + pg_cron	עיבוד משחקים שהסתיימו
POST api/cron/sync-live/	* * * * *	pg_net + pg_cron	תוצאה ודקה למשחקים חיים - רק כשיש משחק בעיצומו
POST api/cron/advisor-pick/	0 5 * * *	Cron Vercel	ניתוח יומי אחד לכל תחרות

שימוש ב-API-Football

Endpoint	פרמטרים	שימוש
GET fixtures/	league, season, from, to	משחקים ותוצאות
GET fixtures/	live	תוצאות חיות, קריאה אחת לכל התחרויות
GET odds/	league, season, date	יחסים
GET standings/	league, season	דירוג הליגה - ליועץ
GET injuries/	fixture	פציעות - ליועץ
GET fixtures/headtohead/	h2h	מפגשים קודמים - ליועץ
GET fixtures/lineups/ · **players/topscorers/**	fixture / league	הרכבים ומבקיעים - רק כשהשאלה דורשת

הלוגיקה העסקית המרכזית

כל הפונקציות בסעיף זה ב-**lib/domain/ - טהורות, בלי גישה ל-DB או לרשת.**

חישוב נקודות

```
// lib/domain/scoring.ts
export function pointsForCorrectPrediction(odds: number, bonusPct = 0): number {
  return round2(odds * (1 + bonusPct / 100));
}
```

דוגמה	חישוב	תוצאה
יחס 7.15, בלי בונוס	**7.15 × 1.00**	**7.15**
יחס 2.10, בונוס תוצאה מדויקת	**2.10 × 3.00**	**6.30**
ניחוש שגוי	-	**0**

היחס **מוקפא בזמן הניחוש (predictions.odds)** - שינוי מאוחר יותר אינו משפיע.

תוצאה מדויקת

```
// lib/domain/exact-score.ts
exactScoreMultiplier(isCorrect, exactScore, scoreHome, scoreAway)
  → isCorrect && exactScore === `${scoreHome}-${scoreAway}` ? 3 : 1
```

isCorrect הוא שער: הבונוס "רוכב" על ניחוש מנצחת שצדק ולעולם אינו "מציל" ניחוש שטעה.

כללי ניחוש

```
// lib/domain/prediction-rules.ts
export function validatePrediction(ctx: {
  game: { kickoffAt: Date; status: string; competitionId: number };
  hasExisting: boolean;
  userCompetitions: number[];
  now: Date;
}): { ok: true } | { ok: false; reason: PredictionRejection }
```

#	כלל	קוד דחייה
1	המשחק טרם התחיל	**GAME_STARTED**
2	סטטוס **scheduled** בלבד	**GAME_NOT_OPEN**
3	אין ניחוש קיים לשאלה	**ALREADY_PREDICTED**
4	המשתמש חבר בליגה של התחרות	**NO_LEAGUE_FOR_COMPETITION**

כללי ביטול

```
export const CANCEL_WINDOW_MINUTES = 10;

export function validateCancellation(ctx: {
  prediction: { userId: string; status: string };
  game: { kickoffAt: Date };
  requesterId: string;
  now: Date;
}): { ok: true } | { ok: false; reason: CancelRejection }
```

#	כלל	קוד דחייה
1	המבקש הוא בעל הניחוש	**NOT_OWNER**
2	הסטטוס **pending**	**ALREADY_SETTLED**
3	נותרו **יותר** מ-10 דקות עד הפתיחה	**CANCEL_WINDOW_CLOSED**

קביעת התשובה הנכונה

```
// lib/domain/settlement.ts
export function resolveOutcome(
  type: QuestionType, scoreHome: number, scoreAway: number
): string {
  switch (type) {
    case 'match_result':
      return scoreHome > scoreAway ? 'home' : scoreHome < scoreAway ? 'away' : 'draw';
    case 'over_under_2_5':
      return scoreHome + scoreAway > 2.5 ? 'over' : 'under';
    case 'btts':
      return scoreHome > 0 && scoreAway > 0 ? 'yes' : 'no';
  }
}
```

עיבוד ניחוש

```
export function settlePrediction(
  pred: { selectedOutcome, odds, bonusPct?, exactScore? },
  correctOutcome: string,
  finalScore?: { home, away },
): { status: 'correct' | 'incorrect'; pointsEarned: number } {
  const isCorrect = pred.selectedOutcome === correctOutcome;
  if (!isCorrect) return { status: 'incorrect', pointsEarned: 0 };

  const base       = pointsForCorrectPrediction(effectiveOdds(pred), pred.bonusPct);
  const multiplier = exactScoreMultiplier(true, pred.exactScore,
                                          finalScore?.home, finalScore?.away);
  return { status: 'correct', pointsEarned: round2(base * multiplier) };
}
```

משחק שבוטל

משחק ב-**cancelled** או **postponed** → כל הניחושים ל-**void** ו-**points_earned = 0**.
המשתמש אינו נפגע ואינו מרוויח מאירוע שאינו בשליטתו.

דירוג

```
// lib/domain/standings.ts
export function rankRows(rows: ScoreRow[]): RankedRow[]
```

	מה נכנס
דירוג ליגה	ניחושי **match_result** בטורניר הליגה, מרגע ההצטרפות
לידרבורד האתר	**profiles.total_points** - הכל

הישגים

```
// lib/domain/achievements.ts
export const ACHIEVEMENTS = [
  { key: 'first_prediction', title: 'ניחוש ראשון',   check: s => s.totalPredictions >= 1 },
  { key: 'ten_predictions',  title: '10 ניחושים',     check: s => s.totalPredictions >= 10
},
  { key: 'first_correct',    title: 'פגיעה ראשונה',   check: s => s.totalCorrect >= 1 },
  { key: 'streak_three',     title: 'רצף 3 נכונים',   check: s => s.currentStreak >= 3 },
  { key: 'underdog',         title: 'ניחוש הפתעה',    check: s => s.bestOdds >= 5 },
  { key: 'first_puzzle',     title: 'אתגר ראשון',     check: s => s.puzzlesSolved >= 1 },
  { key: 'league_joined',    title: 'הצטרפת לליגה',   check: s => s.leaguesJoined >= 1 },
  { key: 'league_leader',    title: 'מקום ראשון',     check: s => s.bestRank === 1 },
] as const;
```

כולם נגזרים מנתונים קיימים - **אין מנגנון מעקב נפרד**. נבדקים בסוף ה-cron של עיבוד הניחוש.

אתגר יומי

```
// lib/domain/puzzle.ts
export const PUZZLE_POINTS = [5, 3, 1] as const;

export function normalizeName(raw: string): string
export function checkAnswer(answer: string, validAnswers: string[]): boolean
```

תצוגה חיה

```
// lib/domain/live-projection.ts
export function projectPrediction(prediction, questionType, score)
  → { points: number; winningNow: boolean }

export function sumLiveByUser(rows: LiveRow[]): Map<string, number>
```

שכבות הגנה ליועץ ה-AI

```
// lib/advisor/guard.ts - Layer 1: pure, deterministic, free
export function guardQuestion(raw: unknown):
  { ok: true; question } | { ok: false; code; message }

// lib/advisor/classifier.ts - Layer 1b: a cheap model gating the expensive one
export function classifyQuestion(question, teams, model)
  → { category, allowed, needs }
```

שתי שכבות "הגנה" ליועץ ה-AI: בדיקת תקינות קלט ו-prompt injections, ושליחת השאילתה ליועץ עם תוכן מינימלי.

ניהול State

סוג State	היכן	מנגנון
נתוני שרת	בשרת	revalidatePath() + Components Server
Session	cookies	middleware.ts + supabase/ssr@
טפסים	לקוח	useFormStatus + useActionState
UI מקומי	לקוח	useState
ערכת נושא	localStorage	next-themes
ערכת צבע	localStorage	class על <html>, נכתב לפני הציור הראשון
מסננים ועמודים	URL	searchParams

החלטה: בלי ספריית state גלובלי

- **הנתונים נטענים בשרת** - אין צורך בcache בלקוח
- **revalidatePath() הוא מנגנון הרענון** - אחרי כל Action
- **State ב-URL** (מסננים, עמוד) - ניתן לשיתוף וחוזר אחרי refresh

הפחתת מורכבות מכוונת, לא חיסכון בעבודה.

טיפול בשגיאות

מיפוי קודי דחייה להודעות

קוד	הודעה למשתמש
GAME_STARTED	המשחק כבר התחיל, לא ניתן לנחש
GAME_NOT_OPEN	המשחק אינו פתוח לניחושים
ALREADY_PREDICTED	כבר ניחשת בשאלה הזו
NO_LEAGUE_FOR_COMPETITION	אינך חבר בליגה של התחרות הזו
INVALID_OUTCOME	התשובה שנבחרה אינה חוקית
INVALID_FORMAT · **NOT_A_DRAW** · **HOME/AWAY_MUST_LEAD**	הודעות בורר התוצאה: פורמט 0-0, התאמה לבחירת המנצחת
CANCEL_WINDOW_CLOSED	לא ניתן לבטל - נותרו פחות מ-10 דקות לפתיחה
ALREADY_SETTLED	הניחוש כבר עבר עיבוד
NOT_OWNER	הניחוש אינו שלך
INVALID_CODE	קוד הזמנה לא תקין
ALREADY_MEMBER	כבר הצטרפת לליגה הזו
NOT_A_MEMBER	אינך חבר בליגה הזו
NOT_SITE_ADMIN	הפעולה מותרת למנהל אתר בלבד
GAME_NOT_STARTED	לא ניתן לרשום תוצאה למשחק שטרם התחיל

כשל בספק החיצוני

אם API-Football לא זמין: ה-cron נכשל בשקט ומתועד, ולא מוחק נתונים קיימים. מנהל האתר יכול לרשום תוצאה ידנית - רשת הביטחון. אם Gemini לא זמין: היועץ מחזיר הודעת "לא זמין כרגע", והמכסה שנצרכה היא ההפסד היחיד.

ולידציה של קלטים

סכימות Zod

```
// lib/validation/schemas.ts
export const signUpSchema = z.object({
  email:       z.string().email('כתובת אימייל לא תקינה'),
  password:    z.string().min(8, 'סיסמה באורך 8 תווים לפחות'),
  displayName: z.string().min(2).max(60),
});
```

```
export const createLeagueSchema = z.object({
  name:          z.string().min(3).max(60),
  description:   z.string().max(500).optional(),
  competitionId: z.number().int().positive(),
});

export const makePredictionSchema = z.object({
  questionId: z.string().uuid(),
  outcome:    z.string().min(1).max(20),
  exactScore: z.string().regex(/^[0-9]-[0-9]$/).optional().nullable(),
});

export const joinLeagueSchema = z.object({
  inviteCode: z.string().length(8).regex(/^[A-Z0-9]+$/),
});

export const prizesSchema = z.object({
  leagueId: z.string().uuid(),
  prizes:   z.array(z.object({
    place: z.number().int().min(1).max(20),
    prize: z.string().min(1).max(120),
  })).max(20),
  note:     z.string().max(500).optional(),
});

export const adminSettleSchema = z.object({
  gameId:    z.string().uuid(),
  scoreHome: z.number().int().min(0).max(99),
  scoreAway: z.number().int().min(0).max(99),
});

export const askAdvisorSchema = z.object({
  gameId:   z.string().uuid(),
  question: z.string().min(1).max(300),
});
```

תכנון חוויית המשתמש

המסע הקריטי - ביצוע ניחוש

```
/dashboard → משחקי התחרות של הליגה שלי
     │ לחיצה על משחק
     ▼
/games/[id]
     │  3 שאלות כאשר לכל שאלה מוצג הניקוד הפוטנציאלי
     │  בונוס "בחירת עורך" יסומן, ותוצאה מדויקת תעניק מכפיל פי 3 לניקוד
     ▼
בחירת אפשרות
     │
     ▼
הצלחה → toast + "בטל כפתור" + "ניחשת למצב השאלה עוברת"
```

החלטת UX: היחס מוצג כנקודות

מוצג **"7.15 נקודות"** ולא "7.15×" ולא "+615". היחס **הוא** הניקוד, ולכן אין סיבה להציג אותו כמכפיל - זה רק היה מבלבל.

מצב כהה וערכות צבע

טוקנים של Tailwind, next-themes בלי צבעים קשיחים בקוד - כל צבע דרך טוקן. בזכות זה ערכת צבעים שלמה היא הגדרה מחדש של אותם טוקנים: 9 ערכות צבע, כל אחת בבהיר וכהה, והבחירה נשמרת ב-localStorage ונכתבת כ-class על <html> לפני הציור הראשון.