Team A - Full Development Brief

App Summary

App Summary:

The Combine Stats Tracker is a tool designed for youth sports combine events, where organizers can register approximately 250 players, collect real-time performance data across five drills, and automatically group players by age for fair ranking. Staff will take player photos at check-in, record scores for each fixed drill, and use a customizable algorithm to rank players within their age group. The app must support offline data entry with sync, and export detailed reports for coaches to use during draft selection. Leaderboards and all data views are staff-only. This version is for a single-event use case, not long-term tracking.

Overview

This document provides a comprehensive technical roadmap for building the Combine Stats Tracker App. It includes functional requirements, expected user flows, backend and frontend structure, and suggested stack.

Core Features

- Player profile creation: Name, number, age, photo via device camera.
- Five fixed drills: 40m dash, vertical jump, catching, throwing, agility.
- Normalized scoring per drill (0-100 scale).
- Customizable drill weightings per event.
- Automatic age grouping and rankings within group.
- Staff-only access to leaderboards and rankings.
- Offline mode with sync when reconnected.
- Coach draft reports (PDF/CSV export).

User Roles

- Admin: Full control over event, inputs, weights, exports.

- Judge: Score entry and player view only.

- Viewer (optional): Read-only mode.

Data Models

Entities:

- Player: id, name, number, age, photo_url

- DrillResult: id, player_id, drill_type, raw_score, normalized_score

- AgeGroup: id, name, min_age, max_age

- Event: id, title, drill_weights, created_at

- Ranking: player_id, event_id, age_group_id, total_score, rank

Backend Requirements

- REST or GraphQL API

- Endpoints for player CRUD, drill result input, leaderboard fetch, weight config, report generation

- Auth middleware for role-based access

- Offline-ready architecture using local queueing

- PDF/CSV generation using server-side tools

Frontend Requirements

- Mobile-first, tablet-friendly responsive UI

- Player check-in with camera access

- Drill input screens with form validation

- Admin panel for weight setting, event start/reset

- Rankings view with filters (age group, top X, by drill)

- Export panel with PDF/CSV options

Tech Stack Suggestions

- Frontend: React or React Native

- Backend: Node.js (Express) or Python (FastAPI)
- Database: PostgreSQL or Firebase
- Media: Cloudinary or Firebase Storage
- Hosting: Vercel/Render for frontend, Railway/Render for backend
- State Management: Redux, Zustand, or built-in React Context
- Sync handling: IndexedDB, Service Workers

Security & Syncing

- JWT-based auth for role access
- Local storage of unsynced changes
- Conflict resolution if sync fails
- Notify users when offline and when data syncs successfully