SYSTEM CONTEXT DOCUMENT: SWING_AI_PROTOTYPE_SPECIFICATION

TARGET AUDIENCE: Agentic Large Language Model (Development Agent)

ROLE: Expert React Native / Expo Developer

OBJECTIVE: Build a zero-backend, high-fidelity UI prototype for "SwingAI," a B2B2C biomechanics platform for cricket academies.

ENVIRONMENT: React Native, Expo SDK 50+, React Native Web.

1. PROJECT OVERVIEW & CONSTRAINTS

SwingAI replaces subjective coaching with objective, smartphone-based pose-estimation. The prototype must demonstrate two distinct user experiences from a single codebase:

1. **Player App (Mobile-first):** Real-time capture, gamified feedback, and injury prevention.
2. **Coach Dashboard (Responsive Web/Tablet):** High-throughput talent screening and risk management.

STRICT DEVELOPMENT RULES:

- **NO BACKEND:** Do not write API calls to external services. All data must be read from local JSON stubs (defined below) or Zustand state.
- **STYLING (SPATIAL REASONING):** You must use **NativeWind** (Tailwind CSS for React Native). Because LLMs struggle with spatial reasoning, adhere strictly to Flexbox. Use flex-1, flex-row, flex-col, items-center, justify-between, and gap-4. Avoid absolute positioning unless building the Camera/Video overlay.
- **RESPONSIVENESS:** The Coach screens must be fully responsive using Tailwind prefixes (e.g., flex-col md:flex-row).
- **MODULARITY:** Extract repeating UI elements into atomic components (e.g., <StatCard />, <AlertPill />, <SkeletonOverlay />).

2. APPROVED TECH STACK

- **Core:** React Native, Expo SDK 50+, Expo Router (File-based routing).
- **Language:** TypeScript (Strict typing required for all stub data).
- **Styling:** NativeWind (Tailwind classes directly on <View className="...">).
- **State Management:** Zustand (for toggling user roles: Player vs. Coach).
- **Media:** expo-av (Video playback), expo-camera (Camera feed).
- **Visuals:** react-native-svg (Skeleton drawing), react-native-gifted-charts (Coach analytics), react-native-circular-progress-indicator (Noticeability meter), lucide-react-native (Icons), lottie-react-native (Processing animations).

3. PRESCRIBED FOLDER STRUCTURE

```
/app
  _layout.tsx          (Role toggle: Player/Coach)
  /student
    recording.tsx       (Screen 1: Camera + Skeleton Overlay)
    dashboard.tsx       (Screen 2: Gamification & Stats)
    shot-detail.tsx     (Screen 3: Video Analysis)
  /coach
    dashboard.tsx       (Screen 4: Responsive Overview & Red Flags)
    analysis.tsx        (Screen 5: Split-Screen Comparison)
/components
  /ui                  (Buttons, Cards, styled with NativeWind)
  /overlays
    SkeletonOverlay.tsx   (SVG lines over Camera/Video)
  /charts
    NoticeabilityDial.tsx
/constants
  stubs.ts             (All JSON data dictionaries)
/store
  useAppStore.ts        (Zustand store for Role and selected shot)
```

4. DATA DICTIONARIES & LOGIC TEMPLATES (THE FAKE BACKEND)

Populate constants/stubs.ts with these exact structures and arrays.

A. The "Flash Card" Metric Generator (Screen 1)

After a shot, randomly select one of these objects to display in the HUD:

```
[
  { "category": "Batting", "label": "Bat Speed", "value": "112", "unit": "km/h", "status": "green" },
  { "category": "Batting", "label": "Impact Timing", "value": "Perfect", "unit": "", "status": "green" },
  { "category": "Batting", "label": "Backlift Angle", "value": "28", "unit": "°", "status": "red" },
  { "category": "Bowling", "label": "Release Speed", "value": "132", "unit": "km/h", "status": "green" },
  { "category": "Bowling", "label": "Release Height", "value": "1.9", "unit": "m", "status": "yellow" }
]
```

(Logic: Green = Good, Yellow = Average, Red = Poor/Risk).

B. "Elite Zone" Text Generator Formula (Screen 2)

Formula: [Action Verb] + [Metric] + by + [Gap Value] + to + [Incentive]

- *Example 1:* "Increase Peak Bat Speed by 8 km/h to enter the State Selection Average."
- *Example 2:* "Improve Release Consistency by 12% to unlock Elite Batch Visibility."

C. Biomechanical "Fix It" Library (Screen 3)

Map these internal errors to UI text:

```
{
  "Head_Fall_Offside": "Your head is falling to the off-side at impact. Keep your chin parallel to the ground to improve balance.",
  "Elbow_Low_Backlift": "Your top elbow is too low. Pull the elbow up towards the sky to generate a straighter downswing.",
  "Front_Foot_Plant_Late": "Late footwork. Trigger your front foot movement 0.2 seconds earlier to handle pace.",
  "Front_Leg_Collapse": "Your front leg is bending at release. Brace the front knee (lock it) to catapult your upper body forward.",
  "Lateral_Flexion_High": "Crunching too much to the side. Keep your spine upright to reduce injury risk and gain height."
}
```

D. Coach "Red Flag" Alerts (Screen 4)

```
[
  { "id": 1, "type": "injury", "color": "red", "student": "Rahul (U-19)", "message": "High Lumbar Stress detected (Hyper-extension > 20°). End session." },
  { "id": 2, "type": "form", "color": "yellow", "student": "Amit", "message": "Batting 10% slower this week. Check for fatigue." },
  { "id": 3, "type": "talent", "color": "green", "student": "Karan", "message": "Just bowled 135km/h (Top 10 in Academy)." }
]
```

5. SCREEN SPECIFICATIONS (PLAYER APP - MOBILE FIRST)

Screen 1: Recording HUD (Live View)

- **Purpose:** Capture video, provide dopamine-hit instant feedback.
- **Layout Rules:**
  - Base <View className="flex-1 bg-black">.

    - <Camera className="flex-1 w-full"> takes up the entire screen.
    - **Skeleton Overlay:** Use absolute positioning (absolute inset-0 items-center justify-center opacity-50). Render a static SVG skeleton in Neon Green (#39FF14).
    - **Status Pill:** Placed at absolute top-12 self-center. Styled as a dark translucent pill (bg-slate-900/80 rounded-full px-4 py-2).
    - **Flash Card:** Appears at absolute bottom-32 w-[90%] self-center. Must use high-contrast text mapping to the "Flash Card Metric" JSON stub.

Screen 2: The Analysis Dashboard (Post-Session Gamification)

- **Purpose:** Show the gap between current skill and professional selection.
- **Layout Rules:** Standard vertical scroll (<ScrollView className="flex-1 bg-slate-50 p-4">).
- **Components:**
    - **Noticeability Meter:** Centered at the top. Use <CircularProgress /> showing a hardcoded percentage (e.g., 75%). Underneath, display the dynamic "Elite Zone Text Generator" string in a <Text className="text-lg font-semibold text-center mt-4">.
    - **Fatigue Battery:** A card flex-row items-center justify-between p-4 bg-white rounded-xl shadow-sm. Show a green battery icon.

Screen 3: Shot Detail View (Deep Dive)

- **Purpose:** Specific error correction.
- **Layout Rules:**
    - Top Half: <Video> player (w-full h-64).
    - Bottom Half: <ScrollView className="flex-1 p-4 bg-white rounded-t-3xl -mt-4">.
    - **Error Box:** A red-bordered card <View className="border-l-4 border-red-500 bg-red-50 p-4">. Populate text from the "Fix It" Library based on a hardcoded error_id.

6. SCREEN SPECIFICATIONS (COACH APP - RESPONSIVE WEB/TABLET)

Screen 4: Academy Command Center (Dashboard)

- **Purpose:** High-throughput screening, "Management by Exception".
- **Layout Rules (CRITICAL FOR SPATIAL REASONING):**
    - Must use the responsive pattern: <View className="flex-1 flex-col md:flex-row bg-slate-100">.
    - **Sidebar (Web) / Top Nav (Mobile):** <View className="w-full md:w-64 bg-slate-900 p-6 flex-col gap-4">.
    - **Main Content:** <ScrollView className="flex-1 p-4 md:p-8">.
- **Components within Main Content:**
    - **Grid:** <View className="flex-col md:flex-row flex-wrap gap-4">.
    - **Red Flag Ticker:** A full-width column displaying the Coach Alerts JSON. Render items with conditional borders (border-l-4 border-red-500 for injury, border-yellow-500 for form).

    - **Roster Quick-View:** A data table or list of students using Tailwind Grid (grid-cols-3 or grid-cols-4).

Screen 5: Split-Screen Comparison ("Ghost Mode")

- **Purpose:** Compare student technique against a professional benchmark.
- **Layout Rules:**
    - <View className="flex-1 flex-col md:flex-row gap-4 p-4">.
    - **Video Containers:** Two identical video containers (flex-1 bg-black rounded-xl overflow-hidden aspect-video). Left is "Student", Right is "Pro Benchmark".
    - **Floating Metrics Table:** Placed below the videos on mobile, or stretching below both on Desktop. Use a Tailwind table structure:
        - Row: <View className="flex-row justify-between border-b border-slate-200 py-3">
        - Columns: Metric Name, Student Value, Pro Value, Delta (Gap). Use Green text for positive deltas, Red for negative.

7. EXECUTION DIRECTIVE FOR THE AGENT

1. **Do not invent features:** Stick strictly to the UI components and JSON stubs listed above.
2. **Ensure flawless rendering:** Rely on Tailwind's Flexbox (flex-1, flex-row, items-center, justify-center, gap-X) to prevent overlapping UI elements.
3. **Color Palette:** Use standard Tailwind colors. Primary backgrounds: bg-slate-50, bg-slate-900. Accents: bg-green-500, bg-red-500, bg-blue-600.
4. **Simulate AI:** When a user clicks "Record" on Screen 1, use a setTimeout of 3000ms, show a Lottie/ActivityIndicator, then display the Flash Card metric to simulate "Processing".