MVP Tech Stack (Client-Only, React Native + NativeWind)

Updated: November 06, 2025

Scope & Principle

- Scope: Connect to device via Bluetooth Low Energy (BLE), store data locally, and show gamified visuals.
- Principle: No backend. Everything runs on-device. Optional SaaS only for crashes and analytics.

Mobile App (React Native)

- Framework: React Native 0.74+ with TypeScript
- Navigation: @react-navigation/native
- State Management: Zustand (lightweight global store)
- Local Storage: react-native-mmkv (fast key/value persistence)
- BLE: react-native-ble-plx (scan, connect, notifications)
- Permissions: react-native-permissions

Background Runtime:

iOS: Xcode → Background Modes → "Uses Bluetooth LE accessories"

Android: react-native-background-actions (foreground service only during active recording)

- UI / Styling: NativeWind (Tailwind utilities for React Native)

Icons: react-native-vector-icons or lucide-react-native

Charts & Visual Effects (choose one path; you can mix later):

Standard charts quickly: react-native-svg + victory-native

High-FPS custom visuals: @shopify/react-native-skia

- Animations: lottie-react-native (celebrations, loaders)
- Testing: Jest (unit). Optional: Detox (single smoke test).

Observability (Hosted, no servers to run)

- Crash Reporting: Sentry (@sentry/react-native)

  Initialize early; upload dSYMs/Proguard in CI; diagnostics opt-in by default.

- Product Analytics: PostHog Cloud (posthog-react-native)

  Enable only after user consent; track screen views and key milestones (connect, session start/stop, celebration).

CI / CD (GitHub Actions)

- PR Checks (no signing): TypeScript compile (tsc), ESLint, Jest, Android assembleDebug, iOS simulator build.

Main Branch Builds (signed artifacts only):

  Android: Gradle bundleRelease → .aab artifact

  iOS: Fastlane build_ipa (via fastlane match) → .ipa artifact

- Artifacts are stored in Actions; manual upload to TestFlight / Play internal when ready.

Accounts Needed

- GitHub (repository and Actions)
- Apple Developer (App Store Connect / TestFlight)
- Google Play Console
- Sentry (project DSN)
- PostHog (project key)
- Optional: Expo account (only if you prefer EAS Build/Updates over Fastlane)

App Data (On-Device Only)

- sessions/{id}: startTs, endTs, stats
- events/{sessionId}: time-series samples (downsample older points)
- settings: theme, units, consent flags (telemetry, crash)

Privacy & Permissions

- iOS (Info.plist): NSBluetoothAlwaysUsageDescription; add location key only if required by your BLE scenario.

- Android: BLUETOOTH_SCAN, BLUETOOTH_CONNECT, (SDK < 31) ACCESS_FINE_LOCATION, FOREGROUND_SERVICE.
- Telemetry is opt-in; collect no PII; no user accounts.

First 7 Days (Milestones)

- 1) BLE connect/subscribe happy path.
- 2) Session recorder with MMKV; throttle live UI updates.
- 3) NativeWind UI shell and game HUD.
- 4) Charts (Victory or Skia) and a celebration animation.
- 5) Background run (iOS mode; Android foreground service).
- 6) Sentry and PostHog behind consent toggles.
- 7) CI builds .aab and .ipa on main; manual upload to TestFlight/Play Internal.