

AASHIR ATHAR

Senior React Native Engineer | Mobile-First Product Builder

Lahore, Pakistan | +92 307 477 8889 | +92 325 420 0010 | aashirathar@gmail.com

[linkedin.com/in/aashirathar](https://www.linkedin.com/in/aashirathar) | github.com/aashir-athar | aashir-athar.github.io

PROFESSIONAL SUMMARY

Senior React Native Developer with 3+ years building and shipping cross-platform mobile applications for iOS and Android, deployed at scale to 50,000+ active users across the United States and Pakistan. Deep specialization in Expo SDK 54, the React Native New Architecture (Fabric + TurboModules), TypeScript strict mode, and offline-first architectures that sustain 60–120 FPS on entry-level Android hardware. Full-stack reach across Node.js, Express, RESTful API design, MongoDB, PostgreSQL, and Supabase, with proven end-to-end ownership from data model through App Store Connect and Google Play Console submission. Author of a published open-source Expo native module on NPM, plus a portfolio spanning fintech, healthcare, conservation tech, and real-time trading. Pairs senior engineering judgment with an AI-augmented development workflow that compounds team velocity without trading away code quality.

TECHNICAL SKILLS

Mobile (Primary Focus): React Native 0.81+, Expo SDK 54, Expo Router v6, New Architecture (Fabric + TurboModules), React Native Reanimated v4, Gesture Handler, FlashList v2, expo-image, expo-notifications, expo-glass-effect, expo-sqlite, EAS Build & Update, Fastlane, native Kotlin (Android module authoring), Deep Linking, Biometric Auth, Push Notifications, Background Tasks

Languages: TypeScript (strict mode), JavaScript (ES2024), Kotlin, SQL, Python (ML pipelines), HTML, CSS

State & Data: Zustand, TanStack Query v5, React Context, AsyncStorage, MMKV, Zod, React Hook Form

Backend & Infrastructure: Node.js, Express 5, Next.js 16 (App Router, RSC), MongoDB, Mongoose, PostgreSQL, Supabase (Auth, RLS, Realtime, Storage), PostGIS, Vercel Serverless, Cron Architectures, REST API Design, JWT, Cloudinary, Arcjet, Firebase (Firestore, FCM, Auth)

Authentication: Clerk (OAuth, MFA), Supabase Auth (OTP), Firebase Auth, JWT

AI / ML: TensorFlow Lite (on-device inference), MobileNetV2 transfer learning, react-native-fast-tflite, transformers.js, LanceDB vector search, Groq SDK, OpenAI / Anthropic APIs, embeddings, RAG pipelines, LoRA fine-tuning

Design & Media: Figma, Adobe XD, Photoshop, After Effects, Premiere Pro, CapCut, design-token systems, pixel-perfect implementation, motion design, brand systems

DevOps & Tooling: Git, GitHub Actions, EAS, Fastlane, Sentry, Flipper, Jira, conventional commits, monorepo workflows

AI-AUGMENTED DEVELOPMENT WORKFLOW

Runs a purpose-built development workflow where every AI tool earns its place. Has trained a custom Claude environment with tailored agents, skill packs, prompt libraries, and UI/UX best-practice modules — resulting in faster, higher-quality code than conventional development alone. Uses Grok for deep technical research with source-backed answers, ChatGPT for pressure-testing architectural ideas, Nano Banana for visual generation (icons, app assets, brand imagery), Veo for video production, and ElevenLabs for voiceover. This isn't AI-assisted work; its AI-multiplied output with engineering judgment still in the driver's seat.

PROFESSIONAL EXPERIENCE

React Native Developer | 3STechLabs

Remote, United States | May 2024 - Present

- Delivered cross-platform React Native features for iOS and Android — including biometric authentication, real-time data synchronization via Firebase, and fault-tolerant REST API integrations engineered to survive intermittent mobile network conditions.
- Drove a 20% lift in user engagement within two months of release through iterative UI/UX refinements grounded in production analytics and session-replay review rather than design instinct.
- Diagnosed and resolved high-priority production crashes using Flipper and Sentry telemetry, cutting iOS and Android crash rates measurably and stabilizing the weekly release train.
- Collaborated daily with backend engineers, product managers, and QA in an Agile/Scrum environment to ship customer-facing features within sprint commitments.

Senior React Native Developer | AppGlide Technologies

Remote | June 2023 - April 2024

- Architected and led three production Expo applications collectively serving 50,000+ active users — owning the full software development lifecycle from data modeling and RESTful API design through App Store and Google Play submission.
- Mentored four junior React Native developers through structured code reviews, a modular component contract, and pair programming sessions — cutting new-hire ramp time by roughly 40% and lifting team-wide code quality.
- Designed and shipped Continuous Integration / Continuous Deployment (CI/CD) pipelines on Fastlane and EAS Build, eliminating ~90% of manual deployment errors and reducing each release to a single command.
- Partnered with product management and design to deliver MVPs on a tight 6–8 week cadence, anchoring technical decisions to business outcomes rather than speculative architecture.
- Cut cold-start time 35% by restructuring the navigation graph, applying lazy loading and code splitting, and rebuilding the image pipeline around expo-image.
- Drove technical decisions across state management (Redux, Zustand), authentication, and offline data strategy — informing patterns adopted across the wider engineering org.

Junior React Native Developer | TechNova Solutions Inc.

Remote, United States | February 2022 - April 2023

- Improved overall mobile application performance by ~25% through a rewrite of the data-fetching layer and the introduction of asynchronous caching against Firebase Firestore — removing redundant network round-trips.
- Released multiple consumer-facing iOS and Android applications to the Apple App Store and Google Play Store with a clean post-launch crash record across both platforms.
- Contributed actively to Agile and Scrum ceremonies — sprint planning, story estimation, daily standups, and retrospectives — and consistently delivered feature work ahead of committed timelines.
- Wrote modular, reusable React Native components and React Hooks that became part of the team's shared component library.

UI/UX & Graphic Designer (Part-Time) | W3 Technologies

Remote | October 2021 - March 2023

- Translated business requirements into responsive, mobile-first UI/UX mockups in Figma and Adobe XD — handed off to engineering as production-ready specs with token-level annotations and full interaction states.
- Embedded directly with React Native development teams to maintain visual consistency across components, screen states, and both iOS and Android targets.

KEY PROJECTS

xMind - Open-Source Social Networking Platform (github.com/aashir-athar/xmind-app)

Expo SDK 54 | React Native 0.81 | React 19 | TypeScript | Express 5 | MongoDB Atlas | Clerk | Cloudinary | Arcjet | NativeWind

- Designed and shipped a production-grade social networking platform end to end — Expo React Native client, stateless Node.js / Express API on Vercel, MongoDB Atlas data layer, and an on-device feed ranking engine — built as a reference implementation rather than a tutorial app.
- Engineered a 12-layer on-device feed ranking algorithm combining TF-IDF cosine similarity, engagement velocity scoring, exponential temporal decay, social-graph signals, and Maximal Marginal Relevance ($\lambda=0.7$) reranking — sustaining sub-3ms per-page ranking latency.
- Modeled reshare semantics on MongoDB \$lookup aggregations with indexed user arrays and atomic \$pull / \$addToSet operations, plus graceful silent-failure handling when source posts are deleted.
- Achieved <1.5s cold start on Pixel 5 and sustained 60 FPS on 2 GB Android devices through FlashList v2 (strict recyclingKey discipline), off-thread image decoding via expo-image, and Reanimated v4 worklets confined to the UI thread.
- Built a token-driven design system (primitive → semantic → component) with platform-aware translucency: iOS 26 Liquid Glass, BlurView fallback for older iOS, and flat tinted surfaces on Android — single-point theme switching across the entire application.
- Hardened the backend with Clerk authentication middleware, Arcjet shield with token-bucket rate limiting, CORS allowlists, and cached Mongoose connections across Vercel serverless cold starts.

BludStack - Real-Time Blood Donor Matching Network (github.com/aashir-athar/BludStack-RN)

Expo SDK 54 | TypeScript | Express | Supabase (Postgres + RLS + Realtime) | PostGIS | expo-notifications

- Built a civic-impact healthcare platform that matches blood donors with recipients in minutes rather than hours, using an expanding geo-ring escalation algorithm (1km → 5km → 15km → 30km → 50km → nationwide) powered by PostGIS ST_DWithin geospatial queries.
- Engineered atomic accept_blood_request and complete_blood_donation RPCs as SECURITY DEFINER PostgreSQL functions — enforcing capacity caps, donor cooldown windows, age eligibility, and the N-donors rule inside a single transaction to eliminate partial-fulfillment race conditions.
- Locked the data layer with Row-Level Security (RLS) across every table plus trigger-based field guards, ensuring donor phone numbers stay private until acceptance even if a client bypasses validation.
- Wired live donor heartbeat streaming over Supabase Realtime channels, giving recipients an Uber-style live-tracking experience on the map view.
- Hit a target of 120 FPS on mid-tier Android using a Reanimated-v4-only animation rule, FlashList v2 with stable keys, and skeleton loaders matched to final content geometry — no spinners anywhere in the app.
- Engineered reliable killed-state push notification delivery on Android through per-OEM channel tuning (Xiaomi, OnePlus, Samsung) and a documented battery-saver bypass strategy.

FlowMoney - AI-Powered Financial Awareness App for Pakistan

(github.com/aashir-athar/FlowMoney)

Expo SDK 54 | TypeScript | Zustand | expo-transaction-sms-reader | Groq (Llama 3.3 70B) | i18n-js | Reanimated 4

- Designed a fintech dashboard that builds itself — Pakistani bank and digital-wallet SMS alerts (HBL, UBL, MCB, Meezan, EasyPaisa, JazzCash, NayaPay, SadaPay, and 22+ more) parsed on-device and auto-categorized into a private transaction ledger with zero manual entry.

- Integrated expo-background-fetch reconciliation so SMS messages received during force-quit still flow into the ledger on next launch, and added a soft-duplicate window to absorb dual-SIM and broadcast-retry noise.
- Built a Groq-powered AI Money Assistant on Llama 3.3 70B with a three-layer client-side rate limiter (1.5s gap, sliding 20 RPM window, retry-after countdown UI) — every response grounded in real transaction data, never fabricated.
- Optimized the entire application for 2 GB-RAM Android hardware: memoized leaf components, narrow Zustand selectors, no LinearGradient inside list rows, no BlurView on Android, and Reanimated worklets used exclusively for animations.
- Shipped trilingual support — English, Urdu (RTL), and Hindi via i18n-js — with material differentiation between iOS Liquid Glass and Android flat surfaces, deliberately rejecting fake-glass emulation that taxes low-end GPUs.
- Privacy-by-default architecture: no cloud backend, no analytics SDKs, all processing on-device — privacy assertions rendered in calm green rather than alarmist red.

expo-transaction-sms-reader - Published Expo Native Module (NPM Package, Android-only)

github.com/aashir-athar/expo-transaction-sms-reader

Expo SDK 54 | Kotlin | TypeScript | Android BroadcastReceiver | Real-Time Banking & Wallet SMS Intelligence

- Authored and published an open-source Expo native module on NPM that converts raw banking SMS into typed transaction objects across 60+ financial institutions spanning Pakistan, India, Bangladesh, and the GCC region.
- Implemented a multi-stage Kotlin → TypeScript pipeline — runtime-registered Android BroadcastReceiver (sidestepping Google Play's default-SMS-handler review), message classifier, heuristic parser, and aggregation layer.
- Designed a 9-factor weighted confidence model (indicator keywords, amount, type, currency, bank code, account mask, reference ID, balance, channel) capped at 0.95 — surfacing the score on every transaction instead of hiding it from consumers.
- Engineered a ref-counted listener pattern so multiple subscribers share a single native receiver with automatic cleanup, plus a sender-registry currency disambiguator that resolves "Rs" across PKR / INR / LKR / NPR.
- Kept the entire parser in dependency-free TypeScript (~500 lines) so it can re-run on any string for unit testing, auditing, and offline analysis without native recompilation.
- Exposed clean APIs for live listening, inbox queries, OTP autofill, and custom parser registration — with typed no-ops on iOS and web for safe cross-platform consumption.

Forest Sentry - Offline-First Conservation App (built for WWF Pakistan)

github.com/aashir-athar/forest-sentry

Expo SDK 54 | TypeScript strict | TensorFlow Lite | MobileNetV2 | Supabase + PostGIS | expo-sqlite | Skia | MIT License

- Built an offline-first conservation field application for forest rangers and researchers operating in connectivity-dead regions like Nathia Gali, Pakistan — every captured record engineered to withstand legal scrutiny.
- Shipped an on-device leaf-health classifier using react-native-fast-tflite with a bundled PlantVillage MobileNetV2 model (39 classes folded into 4 health buckets), backed by a deterministic HSV heuristic fallback when the native module is unavailable.
- Designed a closed active-learning loop: field corrections are stored locally with predicted and corrected labels, synced to Supabase on reconnection, fed into a Python / Keras retraining pipeline, and redistributed as over-the-air (OTA) model updates through Supabase Storage — no app store release required.
- Captured GPS, species, girth, and height for the tree registry in under 20 seconds per record, with PostGIS polygon detection flagging incursions into protected zones in real time.

- Built a dependency-ordered sync engine (zones → trees → inspections → incidents) using stable UUIDs and idempotent upserts — safe for the intermittent connectivity field operators actually face.
- Rendered tree-trajectory projections (linear regression + exponential moving average) with @shopify/react-native-skia for native-canvas performance, and tuned the UI for Fitts's Law compliance under gloved-hand operation.

MindeesAI - Self-Training Open-Source Language Model (In Development)

(github.com/aashir-athar/mindeesai)

Next.js 16 | React 19 | Custom TypeScript Transformer | LanceDB | transformers.js | PyTorch | Vercel — a language model

- Building a genuinely self-improving Large Language Model (LLM) from scratch in TypeScript — not a wrapper around vendor APIs. Implements DeepSeek-V3-class innovations: Mixture of Experts (MoE) with top-K routing, Multi-head Latent Attention (MLA) for ~10x KV cache reduction, Multi-Token Prediction auxiliary heads, Grouped-Query Attention, RoPE, RMSNorm, and SwiGLU.
- Designed a continual-learning loop running real gradient descent every five minutes with eval-gated commits and automatic LoRA rollback when post-training benchmarks regress — preventing degradation from noisy data.
- Engineered a vector-semantic memory layer using LanceDB with BGE-small-en-v1.5 embeddings running locally via transformers.js, plus cross-encoder reranking through ms-marco-MiniLM-L-12-v2 for retrieval precision.
- Built a 7-deep LLM provider fallback chain (Groq tiers → Gemini Flash → Gemini 2.0) with 15-second handshake timeout and 25-second per-chunk stall detection — enabling free-tier production deployment with no subscription risk.
- Designed a 23-dimensional persistent state system (8-D mood vector, 16-D user model, 4-D relationship tensor, knowledge graph, empathy classifier, sentiment EMAs, topic affinity) that evolves conversationally without explicit configuration.
- Implemented Reinforcement Learning pipelines — Group Relative Policy Optimization (GRPO), DPO RLHF, and constitutional self-critique — plus an 8-provider autonomous research loop (Tavily, Exa, JINA, DuckDuckGo, Wikipedia, arXiv, Reddit, HackerNews) that fires whenever the model hedges.

EDUCATION

BSc Software Engineering | Riphah International University

2016 - 2021

CERTIFICATIONS

Mobile App Development (React Native) | Azad Chaiwala Institute

2023 - 2024

Software Engineering Fundamentals | Riphah International University