

Divyansh Goyal

+91-9870764076 — goyaldivyansh1111@gmail.com
github.com/Phoenix1808 — linkedin.com/in/divyansh-goyal — portfolio-divgo.vercel.app

Leadership Experience

Android Lead — GDG on Campus, JSSATEN 2026 – Present
Leading Android-focused sessions and mentoring peers on Jetpack Compose, MVVM, and Clean Architecture within the campus developer community.

Android Domain Recruitment Panel JSSATEN
Evaluated candidates on Android fundamentals, architecture patterns, and system design as part of the technical selection panel.

Achievements & Open Source

Top 20 Finalist — AI Mobile Hackathon 2026 June 2026
Ranked 9th among Top 20 participants in Season 1 · [Certificate No. AIH-1-4-000357](#)

Open Source Contributor — Droid-ify, LibreTube & detekt June 2026

- [Droid-ify #1348](#): Fixed RTL icon mirroring for Arabic locales in F-Droid client (7k+ stars)
- [LibreTube #8551](#): Fixed ungrouped channels not updating after group assignment (12k+ stars)
- [detekt #9424](#): Fixed false-positive guard clause detection in `ReturnCount/ThrowsCount` rules (7k+ stars)

Projects

CrewOrCrook — Real-Time Multiplayer Android Game | Kotlin, Socket.IO, Google Maps [GitHub](#) ↑

- Built an Among Us-style real-time multiplayer Android game supporting live player interactions across multiple devices using WebSockets (Socket.IO) for low-latency event synchronization.
- Integrated Google Maps SDK with GPS to track and render live player positions on a shared map, enabling location-aware gameplay.
- Designed an event-driven game architecture handling kill events, reporting, voting, and meeting flows with consistent multi-device state.
- Implemented a lobby management system with live player synchronization and role-based gameplay logic.

RookNomics — Trading Strategy Simulator | Kotlin, MVVM, MPAndroidChart, Retrofit [GitHub](#) ↑

- Developed a quantitative strategy backtesting simulator that processes historical market data fetched via REST APIs to evaluate trading strategy performance.
- Implemented MVVM architecture with ViewModel and LiveData for clean separation of concerns and lifecycle-safe data handling.
- Visualized financial metrics — returns, drawdowns, and volatility — using MPAndroidChart with custom chart configurations.
- Computed core performance indicators to give users data-driven insights into their simulated strategies.

CrashTrace — Android Crash Monitoring SDK | Kotlin, SDK Dev, JitPack, Lifecycle API [GitHub](#) ↑ [JitPack](#) ↑

- Built and published a lightweight Android SDK for automatic crash detection, session tracking, and user interaction monitoring, distributed via JitPack.
- Leveraged ActivityLifecycleCallbacks for zero-instrumentation screen tracking, requiring no manual setup from the integrating app.
- Designed for plug-and-play integration with minimal performance overhead; captures runtime crashes and interaction events for debugging.

AuraPulse — Voice-First AI Burnout Detection App | Kotlin, Gemini, Agora ConvoAI, Room [GitHub](#) ↑

- Built a voice-first Android wellness companion using Agora ConvoAI + Cartesia STT/TTS for daily 2-minute spoken check-ins; Gemini 2.0 Flash analyses transcripts to extract mood, energy, and stress signals.
- Engineered an on-device burnout prediction engine in pure Kotlin using a 14-day weighted scoring window, producing risk bands (LOW/MODERATE/HIGH) without any cloud dependency.
- Architected with Clean Architecture + MVVM, Hilt DI, Room, and StateFlow; implemented multi-tier AI fallback chain and keyword-aware offline mode to ensure zero-crash demos under API quota limits.

Technical Skills

- **Languages:** Kotlin, Xml, Java, Dart
- **Android Core:** Android SDK, Activity & Fragment Lifecycle, XML Layouts
- **Architecture:** MVVM, ViewModel, LiveData, StateFlow, MVC, Repository Pattern
- **Networking:** Retrofit, REST APIs, JSON Parsing, WebSockets (Socket.IO)
- **Libraries & SDKs:** MPAndroidChart, Google Maps SDK, Firebase, JitPack, Room, Hilt
- **Tools:** Android Studio, Git, GitHub, Firebase, JitPack

Education

JSS Academy of Technical Education
B.Tech in Computer Science Engineering

2024 – 2028