

Umar Akram Mirad

uamirad00@gmail.com — +20 1023651704 —
linkedin.com/in/umarrakram — github.com/umarrakram

Professional Summary

Results-driven **iOS Developer** specializing in **Swift, SwiftUI, UIKit, and Combine**, with a strong foundation in Apple's Human Interface Guidelines and performance tuning. End-to-end ownership of the app lifecycle—architecture, feature delivery, instrumentation, CI/CD, and App Store deployment. Proven impact building production apps and an IDE-class editor for iPhone and iPad. Adept at API design/integration, secure payments (Apple Pay & StoreKit), offline storage, and rigorous testing. Passionate about shipping delightful, accessible, and maintainable iOS experiences.

Education

B.Sc. in Computer Science & Information Technology (Expected Jun 2026)

Egypt-Japan University of Science and Technology — Alexandria, Egypt

Relevant Coursework: Mobile App Development, Data Structures & Algorithms, Software Engineering, Machine Learning

Core Skills

Languages/Frameworks: Swift, SwiftUI, Combine, UIKit, Objective-C (basic), Python, C++

iOS SDKs: Core Data, Core Animation, Core Location, WebKit, PassKit, StoreKit, AVFoundation

Architecture: MVVM, Clean Architecture, Dependency Injection, modular frameworks, Swift Concurrency (async/await)

Networking: REST/JSON, URLSession, Vapor (server), Codable, Request/Response caching, ETags

Data: Core Data, Realm (familiar), SQLite, background sync, migrations

Tooling: Xcode, Instruments (Time Profiler, Allocations, Leaks), Firebase, Git, Fastlane, Bitrise, TestFlight, Postman

Testing: XCTest, XCUITest, snapshot tests, builders/mocks, TDD principles, CI test plans

Other: Localization (Arabic/English), Accessibility, Agile/Scrum, App Store deployment, Privacy & Security

Professional Experience

Onvo

iOS Developer

[Link on App Store](#)

Remote - Egypt

Feb 2024 – Nov 2024

- Designed and shipped SwiftUI-based social/creative features (profiles, music/movie suggestions, drawing messages), driving a **40% lift in DAU**.
- Implemented **Apple Pay (PassKit)** and **StoreKit** for one-time & subscription purchases; added receipt validation and server-side verification, reducing payment failures and improving checkout conversion.
- Refactored network layer and REST integration with **Vapor** back-end: introduced response caching, request coalescing, and retry/backoff—cutting P50 latency by **50%**.
- Built adaptive streaming via **WebKit** and AVFoundation; tuned buffer sizes and ABR heuristics to reduce startup latency by **35%**.
- Migrated legacy MVC to **MVVM + Combine** modules; introduced feature flags, dependency injection, and snapshot testing to raise code coverage and simplify rollbacks.
- Raised product quality with Instruments (leak detection, time profiler), on-device logging, crash triage, and performance budgets (cold start, scroll FPS, memory).
- Led code reviews and knowledge-sharing sessions on Swift Concurrency, result builders, and diffable data sources; aligned UI with Apple's HIG and accessibility best practices.

Selected Projects

Axios — IDE-Class Code/Text Editor for iPhone & iPad *SwiftUI, UIKit, Combine, WebKit, C++, Node.js*

[Link on App Store](#)

- Built a **complete text/code editor** that runs on iPhone and iPad, designed as a practical **VS Code replacement on iOS**.
- Implemented custom **text highlighting algorithms** (incremental lexing + on-the-fly tokenization) for responsive, large-file editing; optimized attributed-string pipelines and layout invalidation for 60 fps scrolling.
- Integrated **Node.js runtime** for project execution (`npm` workflows, GitHub repo cloning, package management) with sandboxed I/O; bridged output to SwiftUI terminals via Combine publishers.
- Embedded **C++ toolchain/compilers** for offline experiments; bridged to Swift with `extern "C"` shims and background execution via OperationQueues/GCD.
- Added multi-pane editing (split view), search/replace with regex, minimap, auto-completion hooks, bracket matching, snippets, theming, and command palette.
- Crafted a robust file abstraction (local + iCloud Drive) with safe saves, conflict detection, and background indexing; designed a plug-in protocol for future language servers.
- Hardened the app with sandboxes, capability scoping, and crash containment; integrated analytics for session funnels, editor latency, and cold start timing.
- Built release automation: versioning, release notes, screenshots, TestFlight distribution; instrumented editor KPIs and crash regression gates in CI.

Swiss Pairing System — Chess Tournament Management *React, Node.js, Express, MongoDB*
[View on GitHub](#) — [Live Demo](#)

- Built a **full-stack web application** for managing chess tournaments using the Swiss pairing algorithm with automated round generation and real-time standings.
- Implemented **React 18** frontend with React Router 6 for seamless navigation; designed admin dashboard for tournament creation, player management (individual/bulk Excel upload), and match result recording.
- Developed **RESTful API** with Node.js/Express backend and MongoDB for persistent storage; integrated Multer for file uploads and XLSX parsing for bulk player imports.
- Engineered Swiss pairing algorithm with tiebreak calculations (Buchholz, Sonneborn-Berger) and match scheduling system; enabled public viewing links for tournament transparency.
- Deployed to **Vercel** with serverless functions for backend and static hosting for frontend; configured MongoDB Atlas for production database with environment-based configuration.
- Designed responsive UI with clean UX for admins and participants; implemented authentication system with secure admin credentials and role-based access control.

WeatherSnap — Forecast App *SwiftUI, Combine, Core Location, OpenWeatherMap*

- Reactive pipeline with Combine (debounce, retry, backpressure) and background refresh; GPS-based forecasts with robust permission and error handling.
- Lottie-powered weather transitions and composable SwiftUI components; designed offline cache, stale-data policies, and graceful degradation.

Notable Achievements

- **2 iOS apps published** with strong adoption and user feedback; cumulative **100K+ downloads** and ~4.4/5 rating history.
- Repeated wins reducing app startup, API latency, and crash rate through profiling, refactors, and CI gates.