

Keyur Savalia

📍 San Francisco ✉ savalia.keyur1091@gmail.com ☎ 415-349-1732 🌐 keyursavalia.com 📱 ksavalia21 in keyursavalia

Education

San Francisco State University
MS in Computer Science

San Francisco
December 2026

Gujarat Technological University
BS in Computer Science

India
May 2024

Experience

iOS Engineer Intern
Qrious Tech Team LLP

India
January – June 2024

- Owned migration of legacy Objective-C modules to Swift, increasing code usability/maintainability by 70%.
- Architected reusable modules & internal frameworks with CocoaPods, improving code reuse by 45%.
- Optimized RESTful API calls, cutting server response times by 40%.
- Applied SOLID and TDD, reducing defects by 34%.
- Drove code reviews, lowering production bugs by 26% and raising code quality across the team.

iOS Engineer Intern
Qrious Tech Team LLP

India
May – August 2023

- Implemented MVVM with Core Data, improving maintainability and cutting bug discovery-to-resolution time by 25%.
- Partnered closely with UX/UI to address feedback, raising user satisfaction by 20% and reducing complaints by 30%.
- Used Xcode to diagnose and fix issues, boosting app stability by 50% and performance by 32%.
- Built responsive interfaces with Auto Layout, ensuring consistent layouts across iPhone and iPad.
- Collaborated via Git (branching/PRs), reducing merge conflicts by 30%.

Technical Skills

Languages: Swift, Objective-C, Python, JavaScript.

Mobile App Development: iOS SDK, UIKit, Core Data, Cocoa Touch, Xcode.

Testing & Debugging: JUnit, Espresso, XCTest, Postman.

APIs & Libraries: RESTful APIs, Alamofire, Firebase, RealmDB.

Frameworks & Tools: SwiftUI, React.js, FastAPI, Figma.

Projects

SplitPro — SwiftUI, Firebase — [GitHub](#)

- Utilized MVVM-based state management and centralized AuthenticationManager, reducing UI update logic by 70%.
- Optimized performance through async/await patterns, reducing API calls by 60% and improving app responsiveness.
- Used Firebase Auth with session management, reducing auth failures 85% through comprehensive error handling.
- Documented codebase with 40% comment coverage, MARK-based organization following industry standards.

moodTracker — SwiftCharts, UserDefaults — [GitHub](#)

- Utilized Swift Charts for aggregating 20+ sample mood entries across 7-day periods for trend analysis.
- Designed data layer with UserDefaults and Codable protocol for JSON serialization, ensuring 100% offline functionality.
- Leveraged Swift's Calendar API and DateFormatter for timestamp handling, grouping mood entries using Dictionary.
- Enabled light/dark theme utilizing UIApplication API, persisting user preferences with didSet patterns.

API Doc Generator — Python, JavaScript — [GitHub](#)

- Used React and FastAPI with async/await, reducing API latency by 40% through non-blocking operations.
- Utilized REST API architecture with CORS and ZIP file generation, delivering 100ms response through async file ops.
- Optimized hooks (useState, useRef, useEffect) with lifecycle cleanup, reducing memory leaks by 85%.