# OpsLens - GitHub Copilot Prompts for Full iOS App Build

## 1. Project Structure & Setup

Create a SwiftUI iOS app called OpsLens with support for iOS 16+.

Include folder structure for Models, Views, ViewModels, and Services.

## 2. Dashboard View UI

Create a SwiftUI dashboard with cards for:

- Crash Rate (iOS, Android)

- Login Time (iOS, Android)

- Account Load Time (iOS, Android)

- App Rating & Sentiment

- Session Count

- Current Release Health

Use VStack and HStack to create responsive tiles.

## 3. Network Service Layer

Build a NetworkService class with async/await URLSession.

Add support for JSON decoding, token headers, and base URL.

Add generic fetch<T: Decodable>() method.

## 4. API Models

Define Codable structs for each API response:

- CrashData (Glassbox)

- PerformanceMetrics (AppDynamics)

- LogInsights (Splunk)

- ReviewSentiment (AppBot)

Use mock sample JSON to infer structures.

## 5. Dashboard ViewModel

Create DashboardViewModel:

- Fetch data from all APIs asynchronously.

- Expose @Published properties for crash rate, sentiment, login time, etc.

- Add loading and error states.

## 6. Charts Integration

Add a SwiftUI Charts view to display:

- Crash rate over time (line chart)

- Login and load times comparison (bar chart)

Bind to ViewModel arrays.

## 7. TabView Navigation

Create a SwiftUI TabView with the following tabs:

- Dashboard

- Logs

- Sessions

- Reviews

- Settings

Use SF Symbols for icons.

## 8. AppBot Review UI

Design a view to display AppBot reviews:

- Group by sentiment (positive, neutral, negative)

- Add search and filter bar

- Use List with custom cells showing rating, text, and platform.

## 9. Auto-Create Jira Ticket (Preview Feature)

Create a CrashMonitor class:

- Detect resurfaced crashes (not seen in last 30 days)

- Extract GIA (Glassbox Insights API)

- Simulate creating a Jira ticket with title, summary, timestamp.

## 10. GitHub Actions CI/CD

Create a GitHub Actions workflow:

- Trigger on push to main

- Build and test iOS app

- Archive and upload to TestFlight using Fastlane.