# **FitGenius Deployment Documentation**

**Version:** 1.3  
**Last Updated:** May 2025  
**Deployment Type:** Internal (Xcode)

## **1. Overview**

This document provides step-by-step deployment instructions for the FitGenius iOS app. The app is currently not available to the public and is deployed through Xcode for internal testing purposes.

## **2. System Requirements**

### **Hardware**

* macOS device with Xcode installed
* iPhone 13 or newer for testing

### **Software**

* macOS Ventura or later
* Xcode (latest stable version)
* iOS 17.6 or later on testing devices

### **Accounts**

* Apple Developer account (free or paid)
* Firebase account with access to Firestore & Authentication services

## 

## **3. Firebase Configuration**

1. Create a Firebase project.
2. Enable the following services:
   * Authentication (Email/Password)
   * Firestore Database
   * Realtime Database (optional for tracking)
3. Download the GoogleService-Info.plist file.
4. Add it to your Xcode project (drag into the root folder).

## **4. HealthKit Setup**

1. In **Xcode**, go to your app target > **Signing & Capabilities**.
2. Click **+ Capability** and add **HealthKit**.
3. In **Info.plist**, add usage descriptions:

<key>NSHealthShareUsageDescription</key>

<string>This app uses HealthKit to track your activity.</string>

<key>NSHealthUpdateUsageDescription</key>

<string>This app uses HealthKit to update your health data.</string>

## **5. Ready Player Me Integration (Avatar)**

* Include the SDK (via Swift Package Manager or manual import).
* Ensure webview-based avatar customization is accessible.
* Store unlocked items in user profiles via Firestore.

## **6. AI Chatbot Integration**

* Use OpenAI or similar API.
* Add environment variable or config for your API key.
* Chatbot input should be filtered to only handle fitness/nutrition queries.
* Store chatbot conversations in Firebase (Firestore).

## 

## **7. Deployment via Xcode**

1. Connect your test iPhone to your Mac.
2. Open the FitGenius project in Xcode.
3. Select the correct team and provisioning profile under **Signing & Capabilities**.
4. Choose your iPhone as the build target.
5. Run the app (Cmd + R).
6. On your iPhone, trust the developer profile: Settings > General > VPN & Device Management > Trust [Your Apple ID]
7. App is now installed and ready for internal testing.

## **8. Optional: Test Data Seeding**

* Use a Python or Node.js script to prepopulate Firestore with test users, workouts, and milestones.
* Avatar data should include locked/unlocked status fields.

## **9. Known Limitations**

* No TestFlight or App Store deployment
* HealthKit updates are asynchronous, not real-time
* Accessibility features not yet implemented

## **10. Future Deployment Considerations**

* Transition to TestFlight for broader testing
* Prepare for App Store review (add privacy policy, app metadata, etc.)