

2025 Swift Student Challenge Written Response

James Kong

1. Tell us about your app in one sentence. What specific problem is it trying to solve? Be concise. (50 words or less)

Home Gym is a fitness app that brings structure, motivation, and convenience to at-home workouts by using real-time body tracking to automatically monitor exercises, track progress, analyze trends, and provide goal-setting and achievement features, helping users stay engaged and consistent without needing a gym membership.

2. Describe the user experience you were aiming for and why you chose the frameworks you used to achieve it. If you used AI tools, provide details about how and why they were used. (500 words or less)

The user experience I envisioned for Home Gym seamlessly integrates technology into home workouts, making exercise more structured, engaging, and accessible. Fitness has always been a big part of my life, as I used to be a track athlete, running a 4:45 mile at my peak. But after developing patella tendonitis, everything changed. Ever since that injury, exercising hasn't felt the same, as I've had to abandon running. Over the years, I've missed the structure and feeling that training at the track facilities gave me. Home workouts never felt as effective because I didn't know which exercises to do, found it tedious to track my progress, and felt like I wasn't getting a "real" workout without gym equipment or big facilities. There's a certain energy and atmosphere in a gym that makes fitness feel more legitimate, and I wanted to replicate that experience at home. My goal with Home Gym was to break down these mental and logistical barriers by creating an app that not only guides users through structured workouts but also provides real-time tracking, progress visualization, and achievements, bringing the structure and motivational vibe of a gym experience into any space.

To achieve this, I primarily used SwiftUI for its declarative nature and streamlined UI-building capabilities. However, key functionalities, particularly camera integration, required bridging SwiftUI with UIKit. I implemented `UIViewControllerRepresentable` to wrap a custom `CameraViewController`, allowing seamless live video capture while integrating smoothly with SwiftUI. This enabled real-time body tracking using the Vision framework while maintaining SwiftUI's flexibility for overlaying UI elements.

For body tracking, I used the Vision framework to estimate key body landmarks, allowing Home Gym to recognize movement patterns and count reps automatically. I combined this with `AVCaptureSession` to process a live video feed while overlaying visual markers on the user's body using Core Animation. To further immerse users, I

added a voice guidance system with AVSpeechSynthesizer, offering real-time verbal feedback on form, rep counts, and encouragement, simulating a personal trainer.

Tracking progress was another crucial aspect of maintaining motivation. To address this, I integrated Swift Charts, providing users with a clear visualization of their workout trends. This allowed them to monitor improvements over time in an intuitive and engaging way, reinforcing consistency and goal-setting.

Since I am relatively new to Swift, I encountered challenges with concurrency and efficiently processing Vision framework outputs. I used AI tools to troubleshoot issues related to asynchronous processing, ensuring the app could handle real-time camera input and UI updates without race condition errors.

Ultimately, Home Gym became more than just a technical challenge, it was a way for me to reconnect with fitness in a new light. By integrating real-time body tracking, structured guidance, and progress visualization, I aimed to capture the structure and motivation I once found in training. Developing this app pushed me to explore new technologies and problem-solve in ways I hadn't before, but more importantly, it reminded me of why I love movement and how technology can enhance that experience.

3. Beyond the Swift Student Challenge

If you've used your coding skills to support your community or an organization in your area, let us know. (200 words or less)

Beyond the Swift Student Challenge, I've used my coding skills to support my community by leading projects that enhance accessibility and engagement. As the Lead Mobile App Developer for Branda, a student-built app unifying essential Brandeis University campus services like shuttle tracking, laundry availability, and dining hours, I stepped up during a critical transition period when many team leads were graduating. Despite the uncertainty, I took initiative, contributed key features, and was promoted to team leader. Under my leadership, Branda continued to serve over 1,600 daily users with continuous updates, improving campus life by making important resources more accessible.

Additionally, I developed a Satisfaction Analyzer during DeisHacks 2022, a tool designed to assess sentiment in feedback for the Clippership Foundation. By analyzing user responses, it helps nonprofits better understand and address community needs. These experiences reinforced my passion for using technology to create meaningful, real-world impact.

4. Apps on the App Store (optional)

If you have one or more apps on the App Store created entirely by you as an individual, tell us about them. This won't influence the judging process. (200 words or less)

N/A

5. Social Media (optional)

If you'd like to share links to your website or social media, please add them below. (200 words or less)

Personal Website: <https://jameskong098.github.io/>

LinkedIn: <https://www.linkedin.com/in/jamesdemingkong/>

6. Comments (optional)

Is there anything else you'd like us to know? (200 words or less)

I have only tested the app on M1 iPads, so the initial orientation may not load correctly on M4 iPads due to the new camera position in landscape mode. If any issues arise, unlocking rotation and rotating the iPad should resolve them.

There are a few known bugs, such as tips not always appearing at the right time and Swift Playgrounds sometimes requiring the app to be rebuilt twice before it starts running properly. Additionally, the behavior of Swift Playgrounds differs from Xcode App Playgrounds and sideloaded preview apps, leading to inconsistencies in animations, alert timing, and other unexpected issues. I have done my best to mitigate these problems.

I originally wanted to integrate a CoreML classifier model for more accurate exercise counting but faced difficulties importing .mlmodel files into the Swift Playground environment.

For reference, sample data has been pre-populated within the app. If you would like to start fresh, you can delete the generated data using the developer tools in the settings.