Project Context

Rather than rushing into a complex idea, I chose to start with a controlled prototype to learn how real systems fail and how to take responsibility for fixing them, and understand the full lifecycle of turning a web app into a mobile application, including deployment, debugging, and real-device testing, before attempting a more complex project.

Debugging & System Failures I Encountered

I built the first version of my app using Base44 as a web app. Later, I converted it into an Android APK using a WebView wrapper. After installing the app on my phone, it did not open properly and kept crashing at launch. The error messages made it look like something was wrong with Android settings or libraries, not with my app.

At first, I tried fixing the problem by checking error messages, rebuilding the app, and changing some configuration files. None of these steps worked. The app still crashed, and the errors were confusing. The Android emulator was not helping much, so I decided to test the app directly on my real Android phone.

To do this, I connected my phone to my Mac and used the terminal to mirror the phone screen. I used AI to help me run the required terminal commands so the mirroring would work. This allowed me to see exactly what was happening when the app tried to open on the phone.

After spending time focused on Android-related errors, I stepped back and checked how the web app was connected to the Android app. I then found the real problem. While converting the web app into an APK, I had entered the web app URL incorrectly. I forgot to include the ".com" part of the domain. Because of this, the Android app was trying to load the wrong link and failed immediately.

Once I fixed the URL, the app worked normally.

This experience taught me that even very small mistakes, especially where different systems connect, can cause big and confusing problems. Now, when I debug, I first check basic setup details and connections instead of only trusting error messages.

What I would do differently:

- Next time, I would first check simple setup mistakes before assuming the problem is deep or technical.

- I would try to understand where the problem is coming from instead of assuming everything is an Android issue.

- I would also double-check basic configuration details like URLs before converting a web app into an APK.

- I would slow down and question my assumptions instead of immediately trusting error messages.

What I learned from this project:

- I learned how to connect my Android phone to my Mac and mirror the screen so I could see what was actually happening when the app launched.

- I learned how to debug an app using a real device instead of relying only on the emulator.

- I learned the very basics of reading logcat errors and understanding when an app crashes early.

- I learned how the backend of an app supports the frontend, and how problems in setup can break the entire app even if the code looks fine.

- I also learned that trying to convert an app written in JavaScript and JSON directly into Python does not work well. When I tried this, the result was a very messy app with a poor user interface, which taught me that tools and languages need to match the problem instead of being forced.