

Acani Internship Information

1. **Main Projects**

- iPhone App
- API Server

2. **Learn by Doing**

I will assign you tasks and give you time to really do an Amazing job and have fun with each of them. Also, things, especially programming tasks, tend to take longer than expected. So, take your time. They say you can have two of Good, Cheap, and Fast but not all three.

You're free to do whatever you'd like (within reason, morality, and the law, of course; and the terms of the [Acani Internship Agreement](#)) to complete the assignments to the best of your ability. This is not like school where you must abide by an [educational honor system](#). There's no such thing as "cheating" here. Actually, I encourage you to seek aid as needed. Use [Google](#), pay someone to assist you, or even ask me questions.

If you have questions after first referencing the docs and then googling, the best way to ask me is by email. If they're generic questions, then please ask them generically on [Stack Overflow](#). (There, you'll most definitely find the answers to many of the programming questions you have.) Then, feel free to email me the link to your question. This will help you & others get better & faster answers to your questions.

3. **Knowledge Share**

I will share valuable information with you to support you in completing the assignments and so that you learn and improve at designing and engineering software. I'll also make myself available a couple times per month to speak with you over the phone.

4. **Code Reviews**

Doing [code reviews](#) and [refactoring](#) will give you a great opportunity to learn and get even better at programming. The main goal of going back and perfecting things is to hone your skills & understanding even further.

Of course, it will also improve your code, but that's secondary. Still, code reviews are good practice. They're something the best engineering teams do. They have at least one other person review their code and then make any changes they like that were suggested. Cleaning up code is called refactoring, and it's very important, especially as the project grows. So, even just learning this code-review process is a great learning experience.

This review practice will also develop your ability to take feedback. Just remember: Any feedback I give you is not to say that you didn't do an outstanding job. Actually, I'm sure you will. Think of it mainly like a great opportunity to learn and get even better at programming. So, the main goal of this going back and cleaning things up/perfecting things is to hone your skills & understanding even further. Mastery. To prepare you.

Also, when people see your code, they'll think it was written by a master iPhone developer. And, when you look back at your code months from now, you'll even amaze yourself.

5. Ownership

I will give you guidance & direction. That way, you can focus on the implementation. But, if you have other ideas you'd like to experiment with, go for it! Because, (1) I want you to feel a sense of autonomy & ownership of your work, and (2) I'd be micromanaging you if I tried to manage all the implementation details anyway.

I do ask that you listen to, understand, and consider all the feedback I give you, asking why (maybe first to yourself and then to me if you don't readily understand the reasoning behind my suggestions. I promise to explain why for every piece of feedback I give.) Then, from a place of understanding, feel free to do what you choose, what you know is best. Because, even though I have general iOS experience, I'm still learning too, just like you, and you'll have more specific experience with the task at hand (because you'll be the one implementing it). Also, sometimes, my feedback might not even be the best solution, for any reason at all.

So, I'll give you feedback, and feel free to take what suits you. And, if one of my suggestions doesn't feel right to you, feel free to leave it. No explanation necessary. If there's something I feel strongly about that I notice you chose to do differently, I may follow up with you on it to understand your reasoning for doing it the way you did.

I might make requests that seem ridiculous to you at the time, perhaps even tedious or trivial. Just be aware if you feel any resistance coming up. This may just be something to work through and may not necessarily mean that you shouldn't implement the request I'm making, but it also may not mean that you should. Such experiences will develop your ability to distinguish between your resistance & intuition. You'll learn to trust yourself in making the best choice given all the information. For example, watch these [lessons from Karate Kid](#), a movie I highly recommend.

Accepting my feedback might be especially challenging because I have high standards and believe in thoroughness. I highly recommend the documentary [Objectified](#) for intelligent design and the book [Don't Make Me Think](#) for usability design & testing.

In the end, the users of the app will give the best feedback. Usability testing is an excellent way to get great user feedback. Perhaps, the users are the true owners.