Fetch Mobile Take Home Project

Thank you for your interest in Fetch! This take home project reflects the kind of challenges you'll encounter while working at Fetch.

This project takes approximately 4-5 hours to complete so please be mindful of how you allocate your time.

Overview

Your task is to build a recipe app that displays recipes from the provided API endpoint.

At a minimum, each recipe should show its name, photo, and cuisine type. You're welcome to display additional details or sort the recipes in any way you see fit.

The app should allow users to refresh the list of recipes at any time, and you can decide how to implement this in the UI. You're free to include any additional UI elements you think would enhance the experience, but the app should consist of just one screen displaying the list of recipes. Ensure efficient network usage when displaying the recipe list. Load images only when needed in the UI to avoid unnecessary bandwidth consumption.

The recipe data should not be stored on disk. Load the recipes only on app launch or when the user manually refreshes the list. Do not load recipes unintentionally. For images, cache them to disk to reduce network usage. You can either use an open-source image caching library or implement your own solution, but do not rely on HTTP caching for this.

We believe that unit tests are essential. As part of your take-home project, include unit tests to demonstrate your approach to testing. Use your judgement to determine what should be tested and the appropriate level of coverage. We're primarily focused on seeing solid unit tests so you can skip UI and integration tests for this exercise.

Details

Endpoints

The following JSON endpoints provide the data for your project.

- **All Recipes**: https://d3jbb8n5wk0qxi.cloudfront.net/recipes.json

You'll also find test endpoints to simulate various scenarios.

- **Malformed Data**: https://d3jbb8n5wk0qxi.cloudfront.net/recipes-malformed.json
- **Empty Data**: https://d3jbb8n5wk0qxi.cloudfront.net/recipes-empty.json

If a recipe is malformed, your app should disregard the entire list of recipes and handle the error gracefully. If the recipes list is empty, the app should display an empty state to inform users that no recipes are available.

Show Off Your Skills

We're excited to see what you excel at and how your unique perspective can contribute to our team. Use this take home project as an opportunity to demonstrate your strengths. Here are a few areas where you can shine:

- **Architecture**: How do you approach structuring a scalable, maintainable codebase?
- **Concurrency**: Show us your expertise with Swift Concurrency and how you handle complex async tasks
- **UI/UX**: How do you bring a clean, intuitive interface to life?
- **Performance Optimization**: What techniques do you use to create high-performing applications?

Feel free to focus on one area you're passionate about or cover multiple, depending on what best showcases your abilities. We want to see what makes you an outstanding iOS developer!

Include a README

Below is a README template to help us understand your approach and decision-making process. Please fill out the following sections as part of your submission and include the README in your project.

```
# Steps to Run the App

# Focus Areas: What specific areas of the project did you prioritize? Wh

# Time Spent: Approximately how long did you spend working on this proje

# Trade-offs and Decisions: Did you make any significant trade-offs in y

# Weakest Part of the Project: What do you think is the weakest part of

# External Code and Dependencies: Did you use any external code, librari

# Additional Information: Is there anything else we should know? Feel fr
```

Submitting the Final Project

When completed, upload your project to a public repository and submit a link to the project through Greenhouse.

FAQ

What are you looking for?

We want to know what production-ready code looks like to you. If you had to make some compromises or shortcuts for the sake of time be sure to list them in the README.

How long should I work on this?

Most candidates spend 4-5 hours working on the take home. Be mindful of how you manage your time.

Can I use ChatGPT or other AI code generation tools?

Yes, you can use any AI code generation tools you wish. Fetch uses machine learning and AI throughout our organization.

Can I use third-party dependencies?

Yes, feel free to include third-party frameworks and libraries if you think they're essential. Please include the list of dependencies in the README.

What minimum versions should I support?

You can decide what minimum version to support. At Fetch, we support iOS 16 and up.