Fetch - iOS Coding Challenge

Please write a native iOS app that allows users to browse recipes using the following API:

https://themealdb.com/api.php

There are 2 endpoints that your app should utilize:
- https://themealdb.com/api/json/v1/1/filter.php?c=Dessert for fetching the list of meals in the
  Dessert category.
- https://themealdb.com/api/json/v1/1/lookup.php?i=MEAL_ID for fetching the meal details by its
  ID.

The user should be shown the list of meals in the Dessert category, sorted *alphabetically*.

When the user selects a meal, they should be taken to a detail view that includes:
- Meal name
- Instructions
- Ingredients/measurements

Please read the following guidelines carefully before starting the coding challenge:
- Be sure to filter out any null or empty values from the API before displaying them.
- UI/UX design is *not* what you're being evaluated on, but the app should be user friendly and
  should take basic app design principles into account.
- Project should compile using the latest version of Xcode.

FAQ

How do I submit my exercise?

Any suitable means of delivering the exercise where it says: Submit Here. Please provide the git link to a public repository that contains your code.

How will this exercise be evaluated?

An engineer will review the exercise you submit. At a minimum, the app must provide the expected results. You should provide any necessary documentation within the repository. While your solution does not need to be fully production ready, you are being evaluated, so put your best foot forward.

I have questions about the problem statement.

For any requirements not specified, use your best judgment to determine expected results.

Can I provide a private repository?

If at all possible, we prefer a public repository because we do not know which engineer will be evaluating your submission. Providing a public repository ensures a speedy review of your submission. If you are still uncomfortable providing a public repository, you can work with your recruiter to provide access to the reviewing engineer.

How long do I have to complete the exercise?

Please complete this exercise within 1 week if possible, but you can take more time if needed.

Can I build the UI of the exercise in either UIKit or SwiftUI?

SwiftUI is preferred. We understand not everyone has experience with SwiftUI, so a UIKit implementation bundled with your enthusiasm to learn SwiftUI works great, too!