**Coding Assignment**

Implement a small iOS app according to the following requirements:

1. Choose any architecture you feel comfortable with, but assume the app will grow in size and complexity (features) over time.
2. Use either UIKit or SwiftUI. There are no Deployment Target constraints.
3. Using base internationalization, the app will support English language, and another language of developer's choice.
4. The app name is "Demo".
5. The app theme color is RGB(77, 116, 217).
6. There should be a launch screen displaying the app name in white, against a solid background of the theme color.
7. The 'home' screen should display a table view embedded in a navigation view. The navigation title should be the app name. The navigation bar should be white, with the control tint color using the app theme color.
8. The data displayed on the table will eventually be retrieved from a web service, but for the time being you will mock that and put the static, dummy data provided in point #12 behind a mock (view model, data provider, etc.)
9. The table view should display the outline of each item: name and price. Choose the format and layout freely.
10. On tapping each table cell, the navigation should push a "detail screen" for the selected item, with the item name as the navigation title. The item price should be displayed under a label titled "price" (12pt), in large characters (24pt) near the center of the screen, and the keywords displayed underneath in some suitable format, under a "keywords" heading. All unspecified layout and design details are left to the developer's choice.
11. Although the data is a dummy immediately available on the client for now, eventually it will be retrieved asynchronously, so implement the data flow around that assumption. By the time the table is displayed, all the data is assumed downloaded, but between launch and initial display of the table, there might be some lengthy netwroking involved.
12. The dummy data is:

{

"items": [

{

"name": "First Item",

"price": 100,

"keywords": [

"red",

"versatile",

"inexpensive"

]

},

{

"name": "Second Item",

"price": 150,

"keywords": [

"yellow",

"interesting",

"available"

]

},

{

"name": "Third Item",

"price": 190,

"keywords": [

"green",

"doorway",

"out-of-stock"

]

}

]

}

**Plus points if your project have:**

* SwiftUI/Combine
* Unit Tests

**Plus points if you don't use:**

* Third party libraries
* Singleton

**What We're Looking For**

* An app that meets all of the functional requirements above.
* Your coding style - show us how you like to write your code.
* Architecture - how you've structured your code.
* Principles - how you believe code should be written.
* Quality - how you guarantee your code is functioning correctly.