Problem Statement

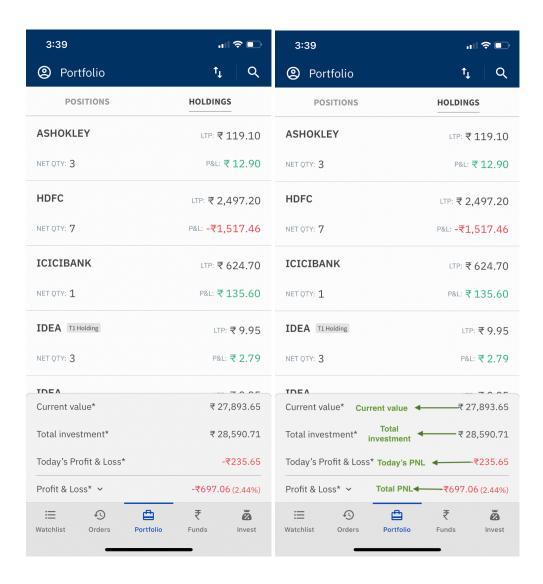
Objective:

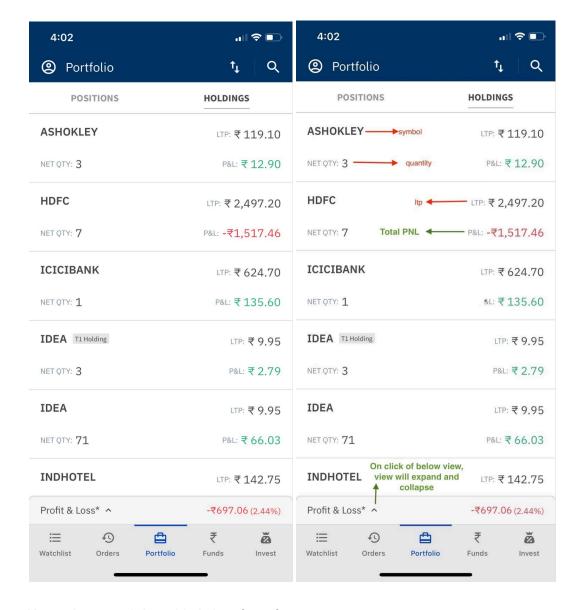
Create a list of holdings (stocks invested) and populate the list as per the below screenshots. Create an expanded & collapsed state UI for showing the summary of your portfolio

Endpoint to fetch data: https://35dee773a9ec441e9f38d5fc249406ce.api.mockbin.io/

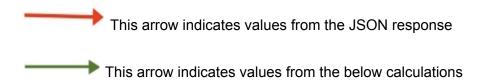
Points to be considered for evaluation:

- 1. Architecture (preferably Clean or MVVM) that can be scaled and used as a production codebase.
- 2. Responsibility segregation in modules
- 3. Specific to iOS: Clean UI (with no constraint warnings, preferably develop UI through code)





Note - Ignore tab bar, this is just for reference



Calculation:

- 1. Current value = sum of (Last traded price * quantity of holding) of all the holdings
- 2. Total investment = sum of (Average Price * quantity of holding) of all the holdings
- 3. Total PNL = Current value Total Investment
- 4. Today's PNL = sum of ((Close LTP) * quantity) of all the holdings

Guidelines for Android

Acceptance Criteria:

Your selection chances are higher if

- 1. You use the latest version of Kotlin.
- 2. You create UI Programmatically using XML. But, you are free to use composeUI.
- 3. You adhere to Google Material Guidelines and coding conventions.
- 4. Your app performs well in terms of CPU, memory and energy usage.
- 5. Your code is loosely coupled, testable and uses an architecture similar to MVVM, MVI, MVP.
- 6. You tastefully customise the UI and add animations etc.
- 7. Your app does not have too many dependencies on third party frameworks.
- 8. Your app runs smoothly offline.
- 9. Try to adhere to SOLID principles as much as possible.
- 10. Cover as much as corner cases and handle error scenarios.

Note: We may reject your application if

- 1. Your project fails to compile or compiled with warnings with the latest version of Android Studio.
- 2. If min sdk is below 21.
- 3. Your app does not layout correctly for all Android devices.
- 4. Your app has Ul/logical flaws that make it difficult to use.
- 5. You do not have unit tests in your application with decent code coverage.
- 6. Your unit tests fail.

Guidelines for iOS

Acceptance Criteria

Your selection chances are higher if

- 1. You use the latest version of Swift.
- 2. You create UI Programmatically using UIKit Framework
- 3. You adhere to Apple's Human Interface Guidelines and coding conventions.
- 4. Your app performs well in terms of CPU, memory and energy usage.
- 5. Your code is loosely coupled, testable and uses an architecture similar to VIPER, MVVM, MVP, etc.
- 6. You tastefully customise the UI and add animations etc.
- 7. Your app does not have too many dependencies on third party frameworks.
- 8. Your app runs smoothly offline.

Note: We may reject your application if

- 1. Your project fails to compile or compiled with warnings with the latest version of Xcode.
- 2. If you use Swift UI
- 3. Your app does not layout correctly for all iOS devices that can run the current release of iOS.
- 4. Your app has UI/logical flaws that make it difficult to use.
- 5. You do not have unit tests in your application with decent code coverage.
- 6. Your unit tests fail.
- 7. If auto layout engine throws constraints error during runtime.