iOS Take-Home Challenge

Part 1 – View Model

Implement a method in ExerciseViewModel to populate the exercises property with exercises fetched from the API client.

Part 2 – Exercise List

Using **UIKit** and UICollectionView, implement a list of exercises in ExerciseListViewController.

Each cell should show the exercise's name, its thumbnail, a list of muscle groups, and a list of equipment.

Implement the list as follows:

- Load data from the view model to populate the list. Make sure to update the list when the data in the view model changes, using Combine observation.
- Use compositional layouts (with the list layout configuration) and diffable data sources to implement the list, instead of classical data sources and flow layouts.
- Use programmatic Auto Layout (NSLayoutConstraint) to layout the list and its cells.
- Use regular UIKit cell configurations, not UIHostingConfiguration.
- Feel free to use a third-party dependency like Kingfisher or Nuke to handle thumbnail image loading.
- Push the SwiftUI exercise detail view for an exercise to the navigation stack when its row is tapped.

Part 3 – Exercise Detail

Using **SwiftUI**, implement a detail view for an exercise.

The detail view should show:

- A video player, only for the first video in the exercise videos array
- A section for the exercise's muscle groups
- A section for the exercise's equipment
- A collapsible section for the exercise's description, if it exists

See the included example screenshots for reference.