Armut Study Case Bonus Questions

● What steps can you take to optimize the load time of photos? How about the scroll performance?

Since the application is generally used on small screens, photo resolutions can be optimized. Thus, although the download process from the internet varies according to the user's internet speed, it will be downloaded in the most optimal way. However, it is important that downloading photos and similar operations are done asynchronously within the “DispatchQueue.main.async” method offered by IOS so that the application does not freeze.

I used the widely used “SDWebImage” library in this project. It downloads the photo quickly from the URL and does not affect the program as scrolling performance.

● Can you handle failed API requests gracefully? Maybe redirect to a 'failed' UI state with a retry option?

If there is a failed API request in the homepage, an alert with two options "Retry" or "Cancel" is displayed to the user. When "Retry" is selected, the API request is sent again, when "Cancel" is selected, it does not take any action.

If there is a failed API request on the Service Details page, an alert with two options "Retry" or "Back" is shown to the user. The "Retry" option resends the API request, while the "Back" option returns to HomePage. It can be redirected to a different controller if desired.