SwiftStockPro

Stock market app build with MVVM on UIKit.

What does it do?

- Real time stock prices, charts and news.
- Save and track favorite stocks.
- Get smart insights on gainers / losers.
- Push notification
- Authentication with Firebase Auth

- API Integration
- Authentication
- Local Storage (Core Data)
- Real life updates
- Chart and animations
- Background Fetch

API of choice is FinancialModelPrep

Folder structure:

```
    Models
    ViewModels
    Views
    Networking
    Extensions
    Resources
```

Project development phases in details

Phase 1 - Authentication (Login / Signup screens)

Objective is to implement Firebase Authentication for email/password sign-up and login.
Steps to follow are:

- Setup Firebase Auth
- Design login/signup screens with basic validation
- Show appropriate error messages for failed login/signup
- Allow used to transition between screens (login - signup - forgot password)
- After successful login, move to Home Screen.

Other details:

The app will start in a splash screen, which will give basic information about the application. And it will animate changing from 1 to 2 and 3. (like it was a 3 step welcome screen but it will be only one and will be animated time based.)
This screen will include a next button which will send the user to the Sign up screen (if first time).
The cases for this are:

- First time user -> Sign up
- Not first time (but didn't activate remember me) -> Log in
- Not first time (but remember me active) -> Home Screen

Phase 2 - Main Screens & API Integration

Home Screen
Objectives for this phase are to design a home screen which will fetch and display data from the api using AlamoFire.
Steps to follow are:

- Design a clean looking HomeScreen,
- Fetch and display stock data
- Allow users to navigate to other screens like: Profile, Portfolio etc.

It will host components like: Search Bar, Stock List (table/collection view), NavBar.

Stock Detail Screen
It will display detailed information about the selected stock
Structure of the screen will be something like:

- Stock Info (Name, Ticker, Current Price, Image)
- Historical Price Chart
- News Feed (latest news related to the stock)
- Recommended stocks based on it (maybe)

Portfolio Screen
User can view and manage their portfolio of stocks.
It will list stocks that user choose to add to their portfolio.
And it will host different information like:

- Name, date bought, purchase price, current price, profit/loss.
- Add stock button (will allow the user to search a stock name to add it)
- Sort stocks based on name, gain/loss, etc.(maybe)

Stock Add/Edit Screen (portfolio management)
Allows users to add or edit stocks in their portfolio/
Stock selection search for stock, price they bought, date they bought, amount, save button.

Profile Screen
Blank screen for now

Settings Screen
Change different setting of the app.

Phase 3 - Portfolio feature

The goal for this phase is to design and make a functional portfolio screen.
Components of the screen.
User can manage their account information and settings.
Display information (Email, username etc.)
Change password (Update password through Firebase)
Logout button

Phase 4 - Code polish and user testing

The purpose of this phase is to clean up and refactor the code. Add comments and documentation. Fix bugs to ensure that everything is working smoothly

Phase 5 - Additional Features

Some of the extra features that i want to add are **Push notification, Watch list.**

ESTIMATED TIME FOR EACH PHASE (by me)

Phase 1 1 / 2 Weeks
Reasoning is that I firstly need to learn Firebase Auth since it's the first time I will use it. And I want to make sure to nail in the first time.

Phase 2 3 / 4 weeks
The API integration will take some time as I didn't work on a project for a time now, and I will need to go back to the past projects to get some help on it.

Phase 3 2 / 3 weeks
This is a crucial feature that involves both UI and data management. Storing the portfolio data and performing calculation might take time.\

Phase 4 1 / 2 weeks
That is just a time range giving more time as it is not known for sure how long it can take. This phase will ensure that the app is ready for the next major features.

Phase 5 3 / 4 weeks
This will add complex features which will take extra time to implement.