🛒 SHOPPING APP

👀 Summary

A shopping app for different categories and products.

You have to use Fake Store API to fetch products and categories.

Fake Store API: https://fakestoreapi.com/

⚙️ Features

🚀 Launch Screen

- The background should be the native color of the application.
- The icon of the application should be in the middle of the screen.
- There should be Activity Indicator under the application icon and the rotation animation should work.

🐣 Onboarding Screen

- There should be screens that inform the user about the features of the application.
- Using UIScrollView and UIPageControl on a single screen, these screens will be scrolled horizontally.
- While the background colors of the screens are the main colors of the application, there should be a description at the top of the screen and below the image.
- Buttons should be located in the lower right and lower left corners of the screen. The user should be able to navigate between the screens either by swiping or using the buttons.
- There should be no back button on the first screen, and the title of the forward button should be named "Finish" when it comes to the last screen. By pressing this button on the last screen, the user should be directed to the login screen.
- There should be another button in the upper right corner to skip these steps and it should direct the user directly to the login screen.
- This screen should be displayed only once.

🔑 Authentication

- Two different options for user login and registration should be presented to the user with UISegmentedControl.
- User login should come by default. Two UITextFields should be used, e-mail and password. “Sign In” should be written in the large font UILabel at the top of the screen.
- When the user registration tab is selected, four UITextFields should appear. The first field should be the username, the second field the email, the third field the password and the fourth field the password repeat information.
- The user should be registered to the system with Firebase Auth and the additional username information received should be saved on the Firebase Firestore with the uid value of the user created by Auth.
- When there are service requests, the screen should be slightly dimmed and the UIActivityIndicator should be displayed in the middle of the screen. When the service call is

finished, the user should be informed about the result of the operation with the UIAlertController.

- The user whose login is successful should be directed to the main screen of the application.
- For any changes that may occur in the session, the observer must be added to the screens in viewWillAppear and the observer must be deleted from the memory in viewDidDisappear.

🏡 Main Page

- UITabBarController should be used on the main screen and there should be three tabs: Products, Search and Profile.
- In the upper right corner of the screen, above the UINavigationBar, a button that will direct you to the cart and the total amount information should be displayed.
- When the Basket button is pressed, the Basket screen should be presented modally.

🧺 Basket

- The contents of the basket should be listed with UITableView and the number of products in the basket should be increased or decreased.
- At the bottom of the screen, there should be a button to complete the payment process and the total amount information should be next to the button.
- When the payment button is pressed, we should get a confirmation from the user with Alert and if the user continues to process, we should empty the cart and send a message with the Alert that the transaction was successful and direct the user to the main screen.

📦 Products

- All products should be displayed using UICollectionView on the Products screen. The UICollectionViewCell view should be a color gradient view with oval edges, an image in the background, black at the top and transparent at the top, and the product name and price information on the bottom (dark part) of this view.
- When the product is clicked, it should go to the detail screen. TabBar should not appear on the detail screen and detailed information about the product (description, score, etc.) should be included.
- There should be an "Add to Cart" button at the bottom of the screen and UIStepper should be used to set the quantity information when adding to the cart is done. Instant product quantity information should be displayed with a UILabel.

🔎 Search

- UISearchController should be used on the screen and products should be filtered when more than two characters are entered in the search box.
- UISegmentedControl should be placed under the search box to easily filter by categories.
- When you click on the product, the flow on the products screen should be followed.

👤 Profile

- User information should be included.
- There must be a button for the user to log out.
- When the logout button is pressed, Alert should be displayed for verification and if the user continues to log out, we must perform the operation and direct the user to the Authentication screen.

✅ Done Criteria

- Smooth running of the overall application flow. **(45 Point)**
  - Onboarding **(5 Point)**
  - Authentication **(10 Point)**
  - Main **(15 Point)**
  - Basket **(10 Point)**

- Using Firebase or Realm Database. **(10 Point)**

- Proper use of VIP architecture and writing clean code. **(10 Point)**

- Establishment of component-based architecture. **(5 Point)**
  - The inclusion of the network layer as a library in the project and the inclusion of service calls in the library.
- Test writing and error handling. **(5 Point)**

- Git Flow management. **(5 Point)**
  - Detailed commit entry. **(2 Point)**
  - Working with feature branches. **(3 Point)**

- Well organized Readme file. **(5 Point)**

- UI/UX **(5 Point)**

- Auto Layout usage. **(5 Point)**

- Cocoapods, Carthage or SPM usage. **(5 Point)**

- Memory management. **(5 Point)**