E-commerce Application

**Tech Stack:**

**Kotlin**: The app is entirely written in Kotlin.

**Retrofit**: Network requests in the app are handled using Retrofit, a type-safe HTTP client for Android and Java. Retrofit simplifies the process of interacting with web services by providing a straightforward way to parse JSON data, handle API calls, and manage network responses.

**MVVM Architecture**: The application follows the MVVM (Model-View-ViewModel) architecture pattern, which promotes a clear separation of concerns and ensures that the UI remains clean and easy to manage.

**Data Sources:**

Data for the application is dynamically fetched from the **DummyJSON API**, which provides mock data for development and testing purposes. The following endpoints are utilized within the app:

**API Endpoints Overview:**

1. Login Endpoint

Endpoint: @POST("auth/login")

Description: This endpoint is used to authenticate users. When a login request is made with the user's credentials, the server returns a `LoginResponse` containing the authentication token, which is stored in SharedPreferences for subsequent requests.

2. Get Categories

Endpoint: @GET("products/categories")

Description: This endpoint fetches a list of all product categories available in the app. The data is used to populate the Categories screen, allowing users to browse products by category.

3. Get User Details

Endpoint: @GET("user/me")

Description: This endpoint retrieves the details of the logged-in user, such as their name, email, and profile picture. The token passed in the header is used to authenticate the request.

4. Refresh Token

Endpoint: @POST("auth/refresh")

Description: When the authentication token expires, this endpoint is used to obtain a new token using a refresh token. This ensures the user remains logged in without having to re-enter their credentials.

5. Get Products by Category

Endpoint: @GET("products/category/{category}")

Description: This endpoint fetches products based on a specific category selected by the user. The category is passed as a path parameter in the request URL.

6. Get Top Rated Products

Endpoint: @GET("products?sortBy=rating&order=desc")

Description: This endpoint retrieves a list of top-rated products, sorted in descending order based on their rating. This data is displayed on the Home screen to showcase highly-rated products to users.

7. Get Product by ID

Endpoint: @GET("products/{id}")

Description: This endpoint fetches detailed information about a specific product, identified by its ID. It is used in the Product Details screen to display comprehensive product information to the user.

8. Search Products

Endpoint: @GET("products/search")

Description: This endpoint allows users to search for products by a specific query string. The search term is passed as a query parameter, and the response includes a list of products that match the search criteria.

**Design Reference:**

The app’s user interface design is inspired by a Figma reference: [Shop - E-commerce Mobile App](<https://www.figma.com/design/Su6SKKsc52hknpY3TyvzQ9/Shop---Ecommerce-Mobile-App-(Community)?node-id=1-11281&t=el4PPdzqhNJEI3IU-0>)

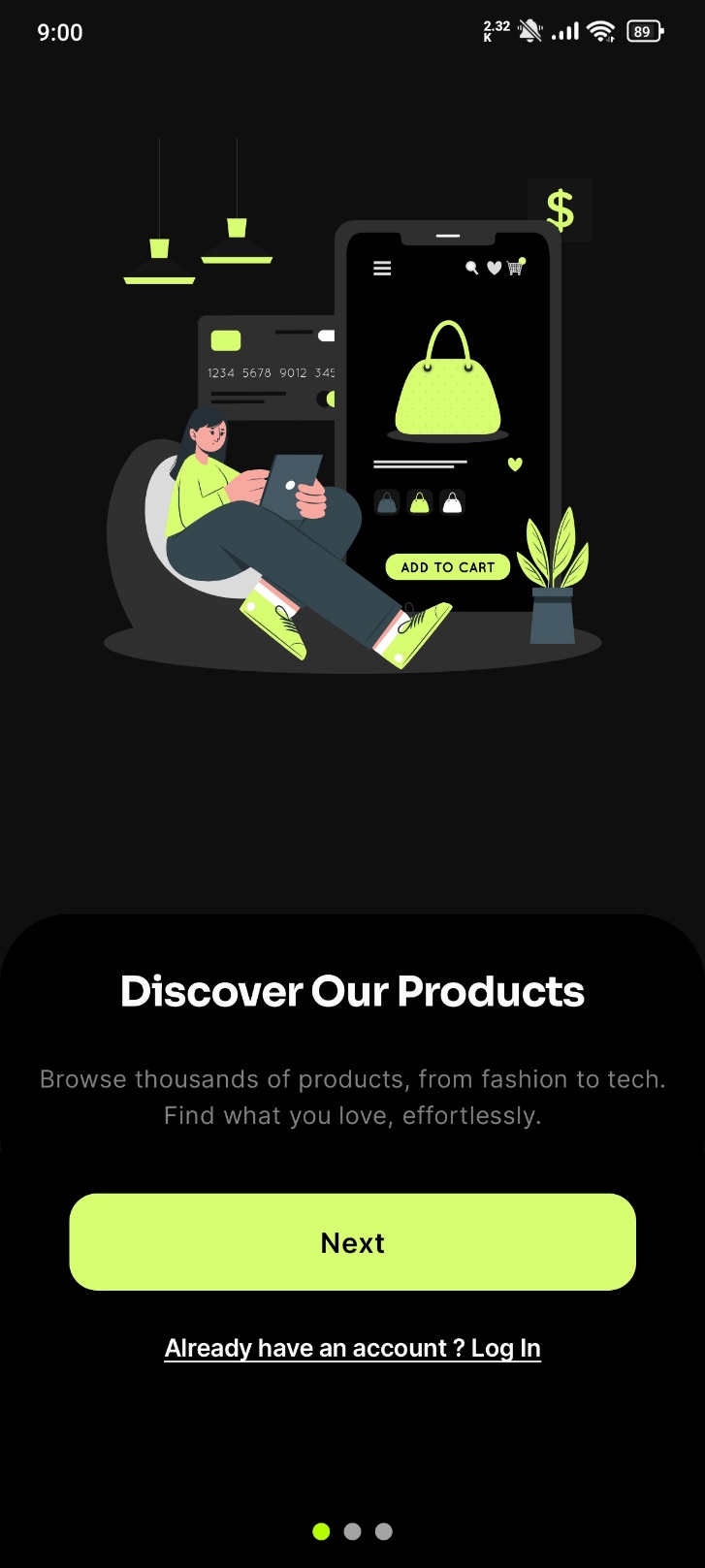
**Splash Screen:**

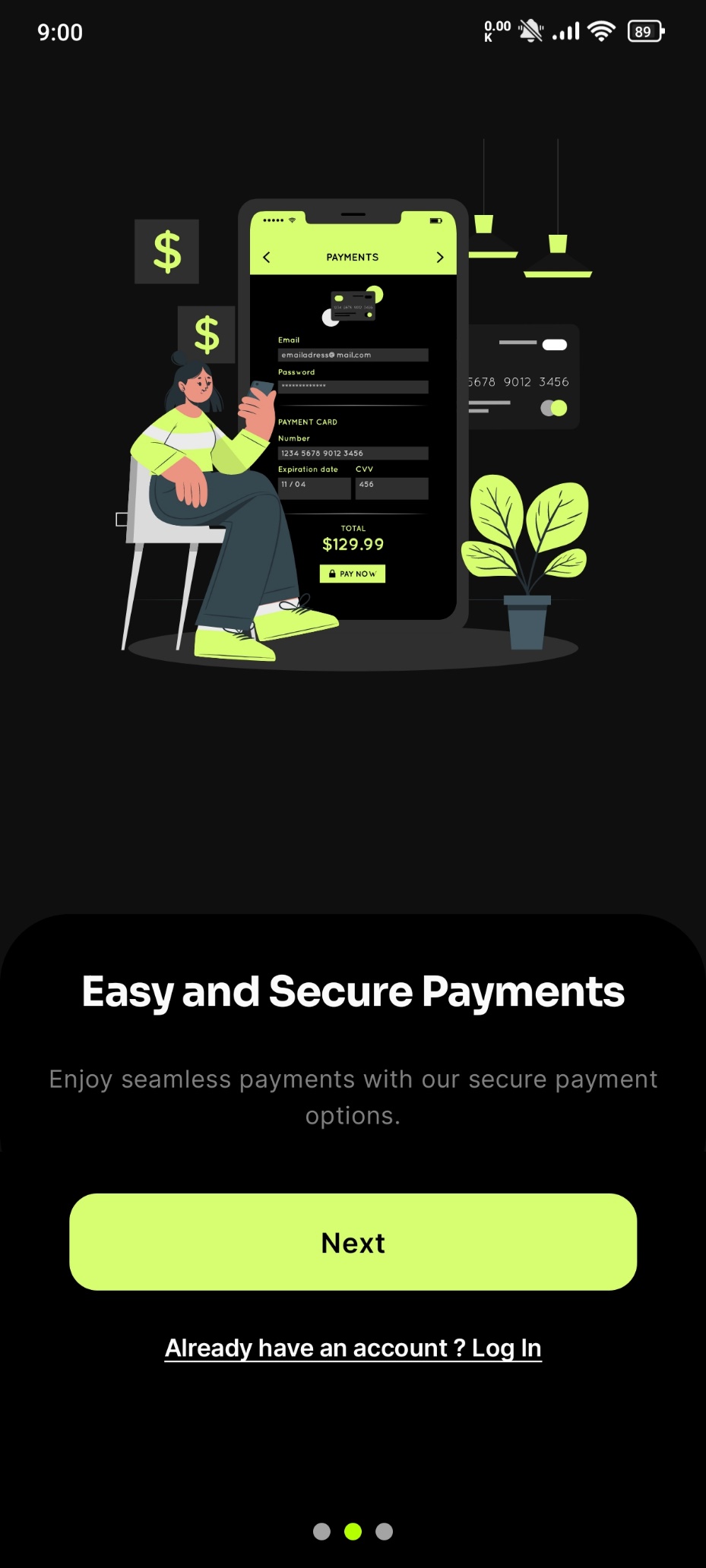
Made using Splash Screen API, this is backward compatible with both older and newer versions.

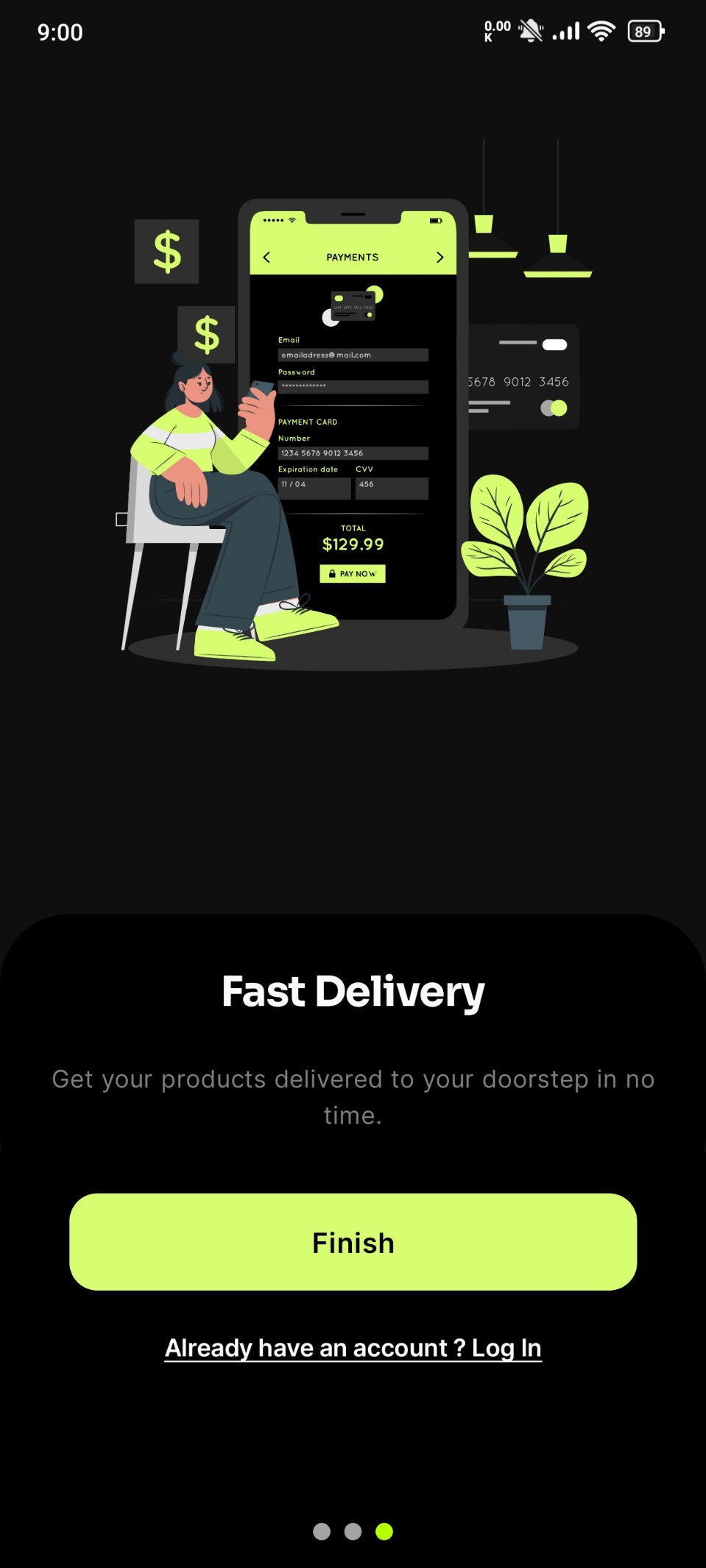


**Onboarding Screen:**

This is the onboarding screen, which is implemented using Fragments and a ViewPager with a tab layout indicator. The onboarding screen is only shown when the user first launches the app, providing an introductory experience that guides them through the app's key features. Once the user has completed the onboarding process, it will not be shown again in subsequent launches.

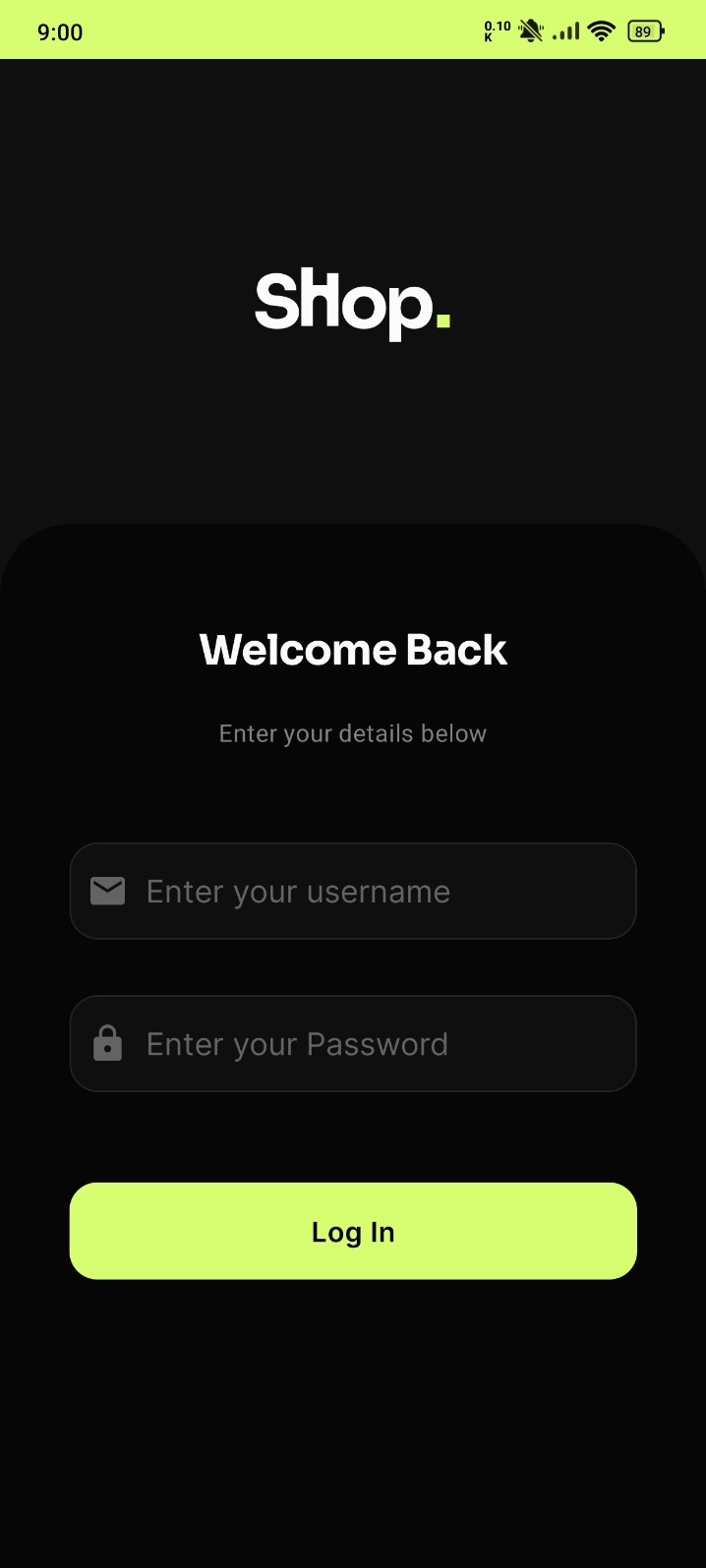


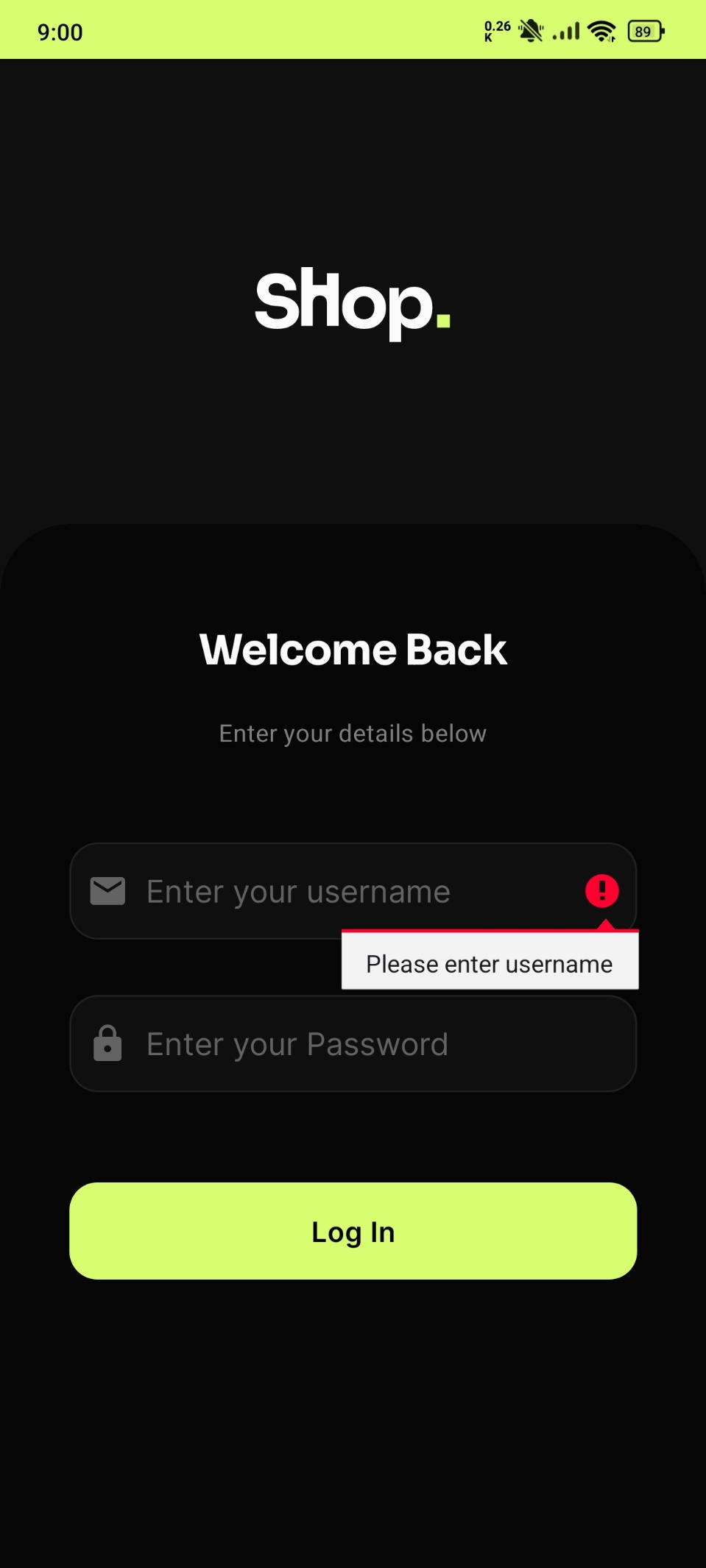


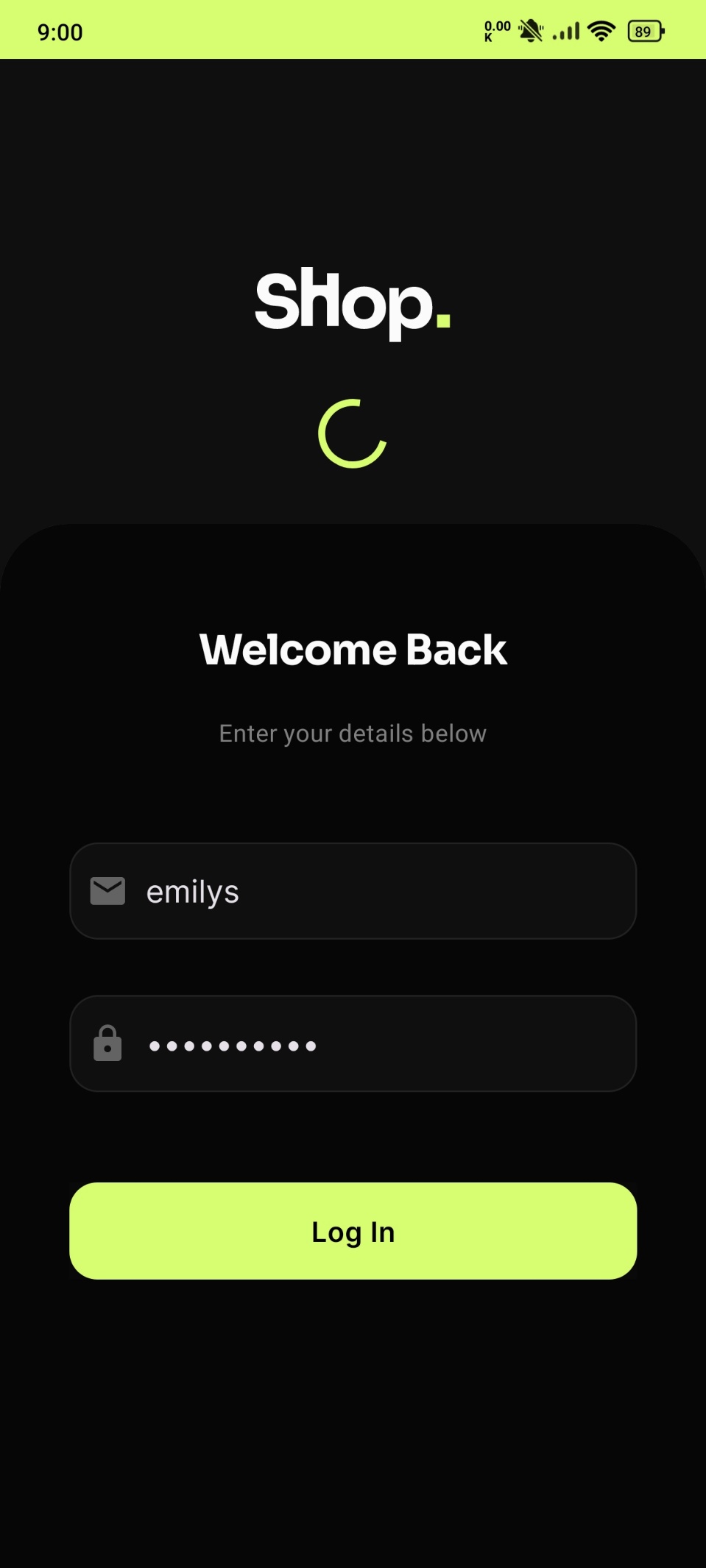


**Login Screen:**

The login screen is implemented using the 'dummyjson.com/auth' endpoint. All users available in the DummyJSON database can log in to the application. Upon successful login, the authentication token is stored in SharedPreferences. This token has an expiry time, and once it expires, a refresh token is used to obtain a new auth token, ensuring that the user doesn't have to log in repeatedly. The login process also includes input validation and a loading screen to enhance the user experience.





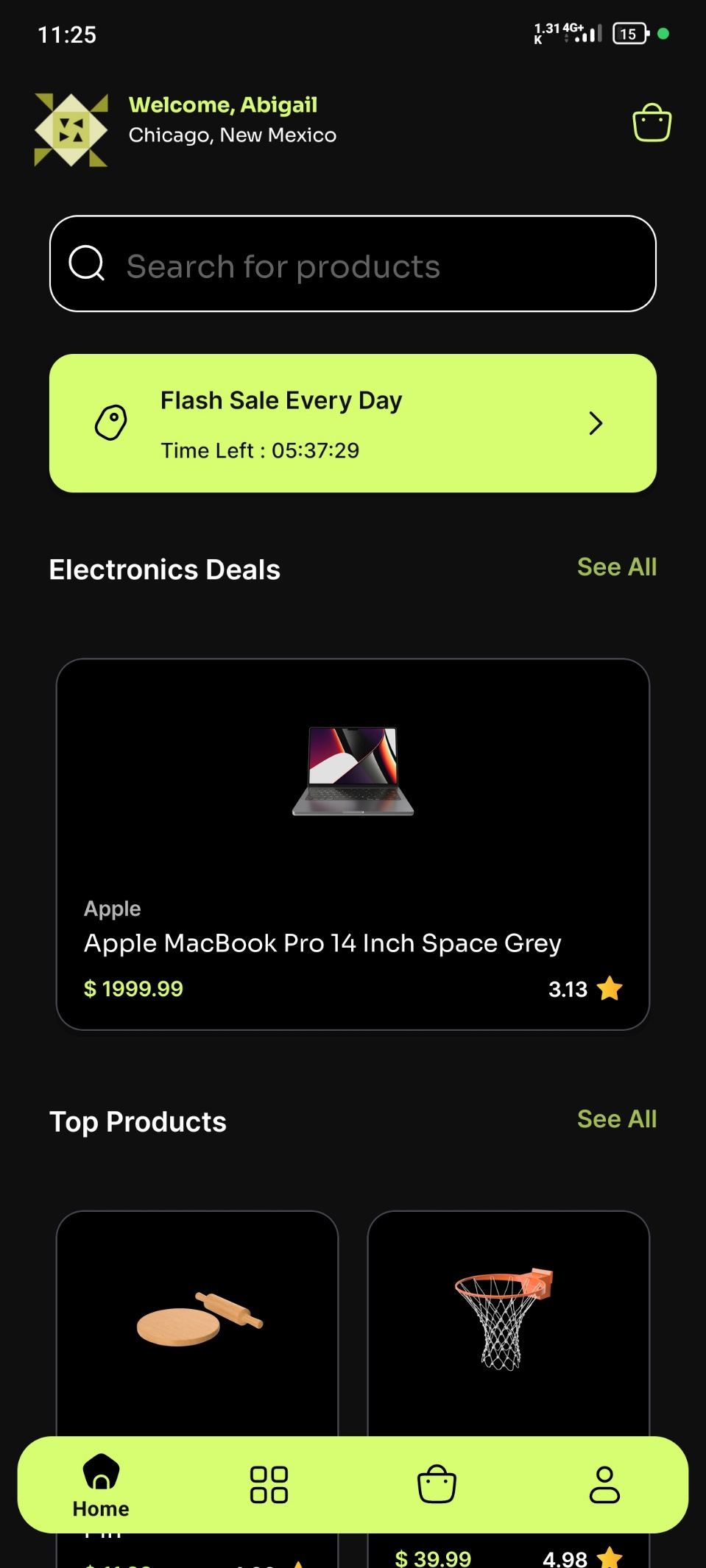


I have built four key screens using the Navigation Graph and Fragments for Home, Categories, Cart, and Profile. These screens are seamlessly connected, allowing for smooth navigation between them. The data displayed in these screens is fetched from the DummyJSON API, specifically from the `dummyjson/products` and `dummyjson/categories` endpoints.

**Home Screen:**

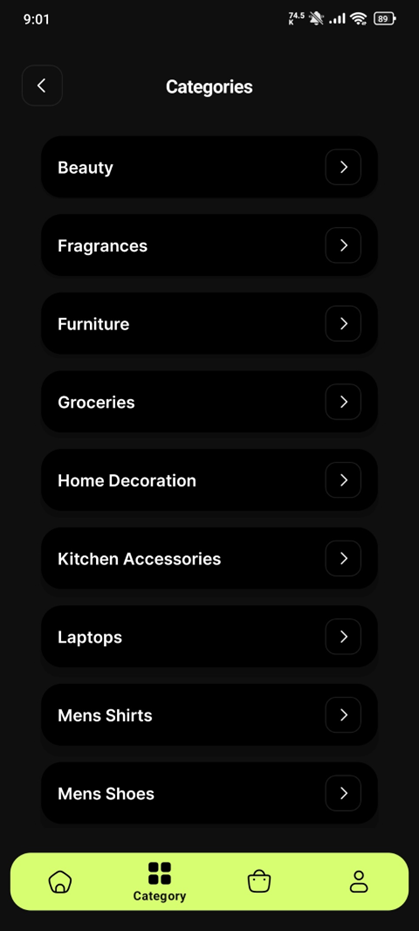
The Home screen is the main landing page of the app. It features a personalized welcome message that greets the user with their name, such as "Welcome, [username]," and displays their profile picture. This screen provides an overview of the top products and offers quick access to various features of the app. The data for the products displayed on this screen is fetched from the `dummyjson/products` endpoint. The Home screen is designed to be user-friendly and engaging, encouraging users to explore the app further.

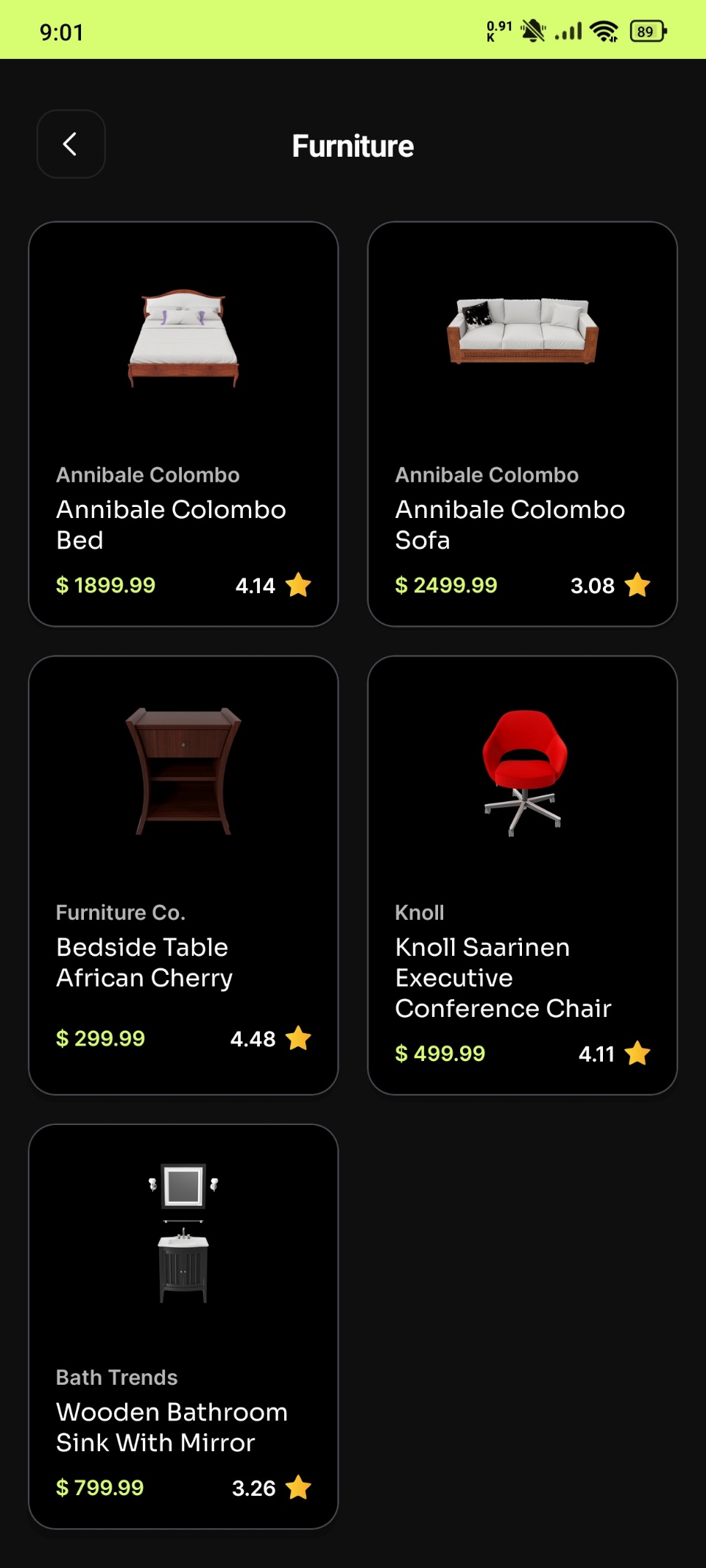
There is also a timer for the flash sale which will Run for every 6 hours and features a random product for that sale.



**Categories Screen:**

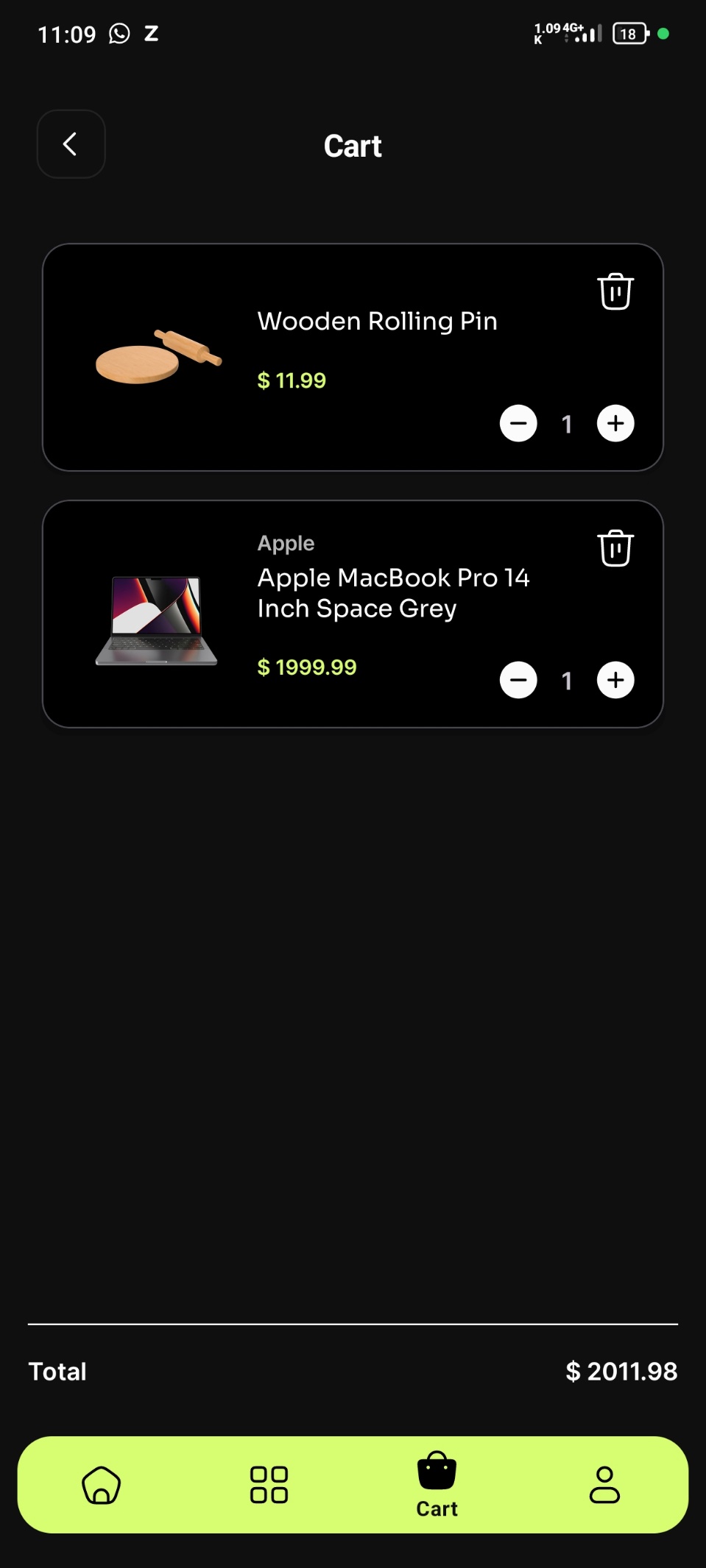
The Categories screen organizes products into different categories, making it easy for users to browse and discover items of interest. The data for the categories is retrieved from the `dummyjson/categories` endpoint. Each category is represented as a clickable item, which, when selected, navigates the user to a list of products within that category. This screen is structured to provide an intuitive and seamless browsing experience.





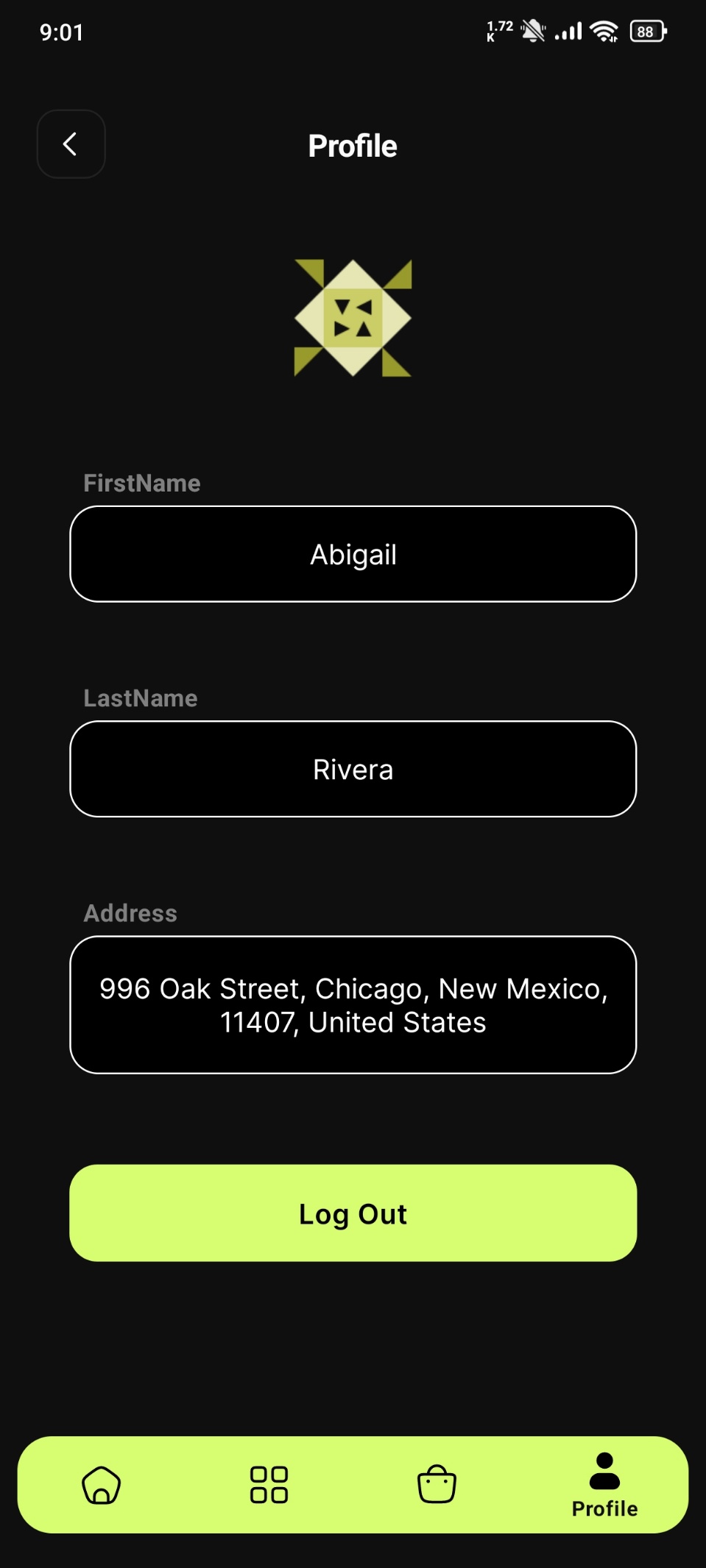
**Cart Screen:**

The Cart screen allows users to manage the items they have selected for purchase. It displays a list of products added to the cart, along with options to update quantities or remove items. The total price of the items in the cart is calculated and displayed at the bottom of the screen. This screen is designed for easy modification and review of selected products before proceeding to checkout.

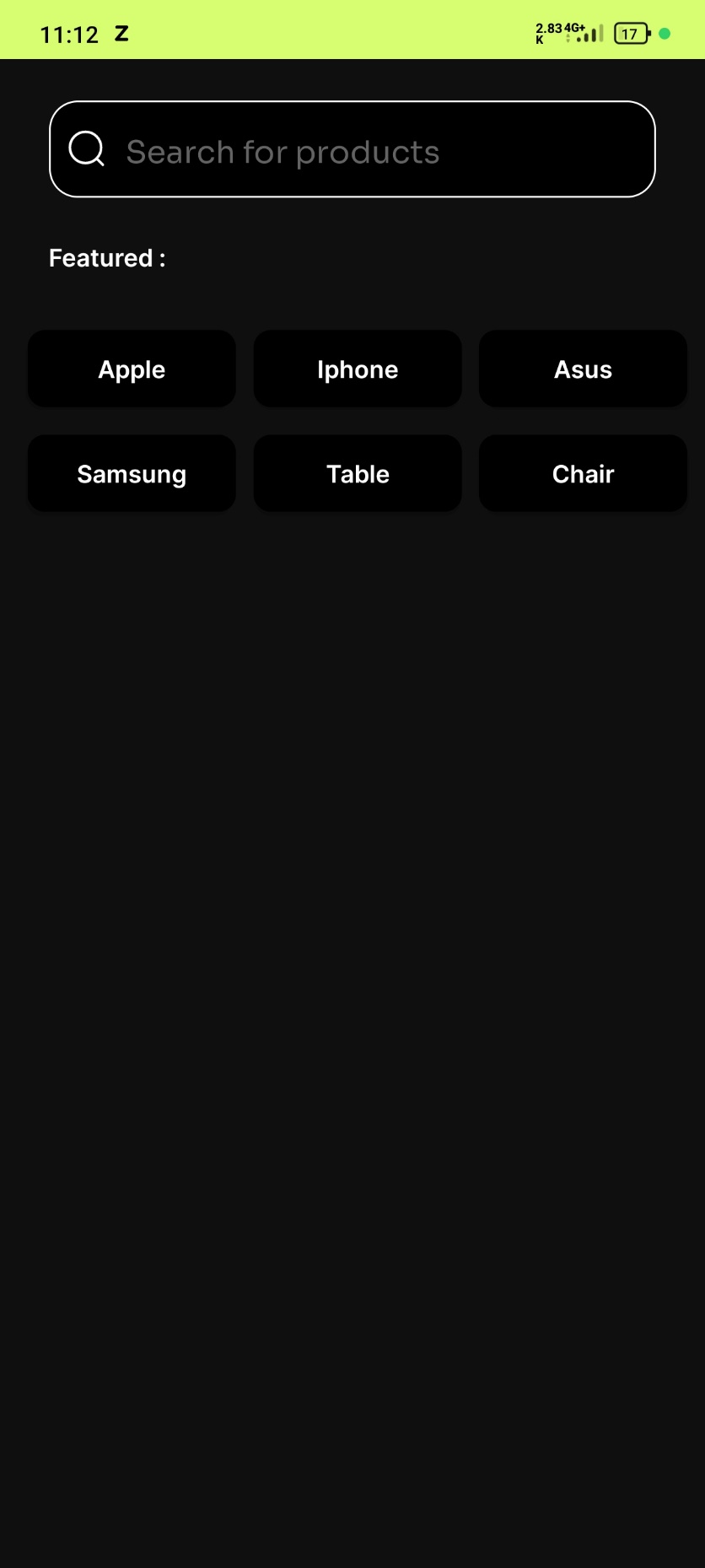


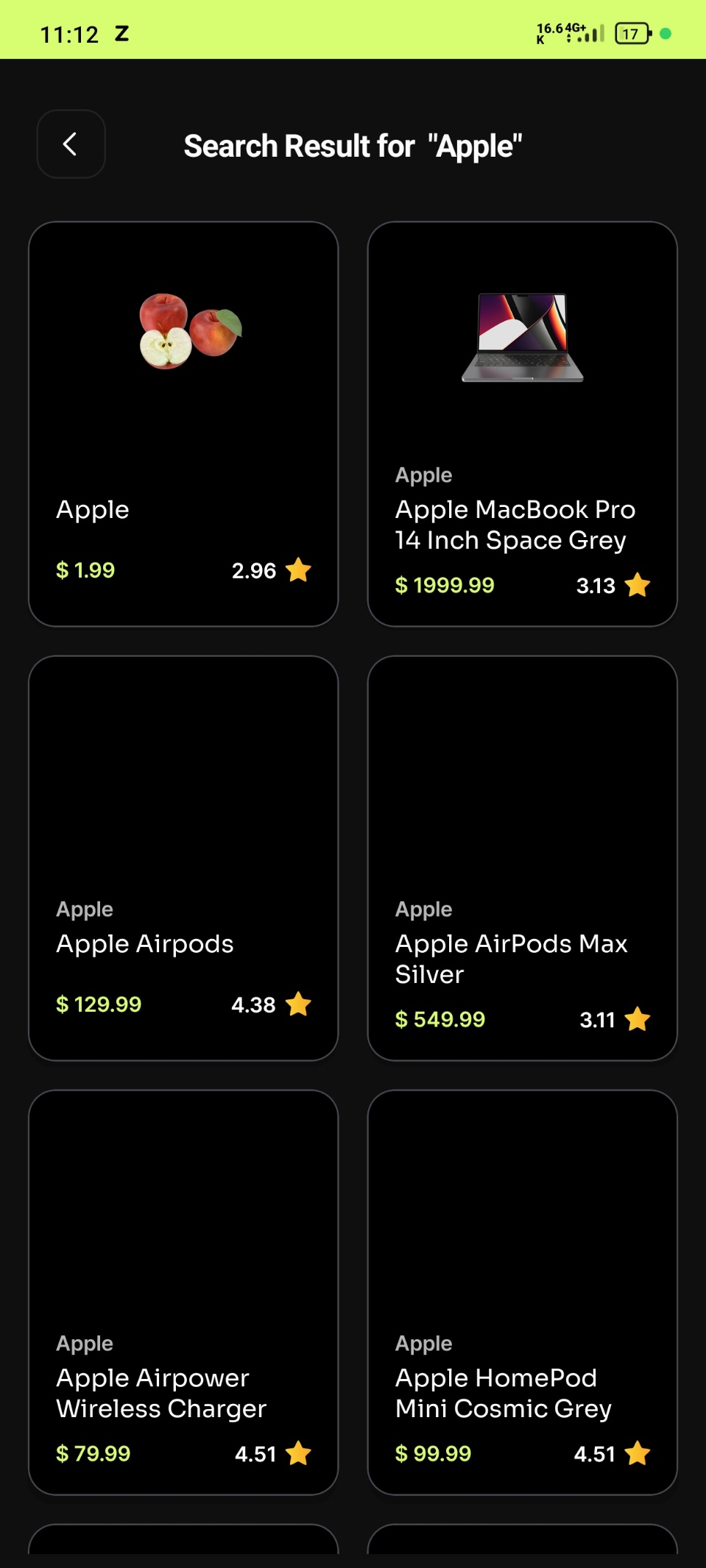
**Profile Screen:**

The Profile screen displays user-specific information and settings. It includes details such as the user's name, email, and profile picture, all retrieved from the user data stored in the app. This screen also provides options for the user to update their profile information, view order history, and log out of the application. The Profile screen is designed to give users control over their account settings and personal information.



**Search Screen:**





**Product Details Screen:**

