# User App

**Note:** While developing the User App, alternative approaches such as using freezed models, auto-generated code, deep linking for automatic routing, and implementing bloc and cubit patterns were considered. However, these options are typically more suitable for larger-scale applications with complex requirements.

**Introduction:** The User App is a Flutter application developed as part of a company assignment. The primary objective of the application is to provide users with a seamless authentication experience, followed by access to their profile information and interactive features such as viewing a list of items. Throughout the development process, special attention has been given to designing an intuitive user interface and ensuring optimal performance.

**Detail about each task:**

1. **Task 1: User Authentication**

**Objective:** To create a login page allowing users to sign in using their email and password, ensuring proper validation and handling authentication errors gracefully.

**Implementation:**

- Designed an aesthetically pleasing login screen with text input fields for email and password, along with a login button.
- Implemented basic validation for the email and password fields to ensure non-empty fields and valid email format.
- Upon successful login, users are seamlessly navigated to the home screen.
- Handled authentication errors gracefully, displaying appropriate error messages for enhanced user experience.
- Additionally, integrated own API for login and register functionalities, providing robust error handling.

2. **Task 2: Profile Screen**

**Objective:** To develop a profile screen displaying user information retrieved from a mock API endpoint, including the user's name, email, and profile picture, with the provision for updating profile information and logging out.

**Implementation:**

- Designed a visually appealing profile screen showcasing user details such as username, email,token, and profile picture.
- Utilized mock API endpoints to fetch user information, ensuring seamless integration of dummy data.

- Displayed fetched user information dynamically on the profile screen for enhanced personalization.
- Implemented a logout button allowing users to seamlessly return to the login screen for enhanced security and convenience.
- Added the functionality for users to update their profile information, including the option to upload profile pictures from the camera or gallery.

3. **Task 3: List View**

**Objective:** To incorporate a feature for displaying a list of items on the home screen, implementing lazy loading for optimal performance and graceful error handling.

**Implementation:**

- Designed an interactive list view on the home screen to display items fetched from mock API endpoints.
- Each list item includes a title, description, and image for improved visual representation.
- Implemented lazy loading functionality to fetch additional items dynamically as users scroll through the list.
- Ensured graceful error handling, displaying informative messages in case of data retrieval failures for enhanced user experience.
- Utilized own API to fetch the list of products, allowing for real-time data updates and integration with the application.

4. **Task 4: Navigation**

**Objective:** To facilitate seamless navigation between screens within the application while ensuring proper authentication and logout functionality.

**Implementation:**

- Implemented navigation between the login screen, profile screen, and home screen using named routes and tag routing for improved code organization and readability.
- Ensured users cannot access the profile screen or home screen without proper authentication, enhancing security.
- Implemented seamless navigation from the profile screen to the home screen upon pressing the logout button, providing users with a smooth transition.
- Incorporated snackbar notifications to display success and error messages for user interactions, enhancing the overall user experience.

5. **Task 5: Bonus Task**

**Objective:** To add an additional feature to the application to enhance its functionality and user experience.

**Implementation:**

- Added a splash screen with animations to provide users with a visually appealing app launch experience.
- Integrated animations into the list view and profile sections to enhance user interaction and engagement.
- Utilized animation effects to create a more dynamic and immersive user experience throughout the application.
- Overall, the User App successfully implements all assigned tasks and bonus features, providing users with a comprehensive and enjoyable mobile experience.

**Models I am using:**

- **ProductModel:**
    - Provides data structure for product information including ID, name, image data, description, price, category, sale status, and stock quantity.
- **UserModel:**
    - Represents user information including username, email, user ID, and authentication token.

**Conclusion:** In conclusion, the User App successfully integrates MVC architecture and component-based programming principles, emphasizing simplicity and efficiency. Leveraging the Provider package for state management ensures responsive performance and seamless data flow. The app's intuitive design, coupled with own API integration, enhances user experience and functionality. Overall, the User App delivers a polished, user-friendly mobile experience that meets and exceeds expectations.