

Laza Mobile Application Project Report

Team Member 1: Botamina Kamel Helmy

Id 1: 2305139

Team Member 2: Jerome Arsany Mansour

Id 2: 2305093

1. Introduction:

The Laza Mobile Application is a Flutter-based e-commerce mobile app inspired by the *Laza* UI design. The project was developed as a final project for the **Mobile Programming** course. The main goal of the application is to provide users with a modern, responsive, and user-friendly shopping experience, while applying real-world mobile development concepts such as API integration, Firebase authentication, and clean project architecture.

2. Project Objectives:

The primary objectives of this project are:

- To design and implement a modern mobile UI inspired by the Laza design.
 - To develop a fully functional Flutter application.
 - To integrate external APIs for product data.
 - To use Firebase for authentication and user management.
 - To apply clean code structure and modular architecture.
 - To gain hands-on experience with mobile app development using Flutter.
-

3. Technologies Used:

3.1 Flutter & Dart

- Flutter framework for cross-platform mobile development.
- Dart programming language for application logic.

3.2 Firebase

- **Firestore Authentication:** User sign-up, login, and logout.
- **Cloud Firestore:** Storing user data and future scalability for cart and favorites.

3.3 REST API

- External REST API (FakeStore API) used to fetch product data dynamically.

3.4 Development Tools

- Visual Studio Code
- Android Emulator
- Flutter SDK
- Firebase Console

4. Application Features:

4.1 Authentication

- User registration using email and password.
- User login and logout.
- Authentication state persistence using Firebase.

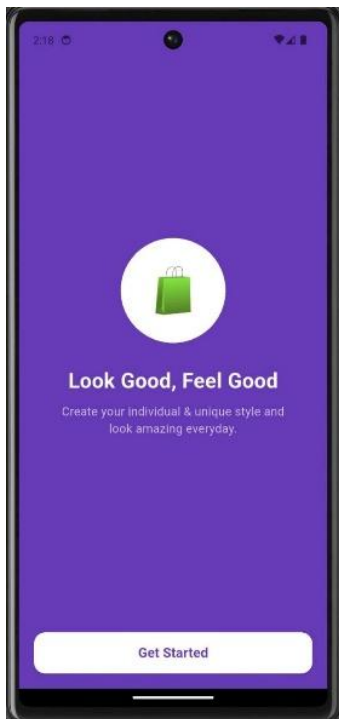
The top screenshot shows the Firebase Authentication console for 'Laza-Project-2025'. It displays a table of users with columns for Identifier, Providers, Created, Signed in, and User UID. The table contains three users: jerome@test.com, test@test.com, and student@test.com, all created on Dec 26, 2025.

Identifier	Providers	Created	Signed in	User UID
jerome@test.com	📧	Dec 26, 2025	Dec 26, 2025	vHw69YYDBJSjDuHMcQg...
test@test.com	📧	Dec 26, 2025	Dec 26, 2025	QX8k4fs44qetWzFSt6plyeA0c1
student@test.com	📧	Dec 26, 2025	Dec 26, 2025	08wOsQ2TWCRXWh6Ez5fsfNtTo1

The bottom screenshot shows the Firebase Database console for the same project. It displays a document view for the 'users' collection, showing a document with a UID of '08wOsQ2TWCRXWh6Ez5fsfNtTo1'. The document contains fields for 'createdAt' (December 26, 2025 at 3:21:30 AM UTC+2) and 'email' ('student@test.com').

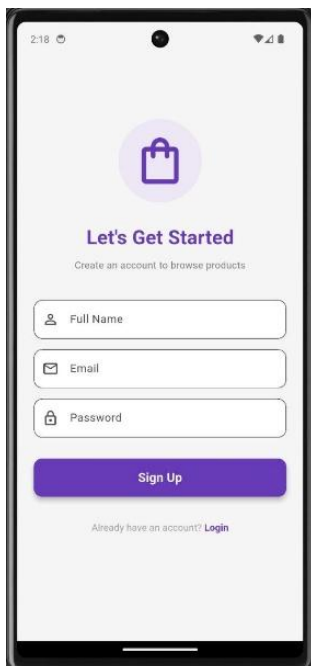
4.2 Welcome / Onboarding Screen

- Presents a motivational message encouraging user engagement.
- Includes a clear call-to-action button (“**Get Started**”) to begin the registration process.



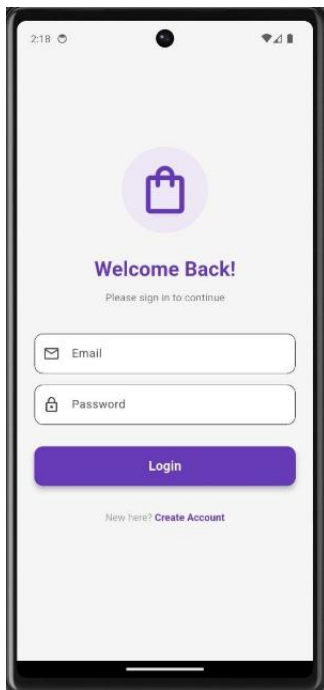
4.3 Sign-Up Screen

- Guides new users through account creation using name, email, and password fields.
- Includes navigation back to the login screen for existing users.



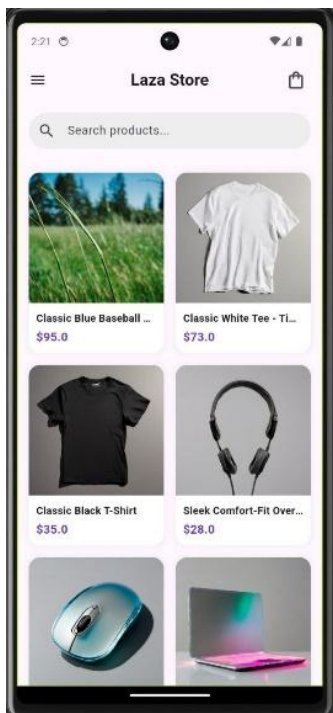
4.4 Login Screen

- Prompts returning users to sign in using email and password.
- Provides an option for new users to navigate to the sign-up screen.



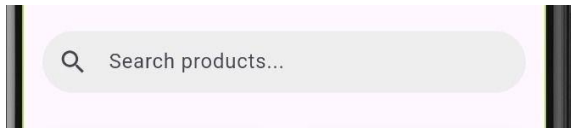
4.5 Home Screen

- Displays a list of products fetched from the API.
- Clean and modern UI layout.



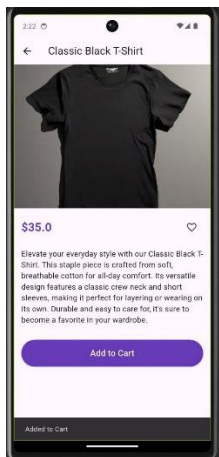
4.6 Home Search Screen

- Displays search functionality with placeholder text.
- Provides quick access to browse available products.



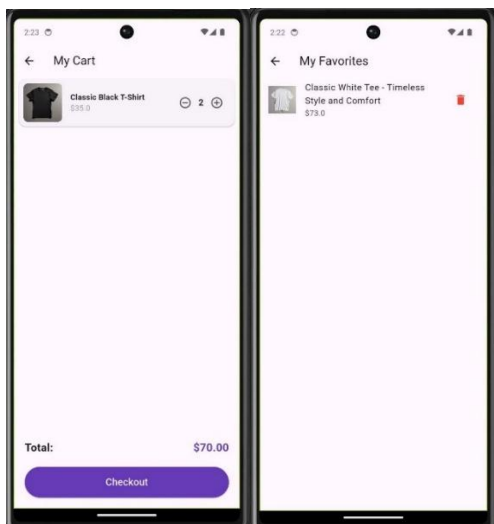
4.7 Product Details

- Detailed view for each product.
- Displays product image, price, and description.



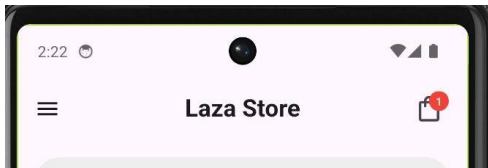
4.8 Cart & Favorites

- Add products to cart.
- Add products to favorites.
- Ready for Firestore integration



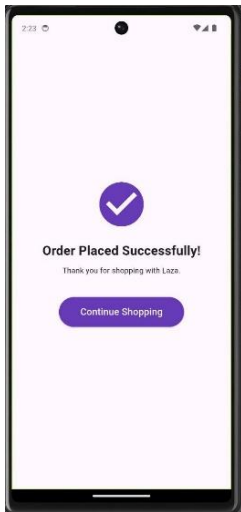
4.9 Store Header

- Features a cart icon with a real-time item counter, allowing users to easily track the number of products added to their cart.



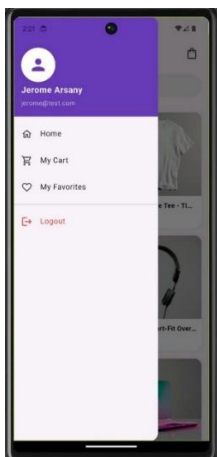
4.10 Order Confirmation Screen

- Displays a success message confirming the successful placement of the order.
- Includes a call-to-action button that allows users to continue shopping.



4.11 Profile Screen

- Displays user information.
- Logout functionality.



5. Project Structure:

The project follows a modular and scalable folder structure:

```
lib/
├── main.dart
├── firebase_options.dart
├── models/
│   ├── product.dart
│   ├── user.dart
│   └── cart_item.dart
├── services/
│   ├── api_service.dart
│   └── firebase_service.dart
├── screens/
│   ├── login_screen.dart
│   ├── home_screen.dart
│   ├── product_details_screen.dart
│   ├── cart_screen.dart
│   ├── favorites_screen.dart
│   └── profile_screen.dart
├── widgets/
│   ├── product_card.dart
│   └── custom_button.dart
├── utils/
└── constants.dart
```

This structure ensures readability, maintainability, and scalability.

6. Database Design:

6.1 Firestore Collections

- **users**
 - userId
 - email
 - createdAt

Future extensions may include:

- cart
 - favorites
 - orders
-

7. Application Flow:

1. App starts and initializes Firebase.
 2. Authentication state is checked.
 3. User is redirected to Login or Home screen.
 4. Products are fetched from API.
 5. User interacts with products (view, add to cart, favorites).
 6. User data is stored securely in Firebase.
-

8. Challenges Faced:

- Firebase setup and configuration issues.
- Android SDK and emulator configuration.
- UI consistency with Laza design.
- Managing state and navigation efficiently.

These challenges were resolved through debugging, documentation review, and testing.

9. Future Improvements:

- Full Firestore integration for cart and favorites.
 - Payment gateway integration.
 - Search and filter functionality.
 - Improved animations and UI enhancements.
 - Admin panel for product management.
-

10. Conclusion:

The Laza Mobile Application successfully demonstrates the use of Flutter for building modern mobile applications integrated with Firebase and REST APIs. The project meets its objectives and provides a strong foundation for future expansion. Through this project, valuable experience was gained in mobile app architecture, Firebase services, and real-world Flutter development.

11. References:

- Flutter Documentation
 - Firebase Documentation
 - FakeStore API
-