1. Introduction

EzBuy is a Flutter-based mobile e-commerce application designed to provide users with a smooth and intuitive shopping experience. The app integrates Firebase Firestore as the backend database and Firebase Authentication for secure user account management. EzBuy allows users to browse products, view product details, manage their cart, mark favorites, and complete purchases.

This project demonstrates practical experience in Flutter development, Firebase integration, state management, and modular code architecture. It also includes an admin module for uploading and managing products.

---

2. Project Objectives

The primary objectives of EzBuy are:

- To build a fully functional mobile e-commerce application using Flutter.
- To integrate Firebase Authentication and Firestore for secure and scalable backend services.
- To apply clean coding practices, modular architecture, and reusable components.
- To implement real-world features such as cart, favorites, checkout, and purchase history.
- To demonstrate understanding of state management using the BLoC pattern.
- To create a polished, user-friendly UI with smooth navigation and responsive layouts.

3. Problem Statement

The project aims to address the need for a simple, lightweight, and intuitive mobile shopping platform that enables users to browse products and make purchases with ease. Many small businesses or individuals need customizable e-commerce apps but often lack access to complex or expensive solutions. EzBuy solves this by providing a scalable mobile app framework with integrated backend services.

---

4. How the Application Works (User Flow)

EzBuy follows a simple and intuitive flow:

1. **User Authentication**
   New users can sign up with email and password. Existing users can log in. Firebase Authentication handles credential management.
2. **Home Screen**
   After logging in, users are redirected to the Home page where they can explore product listings.
3. **Product Browsing**
   Users can scroll through the product list, each showing an image, name, and price.
4. **Product Details**
   Selecting a product opens a details page with more information, quantity selector, and the option to add the item to cart or favorites.
5. **Favorites**
   Users can mark products as favorites and access them anytime.
6. **Cart System**
   Users can add items to the cart, update quantities, view totals, and remove items.
7. **Checkout Process**
   Users review their order, proceed to payment, and complete checkout.
8. **Purchase History**
   Completed orders are stored and displayed for reference.
9. **Profile Management**
   Users can view and update their profile details and change their password.

---

5. System Overview

EzBuy is structured around a combination of UI screens, service classes, models, and BLoC state management.
Key responsibilities include:

- UI screens for presenting data and user interaction
- Service classes for handling Firestore operations
- Models for representing structured data
- BLoC for managing complex UI state transitions

- Firebase for authentication and database storage

---

6. Key Features

User Features

- User registration and login
- Persistent authentication
- Browse products from Firestore
- View detailed product information
- Add to cart
- Add to favorites
- Remove items from cart or favorites
- Adjust product quantity
- Checkout and simulate payment
- View past purchases
- Manage profile and change password

App Structure & UX Features

- Clean UI layout and smooth navigation
- Modular and maintainable code
- Persistent Firestore storage
- State management using BLoC

---

7. Technology Stack

- **Flutter (Dart)** – Mobile app framework
- **Firebase Authentication** – User login and signup
- **Firestore Database** – Product, cart, and favorites storage
- **Firebase Storage** – Product image uploading
- **BLoC Pattern** – State management
- **Material Design** – UI components

---

8. Firebase Integration

EzBuy uses Firebase across multiple modules:

Firebase Authentication

Used for:

- Email/password signup
- Login
- Password change
- User session persistence

Cloud Firestore

Used for:

- Product collection
- Cart collection
- Favorites collection
- Order history

Every user gets their own Firestore document path, ensuring secure and scalable data separation.

---

9. Implementation Details

Authentication

- Signup and login implemented with FirebaseAuth.
- Error handling for invalid credentials.
- Users redirected after successful sign-in.
- Password change implemented in profile settings.

Product Management

- Products are fetched from Firestore in real time.
- Product images uploaded to Firebase Storage.

Cart System

- Each user has a personal cart collection.
- Items include quantity and calculated totals.
- Users can add, update, or remove items.

Favorites

- Users can mark products as favorites.
- Favorites stored in a separate Firestore collection.

Checkout Flow

- Cart items summarized and saved into a purchase history.
- Cart is cleared after checkout.

State Management

- Product details use a BLoC to handle state changes (loading, success, error).
- Improves performance and separation of concerns.

Profile Management

- Users can update profile details.
- Password change uses FirebaseAuth’s secure update method.

11. Challenges and Solutions

1. Managing Complex UI States

Challenge: Handling product loading, errors, quantity updates.
Solution: Implemented BLoC pattern for predictable state management.

2. Firebase Data Sync

Challenge: Keeping cart and favorites in sync across updates.
Solution: Used Firestore streams for real-time updates.

3. Secure Authentication

Challenge: Handling password changes and login errors.
Solution: Used FirebaseAuth's secure methods with proper validation.

4. Code Organization

Challenge: Preventing large, unmanageable files.
Solution: Broke the app into models, services, screens, and BLoC components.

12. Future Improvements

- Implement real payment gateway
- Add order tracking
- Introduce admin dashboard
- Improve UI animations

13. Conclusion

EzBuy is a fully functional e-commerce Flutter application that demonstrates strong understanding of mobile development, Firebase integration, and clean software architecture. The app offers a wide range of essential shopping features, including authentication, product browsing, cart management, favorites, checkout, and purchase history. The modular design and state management using BLoC make the application maintainable, scalable, and production-ready.

This project successfully meets the requirements of a modern mobile e-commerce system and reflects practical proficiency in Flutter development.