DEPI

Mobile Application Development

S H A R Y

Your Story, Your Style

E-Commerce App

by

Ahmed Gomaa Abdelhamed Sheiba

Mahmoud Mohamed Hassan Ahmed

Ahmed Saed Saad Hassan Khattab
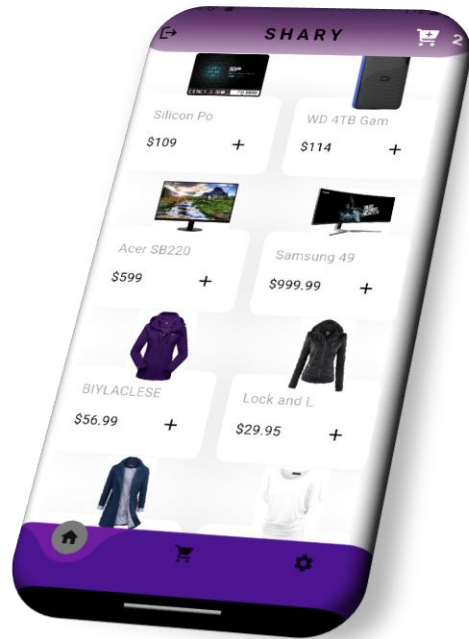
Ahmed Mohamed Nasr-Eldin

Islam Mohamed Abdelhalem

SHARY APP

1. Introduction

Shary is an e-commerce mobile application designed and built using Flutter and Dart, providing a seamless shopping experience with modern UI/UX features. The app offers functionalities such as product browsing, cart management, payment options, and state management with Provider. The app is integrated with Firebase and Firestore for backend services, ensuring secure and scalable data management.
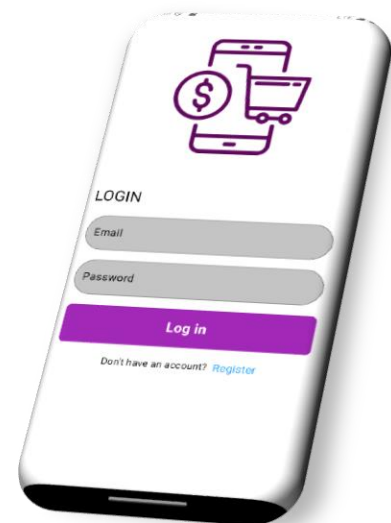
2. Features Overview

- **Browse Products:** Users can view a variety of products available for purchase.
- **Add to Cart:** Products can be added to a cart for purchasing now or at a later time.
- **Payment:** Integrated with credit card options, allowing users to complete their purchase using their bank account details.
- **User Authentication:** Users can register and log in to their accounts.
- **Light/Dark Mode:** A theme switcher allows users to toggle between light and dark modes.
- **Splash Screen:** A welcoming splash screen that enhances the user experience.
- **State Management:** The app uses Provider for managing state efficiently.

3. Technologies Used

- **Flutter:** Framework used for building the cross-platform app.
- **Dart:** Programming language used in the development of the app.
- **Firebase Authentication:** Used for user registration and login.
- **Firestore:** NoSQL database used for managing app data.
- **Provider:** State management solution for handling UI updates based on data changes.
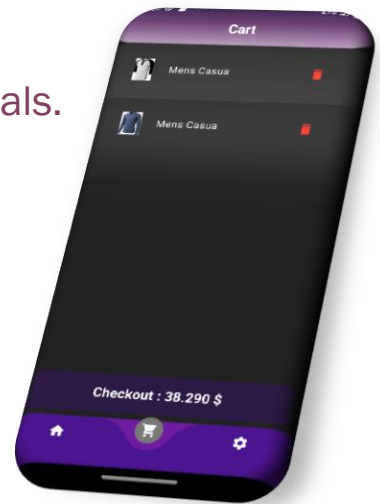
4. User Flow

4.1 Registration & Login

- Users need to create an account by providing necessary credentials (email, password).
- Alternatively, existing users can log in using their credentials.

4.2 Browsing Products

- The home screen displays a list of available products.

4.3 Adding Products to Cart

- Users can select products and add them to their cart.
- The cart screen shows all selected products,
- where users can adjust quantities or remove items.

4.4 Payment Gateway

- The payment screen allows users to input credit card information and finalize the purchase.
- Secure integration with payment services ensures data privacy **and** security.

4.5 Dark/Light Mode

- Users can toggle between light and dark modes via the settings screen for a personalized experience.
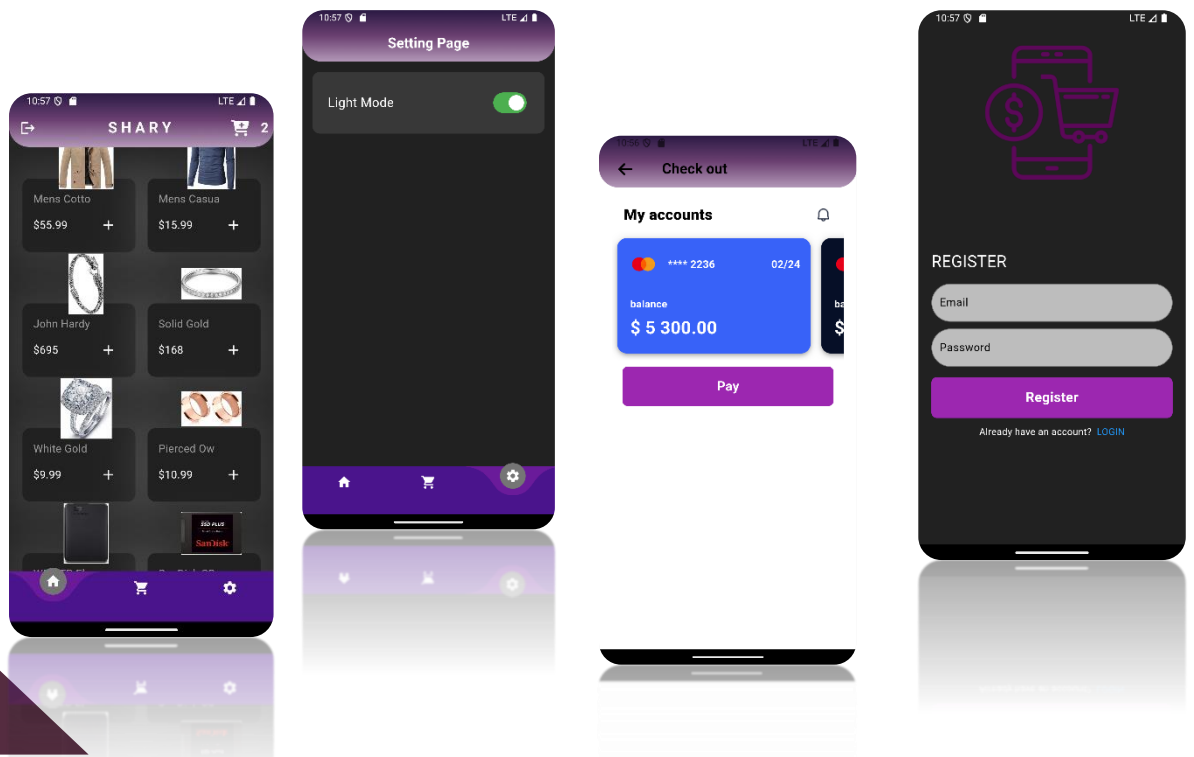
5. App Architecture

5.1 State Management

- The app utilizes the Provider package for efficient state management across different screens and components. This approach ensures a responsive UI by reacting to changes in the app's state.

5.2 Firebase Integration

- Firebase Authentication is used for managing user accounts and secure login.
- Firestore is used for storing product data, user information, and order details. This scalable NoSQL database ensures efficient data retrieval and management.

5.3 UI/UX Design

- Splash Screen: A visually appealing splash screen to greet users when the app launches.
- Theme Customization: Users can switch between light and dark modes, offering flexibility in user experience.

6. Areas for Improvement

6.1 Performance Optimization

- **Issue:** The app may experience performance lags when there are many items in the cart or a large number of products on the home screen.
- **Solution:** Implement lazy loading for product lists and optimize image loading by using caching strategies, such as cached_network_image.

6.2 Payment Integration

- **Issue:** Currently, the payment feature is limited to credit card transactions.
- **Solution:** Consider integrating multiple payment options like mobile wallets (Google Pay, Apple Pay) or third-party services (Stripe, PayPal) to provide more flexibility to users.

6.4 Security Enhancements

- **Issue:** Although Firebase Authentication is secure, there are additional security measures that can be taken to protect user data.
- **Solution:** Implement two-factor authentication (2FA) for login, use Firebase Security Rules for fine-grained control over Firestore data, and encrypt sensitive data stored locally.

6.5 UI/UX Refinements

- **Issue:** The app's design can be further improved for better user engagement, especially on small-screen devices.
- **Solution:** Conduct user testing to gather feedback on UI/UX design and optimize the app for various screen sizes and orientations.

7. Future Enhancements

- **Wishlist Feature:** Add the ability for users to save products for future reference without adding them to the cart.
- **Push Notifications:** Send notifications to users about offers, discounts, and abandoned carts.
- **Multi-language Support:** Allow the app to support multiple languages for a global audience.

Ahmed Gomaa Abdelhamed Sheiba	Home View – API - Firebase
Mahmoud Mohamed Hassan Ahmed	Cart View
Ahmed Saed Saad Hassan Khattab	Login View – Register View
Ahmed Mohamed Nasr-Eldin	Payment
Islam Mohamed Abdelhalem	Setting - Firebase

GitHub link:
- https://github.com/ahmedsheiba/Shary_App_DEPI.git