📱 SmartCart Project Documentation

📖 Project Overview

SmartCart is a mobile application developed to enhance the in-store shopping experience. It allows users to scan product barcodes using their smartphones, manage a virtual shopping cart, and streamline the checkout process. The app aims to:

- **Simplify Shopping**: Users can effortlessly add items to their cart by scanning barcodes.
- **Real-Time Updates**: View and manage cart contents in real-time.
- **Seamless Checkout**: Proceed to payment without the need for traditional checkout lines.

🛠️ Technologies & Frameworks

The SmartCart application is built using the following technologies:

- **Android Development**: Utilizing Android Studio for development.
- **Jetpack Compose**: For building modern, declarative UIs.
- **Firebase**: To handle user authentication, real-time database storage, and notifications.
- **ML Kit**: For barcode scanning functionality.

📱 Screens & Features

1. HomeScreen

- **Purpose**: Serves as the landing page of the app.
- **Features**:
  - Welcome message.
  - Navigation buttons to other sections (Scan, Cart, Profile).

2. ScanScreen

- **Purpose**: Allows users to scan product barcodes.
- **Features**:
  - Camera viewfinder for scanning.
  - Real-time barcode detection using ML Kit.
  - Fetches product details from Firebase upon successful scan.

3. CartScreen

- **Purpose**: Displays items added to the shopping cart.

- **Features**:
    - List of products with names, quantities, and prices.
    - Option to modify quantities or remove items.
    - Total price calculation.
    - Proceed to the payment button.

4. BillPage

- **Purpose**: Shows the receipt after a purchase.
- **Features**:
    - Itemized list of purchased products.
    - Total amount and tax details.
    - Store information and timestamp.

5. PaymentFormPage

- **Purpose**: Collects payment information from the user.
- **Features**:
    - Input fields for payment details (e.g., card number, expiration date).
    - Submit button to process payment.

6. ProfileScreen

- **Purpose**: Displays user profile information.
- **Features**:
    - View and edit personal details (name, email, address).

7. NotificationsScreen

- **Purpose**: Displays notifications related to user activities.
- **Features**:
    - List of notifications with titles and timestamps.
    - Option to clear or mark notifications as read.

🔄 Data Flow & Architecture

The app follows the **MVVM (Model-View-ViewModel)** architecture:

- **Model**: Represents the data layer, including data models and repository classes.
- **View**: Represents the UI components (screens).
- **ViewModel**: Acts as a mediator between the Model and View, handling business logic and data transformation.

Data Flow:

1. **Scanning**: The user scans a product barcode using the ScanScreen. ML Kit processes the barcode, and the app fetches product details from Firebase.
2. **Cart Management**: Scanned products are added to the CartScreen. Users can modify quantities or remove items. The total price is updated in real-time.
3. **Payment**: Upon proceeding to payment, the PaymentFormPage collects payment details. The app processes the payment and displays the BillPage with the receipt.
4. **Profile & Notifications**: Users can view and edit their profile information on the ProfileScreen. Notifications are displayed on the NotificationsScreen.

🔐 Firebase Integration

Firebase is utilized for:

- **Authentication**: Managing user sign-in and sign-up.
- **Realtime Database**: Storing and retrieving product and cart data.
- **Cloud Messaging**: Sending notifications to users.

Setup:

1. Create a Firebase project in the Firebase Console.
2. Add the `google-services.json` file to the `app/` directory.
3. Enable Firebase Authentication and Realtime Database in the Firebase Console.

📦 Dependencies

The project includes the following dependencies:

- **Firebase SDK**: For authentication and database services.
- **ML Kit**: For barcode scanning functionality.
- **Jetpack Compose**: For building the UI.
- **Navigation Component**: For handling in-app navigation.
- **Lifecycle Components**: For managing UI-related data lifecycle-consciously.

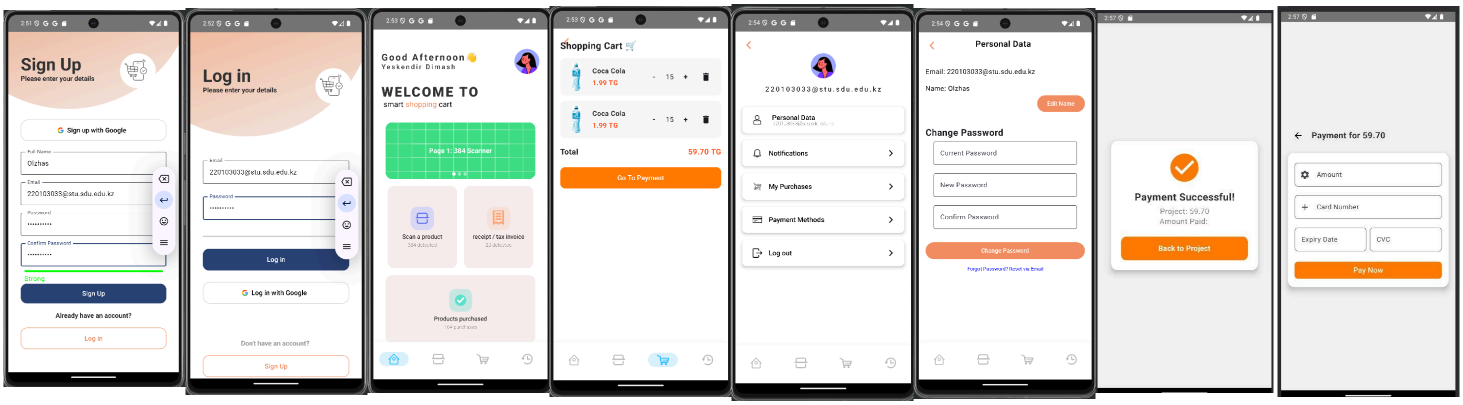
🚀 Setup Instructions

Clone the Repository:

```
 git clone https://github.com/D-Arshad-Dazai-Dimash/SmartCartProject_16-P.git
cd SmartCartProject_16-P
```

1.
2. **Open in Android Studio**:
   - ○ Launch Android Studio.
   - ○ Open the cloned project.
3. **Sync Gradle**:
   - ○ Click on "Sync Project with Gradle Files" to download dependencies.
4. **Run the Application**:
   - ○ Select a device or emulator.
   - ○ Click on the "Run" button.

📸 Screenshots



Full Name: Olzhas
Email: 220103033@stu.sdu.edu.kz
Password: ••••••••••
Confirm Password: ••••••••••

Email: 220103033@stu.sdu.edu.kz
Password: ••••••••••

Good Afternoon 👋
Yeskendir Dimash
WELCOME TO
smart shopping cart

Coca Cola	1.99 TG	- 15 +
Coca Cola	1.99 TG	- 15 +
Total	59.70 TG	

Email: 220103033@stu.sdu.edu.kz
Name: Olzhas

Payment Successful!
Project: 59.70
Amount Paid:

📚 References

[Firebase Realtime Database Documentation](https://smart-9f34e-default-rtdb.firebaseio.com/)
[ML Kit Barcode Scanning Documentation](https://developers.google.com/ml-kit/vision/barcode-scanning?hl=ru)
[CameraX Documentation](https://developer.android.com/media/camera/camerax?hl=ru)
[Jetpack Compose Official Docs](https://developer.android.com/develop/ui/compose/documentation?hl=ru)

👥 Team Members

Dimash Yeskendir, 220103327, 16-P
Olzhas Musakhan, 220103033, 16-P
Asylzhan Bitore, 220103181, 14-P
Ayan Amantay, 123456, 16-P
Damir Turgambekov, 220103278, 17-P