**#** **Smart Bill Payment App**

**##** **Project Theme**

**\*\*Smart Bill Payment App\*\***

**##** **App Name**

**\*\*Smart Bill\*\***

**##** **1. Project Overview**

Smart Bill is a cross-platform mobile and web application designed to simplify and unify bill payments for users in Cameroon. The app allows users to pay all their utility bills (electricity, water, TV, school fees, etc.) from a single platform, using their preferred payment method. It leverages modern technologies to provide a seamless, secure, and user-friendly experience.

**##** **2. Problem Statement**

Paying bills in Cameroon is often stressful and time-consuming. Users must:

- Visit different payment points or use various USSD codes for each biller

- Stand in long queues or face service interruptions due to late payments

- Struggle to track payment history and receipts

- Lack reminders for upcoming bills

There is no unified, user-friendly solution for managing all bills in one place, especially with secure integration for mobile money payments.

**##** **3. Solution**

Smart Bill provides a one-stop solution for all bill payments. Users can:

- Register and log in securely

- Add and manage multiple bill accounts

- Pay bills using MTN Mobile Money, Orange Money, or other methods

- Receive reminders before bills are due

- View and download payment history and receipts

- Get customer support within the app

**##** **4. Key Features**

- **\*\*User Registration/Login\*\*** (with Firebase Auth)

- **\*\*Dashboard:\*\*** View all available billers

- **\*\*Bill Payment:\*\*** Pay any bill using mobile money or other methods

- **\*\*Smart Reminders:\*\*** Notifications before bills are due

- **\*\*Payment History:\*\*** View/download receipts

- **\*\*Auto-Pay:\*\*** Set up recurring payments

- **\*\*Customer Support:\*\*** In-app help for issues

- **\*\*Multi-language Support:\*\*** English, French, and local languages

- **\*\*Admin Panel:\*\*** Manage billers, users, and transactions

**##** **5. System Architecture**

**###** **\*\*Architecture Diagram\*\***

```mermaid

graph TD

    A[Flutter App (iOS/Android/Web)] -- HTTP/JSON --> B[Firebase Backend]

    A -- HTTP/JSON --> C[Custom Django Backend (Payment Service)]

    C -- API --> D[MTN/Orange Mobile Money APIs]

    B -- Sync --> E[Firestore Database]

    B -- Auth --> F[Firebase Auth]

    C -- Update --> B

```

**###** **\*\*Explanation\*\***

- **\*\*Flutter App:\*\*** User interface for Android, iOS, and Web

- **\*\*Firebase:\*\*** Handles authentication, main data storage, notifications

- **\*\*Custom Django Backend:\*\*** Handles secure payment processing and integration with MTN/Orange Money APIs

- **\*\*Firestore Database:\*\*** Stores user, bill, and transaction data

- **\*\*Firebase Auth:\*\*** Manages user authentication

- **\*\*Mobile Money APIs:\*\*** External services for processing payments

**##** **6. Technology Choices & Justification**

- **\*\*Flutter (Dart):\*\*** One codebase for Android, iOS, and Web; beautiful UI; fast development

- **\*\*Firebase:\*\*** Easy authentication, real-time database, notifications, and hosting; no server management

- **\*\*Django (Python):\*\*** Full control for secure payment integration; handles payment callbacks/webhooks; updates Firebase with payment status

**##** **7. User Flow**

1. **\*\*User registers/logs in\*\*** via Firebase Auth

2. **\*\*Adds bill accounts\*\*** (e.g., ENEO meter number)

3. **\*\*Views dashboard\*\*** with all billers

4. **\*\*Selects a bill to pay\*\*** and enters amount

5. **\*\*Chooses payment method\*\*** (MTN/Orange Money)

6. **\*\*Flutter app sends payment request\*\*** to Django backend

7. **\*\*Django backend processes payment\*\*** with mobile money API

8. **\*\*On success, Django updates Firebase\*\*** with payment status

9. **\*\*User receives confirmation and receipt\*\*** in the app

10. **\*\*App sends reminders\*\*** for upcoming bills

**##** **8. Payment Integration (Firebase + Django)**

- **\*\*Why combine both?\*\***

  - Firebase is great for user management, data, and notifications, but cannot securely handle payment API secrets or webhooks.

  - Django backend acts as a secure bridge between the app and mobile money APIs.

- **\*\*How it works:\*\***

  - Flutter app sends payment details to Django backend

  - Django backend communicates with MTN/Orange Money APIs

  - Django receives payment confirmation (webhook/callback)

  - Django updates payment status in Firebase

  - App reads updated status from Firebase and notifies user

**##** **9. Summary & Next Steps**

Smart Bill is a modern, scalable solution to the bill payment challenges faced by Cameroonians. By combining Flutter, Firebase, and a custom Django backend, the app delivers a seamless, secure, and user-friendly experience across all devices. The architecture allows for rapid development, easy scaling, and secure payment integration.

**\*\*Next Steps:\*\***

- Design UI wireframes

- Set up Firebase project and authentication

- Develop Django backend for payment integration

- Build and test Flutter app

- Deploy and gather user feedback

Primary k