Flutter Project
Government ID Card Generator

Paranthaman S
1948TM0494

Project Specifications to be covered

- 1.Navigation
- 2. Events
- 3. Images
- 4. Layouts
- 5. Database(Firebase)
- 6. Localization(Multi language support)

Introduction

I developed a **Government ID App** using **Flutter**, designed to provide a seamless digital identity solution. The app allowed users to register, generate a unique ID, and view their digital ID card with essential details. It featured smooth navigation between screens, interactive event handling for dynamic functionality, and visually appealing layouts with background images and QR codes for identity verification. User details were securely stored in **FirebaseFirestore**, ensuring real-time access and data safety. Additionally, the app supported **multi-language localization** in **English and Tamil**, making it accessible to a broader audience. My goal was to create a **secure, efficient, and user-friendly** platform for digital identity management.

Project Structure

- gov_id/
- │── lib/
- │   ├── l10n/
- │   ├── database_helper.dart
- │   ├── firebase_options.dart
- │   ├── home_page.dart
- │   ├── login_page.dart
- │   ├── main.dart
- │   ├── register_page.dart
- │   ├── registered_details_page.dart
- │   ├── view_card_page.dart
- │ ── pubspec.lock
- │── pubspec.yaml

Home Page

The **home_page.dart** file served as the landing screen of our Flutter application, providing navigation to key functionalities. It featured a multilingual interface with a dropdown to switch between **English** and **Tamil**, ensuring localization support. The app bar displayed the title dynamically based on the selected language. The body of the page consisted of two main buttons: **Register** and **Login**, allowing users to either sign up or access their accounts. The layout was centered for a clean and structured appearance, making navigation seamless. Additionally, the **onLogout callback** mechanism was integrated to handle user authentication states efficiently.

Register Page

The **register_page.dart** file handled user registration, allowing new users to create an account with essential details. It featured a structured form with input fields for **name, date of birth, email, address, and password**, ensuring comprehensive data collection. The form utilized **validation mechanisms** to prevent incorrect inputs and improve user experience. Upon submission, the entered data was stored in the **Firebase database**, enabling secure and persistent user management. The page followed a clean and responsive layout with proper spacing and styling. Additionally, a navigation button directed users to the **login page**, ensuring a smooth registration flow within the application.

Login Page

The **login_page.dart** file facilitated user authentication by allowing registered users to log in using their **email and password**. It incorporated input validation to ensure correct credentials were entered before submission. Upon a successful login attempt, the app verified user details stored in the **Firebase database**, granting access to the main sections of the application. In case of incorrect credentials, appropriate error messages were displayed to guide the user. The page featured a **user-friendly layout**, including a navigation option to the **registration page** for new users. Additionally, it maintained a secure authentication process, ensuring data protection and smooth user access.

View Generated Card Page

The **view_card_page.dart** file was responsible for displaying a **digital government ID card** for a registered user. It retrieved user details from the **database** based on the logged-in email and presented them in a structured card layout. The card featured key information such as **name, date of birth, address, and a unique ID number**, along with a dynamically generated **QR code** that encoded user details for quick scanning. The page also included a **logout option** with a confirmation dialog to ensure a secure exit. A visually appealing design was implemented using **Flutter layouts and gradients**, making the digital ID both professional and user-friendly.

Firebase DataBase

In our **Gov ID** project, Firebase played a crucial role in handling user authentication and data storage. We used **Firebase Authentication** for user login and registration, ensuring secure access. The **register_page.dart** file allowed users to sign up by entering their details, which were then stored in **Firebase Realtime Database** under a unique user ID. The **login_page.dart** file authenticated users using signInWithEmailAndPassword, ensuring only registered users could access their ID cards. In **view_card_page.dart**, we retrieved user details from Firebase using their email ID and displayed them in a digital **ID card format**. The page also dynamically generated a **QR code** containing user details. Firebase allowed seamless **real-time data retrieval**, ensuring up-to-date user information. Additionally, Firebase was used to store **user preferences**, such as multi-language settings, in the l10n folder. This helped in implementing **localization**, enabling users to switch between **English and Tamil**. By integrating Firebase, we ensured **data security, scalability, and real-time access**, making our app efficient. The cloud-based database enabled smooth data transactions, while Firebase Authentication enhanced user security. Overall, Firebase provided a **robust backend** that streamlined user management and ID card generation.

Navigation

- **[Home Page]**
- **| **
- **| **
- **| → [Register Page] → [View Card Page]**
- **|**
- **→ [Login Page] → [View Card Page]**
-
- **[View Card Page]**
- **|**
- **→ (Logout) → [Home Page]**

Commands Involved

- **flutter create gov_id**
- **cd gov_id**
- **flutterfire configure**
- **flutter pub add firebase_core firebase_auth cloud_firestore**
- **flutter pub add sqflite path_provider**
- **flutter pub add qr_flutter**
- **flutter pub add flutter_localizations intl**
- **flutter gen-l10n**
- **flutter run**