Bibliophilic

By

Rajvi Dave(92200133038)

Project Submitted to

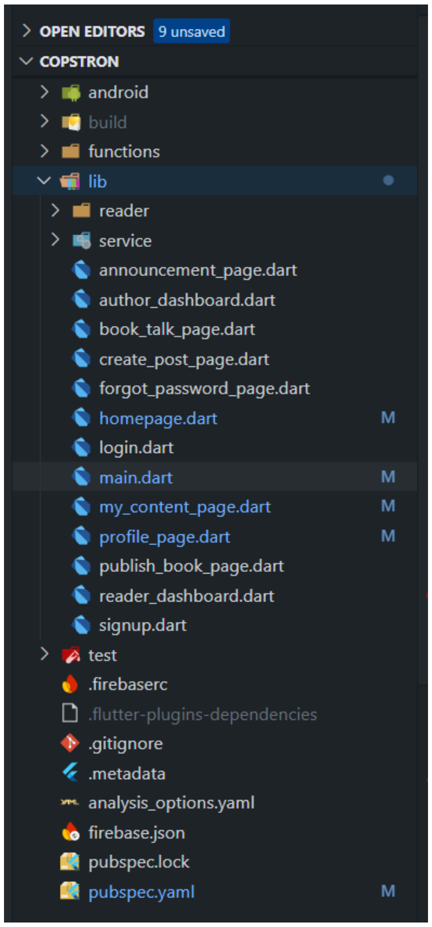
Marwadi University in Partial Fulfillment of the Requirements for the subject Capstrone Project

September 25

MARWADI UNIVERSITY
Rajkot-Morbi Road, At & Po. Gauridad,
Rajkot-360003, Gujarat, India

Code structure and organization:-

File Hierarchy:-



```
> OPEN EDITORS  9 unsaved
v COPSTRON
  > android
  > build
  > functions
  v lib
    > reader
    > service
      announcement_page.dart
      author_dashboard.dart
      book_talk_page.dart
      create_post_page.dart
      forgot_password_page.dart
      homepage.dart                M
      login.dart
      main.dart                    M
      my_content_page.dart         M
      profile_page.dart            M
      publish_book_page.dart
      reader_dashboard.dart
      signup.dart
  > test
    .firebaserc
    .flutter-plugins-dependencies
    .gitignore
    .metadata
    analysis_options.yaml
    firebase.json
    pubspec.lock
    pubspec.yaml                   M
```

<u>Modular Description:-</u>

Lib\reader - has all the files for reader side
Lib\service - has all the backend connected files
Lib - has all the basic files like signup and login as well as author side files

<u>Implementation Details:-</u>

Key Technologies Used

- **Front-End:** The app is built using **Flutter**, Google's cross-platform UI toolkit, which allows us to have one beautiful codebase for both Android and iOS. The programming language used is **Dart**.

- **Back-End:** We are using **Google Firebase** as our all-in-one backend, which includes:
    - **Firebase Authentication** for user management.
    - **Cloud Firestore** as our real-time NoSQL database.
    - **Firebase Storage** for file and media hosting.

Core Algorithm: Role-Based Authentication Flow

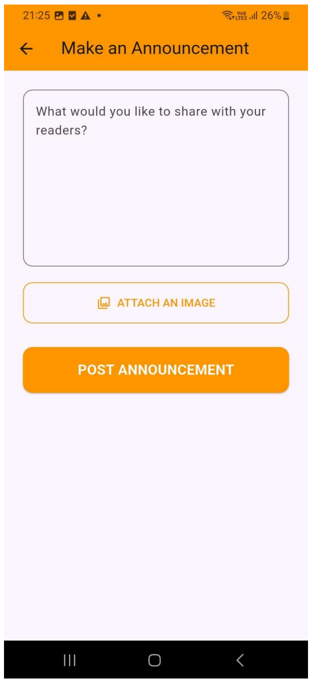
The most critical process in our app is ensuring users are directed to the correct dashboard based on their role. This is handled by a clear, two-part flow:

1. **On Signup:**
    - The user fills out the form on the SignUpPage, including selecting their role ("Author" or "Reader").
    - When they click "Sign Up," the app first calls our AuthService to create a new account in **Firebase Authentication**.
    - If that is successful, the app immediately calls our DatabaseService to create a new user document in the **Firestore Database**, saving their name, email, and the chosen role.
    - The app then navigates them to the correct dashboard based on the role they selected.
2. **On Login:**
    - The user enters their credentials on the LoginPage.
    - The app calls our AuthService to sign them in with **Firebase Authentication**.
    - If successful, the app immediately calls our DatabaseService to fetch that user's document from **Firestore** and read their role field.
    - Based on the role returned ("Author" or "Reader"), the app navigates them to the appropriate dashboard.

3. **Real time data protocol:**

The Process: Instead of asking the database for data just once, the MyContentPage opens a persistent, live connection (a "stream") to the author's content collection in Firestore.

The Result: The StreamBuilder widget automatically listens for any changes. The moment a new book or announcement is added to the database, Firestore pushes that update down the stream to the app, and the StreamBuilder automatically rebuilds the list on the screen to show the new content. This happens without needing a "pull-to-refresh" action, creating a seamless and modern user experience

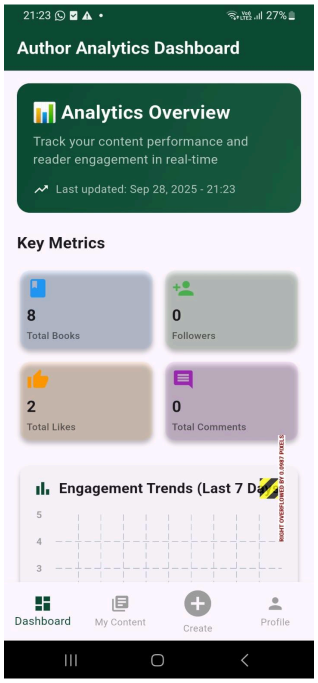

21:25 26%

Make an Announcement

What would you like to share with your readers?

ATTACH AN IMAGE

POST ANNOUNCEMENT


21:23 27%

Author Analytics Dashboard

Analytics Overview

Track your content performance and reader engagement in real-time

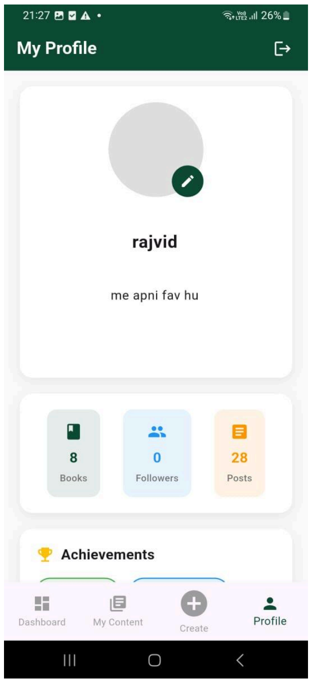
Last updated: Sep 28, 2025 - 21:23

Key Metrics

8 Total Books	0 Followers
2 Total Likes	0 Total Comments

Engagement Trends (Last 7 D

Dashboard · My Content · Create · Profile


21:27 26%

My Profile

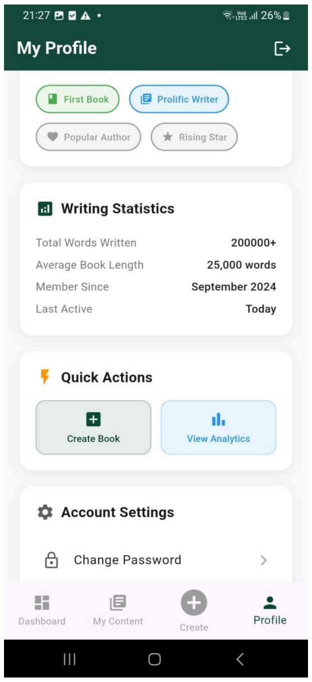
rajvid

me apni fav hu

8 Books	0 Followers	28 Posts

Achievements

Dashboard · My Content · Create · Profile


21:27 26%

My Profile

First Book · Prolific Writer · Popular Author · Rising Star

Writing Statistics

Total Words Written	200000+
Average Book Length	25,000 words
Member Since	September 2024
Last Active	Today

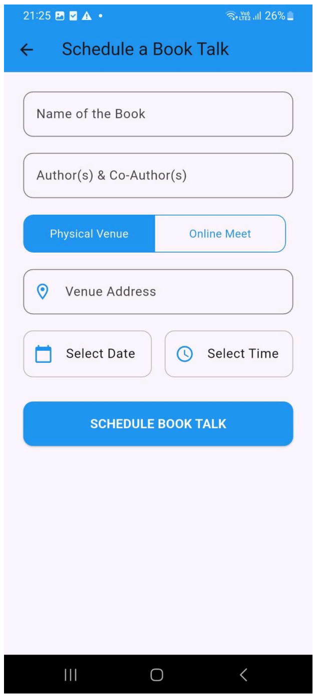
Quick Actions

Create Book · View Analytics

Account Settings

Change Password

Dashboard · My Content · Create · Profile


21:25 26%

Schedule a Book Talk

Name of the Book

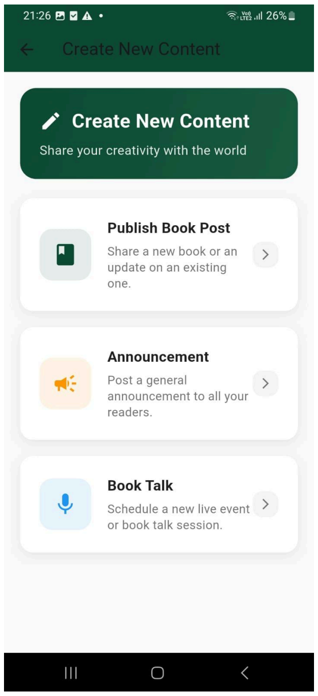
Author(s) & Co-Author(s)

Physical Venue · Online Meet

Venue Address

Select Date · Select Time

SCHEDULE BOOK TALK


21:26 26%

Create New Content

Create New Content

Share your creativity with the world

Publish Book Post

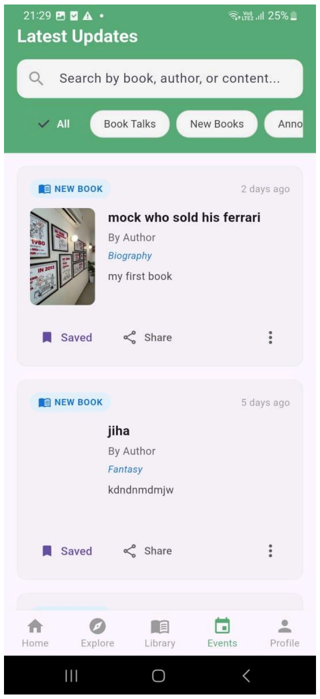
Share a new book or an update on an existing one.

Announcement

Post a general announcement to all your readers.

Book Talk

Schedule a new live event or book talk session.



21:29 25%

Latest Updates

Search by book, author, or content...

All | Book Talks | New Books | Anno

NEW BOOK 2 days ago

mock who sold his ferrari

By Author

Biography

my first book

Saved Share

NEW BOOK 5 days ago

jiha

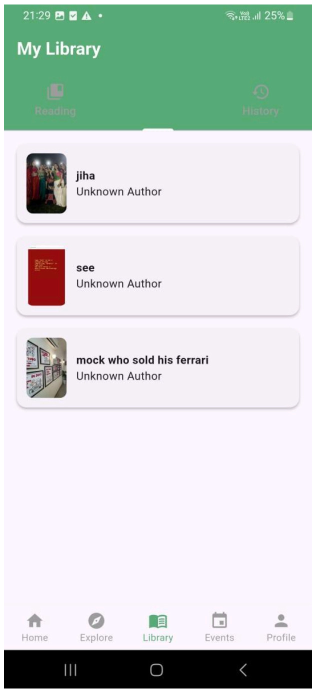
By Author

Fantasy

kdndnmdmjw

Saved Share

Home Explore Library Events Profile



21:29 25%

My Library

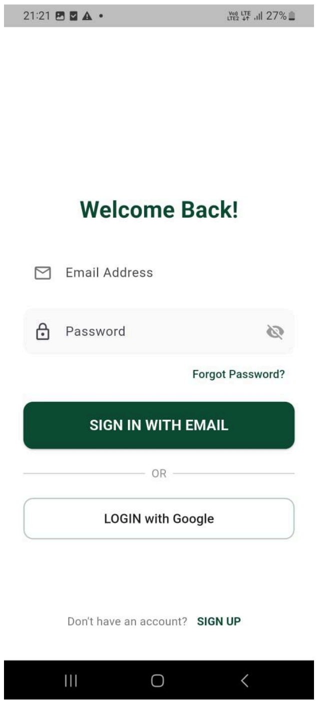
Reading History

jiha
Unknown Author

see
Unknown Author

mock who sold his ferrari
Unknown Author

Home Explore Library Events Profile



21:21 27%

Welcome Back!

Email Address

Password

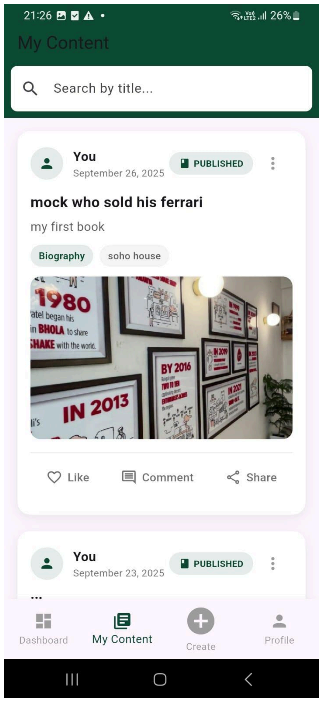
Forgot Password?

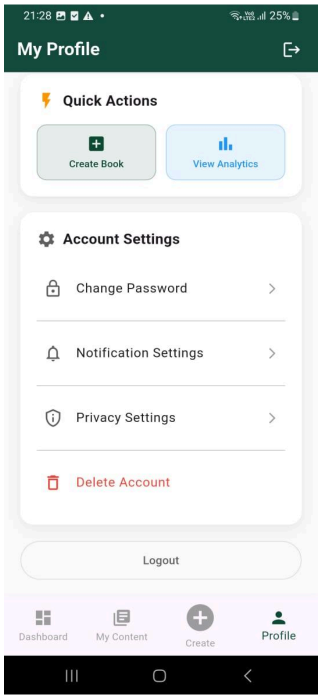
SIGN IN WITH EMAIL

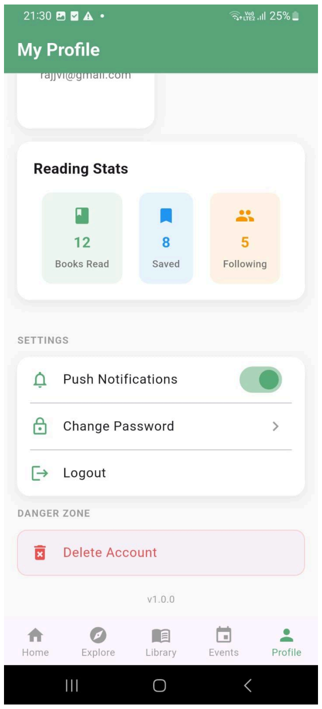
OR

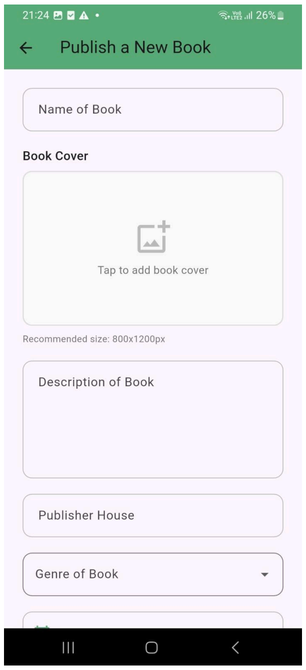
LOGIN with Google

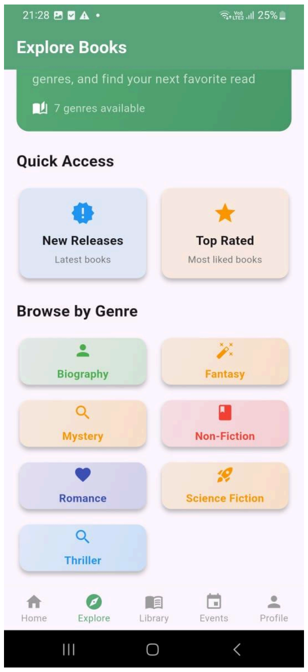
Don't have an account? SIGN UP

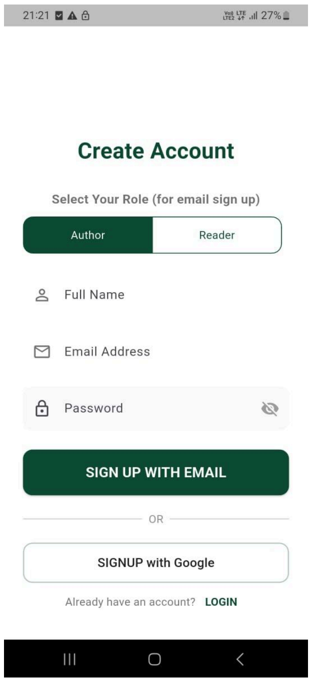












Setup Guide:-

To set up and run this project, you will need the following:

1. **Flutter SDK:** Ensure you have the Flutter SDK installed on your machine. You can find instructions at flutter.dev.
2. **Firebase Project:**
   - Create a new project on the Firebase Console.
   - Add an **Android application** to the project. Ensure the package name matches the one in android/app/build.gradle (e.g., com.example.cp_final).
   - Download the generated google-services.json file and place it in the android/app/ directory of your project.
   - For Google Sign-In, you must add your computer's **SHA-1 fingerprint** to the Android app settings in the Firebase console.
3. **Enable Firebase Services:**
   - In the Firebase console, go to **Authentication** and enable the **Email/Password** and **Google** sign-in providers.
   - Go to **Firestore Database** and create a new database. Start in **Test Mode** for initial setup.
   - Go to **Storage** and create a new storage bucket.