Mobile Application Development - Mini-Project

-Biomark-

Team : Prathibhapoorna T.A.P – 21001431

Github: https://github.com/Thenuwara07/Mini_Project_App.git

Video: https://drive.google.com/drive/folders/139dT-Y1N-xqQeGAHeMxaMn4u7_tPijCh?usp=sharing

Introduction

As part of this project, we have to create a mobile application for a research company named "Biomark." Biomark specializes in collecting large quantities of personal information from adult volunteers to build predictive machine learning models aimed at personalizing digital services.
To make it easier to register, manage, and protect the personal information volunteers contribute, the organisation needs a safe and intuitive mobile application.

Requirements and Their Implementation

1. Design a simple and intuitive UI with screens for user registration, login and profile management.

- Then I implemented those UIs using Flutter

2. Provide a Participator Account (PAC) for each volunteer, allowing them to securelycreate and manage profiles.

- We developed a registration system that allows volunteers to create accounts using a valid email address and password. After successfully registering, they are assigned a unique PAC (Participator Account).
- Volunteers can log in with their PAC credentials to view their profile, which includes personal and security information.
- Profile screens are designed to let volunteers view their personal data, which is non-editable and used for model building, ensuring data integrity after submission.

3. Collect specific personal information (e.g., Date of Birth, Blood Group, Ethnicity)from volunteers.

- A profile creation form was built to gather the required personal information after the registration.
- The form captures key data such as Date of Birth, Time of Birth, Blood Group, and optional data, all stored securely in external database (Firebase).
- Additional data collection is planned to be integrated in future updates on a subscription basis.

4. Securely store sensitive data (Full Name, Email, Password, Recovery Information) while preventing unauthorized access or identification.

- To protect volunteer identities, no direct mapping between personal information and recovery data is maintained.
- Recovery information and data used for model building are stored separately, ensuring both security and anonymity.

5. Ensure data persistence using a MongoDB for local storage and an externalservice for long-term storage.

- Personal information is securely stored locally in a MongoDB.
- We implemented interfaces to connect with the external Biomark database for future expansion, ensuring data persistence beyond the app's local storage and store Biodata.
- Encryption was applied to sensitive fields stored in the local database to prevent unauthorized access.

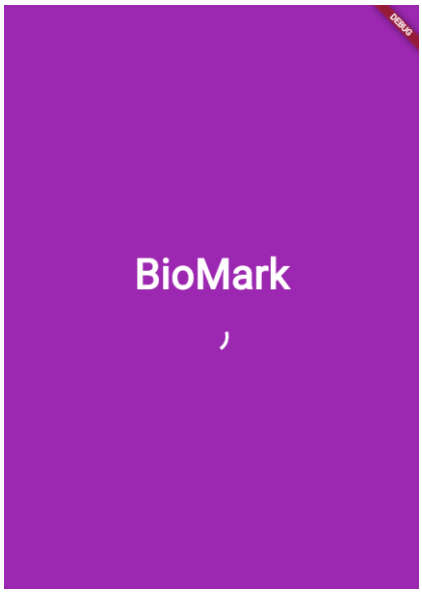
6. Allow users to Delete from Biomark and remove their profile data from thesystem.

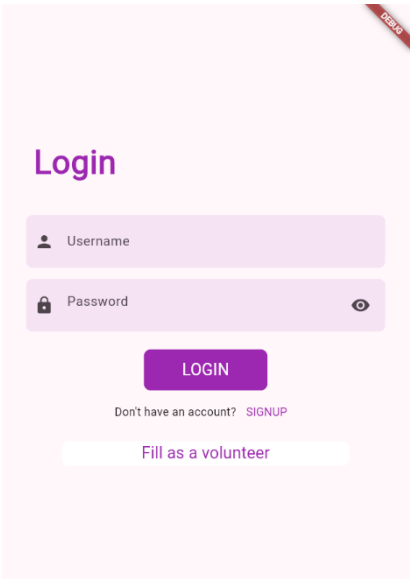
- A dedicated feature allows users to delete from the program, which deletestheir profile from the Biomark database.
- Simply uninstalling the app will not delete the user's data; explicit delete is required to remove their profile data permanently.

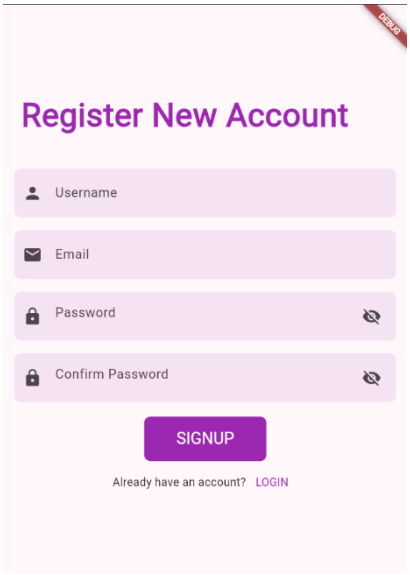
7. Develop the app using a specific programming language and ensure data persistence through MongoDB, Firebase and external services.

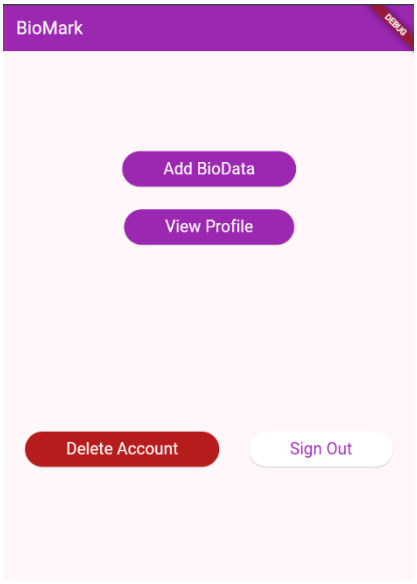
- We developed the app using Flutter, Dart leveraging MongoDB for local data persistence and designed an interface for future integration with an external service for long-term data storage.
- The technologies used ensure seamless performance and data handling while maintaining security and privacy standards.
- Use external database to save biodata.

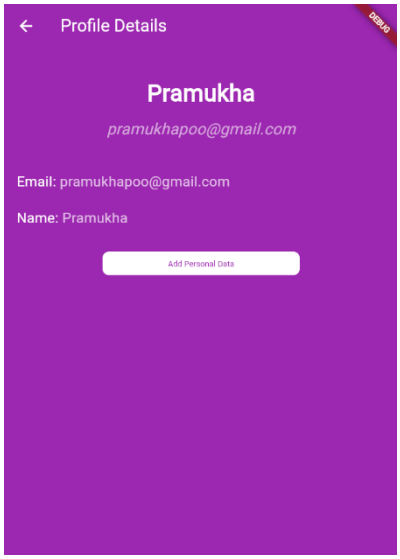
Biomark User Interfaces

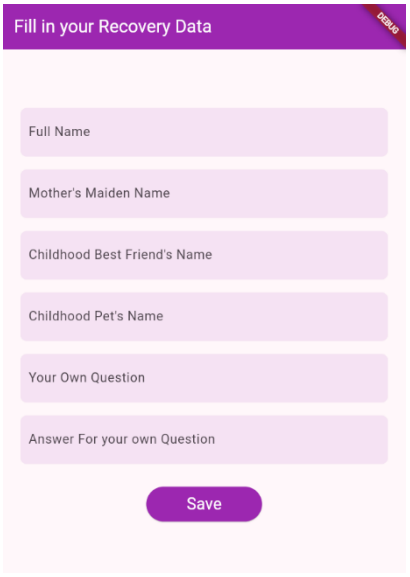


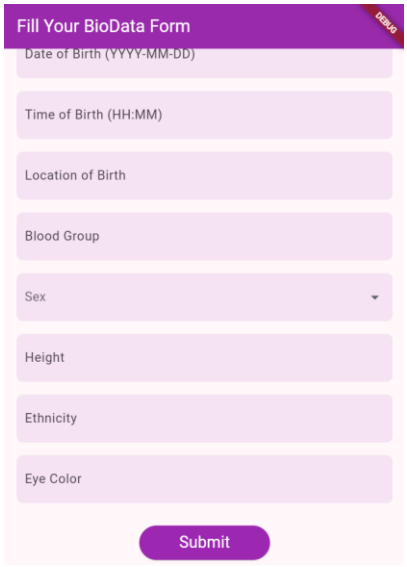












Project Workflow

1. Requirements Analysis - Identify and document key features and functionalities required for the app.

2. UI/UX Design Using Figma - Create wireframes and prototypes for the app in Figma.

3. Implementing UIs Using Flutter - Convert Figma designs into functional UI components using Flutter.

4. Backend Development with Flutter, Dart, MongoDb and Firebase - Set up local data storageusing MongoDB for persistence of user data.

5. Version Control with GitHub - Regularly commit and push changes to keep track of updates and collaboration.

Q & A

Does the application meet the requirements outlined above?

Yes, the application meets the requirements outlined in the project specification.

Requirement	Availability
User Interface (UI) Requirements	✓
Account System(PAC)	✓
Data Collection	✓
Data Security and Privacy	✓
Account Recovery	✓
Unsubscription and Data Removal	✓
Edit Email	
Edit Password	

What technologies are utilized and explained in the document?

1. **Flutter** : Flutter was used as the primary framework for developing the mobile application. It provides a fast and efficient way to create cross-platform apps with a single codebase.
2. **Dart** : Dart is the programming language used with Flutter to implement both the front-end UI logic and the back-end processes.
3. **Firebase** : Firebase is a platform by Google that provides tools for app development, including authentication, real-time databases, and cloud functions, allowing easy integration with Flutter for both front-end and back-end tasks.
4. **MongoDB:** MongoDB is a NoSQL database that stores data in flexible, JSON-like documents instead of traditional tables. It allows for easy scaling and fast performance, making it ideal for handling large amounts of unstructured data in modern applications.