Biomark Project Development Overview

Project Overview

The Biomark project is a mobile application aimed at streamlining voluntary data collection users, *to build predictive machine learning models for personalising digital services.* by providing functionalities such as user authentication, data management, and profile maintenance. The app uses modern frameworks and tools, including Flutter for cross-platform mobile development, Firebase for backend services, and SQLite for local data storage, ensuring smooth and efficient functionality.

A comprehensive explanation video of this project can be accessed here: https://youtu.be/LrCvDk-Ew0g

---

1. User Authentication

The application utilizes Firebase Authentication to handle user registration, login, and password recovery. Firebase provides a simple way to manage user credentials, ensuring that the app can securely store and retrieve users' information.

- Sign-Up: Users register by providing an email and password.
- Sign-In: Users authenticate using the email and password associated with their account.
- Password Recovery: In case of a forgotten password, users can reset it using their email, managed through Firebase Authentication.

---

2. Firestore Integration

The app uses Firebase Firestore to store user data and preferences, ensuring that user information is persistently available across sessions. The Firestore database holds crucial information like:

- User Health Data: Medical history, prescriptions, test results, etc.
- Profile Details: User’s name, date of birth, contact information, and health-related attributes.
- Security Questions: Custom security questions are stored after they are encrypted to enhance security. These security questions are later used for password recovery.

---

3. SQLite for Local Storage

To ensure offline availability of certain data, SQLite is used to store users' security question answers on the device. This data is encrypted to maintain security and is queried when the user attempts to recover their account by answering the security questions.

•Encrypted Storage: Security answers are encrypted before being saved to SQLite, protecting sensitive information from unauthorized access.
•Security Question Verification: When a user tries to recover their account, their input is compared against the encrypted values stored in SQLite.

---

4. Password Reset Functionality

The application features a password reset functionality, allowing users to update their passwords securely. The reset process involves validating the user's identity using security questions, and once verified, the new password is updated in Firebase Authentication.

---

5. Encryption for Security

To ensure the confidentiality of sensitive information such as the answers to security questions, the application uses encryption techniques:

•AES Encryption: Security question answers are encrypted using AES before they are stored in SQLite.
•Decryption: During account recovery, the app decrypts the answers stored in SQLite for comparison with the answers provided by the user.

---

6. Database Helper

The DatabaseHelper class provides an interface to interact with SQLite. This class contains methods for initializing the database, creating tables, and inserting/retrieving data:

•saveSecurityQuestions: This method stores security question answers in the SQLite database.
•querySecurityQuestions: This method is used to fetch the answers when verifying the user during account recovery.

---

7. Flutter UI Implementation

The front-end of the application is built using Flutter, ensuring cross-platform compatibility with both Android and iOS. The app features:

•Login/Sign-Up Screens: Users can create an account or log in using their credentials.
•Recovery Screen: A screen where users can recover their account by answering security questions.
•Password Reset Screen: A screen that allows users to reset their password after successfully verifying their identity.

---

8. Integration with Firebase Authentication

Firebase Authentication is integrated to handle user sign-ups, sign-ins, and password resets. The app interacts with Firebase to:

- Sign up new users: Email and password-based registration.
- Authenticate users: Verifying users when they attempt to log in.
- Reset passwords: Managing the password reset process securely.

---

9. Error Handling

To provide a smooth user experience, the app includes comprehensive error handling throughout the authentication and recovery processes. If errors are encountered, the app displays user-friendly messages via snack bars to inform the user about issues, such as incorrect credentials or security answers.

---

10. Testing and Debugging

The app was tested extensively to ensure that all functionalities work as expected. This included:

- Unit Testing: Testing the individual components like authentication and database operations.
- UI Testing: Ensuring that the user interface is responsive and user-friendly.
- Security Audits: Verifying that sensitive data is securely encrypted and stored.

---

11. Conclusion

The Biomark project successfully integrates Firebase for backend services and SQLite for local data management, providing a seamless and secure platform for users to manage their health-related data. The application's architecture ensures that sensitive information is encrypted and stored securely, making it both user-friendly and privacy-conscious.

12. Group Members

Mira A.M.H.	21020639
Wasfan M.W.M	21021023