In the case of Flutter, there are several techniques that we can use to store data. Different methods, ranging from the device storage to the use of remote storage, cloud storage. Here are some common storage management approaches in Flutter:

1. Local Storage

Flutter provides several options for local storage

Shared Preferences: The key-value store design is simple and convenient for storing small-scale data. It can be used respectively to store the user settings and preferences.
SQLite: A database tool that makes it possible to store local and find any needed data structure. Libraries like sqlite not only provide the interface for Flutter to use SQLite, but they are also responsible for the creation of the tables, the retrieval and the update of data.
File Storage: You can load files directly into the device through paths, libraries such as path_provider can give you the path to the file, I/O libraries standard can then read and write it.

2. Remote Storage
Firebase Firestore: A NoSQL cloud database that is well-suited for the real-time syncing and easy integration with Flutter framework.
Firebase Realtime Database: For real-time cloud-based database Firebase provides its databases as a second option, which have a Keys-Values model of data storage.
3. Cloud Storage
Firebase Storage: The cloud storage provided is exclusively for files and media at large with the capability to make seamless integration into the apps you make using Flutter.
Other cloud storage services: AWS S3 and Google Cloud Storage, providers, with space for unlimited growth for files, and media are some of the scalable Storage options available.
4. In-Memory Storage
Provider: Set up provider package of flutter to keep the state and the data cache in memory for lesser time for the access to them.

In the context of an app created using Flutter for a blood donation app, we can effectively manage the data by applying various storage techniques. Here are some options, along with their pros and cons for your project

1. Local Storage

Pros:
Fast access: Local storage which gives on-device fast data retrieval and storage without need of an external network is beneficial in the improvement of app performance.
Offline availability: Data stored at local level can be easily accessed even on the occasions of internet connection breakage.
Ease of implementation: Flutter has sqlite-based sqflite library that takes care of some routine stuff like file storage and file retrieval.

Cons:
Limited storage space: The mobile devices usually have minimal storage capacity and hence they place your local storage space in a difficult situation.
Data synchronization: Local storage and cloud server may share some data so proper data sync may be quite involved.

Thus this is not a practical option for our project

2. Remote storing system (for example, Firebase Firestore)
Pros:
Real-time data synchronization: Real-time data updates provided by Firebase Firestore is beneficial to blood bank donation app as data needs to be up-to-date due to real-time working nature of the application.
Easy integration with Flutter: Firebase is honoured for its ability to provide Flutter with good libraries and support.
Scalability: What cloud-based databases, like Firestore, offer is scalability to process the accelerating data request flows.

Cons:
Network dependency: Tends performance issues with net connectivity.
Data privacy and security: Privacy and security of data is an important issue which needs to be considered while using third party applications.

3. Cloud Storage

Pros:
Scalability: In contrast with traditional storage, solutions as Firebase Storage are easily scalable to deal with fluctuating data.
Cost-effective: Pay-as-you-use pricing plans could be instrumental in the budgeting.
Easy integration: Flutter has an integration with diverse cloud storage suppliers which makes the storage of files for mobile apps quite seamless.

Cons:

Network latency: Delay associated with transmission is another detrimental factor, particularly in case of large data transfers.
Security and privacy: Keeping personal and business data safe and private while using cloud services will be achieved by proper assessment.

4. Hybrid Approach

Pros:

Flexibility: A hybrid model features both on-site and cloud storage, to capitalise on the advantages of the two.
Offline-first: Data storing locally can be used offline and will be synchronized to cloud whenever the internet is accessible.
Scalability and performance: Presenting a mix of the fast local storage and the large data capacity in the remote storage.

Cons:
Complexity: Integrating and handling the conflict and synchronizes data among the local and remote storage has been an issue to figure out.

Thus, based on my analysis I feel cloud storage or remote storage like firebase is the best suited for our project

Home screen:

Allows you to select image and display, download image and user setting.

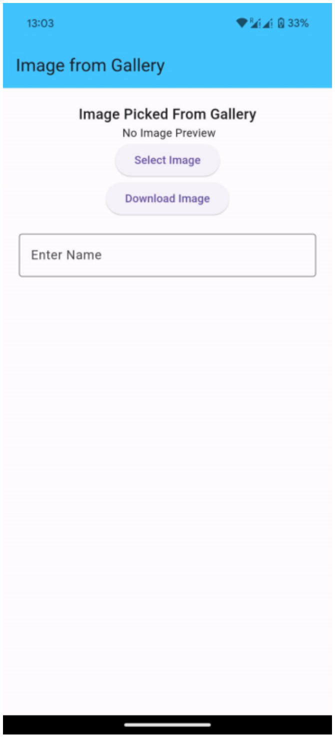


Image from Gallery

Image Picked From Gallery

No Image Preview

Store AND load a media item locally (e.g. music, video, image)

When you click on select image: it shows the gallery picker

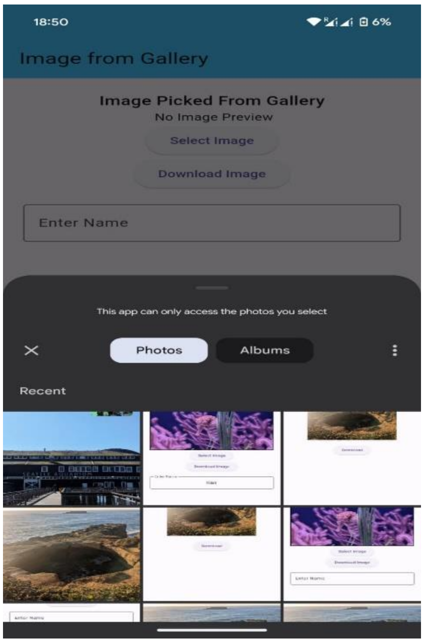


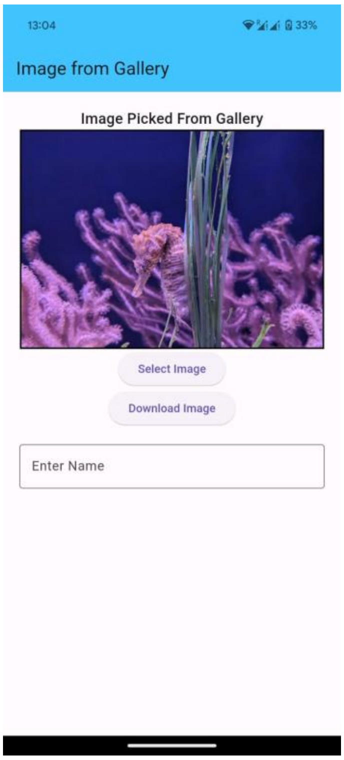
Image from Gallery

Image Picked From Gallery

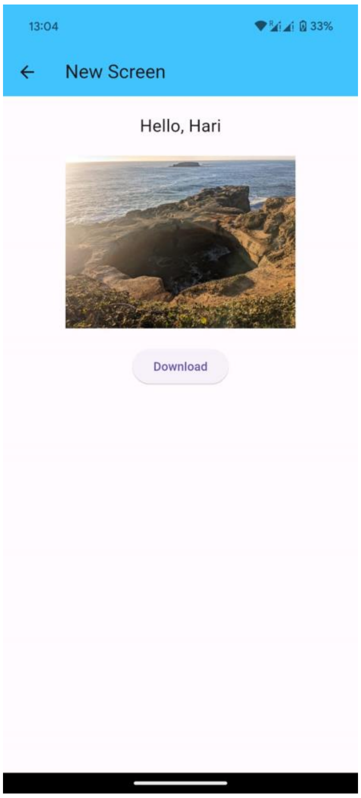
No Image Preview

This app can only access the photos you select

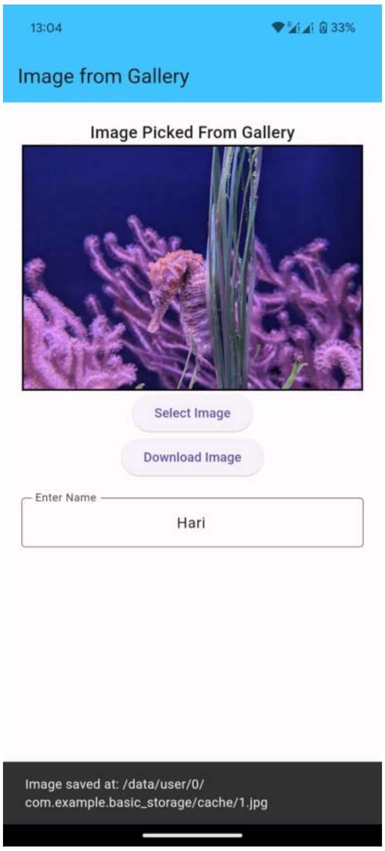
After you select the image: it will load the image as below



When you click on download image in the above it will take you to the next screen:



It shows the image bundled with the app, when you click on download it will download as 1.jpg, if you click again it will download as 2.jpg, its like a counter.

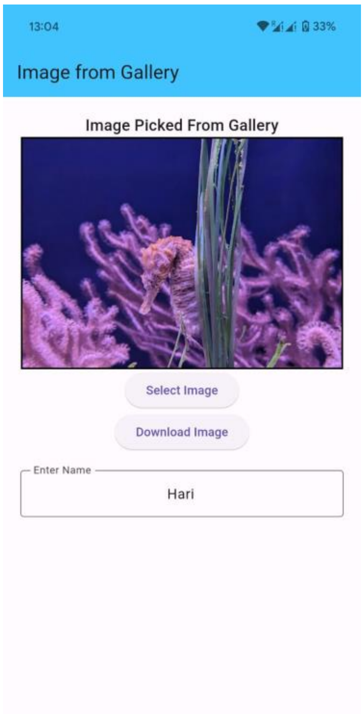


Enter Name
Hari

Image saved at: /data/user/0/
com.example.basic_storage/cache/1.jpg

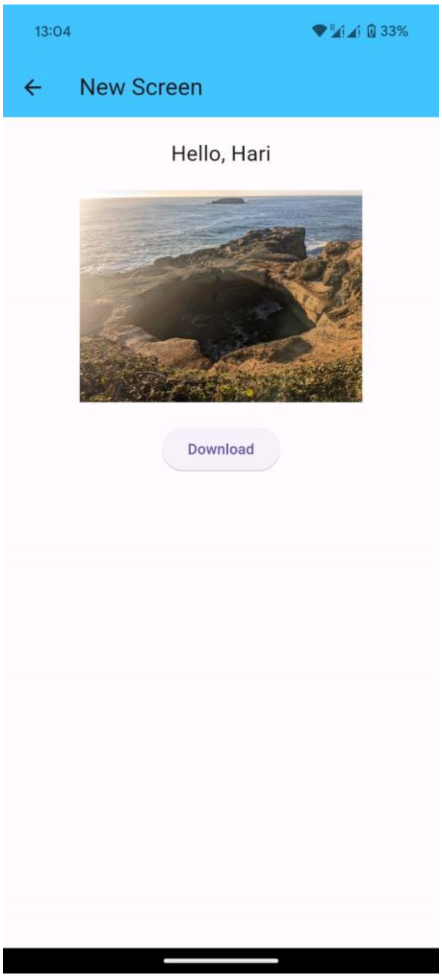
Store AND Load user settings locally:

In the home screen you can enter name, this will persist when you go back and forth and also when you close the phone.,



Enter Name
Hari

The name you enter will be also displayed in the next screen



Hello, Hari