Project-Name: Posts Application
Technologies Used: API, SharedPreferences, Bloc-StateMangement.

App Tasks

1.	CRUD-Operation(AddPost, UpdatePost, DeletePost, GetAllPost).
2.	Verify before sending any request to the server
3-	Cache data in Local DB and get data from it.
4-	Check the connection before opening the apps and connect to the server
5-	Handel all possible errors and exceptions
6-	Using Functional Programming by Equatable Package.

How My App Applies SOLID Principles :

1st.	✓ Bloc classes handle state management, and UseCase classes handle business logic. ✓ The core directory handles specific types of errors.
2nd.	✓ PostBloc Class can be extended without modifying its existing code
3rd.	✓ Objects of the base class (Post) can be replaced with objects of the derived class (PostModel) without affecting the correctness of the program.
4th.	✓ Failure and NetWorkInfo, segregate the responsibilities. They provide a clear contract for implementing specific functionalities.
5th.	✓ Domain layer depends on the abstraction class(Repository not Data layer).

How My App Uses Design Patterns:

1.	Factory Pattern: `s1.registerFactory(() => PostBloc(getAllPosts:s1()));`
2.	Singleton Pattern: `s1.registerLazySingleton(() => http.Client());`

✓ **Note:** The App only applies the clean code and does not care about UI, you can Watch y other Projects if you want..