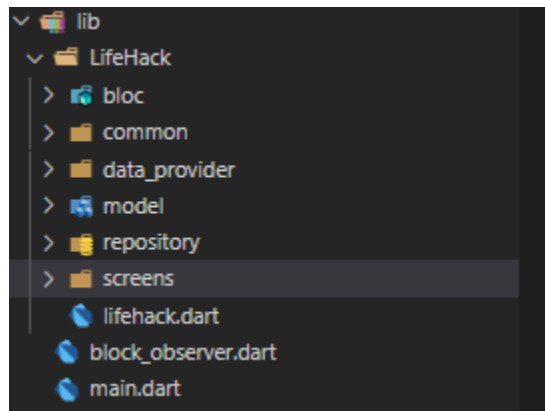# Lifehack

Dependencies

1. Required
   - Sqfilite
   - Path_provider
   - Flutter_bloc
   - Equitable
   - Dropdown_formfield
   - Share

The app's state management is based on the Bloc/Rx

I structured the app in the following folder structure **Lifehack** as parent structure



1. **Model**  folder contains lifhack.dart file that contains the model of the lifehack object

2. **Data_provider** folder contains codes that interact with the database

3. **Repository** folder contains codes that call the database **CRUD** operations by with the UI interact with the repository not the database

4. **Bloc** folder contains the application logic

   That is the event , the state and the bloc

5. **Screens** folder contains the UI of the application

## How the app's Logic words?

First the app starts on the **loading** state based on this state the bloc provider provides the states.

After the app runs the user can add **Create, Delete, Update** events , then the app's logic maps this events to the necessary states. And based on these states the application returns the data needed.

And then, the app displays this data based on the state.

## How the app works?

The app has its own lifehacks in the database when the app is installed first.

The app has Five sections on the home page.

1. All Lifehacks – in this the app displays all the category of life hacks in one page and users can view these life hacks one by one
2. Category – in this section the app displays life hacks based on the available categories
3. Favorite – when users see the life hacks they can like a life hack as they want and these life hacks will be displayed in the Favorite section.
4. My Favorite – Users can also add their own lifehack as they want. And these life hacks will be displayed in this section.
5. Share – this section shares the app via share methods.

Users can view theses lifehacks, like copy to clipboard and share the life hacks.