Local Storage Management :-

1. Storage Options Used:
    - Hive for local database storage.
    - Flutter Secure Storage for storing sensitive data like encryption keys.
2. App Initialization:
    - In the main function, a FutureBuilder is used to call the checkStatus() function.
    - checkStatus() checks whether the user is logged in or not. The login status is stored in a Hive box, updated during login or logout.
    - Based on the user's login status, they are navigated to either the Home Screen or Login Screen.
3. Handling Encryption Keys:
    - Key Storage: Encryption keys are stored in Flutter Secure Storage.
    - Key Management:
        - Before navigation, the getKey() function executes to check if an encrypted key is stored in secure storage.
        - If no key exists, a new encryption key is generated, and the Hive box is opened with this new key.
        - If a key is already present, the box is opened with the existing key.
4. Login Behavior:

- If the user is logged in, the encrypted key is already stored, and the existing Hive box is opened using that key.

5. Account Creation:
    - During account creation, a key-value pair of the user's email and phone number is stored in the Hive box.
6. Account Login:
    - During login, a key-value pair of the user's User ID and email is stored in the Hive box.
    - The authentication status is updated to Logged In in another Hive box.
7. Account Logout:
    - The clearKey() function is called to:
        - Remove the encryption key from secure storage.
        - Close the existing open Hive box.
    - The authentication status is updated to Logged Out in the Hive box.
    - The user is navigated back to the Login Screen.