# SỬ DỤNG FIREBASE(CRUD)

1. Thêm document

```dart
Future<void> addUserToFB(
    {required String name,
    required String nickname,
    required DateTime birthday}) async {
  // Add a new document with random id to collection "User"
  final docUser = FirebaseFirestore.instance
      .collection('User'); // reference to collection "User"
  final json = {
    'id': docUser.id, // use id of document as id of user
    'name': name,
    'nickname': nickname,
    'birthday': birthday
  };
  await docUser.add(json);

  // Add a new document with id my-id to collection "User"
  final docUser = FirebaseFirestore.instance.collection('User');
  await docUser.doc("my-id").set(json); // reference to document "my-id" and add json
}
```

2. Lấy dữ liệu từ Firebase:
    2.1. Dùng StreamBuilder:

```dart
Stream<List<User>> getUserFromFBByFirstWay() {
  // tạo stream
  final docUser = FirebaseFirestore.instance
      .collection('User'); // reference to the document User

  // docUser.snapshots(): get snapshots real time
  // snapshot: collection at a specific time
  return docUser.snapshots().map((snapshot) {
    return snapshot.docs.map((doc) => User.fromJson(doc.data())).toList();
    // .docs: to get list of all documents in collection User in json
    // doc.data() to get the data of document in json
  });
}
```

    2.2. Đọc dữ liệu ra list:

```dart
getUserFromFBBySecondWay() async {
  final docUser = FirebaseFirestore.instance.collection('User');
  final snapshot = await docUser.get();
  list = snapshot.docs.map((doc) => User.fromJson(doc.data())).toList();
}
```

    2.3. Đọc dữ liệu bằng ID:

```dart
Future<User?> readUserByID() async {
  final docUser = FirebaseFirestore.instance.collection('User').doc("my-id");
  final snapshot = await docUser.get();
  if (snapshot.exists) {
    return User.fromJson(snapshot.data()!);
  }
}
```

*** ĐƯA DỮ LIỆU VÀO

```dart
// dùng StreamBuilder để đưa stream vào listview
      Expanded(
        child: StreamBuilder<List<User>>(
          stream: getUserFromFBByFirstWay(),
          builder: (context, snapshot) {
            if (snapshot.hasError) {
              return Text('Error: ${snapshot.error}');
            } else if (snapshot.hasData) {
              final users = snapshot.data!;

              return ListView(
                shrinkWrap: true,
                children: users.map((user) {
                  return ListTile(
                    title: Text(user.name),
                    subtitle: Text(user.nickName),
                    trailing: Text(user.birthday.toString()),
                  );
                }).toList(),
              );
            } else {
              return const Center(child: CircularProgressIndicator());
            }
          }),
      ),
```

```dart
// Dùng FutureBuilder để đưa 1 doc ra
      Expanded(
        child: FutureBuilder<User?>(
          future: readUserByID(),
          builder: (context, snapshot) {
            if (snapshot.hasError) {
              return Text('Error: ${snapshot.error}');
            } else if (snapshot.hasData) {
              final user = snapshot.data;

              return user == null
```

```
                      ? const Center(child: Text('No user'))
                      : ListTile(
                          title: Text(user.name),
                          subtitle: Text(user.nickName),
                          trailing: Text(user.birthday.toString()),
                      );
                } else {
                  return const Center(child: CircularProgressIndicator());
                }
            }))
```

3. Update:

```
updateUser() {
  final docUser = FirebaseFirestore.instance.collection('User').doc("my-id");
  docUser.update({
    // 'name' : FieldValue.delete(), // delete a field of doc
    'name': 'Nguyen Van A',
    'nickname': 'NVA',
    'birthday': DateTime.now()
  });
}
```

4. Delete:

```
delteUser() {
    final docUser = FirebaseFirestore.instance.collection('User').doc("my-id");
    docUser.delete();
}
```