배달 앱(패캠마트)

2 Firebase 알아보기
Firebase Authentication

Firebase Authentication

설치 필요 패키지

1. https://pub.dev/packages/firebase_core
2. https://pub.dev/packages/firebase_auth
3. https://pub.dev/packages/google_sign_in

```
dependencies:
  firebase_core: ^2.14.0
  google_sign_in: ^6.1.4
  firebase_auth: ^4.6.3
```

Firebase Authentication

익명 로그인/인증하기

```
try {
  final userCredential = await FirebaseAuth.instance.signInAnonymously();
  print("Signed in with temporary account.");
} on FirebaseAuthException catch (e) {
  switch (e.code) {
    case "operation-not-allowed":
      print("Anonymous auth hasn't been enabled for this project.");
      break;
    default:
      print("Unknown error.");
  }
}
```

Firebase Authentication

이메일/비밀번호 회원가입(등록)

```
try {
  final credential = await FirebaseAuth.instance.createUserWithEmailAndPassword(
    email: emailAddress,
    password: password,
  );
} on FirebaseAuthException catch (e) {
  if (e.code == 'weak-password') {
    print('The password provided is too weak.');
  } else if (e.code == 'email-already-in-use') {
    print('The account already exists for that email.');
  }
} catch (e) {
  print(e);
}
```

Firebase Authentication

이메일/비밀번호 로그인

```
Future signInEmailAndPassword() async {
    try {
      final email = emailTextEditingController.text.trim();
      final pwd = pwdTextEditingController.text.trim();
      final credential =
          await auth.signInWithEmailAndPassword(email: email, password: pwd);
      print("SignIn: ${credential.toString()}");
      setState(() {
        user = credential.user;
      });
    } on FirebaseAuthException catch (e) {
      if (e.code == 'weak-password') {
        print('The password provided is too weak.');
      } else if (e.code == 'email-already-in-use') {
        print('The account already exists for that email.');
      }
    } catch (e) {
      print(e);
    }
}
```

Firebase Authentication

구글 로그인

```dart
import 'package:google_sign_in/google_sign_in.dart';

Future<UserCredential> signInWithGoogle() async {
  // 구글 인증 시작하기
  final GoogleSignInAccount? googleUser = await GoogleSignIn().signIn();
  // 구글 로그인 결과 받아 처리하기
  final GoogleSignInAuthentication? googleAuth = await googleUser?.authentication;
  // 새로운 credential 정보 생성
  final credential = GoogleAuthProvider.credential(
    accessToken: googleAuth?.accessToken,
    idToken: googleAuth?.idToken,
  );
  return await FirebaseAuth.instance.signInWithCredential(credential);
}
```