1. สร้าง Project
   1. สร้าง Project ใหม่
   2. ลบข้อมูลในไฟล์ lib/main.dart แทนที่ด้วย และแก้ไข path ของ ไฟล์ login.dart ตามชื่อโฟลดอร์ของ project

import 'package:flutter/material.dart';

import 'package:lab8/login.dart';

import 'package:firebase\_core/firebase\_core.dart';

void main() async {

  WidgetsFlutterBinding.ensureInitialized();

  await Firebase.initializeApp();

  runApp(const MyApp());

}

class MyApp extends StatelessWidget {

  const MyApp({Key? key}) : super(key: key);

  // This widget is the root of your application.

  @override

  Widget build(BuildContext context) {

    return buildMaterialApp();

  }

  MaterialApp buildMaterialApp() {

    return MaterialApp(

      title: 'Flutter Demo',

      theme: ThemeData(

        primarySwatch: Colors.amber,

      ),

      initialRoute: '/',

      routes: {

        '/': (context) => LoginPage(),

        // '/register': (context) => RegisterPage(),

      },

    );

  }

}

1. สร้าง file ใหม่ ชื่อ login.dart เพื่อใช้เป็น หน้า login ตามโปรแกรมด้านล่าง

import 'package:flutter/material.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

class LoginPage extends StatefulWidget {

  const LoginPage({Key? key}) : super(key: key);

  @override

  State createState() => \_LoginPageState();

}

class \_LoginPageState extends State<LoginPage> {

  final \_formstate = GlobalKey<FormState>();

  String? email;

  String? password;

  final auth = FirebaseAuth.instance;

  @override

  Widget build(BuildContext context) {

    return Scaffold(

        body: Form(

      autovalidateMode: AutovalidateMode.always,

      key: \_formstate,

      child: ListView(

        children: <Widget>[

          emailTextFormField(),

          passwordTextFormField(),

          loginButton(),

          registerButton(context),

        ],

      ),

    ));

  }

  ElevatedButton registerButton(BuildContext context) {

    return ElevatedButton(

      // ignore: prefer\_const\_constructors

      child: Text('Register new account'),

      onPressed: () {

        print('Goto  Regis pagge');

        Navigator.pushNamed(context, '/register');

      },

    );

  }

  ElevatedButton loginButton() {

    return ElevatedButton(child: Text('Login'), onPressed: () {});

  }

  TextFormField passwordTextFormField() {

    return TextFormField(

      onSaved: (value) {

        password = value!.trim();

      },

      validator: (value) {

        if (value!.length < 8) {

          return 'Please Enter more than 8 Character';

        } else {

          return null;

        }

      },

      obscureText: true,

      keyboardType: TextInputType.text,

      decoration: const InputDecoration(

        labelText: 'Password',

        icon: Icon(Icons.lock),

      ),

    );

  }

  TextFormField emailTextFormField() {

    return TextFormField(

      onSaved: (value) {

        email = value!.trim();

      },

      validator: (value) {

        if (!validateEmail(value!)) {

          return 'Please fill in E-mail field';

        } else {

          return null;

        }

      },

      keyboardType: TextInputType.emailAddress,

      textInputAction: TextInputAction.next,

      decoration: const InputDecoration(

        labelText: 'E-mail',

        icon: Icon(Icons.email),

        hintText: 'x@x.com',

      ),

    );

  }

  bool validateEmail(String value) {

    RegExp regex = RegExp(

        r'^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)\*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$');

    return (!regex.hasMatch(value)) ? false : true;

  }

}

2.1 Extract method ของ Textfield และ Elevated button เพื่อให้ง่ายต่อการจัดการ code แล้ว run flutter run ที่ terminal และแสดงผลลัพธ์

|  |
| --- |
|  |

1. แก้ไข onPressed ของ login button ใน login.dart เพื่อนำค่า email และ password ส่งไป verify ที่ firebase

ElevatedButton loginButton() {

    return ElevatedButton(

        child: const Text('Login'),

        onPressed: () async {

          if (\_formstate.currentState!.validate()) {

            print('Valid Form');

            \_formstate.currentState!.save();

            try {

              await auth

                  .signInWithEmailAndPassword(

                      email: email!, password: password!)

                  .then((value) {

                if (value.user!.emailVerified) {

                  ScaffoldMessenger.of(context).showSnackBar(

                      const SnackBar(content: Text("Login Pass")));

                } else {

                  ScaffoldMessenger.of(context).showSnackBar(

                      const SnackBar(content: Text("Please verify email")));

                }

              }).catchError((reason) {

                ScaffoldMessenger.of(context).showSnackBar(

                    const SnackBar(content: Text("Login or Password Invalid")));

              });

            } on FirebaseAuthException catch (e) {

              if (e.code == 'user-not-found') {

                print('No user found for that email.');

              } else if (e.code == 'wrong-password') {

                print('Wrong password provided for that user.');

              }

            }

          } else {

            print('Invalid Form');

          }

        });

  }

3.1 กด r เพื่อใช้ hot reload ที่ terminal และแสดงผลลัพธ์ อธิบายการทำงานของ ส่วนการ login

|  |
| --- |
|  |

1. สร้างไฟล์ register.dart โดยโปรแกรมดังนี้ พร้อมทั้งแก้ไข path ของ login.dart ตามชื่อ folder ของโปรเจค

import 'package:lab65\_8/login.dart';

import 'package:flutter/material.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

class RegisterPage extends StatefulWidget {

  const RegisterPage({Key? key}) : super(key: key);

  @override

  State createState() => \_RegisterPageState();

}

class \_RegisterPageState extends State<RegisterPage> {

  final \_formstate = GlobalKey<FormState>();

  TextEditingController email = TextEditingController();

  TextEditingController password = TextEditingController();

  final auth = FirebaseAuth.instance;

  @override

  Widget build(BuildContext context) {

    return Scaffold(

        body: Form(

      key: \_formstate,

      child: ListView(

        children: <Widget>[

          buildEmailField(),

          buildPasswordField(),

          buildRegisterButton(),

        ],

      ),

    ));

  }

  ElevatedButton buildRegisterButton() {

    return ElevatedButton(

      child: const Text('Register'),

      onPressed: () {},

    );

  }

  TextFormField buildPasswordField() {

    return TextFormField(

      controller: password,

      validator: (value) {

        if (value!.length < 8) {

          return 'Please Enter more than 8 Character';

        } else {

          return null;

        }

      },

      obscureText: true,

      keyboardType: TextInputType.text,

      decoration: const InputDecoration(

        labelText: 'Password',

        icon: Icon(Icons.lock),

      ),

    );

  }

  TextFormField buildEmailField() {

    return TextFormField(

      controller: email,

      validator: (value) {

        if (value!.isEmpty) {

          return 'Please fill in E-mail field';

        } else {

          return null;

        }

      },

      keyboardType: TextInputType.emailAddress,

      textInputAction: TextInputAction.next,

      decoration: const InputDecoration(

        labelText: 'E-mail',

        icon: Icon(Icons.email),

        hintText: 'x@x.com',

      ),

    );

  }

}

4.1 แก้ไข onPressed ของ register button ใน register.dart เพื่อนำค่า email และ password ส่งไป เพิ่ม user ที่ firebase และกลับไปยังหน้า login

  ElevatedButton buildRegisterButton() {

    return ElevatedButton(

      child: const Text('Register'),

      onPressed: () async {

        print('Register new account');

        if (\_formstate.currentState!.validate()) print(email.text);

        print(password.text);

        final user = await auth.createUserWithEmailAndPassword(

            email: email.text.trim(), password: password.text.trim());

        await user.user!.sendEmailVerification();

        Navigator.pushAndRemoveUntil(

            context,

            MaterialPageRoute(builder: (context) => const LoginPage()),

            ModalRoute.withName('/'));

      },

    );

  }

4.2 แก้ไข route ที่ main.dart เพื่ม route register page โดยเอาเครื่องหมาย // ออก และ import register.dart

4.3 กด Rเพื่อใช้ Restart ที่ terminal และแสดงผลลัพธ์ของ resigter และผลลัพธ์ที่ firebase

|  |
| --- |
|  |

4.4 จงอธิบาย async และ await และอธิบายการใช้ async กับ await ในข้อ 4.1

|  |
| --- |
|  |

1. สร้างไฟล์ homepage.dart ในส่วนของ body

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

class Homepage extends StatefulWidget {

  const Homepage({Key? key}) : super(key: key);

  @override

  State createState() => \_HomepageState();

}

class \_HomepageState extends State<Homepage> {

  @override

  Widget build(BuildContext context) {

    return Scaffold(

        appBar: AppBar(

          title: const Text("Firebase Authen App",

              style: TextStyle(color: Colors.white)),

          actions: <Widget>[

            IconButton(

                onPressed: () {

              },

                icon: const Icon(Icons.exit\_to\_app))

          ],

        ),

        body: Center(

            child:

                Column(mainAxisSize: MainAxisSize.min, children: const <Widget>[

          Text("Hello", style: TextStyle(fontSize: 26)),

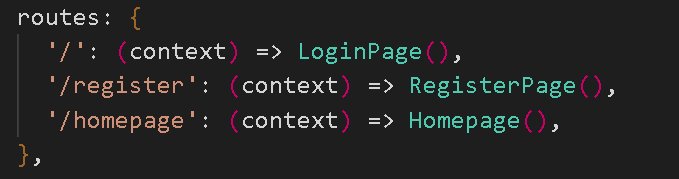
        ])));

  }

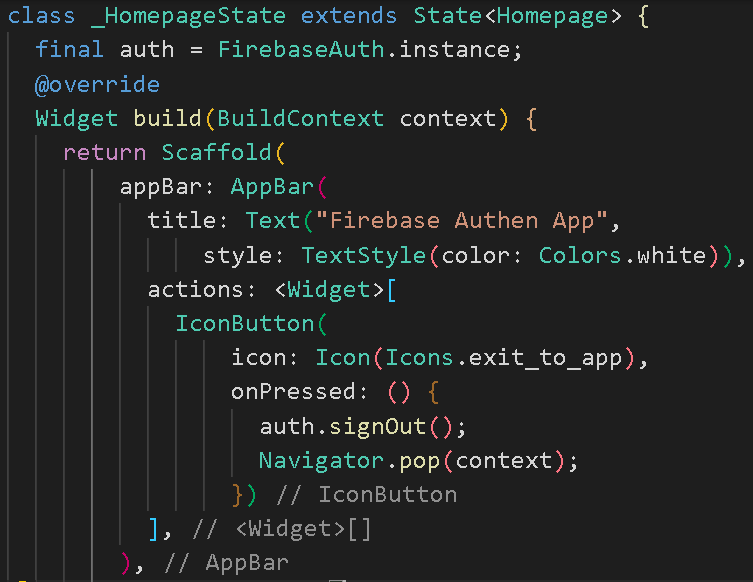
}

5.1 เรียกใช้งานหน้า homepage เมื่อ login ผ่าน ที่ loginbutton ใน login.dart

5.2 เพิ่ม route homepage ที่ main.dart



5.3 เพิ่ม ปุ่ม Icon logout ที่ appbar ใน homepage.dart และกด r เพื่อใช้ hot reload ที่ terminal และแสดงผลลัพธ์



|  |
| --- |
|  |