**BÁO CÁO DỰ ÁN ỨNG DỤNG FLUTTER TÍCH HỢP FIREBASE**

**1. Mục tiêu dự án**

Mục tiêu của dự án này là xây dựng một ứng dụng Flutter đơn giản cho phép người dùng quản lý danh sách sản phẩm. Ứng dụng sẽ tích hợp với Firebase để thực hiện các thao tác CRUD (Create - Thêm, Read - Đọc, Update - Cập nhật, Delete - Xóa) trên dữ liệu thời gian thực. Dữ liệu bao gồm tên sản phẩm, loại sản phẩm, giá và hình ảnh của sản phẩm.

**2. Các bước thực hiện**

**2.1. Tích hợp Firebase vào dự án Flutter**

**Bước 1: Chuẩn bị môi trường làm việc**

Trước tiên, cài đặt **Flutter SDK** và thiết lập môi trường phát triển cho Flutter.

**Bước 2: Cài đặt và chạy Flutter CLI**

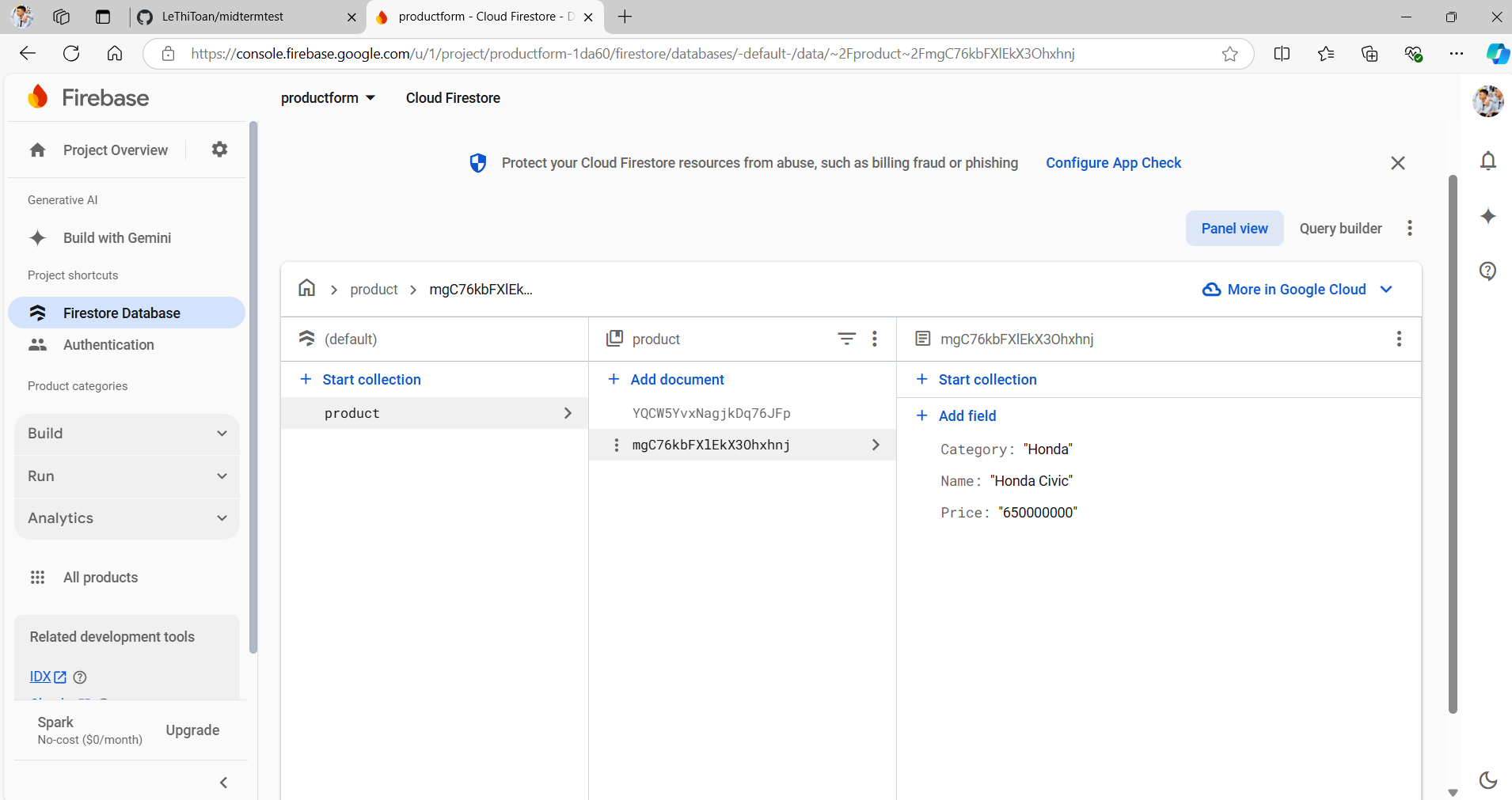
**-Tạo project flutter trên visual studio**

Chạy lệnh sau trong terminal để tạo một dự án Flutter mới:

flutter create gk\_flutter

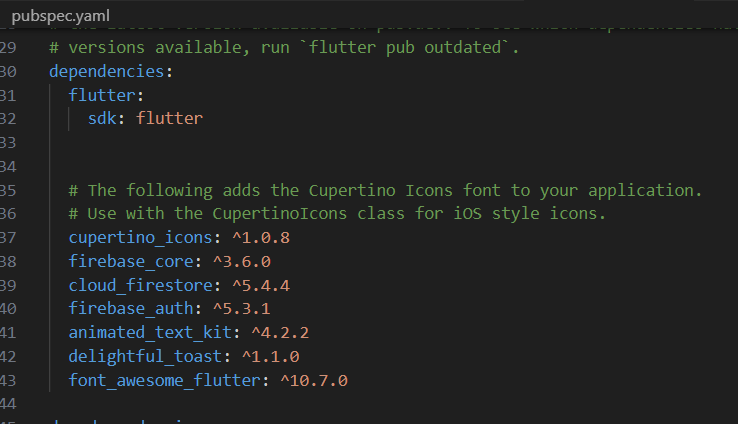
**Bước 3: Tích hợp Firebase vào dự án**

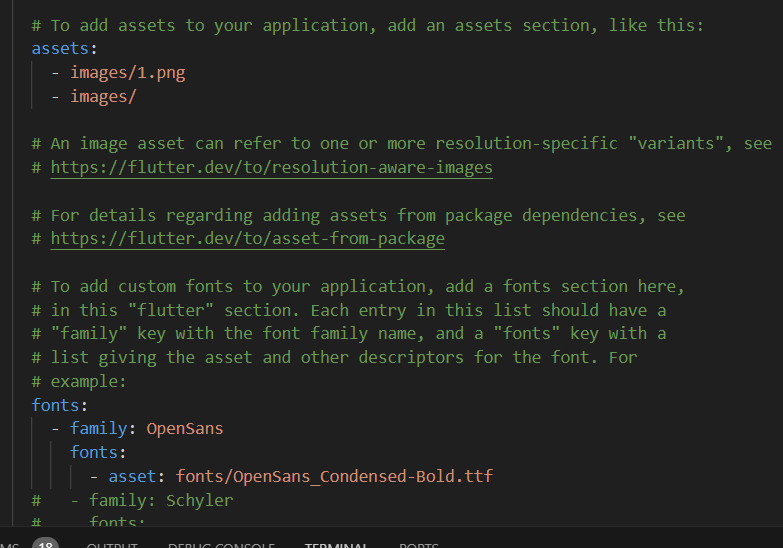
Truy cập vào Firebase Console, tạo một dự án Firebase mới và thêm ứng dụng Flutter vào Firebase.



 Sau khi thêm Firebase vào dự án, tải xuống các tệp cấu hình (google-services.json cho Android và GoogleService-Info.plist cho iOS) và thêm vào thư mục tương ứng trong dự án Flutter.

Thêm các thư viện Firebase vào tệp pubspec.yaml để cài đặt Firebase cho Flutter:





Sau đó chạy lệnh sau để cài đặt các thư viện:

flutter pub get

**2.2. Khởi tạo Firebase trong ứng dụng**

Trong file firebase\_options.dart, khởi tạo Firebase trong ứng dụng. Việc khởi tạo này là cần thiết để sử dụng các dịch vụ của Firebase như Firebase Realtime Database và Firebase Storage.

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

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

import 'package:flutter/foundation.dart'

    show defaultTargetPlatform, kIsWeb, TargetPlatform;

/// Default [FirebaseOptions] for use with your Firebase apps.

///

/// Example:

/// ```dart

/// import 'firebase\_options.dart';

/// // ...

/// await Firebase.initializeApp(

///   options: DefaultFirebaseOptions.currentPlatform,

/// );

/// ```

class DefaultFirebaseOptions {

  static FirebaseOptions get currentPlatform {

    if (kIsWeb) {

      return web;

    }

    switch (defaultTargetPlatform) {

      case TargetPlatform.android:

        return android;

      case TargetPlatform.iOS:

        return ios;

      case TargetPlatform.macOS:

        return macos;

      case TargetPlatform.windows:

        return windows;

      case TargetPlatform.linux:

        throw UnsupportedError(

          'DefaultFirebaseOptions have not been configured for linux - '

          'you can reconfigure this by running the FlutterFire CLI again.',

        );

      default:

        throw UnsupportedError(

          'DefaultFirebaseOptions are not supported for this platform.',

        );

    }

  }

  static const FirebaseOptions web = FirebaseOptions(

    apiKey: 'AIzaSyAwhdonjDLA0YUL-Hx8kUh45ASJzlHwGK8',

    appId: '1:907615055976:web:5c59c5b168bfc9950fcbdd',

    messagingSenderId: '907615055976',

    projectId: 'productform-1da60',

    authDomain: 'productform-1da60.firebaseapp.com',

    storageBucket: 'productform-1da60.appspot.com',

    measurementId: 'G-6D316SD0T7',

  );

  static const FirebaseOptions android = FirebaseOptions(

    apiKey: 'AIzaSyC5OFPJjy7l5TckECidXJMzSTfcWTiAA-0',

    appId: '1:907615055976:android:9ddec9068457f26f0fcbdd',

    messagingSenderId: '907615055976',

    projectId: 'productform-1da60',

    storageBucket: 'productform-1da60.appspot.com',

  );

  static const FirebaseOptions ios = FirebaseOptions(

    apiKey: 'AIzaSyA8SKyBpX0iWUI8GANQuyuMaSLgONW4JBQ',

    appId: '1:907615055976:ios:c064584c7e923b800fcbdd',

    messagingSenderId: '907615055976',

    projectId: 'productform-1da60',

    storageBucket: 'productform-1da60.appspot.com',

    iosBundleId: 'com.example.productform',

  );

  static const FirebaseOptions macos = FirebaseOptions(

    apiKey: 'AIzaSyA8SKyBpX0iWUI8GANQuyuMaSLgONW4JBQ',

    appId: '1:907615055976:ios:c064584c7e923b800fcbdd',

    messagingSenderId: '907615055976',

    projectId: 'productform-1da60',

    storageBucket: 'productform-1da60.appspot.com',

    iosBundleId: 'com.example.productform',

  );

  static const FirebaseOptions windows = FirebaseOptions(

    apiKey: 'AIzaSyAwhdonjDLA0YUL-Hx8kUh45ASJzlHwGK8',

    appId: '1:907615055976:web:e754ea2bc1728d610fcbdd',

    messagingSenderId: '907615055976',

    projectId: 'productform-1da60',

    authDomain: 'productform-1da60.firebaseapp.com',

    storageBucket: 'productform-1da60.appspot.com',

    measurementId: 'G-277LBCFC1Z',

  );

}

**2.4. Tạo trang Home\_screen**

Đây là một trang chính của ứng dụng Flutter, cho phép người dùng xem danh sách sản phẩm được lưu trữ trong Firebase Firestore, cũng như thực hiện các thao tác thêm, chỉnh sửa và xóa sản phẩm.

import 'dart:ui';

import 'package:animated\_text\_kit/animated\_text\_kit.dart';

import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'package:flutter/material.dart';

import 'package:productform/showcustomdelightbar.dart';

class HomeScreen extends StatefulWidget {

  const HomeScreen({super.key});

  @override

  State<HomeScreen> createState() => \_HomeScreenState();

}

class \_HomeScreenState extends State<HomeScreen> {

  final TextEditingController \_nameController = TextEditingController();

  final TextEditingController \_categoryController = TextEditingController();

  final TextEditingController \_priceController = TextEditingController();

  CollectionReference products =

      FirebaseFirestore.instance.collection('product');

    void \_addProduct() {

  products.add({

    'Name': \_nameController.text,

    'Category': \_categoryController.text,

    'Price': \_priceController.text,

  }).then((\_) {

    // Show success toast

   showCustomDelightToastBar(context, "Thêm thành công sản phẩm !");

  }).catchError((error) {

    // Show error toast if there's an issue

    showCustomDelightToastBar(context, "Thêm sản phẩm thất bại !");

  });

  \_nameController.clear();

  \_categoryController.clear();

  \_priceController.clear();

}

      void \_deleteProduct(String productId) {

        products.doc(productId).delete();

      }

      void \_editProduct(DocumentSnapshot product) {

        \_nameController.text = product['Name'];

        \_categoryController.text = product['Category'];

        \_priceController.text = product['Price'];

        showDialog(context: context, builder: (context) {

          return AlertDialog(

            title: const Text('Edit product'),

            content: Column(

              mainAxisSize: MainAxisSize.min,

              children: [

                TextFormField(

                  controller: \_nameController,

                  decoration: const InputDecoration(labelText: 'Product Name'),

                ),

                const SizedBox(height: 8),

                TextFormField(

                  controller: \_categoryController,

                  decoration: const InputDecoration(labelText: 'Category Name'),

                ),

                 const SizedBox(height: 8),

                TextFormField(

                  controller: \_priceController,

                  decoration: const InputDecoration(labelText: 'Price'),

                ),

              ],

            ),

            actions: [

              TextButton(onPressed: (){

                Navigator.pop(context);

              }, child: const Text('Cancel')),

              ElevatedButton(onPressed: () {

                \_updateProduct(product.id);

              }, child: const Text('Update')),

            ],

          );

        });

      }

      void \_updateProduct(String productId) {

        products.doc(productId).update({

          'Name': \_nameController.text,

          'Category': \_categoryController.text,

          'Price': \_priceController.text,

        }

        );

      }

 @override

Widget build(BuildContext context) {

  return Scaffold(

    body: Stack(

      children: [

     ImageFiltered(

        imageFilter: ImageFilter.blur(sigmaX: 5.0, sigmaY: 5.0), // Điều chỉnh giá trị sigma để tăng/giảm độ mờ

        child: Container(

          height: double.infinity,

          width: double.infinity,

          decoration: const BoxDecoration(

            image: DecorationImage(

              image: AssetImage('images/bg4.webp'),

              fit: BoxFit.fill,

            ),

          ),

        ),

      ),

      SingleChildScrollView(

        child: Padding(

          padding: const EdgeInsets.only(bottom: 20,top: 20,left: 10, right: 10),

          child: Column(

            children: [

              Center(

                child: SizedBox(

                  width: MediaQuery.of(context).size.width\*0.8,

                  height: 50,

                   child: const  Row(children: [Text("Danh Sách",style: const TextStyle(

                          fontSize: 30,

                          color: Colors.blue,

                          fontFamily: 'OpenSans',

                        ),),SizedBox(width: 8,),Text("Sản Phẩm",style: const TextStyle(

                          fontSize: 30,

                          color: Color.fromARGB(255, 52, 51, 51),

                          fontFamily: 'OpenSans',

                        ),)],),

                ),

              ),

              const SizedBox(height: 10,),

              TextField(

                controller: \_nameController,

                decoration: const InputDecoration(labelText: 'Tên sản phẩm'),

              ),

              const SizedBox(height: 16),

              TextField(

                controller: \_categoryController,

                decoration: const InputDecoration(labelText: 'Loại sản phẩm'),

              ),

              const SizedBox(height: 16),

              TextField(

                controller: \_priceController,

                keyboardType: TextInputType.number,

                decoration: const InputDecoration(labelText: 'Giá sản phẩm'),

              ),

              const SizedBox(height: 16),

              ElevatedButton(

                onPressed: \_addProduct,

                child:   Center(

                              child: Container(

                                width: MediaQuery.of(context).size.width \* 0.7,

                                height: 50,

                                decoration: BoxDecoration(

                                    borderRadius: BorderRadius.circular(20)),

                                child: const Center(

                                    child: Text(

                                  "Thêm sản phẩm",

                                  style: TextStyle(

                                      fontSize: 15,

                                      color: Colors.blue),

                                )),

                              ),

                            ),

              ),

              const SizedBox(height: 16),

              StreamBuilder(

                stream: products.snapshots(),

                builder: (context, snapshot) {

                  if (snapshot.connectionState == ConnectionState.waiting) {

                    return const CircularProgressIndicator();

                  } else if (snapshot.hasError) {

                    return Text('Error: ${snapshot.error}');

                  } else if (!snapshot.hasData || snapshot.data!.docs.isEmpty) {

                    return const Text('No products available.');

                  }

                  return ListView.builder(

                    physics: const NeverScrollableScrollPhysics(),

                    shrinkWrap: true,

                    itemCount: snapshot.data!.docs.length,

                    itemBuilder: (context, index) {

                      var product = snapshot.data!.docs[index];

                      return Dismissible(

                        key: Key(product.id),

                        background: Container(

                          color: Colors.redAccent,

                          alignment: Alignment.centerRight,

                          padding: const EdgeInsets.only(right: 16),

                          child: const Icon(Icons.delete, color: Colors.white),

                        ),

                        onDismissed: (direction) {

                          \_deleteProduct(product.id);

                        },

                        direction: DismissDirection.endToStart,

                        child: Container(

                          decoration: BoxDecoration(  borderRadius: BorderRadius.circular(10),border: Border.all(color: Colors.black),

                            boxShadow: [BoxShadow(color: Colors.grey.withOpacity(0.2),spreadRadius: 1,blurRadius: 2)], ),

                          margin: const EdgeInsets.symmetric(vertical: 8),

                          child: ListTile(

                            title: Text('Tên sản phẩm : ${product['Name']}'),

                            subtitle: Column(

                              crossAxisAlignment: CrossAxisAlignment.start,

                              children: [

                                Text('Loại: ${product['Category']}'),

                                Text('Giá : ${product['Price']} VND'),

                              ],

                            ),

                            trailing: IconButton(

                              onPressed: () => \_editProduct(product),

                              icon: const Icon(Icons.edit),

                            ),

                          ),

                        ),

                      );

                    },

                  );

                },

              ),

            ],

          ),

        ),

      ),

      ]

    ),

  );

  }

}

extension on Object {

  get docs => null;

}

**3. Kết luận**

Dự án này đã hoàn thành việc tích hợp Firebase với ứng dụng Flutter và cung cấp các chức năng CRUD cho quản lý sản phẩm. Thông qua ứng dụng, người dùng có thể thêm, xem, chỉnh sửa và xóa sản phẩm.  
  
Source code: https://github.com/LeThiToan/midtermtest.git