BÁO CÁO THỰC HIỆN ỨNG DỤNG CRUD SỬ DỤNG NGÔN NGỮ FLUTTER KẾT HỢP VỚI FIREBASE

Họ tên: Trần Đức Công

Mã SV: 21IT535

1. Giới thiệu

-Ứng dụng này sử dụng Flutter kết hợp với Firebase để quản lý sản phẩm. Người dùng có thể thực hiện các chức năng như thêm, sửa, xóa sản phẩm. Các dữ liệu sản phẩm được lưu trữ trên Firebase Realtime Database và cập nhật theo thời gian thực.

2. Các bước thực hiện

-Tạo một dự án Firebase trên trang Firebase Console (<https://console.firebase.google.com>).

-Kết nối Firebase với ứng dụng Flutter bằng cách thêm file cấu hình google-services.json cho Android.

-Cài đặt các package trong Flutter bằng cách vô file pubspec.yaml và nhập các phiên bản cho package

firebase\_core: ^3.6.0

firebase\_auth: ^5.3.1

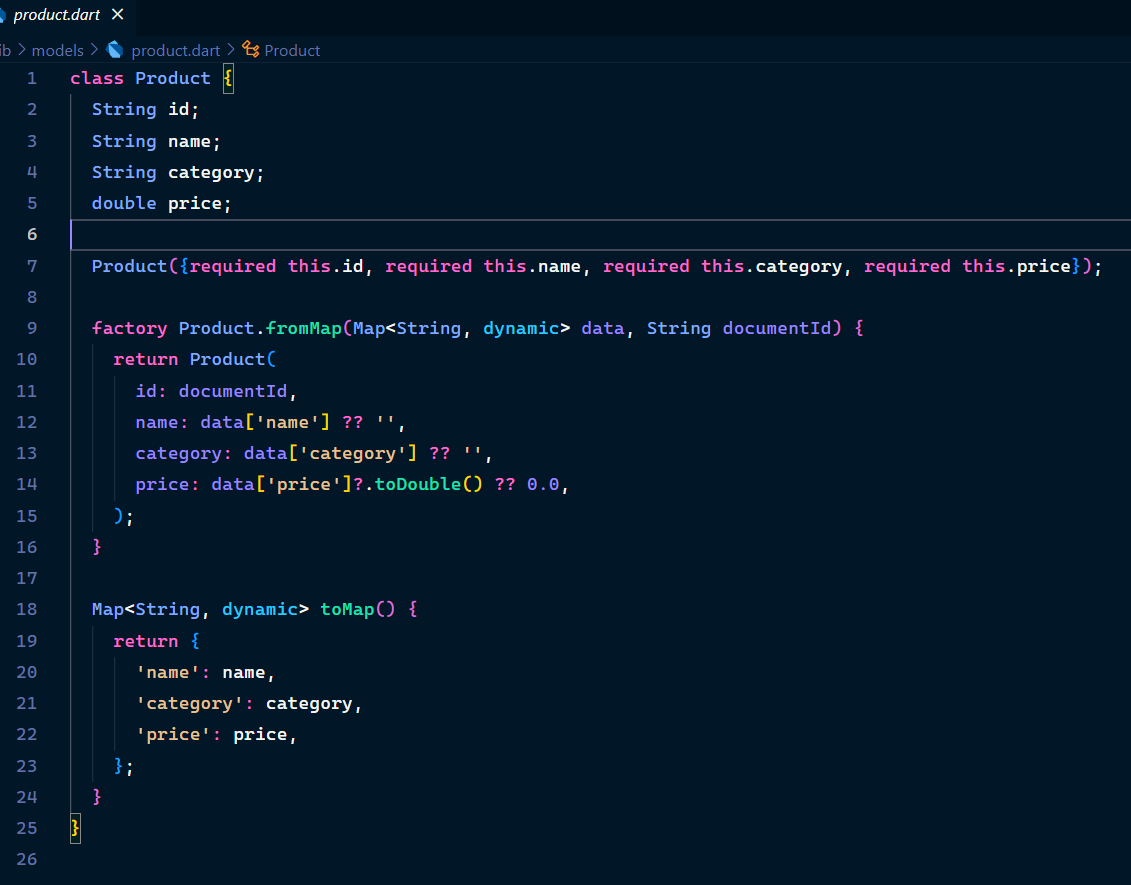
cloud\_firestore: ^5.4.4

firebase\_storage: ^12.3.3

image\_picker: ^0.8.7+4

firebase\_database

-Xây dựng ứng dụng



Product



Kết nối firebase



Thêm sửa xóa sản phẩm

import 'package:flutter/material.dart';

import '../models/product.dart';

import '../services/firestore\_service.dart';

class ProductScreen extends StatefulWidget {

  @override

  \_ProductScreenState createState() => \_ProductScreenState();

}

class \_ProductScreenState extends State<ProductScreen> {

  final FirestoreService \_firestoreService = FirestoreService();

  final TextEditingController \_nameController = TextEditingController();

  final TextEditingController \_categoryController = TextEditingController();

  final TextEditingController \_priceController = TextEditingController();

  List<Product> \_products = [];

  String? \_editingProductId;

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text('Product Management')),

      body: Padding(

        padding: const EdgeInsets.all(16.0),

        child: Column(

          children: [

            TextField(controller: \_nameController, decoration: InputDecoration(labelText: 'Tên Sản Phẩm')),

            TextField(controller: \_categoryController, decoration: InputDecoration(labelText: 'Loại sản phẩm')),

            TextField(controller: \_priceController, decoration: InputDecoration(labelText: 'Giá sản phẩm'), keyboardType: TextInputType.number),

            SizedBox(height: 10),

            ElevatedButton(

              onPressed: \_addOrUpdateProduct,

              child: Text(\_editingProductId == null ? 'Thêm sản phẩm' : 'Cập nhật sản phẩm'),

            ),

            SizedBox(height: 20),

            Text(

              'Danh sách sản phẩm',

              style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),

            ),

            Expanded(

              child: StreamBuilder<List<Product>>(

                stream: \_firestoreService.getProducts(),

                builder: (context, snapshot) {

                  if (!snapshot.hasData) return Center(child: CircularProgressIndicator());

                  \_products = snapshot.data!;

                  return ListView.builder(

                    itemCount: \_products.length,

                    itemBuilder: (context, index) {

                      return ListTile(

                        title: Row(

                          mainAxisAlignment: MainAxisAlignment.spaceBetween,

                          children: [

                            Expanded(

                              child: Column(

                                crossAxisAlignment: CrossAxisAlignment.start,

                                children: [

                                  Text('Tên SP: ${\_products[index].name}'),

                                  Text('Loại SP: ${\_products[index].category}'),

                                  Text('Giá SP: \$${\_products[index].price}'),

                                ],

                              ),

                            ),

                            Row(

                              mainAxisSize: MainAxisSize.min,

                              children: [

                                IconButton(

                                  icon: Icon(Icons.edit),

                                  onPressed: () => \_editProduct(\_products[index]),

                                ),

                                IconButton(

                                  icon: Icon(Icons.delete),

                                  onPressed: () => \_deleteProduct(\_products[index].id),

                                ),

                              ],

                            ),

                          ],

                        ),

                      );

                    },

                  );

                },

              ),

            ),

          ],

        ),

      ),

    );

  }

  void \_addOrUpdateProduct() {

    if (\_nameController.text.isNotEmpty && \_categoryController.text.isNotEmpty && \_priceController.text.isNotEmpty) {

      if (\_editingProductId == null) {

        // Thêm sản phẩm mới

        Product newProduct = Product(

          id: DateTime.now().millisecondsSinceEpoch.toString(),

          name: \_nameController.text,

          category: \_categoryController.text,

          price: double.parse(\_priceController.text),

        );

        \_firestoreService.addProduct(newProduct);

      } else {

        // Cập nhật sản phẩm

        Product updatedProduct = Product(

          id: \_editingProductId!,

          name: \_nameController.text,

          category: \_categoryController.text,

          price: double.parse(\_priceController.text),

        );

        \_firestoreService.updateProduct(updatedProduct);

        \_editingProductId = null; // Reset sau khi cập nhật

      }

      \_clearInputs();

      setState(() {}); // Cập nhật giao diện

    }

  }

  void \_clearInputs() {

    \_nameController.clear();

    \_categoryController.clear();

    \_priceController.clear();

  }

  void \_editProduct(Product product) {

    \_editingProductId = product.id;

    \_nameController.text = product.name;

    \_categoryController.text = product.category;

    \_priceController.text = product.price.toString();

    setState(() {}); // Cập nhật giao diện nếu cần

  }

  void \_deleteProduct(String productId) {

    \_firestoreService.deleteProduct(productId);

  }

}

Giao diện

* Giao diện danh sách sản phẩm

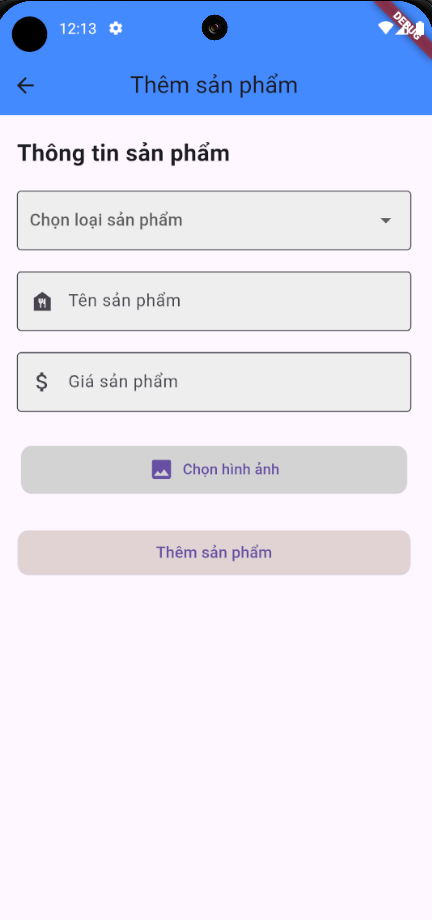
+Khi ấn vô ứng dụng, giao diện danh sách sản phẩm sẽ hiện ra, mới đầu thì chưa có sản phẩm nào nên sẽ hiện dòng chữ “Không có sản phẩm nào”



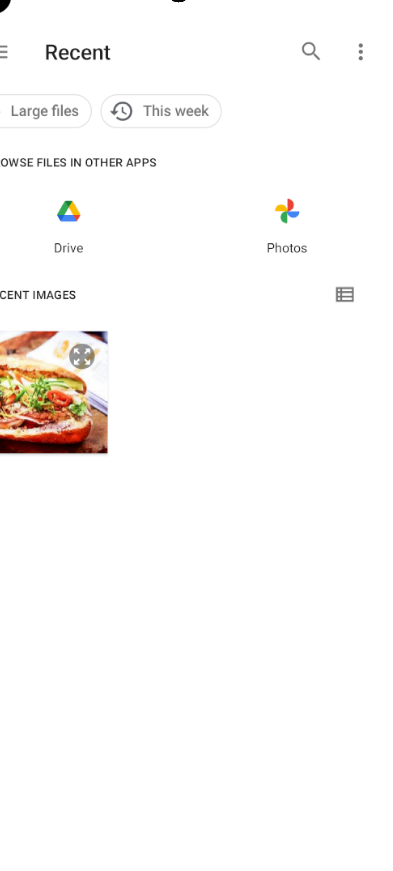
* Giao diện thêm sản phẩm

+Nhấn dấu cộng ở góc trên bên phải ở trang danh sách sản phẩm, sẽ chuyển hướng qua “Thêm sản phẩm”

+Ở giao diện này sẽ có các mục liên quan đến thông tin sản phẩm bao gồm Loại sản phẩm, tên sản phẩm, giá sản phẩm, và chọn hình ảnh để hiển thị cho sản phẩm đó  
+Sau khi nhấn nút thêm sản phẩm, sản phẩm sẽ được thêm vào mục “Danh sách sản phẩm” và giao diện quay trở lại trang ban đầu là trang danh sách sản phẩm



-Ở giao diện thêm sản phẩm, sẽ có mục chọn hình ảnh, khi nhấn vào, hình ảnh từ trong mục Download của điện thoại sẽ hiển thị các hình ảnh mà mình hiện có để có thể thêm sản phẩm vào



3. Kết luận

-Ứng dụng Flutter kết hợp với Firebase đã hoàn thiện các chức năng cốt lõi của một hệ thống quản lý sản phẩm. Nhờ tích hợp Firebase Realtime Database, ứng dụng đảm bảo khả năng lưu trữ và truy xuất dữ liệu theo thời gian thực, mang đến trải nghiệm nhanh chóng, mượt mà cho người dùng. Các chức năng như đăng nhập, đăng ký, thêm mới, chỉnh sửa và quản lý sản phẩm được xây dựng hiệu quả, giúp người dùng dễ dàng quản lý danh sách sản phẩm cá nhân một cách trực quan và thuận tiện. Một trong những điểm nổi bật của ứng dụng là việc xử lý hình ảnh sản phẩm thông qua thư viện image\_picker, cho phép người dùng dễ dàng chọn ảnh từ thư viện cá nhân, tăng tính tương tác và tính thẩm mỹ cho giao diện. Việc quản lý dữ liệu sản phẩm được thực hiện một cách nhanh chóng, đồng bộ và luôn cập nhật theo thời gian thực. Điều này giúp người dùng theo dõi sát sao thông tin sản phẩm, đảm bảo tính chính xác và kịp thời trong quá trình quản lý.

-Tổng thể, ứng dụng đã hoàn thành tốt các mục tiêu đề ra, cung cấp một hệ thống quản lý sản phẩm đầy đủ chức năng với khả năng mở rộng trong tương lai. Nhờ sự linh hoạt của Firebase, ứng dụng có thể dễ dàng nâng cấp để bổ sung thêm các tính năng phức tạp hơn, như phân tích dữ liệu, tối ưu hóa trải nghiệm người dùng, và tích hợp thêm các dịch vụ khác.