**1. Cấu hình Firebase trong Flutter**

Tạo dự án Firebase:

Truy cập vào Firebase Console và tạo dự án mới.

Thêm Firebase vào ứng dụng Flutter:

Thêm Firebase SDK vào dự án bằng cách cài đặt các package: firebase\_core, cloud\_firestore, firebase\_storage, firebase\_auth và firebase\_analytics (nếu cần).

Thêm file google-services.json vào thư mục android/app và cập nhật cấu hình Firebase trong dự án (bằng cách chỉnh sửa build.gradle).

**2. Cài đặt các package cần thiết**

Cài đặt các package cần thiết trong file pubspec.yaml:

dependencies:

flutter:

sdk: flutter

firebase\_core: ^latest\_version

cloud\_firestore: ^latest\_version

firebase\_storage: ^latest\_version

image\_picker: ^latest\_version

Chạy lệnh flutter pub get để tải các package.

**3. Chọn và tải ảnh lên Firebase Storage**

Sử dụng ImagePicker để cho phép người dùng chọn ảnh từ thư viện.

Sau khi chọn ảnh, sử dụng Firebase Storage để tải ảnh lên và lấy URL của ảnh.

Future<void> \_pickAndUploadImage() async {

final file = await ImagePicker().pickImage(source: ImageSource.gallery);

if (file == null) return;

setState(() {

\_selectedImage = File(file.path);

});

// Tải ảnh lên Firebase Storage

String fileName = DateTime.now().microsecondsSinceEpoch.toString();

Reference ref = FirebaseStorage.instance.ref().child('images/$fileName');

UploadTask uploadTask = ref.putFile(\_selectedImage!);

TaskSnapshot snapshot = await uploadTask;

downloadUrl = await snapshot.ref.getDownloadURL();

}

**4. CRUD với Cloud Firestore**

Thêm sản phẩm: Sau khi tải ảnh lên Firebase Storage, lưu thông tin sản phẩm và URL ảnh vào Firestore.

void \_addsanpham() async {

if (downloadUrl == null) {

print("Vui lòng chọn và tải ảnh trước khi thêm sản phẩm.");

return;

}

await \_sanpham.add({

'tensp': \_tensp.text,

'loaisp': \_loaisp.text,

'gia': \_giasp.text,

'image\_url': downloadUrl,

});

\_resetFields();

}

Hiển thị sản phẩm: Sử dụng StreamBuilder để lắng nghe thay đổi từ Firestore và hiển thị danh sách sản phẩm theo thời gian thực.

StreamBuilder(

stream: \_sanpham.snapshots(),

builder: (context, snapshot) {

if (!snapshot.hasData) {

return Center(child: CircularProgressIndicator());

}

return ListView.builder(

itemCount: snapshot.data?.docs.length ?? 0,

itemBuilder: (context, index) {

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

// Hiển thị sản phẩm

},

);

},

);

Chỉnh sửa sản phẩm: Cho phép người dùng chỉnh sửa tên, loại, giá, và thay đổi ảnh.

void \_editsanpham(DocumentSnapshot sanpham) {

\_tensp.text = sanpham['tensp'];

\_loaisp.text = sanpham['loaisp'];

\_giasp.text = sanpham['gia'];

downloadUrl = sanpham['image\_url'];

// Hiển thị Dialog chỉnh sửa sản phẩm

}

Xóa sản phẩm: Khi xóa sản phẩm, cần xóa cả thông tin sản phẩm từ Firestore và ảnh từ Firebase Storage.

void \_deletesanpham(String sanphamId, String imageUrl) async {

await \_sanpham.doc(sanphamId).delete();

if (imageUrl.isNotEmpty) {

Reference photoRef = FirebaseStorage.instance.refFromURL(imageUrl);

await photoRef.delete();

}

}

**5. Thiết kế giao diện người dùng**

Sử dụng các widget như TextFormField, IconButton, và ListView.builder để tạo giao diện thêm, chỉnh sửa, và xóa sản phẩm.