Để kết nối ứng dụng Flutter của bạn với Firebase và thực hiện bài tập quản lý sản phẩm như trên, bạn cần thực hiện theo các bước sau:

1. Tạo Dự Án Firebase

Đăng Nhập Firebase:

Truy cập Firebase Console.

Đăng nhập bằng tài khoản Google của bạn.

Tạo Dự Án Mới:

Nhấn vào "Add project" để tạo dự án mới.

Nhập tên dự án la giua\_ky và chọn các tùy chọn cần thiết, sau đó nhấn "Create project".

2. Thêm Firebase vào Ứng Dụng Flutter

Chọn Nền Tảng:

Sau khi dự án được tạo, chọn biểu tượng Android để thêm Firebase vào ứng dụng Android của bạn.

Cài Đặt Android App:

Nhập tên gói ứng dụng (ví dụ: com.example.productmanager).

Nhấn "Register app".

Tải Tệp google-services.json:

Tải tệp google-services.json và đặt nó vào thư mục android/app của dự án Flutter.

Cài Đặt Thư Viện Firebase:

Mở tệp android/build.gradle và thêm dòng sau vào phần dependencies:

groovy

Sao chép mã

id 'com.google.gms.google-services' version '4.4.2' apply false

nhất

Mở tệp android/app/build.gradle và thêm dòng sau vào cuối tệp:

groovy

Sao chép mã

apply plugin: 'com.google.gms.google-services'

Cài Đặt Thư Viện Flutter:

Mở tệp pubspec.yaml và thêm các thư viện sau vào phần dependencies:

yaml

Sao chép mã

dependencies:

flutter:

sdk: flutter

firebase\_core: ^3.6.0  
firebase\_database: ^11.1.4  
firebase\_storage: ^12.3.3  
cloud\_firestore: ^5.4.4 *# Adjust to the latest compatible version*image\_picker: ^1.1.2 *# Adjust to the latest compatible version*uuid: ^3.0.6

Chạy lệnh flutter pub get để cài đặt các thư viện.

3. Cấu Hình Firebase Realtime Database và Firebase Storage

Cấu Hình Realtime Database:

Quyền truy cập rules:

{

"rules": {

".read": true,

".write": true

}

}

Trong Firebase Console, chọn "Realtime Database" từ menu bên trái.

Nhấn vào "Create Database" và chọn "Start in Test Mode" (để dễ dàng phát triển, bạn có thể thay đổi quyền sau).

Nhấn "Next", sau đó "Done".

Cấu Hình Firebase Storage:

Quyền truy cập rules:

rules\_version = '2';

// Craft rules based on data in your Firestore database

// allow write: if firestore.get(

// /databases/(default)/documents/users/$(request.auth.uid)).data.isAdmin;

service firebase.storage {

match /b/{bucket}/o {

match /{allPaths=\*\*} {

allow read, write: if true;

}

}

}

Trong Firebase Console, chọn "Storage" từ menu bên trái.

Nhấn vào "Get Started" và chọn "Start in Test Mode" (có thể thay đổi quyền sau).

4.Mở Tệp AndroidManifest.xml

Tệp này nằm trong thư mục android/app/src/main/. Mở tệp AndroidManifest.xml trong trình soạn thảo mã của bạn.

Thêm Quyền Truy Cập

Thêm các quyền sau vào phần <manifest> trước thẻ <application>:

<uses-permission android:name="android.permission.INTERNET"/> <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/> <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

5. Build app:

**a. Xây dựng Giao diện Người Dùng**

**Tạo Widget chính**

* **MyApp**: Widget này định nghĩa cấu trúc của ứng dụng, bao gồm tiêu đề và chủ đề. Trong hàm build, nó trả về widget MaterialApp với một home là ProductManager.
* **ProductManager**: Là một StatefulWidget chứa trạng thái cho danh sách sản phẩm và các trường nhập liệu.

**Kết cấu Giao diện**

* Giao diện bao gồm:
  + Các trường nhập liệu cho tên sản phẩm, loại sản phẩm, và giá.
  + Một nút để chọn hình ảnh từ thư viện ảnh.
  + Một nút để thêm sản phẩm.
  + Danh sách sản phẩm được hiển thị bằng ListView.builder.

**b. Quản lý Sản phẩm**

**Thêm sản phẩm**

* **Lấy dữ liệu từ người dùng**: Sử dụng TextEditingController để quản lý dữ liệu từ các trường văn bản.
* **Kiểm tra và lưu thông tin**:
  + Kiểm tra xem các trường có dữ liệu hay không.
  + Tải hình ảnh lên Firebase Storage và lưu đường dẫn hình ảnh vào Realtime Database:

dart

Sao chép mã

Future<void> addProduct() async {

if (\_nameController.text.isNotEmpty &&

\_typeController.text.isNotEmpty &&

\_priceController.text.isNotEmpty &&

\_imageUrl != null) {

final productId = Uuid().v4(); // Tạo ID duy nhất cho sản phẩm

// Tải hình ảnh lên Firebase Storage

final ref = FirebaseStorage.instance.ref().child('product\_images').child('$productId.jpg');

await ref.putFile(File(\_imageUrl!));

final imageUrl = await ref.getDownloadURL(); // Lấy URL hình ảnh đã tải lên

// Lưu thông tin sản phẩm vào Realtime Database

await databaseReference.child('products').child(productId).set({

'name': product.name,

'type': product.type,

'price': product.price,

'imageUrl': product.imageUrl,

});

}

}

**Chỉnh sửa sản phẩm**

* **Lấy thông tin sản phẩm**: Khi người dùng chọn sửa một sản phẩm, lấy thông tin sản phẩm từ Firebase và hiển thị trong các trường nhập liệu để người dùng có thể chỉnh sửa.
* **Cập nhật thông tin**: Sau khi người dùng sửa xong, gọi hàm để cập nhật thông tin vào Firebase.

**Xóa sản phẩm**

* **Xóa từ Realtime Database**: Xóa sản phẩm khỏi cơ sở dữ liệu và hình ảnh khỏi Firebase Storage. Sử dụng await ref.delete() để xóa hình ảnh.

dart

Sao chép mã

Future<void> deleteProduct(String id) async {

// Xóa sản phẩm từ Realtime Database

await databaseReference.child('products').child(id).remove();

}

**c. Chọn Hình ảnh**

**Chọn hình ảnh**

* **Sử dụng ImagePicker**: Cho phép người dùng chọn hình ảnh từ thư viện ảnh của thiết bị.

dart

Sao chép mã

Future<void> \_pickImage() async {

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

if (pickedFile != null) {

setState(() {

\_imageUrl = pickedFile.path; // Lưu đường dẫn hình ảnh

});

}

}

**d. Cập nhật Giao diện Người Dùng**

**Hiển thị danh sách sản phẩm**

* **ListView.builder**: Dùng để hiển thị danh sách sản phẩm. Mỗi mục trong danh sách có thể được chỉnh sửa hoặc xóa:

dart

Sao chép mã

Expanded(

child: ListView.builder(

itemCount: \_products.length,

itemBuilder: (context, index) {

final product = \_products[index];

return ListTile(

title: Text(product.name),

subtitle: Text('${product.type} - \$${product.price}'),

leading: product.imageUrl != null ? Image.network(product.imageUrl!, width: 50) : null,

);

},

),

)

**Hiển thị Thông báo**

* Sử dụng ScaffoldMessenger để hiển thị thông báo khi có thao tác thành công hoặc thất bại. Điều này giúp người dùng biết được trạng thái của các hành động như thêm, chỉnh sửa, hay xóa sản phẩm.

**e. Triển khai và Kiểm tra**

**Chạy ứng dụng**

* Chạy ứng dụng trên thiết bị thực hoặc trình giả lập. Kiểm tra từng chức năng để đảm bảo mọi thứ hoạt động như mong đợi.