**BÁO CÁO MÔ TẢ CÁC BƯỚC TRIỂN KHAI BÀI THI GIỮA KÌ**

**Ngô Thị Mỹ Tiên – 21IT110 – PTUDDĐ Đa Nền Tảng (7)**

**Ngày 10/10/2024**

**Link github:** <https://github.com/ntmtien-15/Mdtern_exam.git>

1. **Tạo Project**

Ảnh có chứa văn bản, phần mềm, ảnh chụp màn hình, máy tính

Mô tả được tạo tự động

1. **Kết nối Firebase Store, Storage**

- Tải Firebase CLI để cấu hình Fisebase.

- Tích hợp và cấu hình Firebase vào ứng dụng Flutter

Ảnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

- Storage: thiết lập quyền truy cập cho phép truy cập đọc và ghi vào

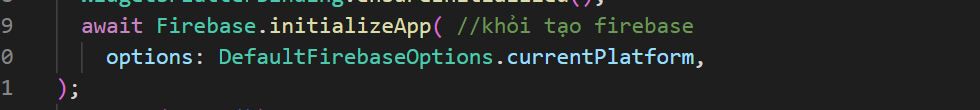
Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động

- Cloud FireStore: thiết lập quyền truy cập để cho phép đọc và ghi vào collection products.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động

- Khởi tạo Firebase

- Trong mã nguồn, kết nối với FireStore, sử dụng lớp FirebaseFireStore để truy cập vào cơ sở dũ liệu FireStore.

- Đồng thời thêm các packet trong phần dependencies.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

1. **Xây dựng giao diện**

Ảnh có chứa văn bản, ảnh chụp màn hình, đa phương tiện, Thiết bị liên lạc

Mô tả được tạo tự động

- Giao diện gồm 3 TextEdit để nhập tên 3 field: tên sản phẩm, loại sản phẩm, giá:

- Một button Chọn hình ảnh được truy cập vào thư viện ảnh.

- Một button Thêm sản phẩm để thêm những thông tin trên vào cơ sở dữ liệu FireStore

- Vùng hiển thi thông tin sản phẩm đã được thêm. (gồm hình ảnh, tên, loại, giá)

- Một icon xóa sản phẩm.

**Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Hệ điều hành

Mô tả được tạo tự động**

1. **Xử l‎í các chức năng**

* Lấy ảnh từ thư viện máy: sủ dụng hàm pickImage để truy cập ảnh trong thư viện máy và lưu dưới dạng File
* Future<void> pickImage() async {
* final pickedFile = await \_picker.pickImage(source: ImageSource.gallery);
* if (pickedFile != null) {
* setState(() {
* imageFile = File(pickedFile.path);
* });
* }
* }
* Tải ảnh lên FireStore và lưu đường dẫn: lấy tên file ở thơi gian được tải và tham chiếu đến nói lưu trữ trong FireStore -> tải lên FireStore và lưu lại đường dẫn của file ảnh.

Future<void> uploadImage() async {

    if (imageFile != null) {

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

      Reference storageRef =

          FirebaseStorage.instance.ref().child('products/$fileName');

      UploadTask uploadTask = storageRef.putFile(imageFile!);

      TaskSnapshot snapshot = await uploadTask;

      imageUrl = await snapshot.ref.getDownloadURL();

    }

  }

* Nhập thông tin sản phầm và lưu vào FireStore: lấy thông tin được nhập lưu vào fireStore tương ứng với các miền

await FirebaseFirestore.instance.collection('products').add({ //them sp vao FS

        'name': nameController.text,

        'category': categoryController.text,

        'price': double.parse(priceController.text),

        'imageUrl': imageUrl,

* Xử lí lỗi khi nhập thông tin sản phẩm

 if (nameController.text.isEmpty || categoryController.text.isEmpty || priceController.text.isEmpty) {

      ScaffoldMessenger.of(context).showSnackBar(

        SnackBar(content: Text('Vui lòng điền đầy đủ thông tin.')),

      );

      return;

    }

* Hiển thị sản phẩm đã được thêm
* Khi san phẩm có sẵn, sẽ lấy danh sách các document trong FireStore, mỗi document tương ứng với 1 sản phẩm

var products = snapshot.data.docs;

* Xây dưng 1 ListView để hiển thị ds

return ListView.builder( //hien thi ds sp

                    itemCount: products.length,

                    itemBuilder: (context, index) {

                      var product = products[index];

                      return Card(

                        child: ListTile(

                          leading: product['imageUrl'] != null

                              ? Image.network(product['imageUrl'])

                              : Container(width: 50, height: 50),

                          title: Text('Tên sp: ${product['name']}'),

                          subtitle: Column(

                            crossAxisAlignment: CrossAxisAlignment.start,

                            children: [

                              Text('Giá sp: ${product['price']}'),

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

                            ],

                          ),

                          trailing: Row(

                            mainAxisSize: MainAxisSize.min,

                            children: [

                              IconButton(

                                icon: Icon(Icons.delete, color: Colors.red),

                                onPressed: () {

                                  FirebaseFirestore.instance

                                      .collection('products')

                                      .doc(product.id)

                                      .delete();

                                },

                              ),

                            ],

                          ),

                        ),

* Hiển thị ds sản phẩm theo định dạng có cấu trúc.
* Xóa sảm phẩm:

 IconButton(

     icon: Icon(Icons.delete, color: Colors.red),

           onPressed: () {

  FirebaseFirestore.instance

          .collection('products')

           .doc(product.id)

           .delete();

},

),

1. **Cơ sở dữ liệu Firebase** Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

   Mô tả được tạo tự động