# BÁO CÁO THI GIỮA KỲ

## Mô tả tổng quan dự án

Dự án là một ứng dụng quản lý sản phẩm được xây dựng bằng Flutter và kết nối với Firebase để lưu trữ và quản lý dữ liệu. Người dùng có thể đăng nhập, đăng ký tài khoản, thêm, sửa, xóa sản phẩm và quản lý các sản phẩm này theo thời gian thực. Dự án sử dụng Visual Studio Code làm môi trường phát triển chính.

## Quy trình thực hiện

* Bước 1: Cài đặt môi trường phát triển Flutter và các plugin cần thiết trên Visual Studio Code.
* Bước 2: Thiết lập Firebase, bao gồm Authentication và Realtime Database, Storage tích hợp với dự án Flutter.
* Bước 3: Xây dựng giao diện người dùng với các trang home\_page.dart, login\_page.dart, và register\_page.dart.
* Bước 4: Tạo các dịch vụ xử lý việc đăng nhập, đăng ký và quản lý dữ liệu từ Firebase thông qua các file trong folder services và database.
* Bước 5: Tạo mô hình dữ liệu sản phẩm với class Product để dễ dàng thao tác và lưu trữ thông tin.
* Bước 6: Kiểm tra và đảm bảo các chức năng thêm, sửa, xóa sản phẩm hoạt động tốt với Firebase Realtime Database.

## Chi tiết các bước thực hiện

Bước 1: Mở Visual Studio Code và khởi tạo 1 project flutter mới

Bước 2: Tạo các Folder trong lib bao gồm: components, images, model, pages, services, database.

* Trong Folder tạo các file .dart như sau:
* Folder components:
* Chứa file my\_textfield.dart, định nghĩa và thiết lập form nhập liệu với các tham số tùy chỉnh, bao gồm việc định hình giao diện của các trường nhập dữ liệu (text fields).
* Folder images:
* Chứa các hình ảnh sử dụng làm background cho ứng dụng.
* Folder model: Chứa file product.dart định nghĩa class Product với các thuộc tính sau:
* id: Mã sản phẩm.
* name: Tên sản phẩm.
* category: Loại sản phẩm.
* price: Giá sản phẩm.
* imageUrl: Đường dẫn hình ảnh sản phẩm.
* Class này có các hàm:
* Product.fromMap: Chuyển đổi dữ liệu từ dạng JSON khi lấy từ Firestore.
* Product.toMap: Chuyển đổi dữ liệu từ đối tượng Product sang dạng JSON để lưu vào Firestore.
* Folder pages:
* Chứa các trang chính của ứng dụng:
* home\_page.dart: Thực hiện các chức năng quản lý sản phẩm (thêm, sửa, xóa, thêm hình ảnh) và lưu dữ liệu vào Firebase. Trang này cũng lấy dữ liệu từ Firebase để hiển thị danh sách sản phẩm.
* login\_page.dart và register\_page.dart: Dùng để đăng nhập và đăng ký tài khoản. Cả hai trang này kết nối với Firebase để xác thực người dùng.
* Folder services:
* Chứa các dịch vụ hỗ trợ việc quản lý tài khoản:
* auth\_gate.dart, auth\_service.dart: Đảm bảo việc chuyển đổi qua lại giữa màn hình đăng nhập và đăng ký.
* login\_or\_register.dart: Kiểm tra người dùng đã đăng nhập hay chưa và điều hướng đến trang tương ứng (đăng nhập hoặc đăng ký).
* Folder database:
* Chứa file firestore.dart kết nối với Firebase Realtime Database để quản lý sản phẩm với các phương thức:
* saveProduct: Thêm mới sản phẩm.
* updateProduct: Cập nhật thông tin sản phẩm.
* deleteProduct: Xóa sản phẩm khỏi database.
* getProducts: Lấy danh sách sản phẩm theo thời gian thực từ Firebase.

Bước 3: Kết nối Firebase

* Lên Firebase tạo tên project của mình: demo gk.
* Sau khi tạo project thì Authetication, Storage, Realtime Database
  + Authetication: Chọn Email/Password : Kiểm tra phần đăng nhập
  + Storage: Dùng để lưu ảnh
  + Realtime Database: Thêm, sửa, xóa dữ liệu được đưa vào và lấy dữ liệu đó hiện thị ra trên giao diện home\_page.dart
* Mở Terminal trong Visual Studio Code nhập các câu lệnh sau để kết nối với firebase:
  + npm install -g firebase-tools : Cái này để cài firebase tools
  + firebase login : Kiểm tra đã kết nối đến firebase chưa
  + flutter pub global activate flutterfire\_cli : Cái này để các file cần thiết
  + flutterfire configure : Cái này thì cần chọn project mình đang thực hiện là: demo gk

Bước 4: Tạo giao diện

* Trang đăng nhập: login\_page.dart sau khi nhập email và password thì sẽ dẫn đến trang home\_page.dart(trang nhập thêm, sửa, xóa dữ liệu)

A screenshot of a phone

Description automatically generated

* Trang thêm dữ liệu: Sau khi nhấn lưu sẽ hiện thị sản phẩm ở phía dưới và lưu dữ liệu vào Realtime Database

A cell phone with a hamburger and a purple text

Description automatically generated

* Trang sửa dữ liệu:
  + Ảnh đầu tiên: khi chọn Hamburger thì nó lại hiện thị lại toàn bộ dữ liệu đã thêm trước đó
  + Ảnh thứ 2: Sau khi sửa lại dữ liệu và nhấn cập nhật sản phẩm thì dữ liệu sản phẩm sẽ được cập nhật trên Realtime Database và hiện thị như ảnh thứ 3

A screenshot of a phone

Description automatically generated A screenshot of a phone

Description automatically generated A screenshot of a phone

Description automatically generated

* Trang xóa dữ liệu: Sau khi xóa sản phẩm thì nó sẽ xóa dữ liệu khỏi Realtime Database và không hiện dữ liệu trên trang

A screenshot of a chat

Description automatically generated

## Video demo



## Link Github

<https://github.com/votiendung1/demogk.git>