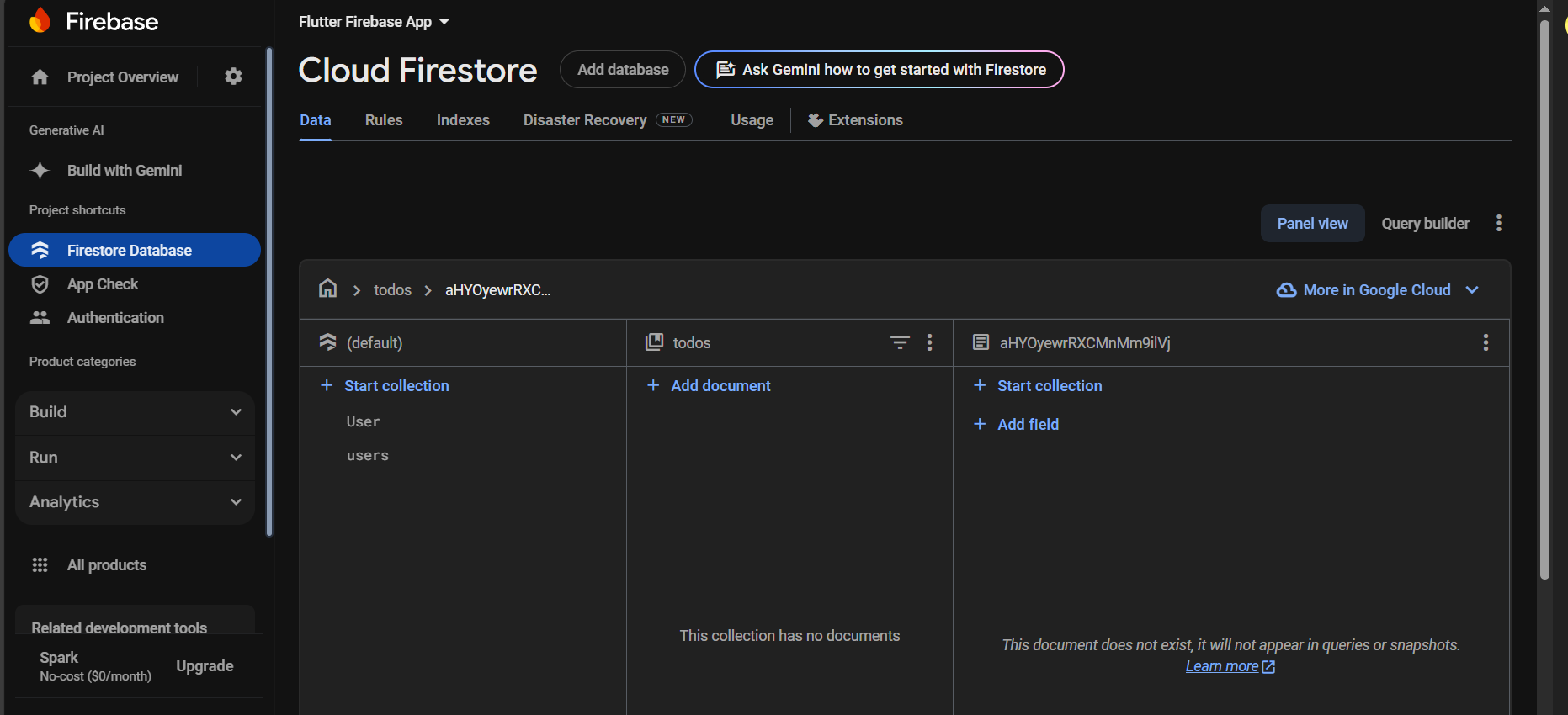
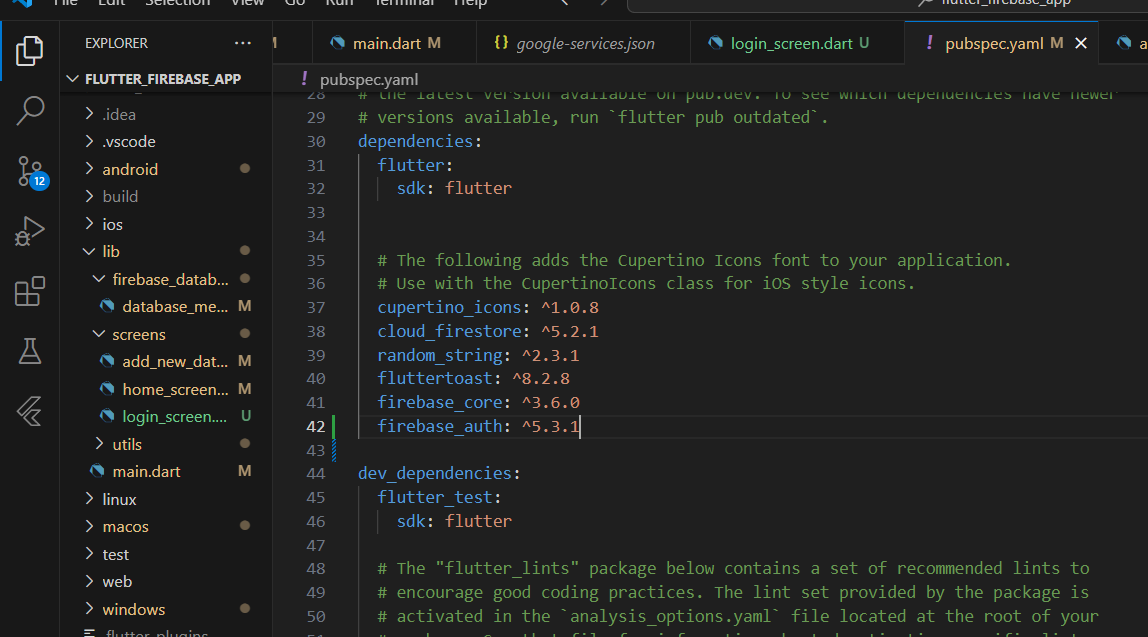
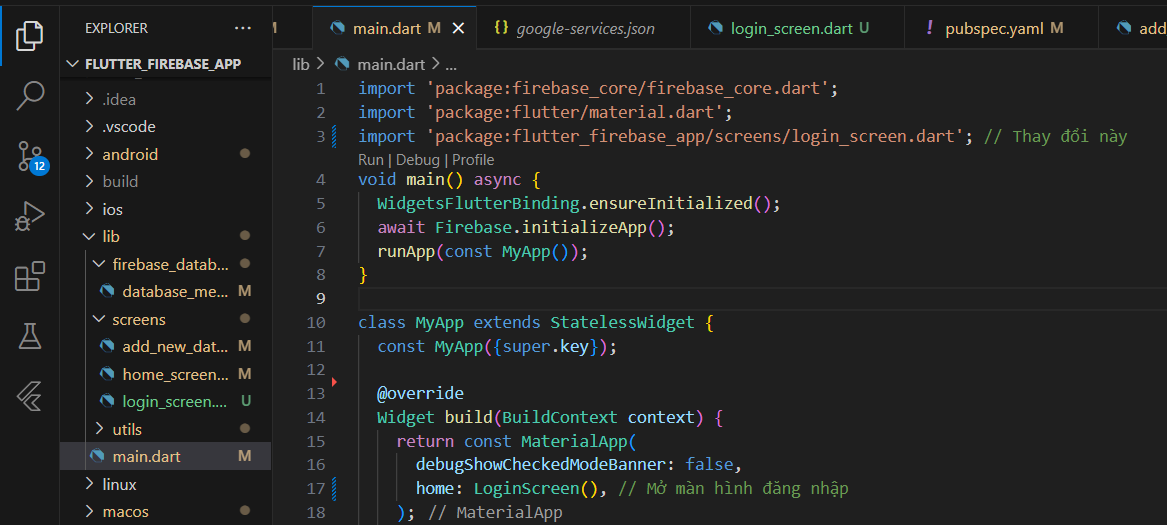
**Họ tên:** Trần Phước Thông   
**Mã SV:** 21IT450   
**github:** [phuocthong/CRUD-with-flutter (github.com)](https://github.com/phuocthong/CRUD-with-flutter)

Các bước triển khai dự án

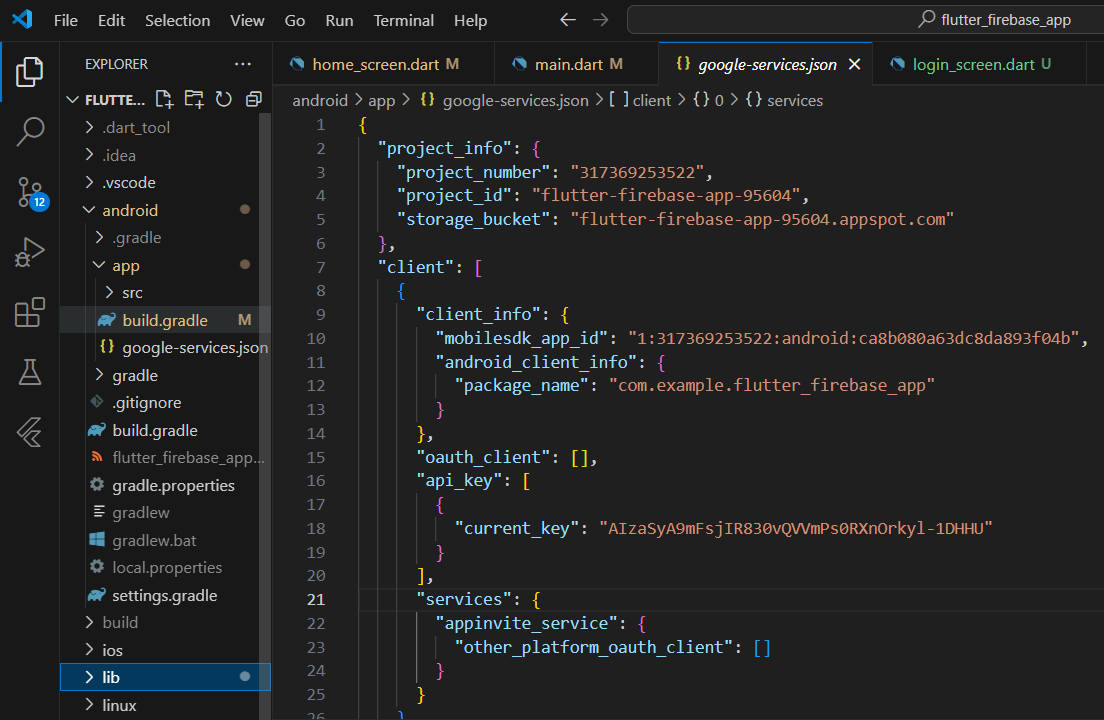
* Truy cập firebase console tạo dự án với tên Flutter Firebase App  
  
* Cài đặt thư viện Firebase cho flutter



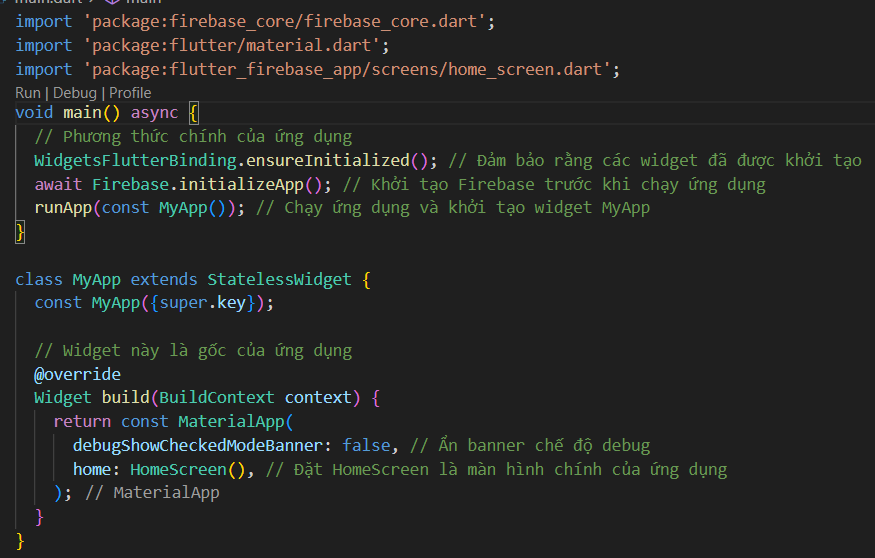
* Thiết lập firebase trong dự án flutter



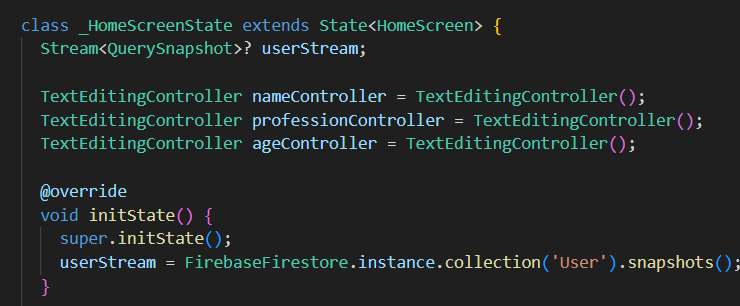
* Kết nối dự án với Firebase



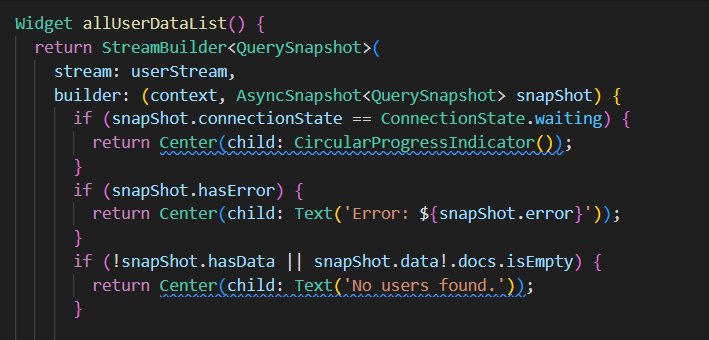
* Thêm pub package
* Xây dựng file **main.dart**   
   Khởi tạo ứng dụng Flutter với Firebase và định nghĩa một widget không có trạng thái (MyApp) để thiết lập cấu trúc chính của ứng dụng, với màn hình chính là HomeScreen.



* File **home\_screen.dart**: giao diện chính của ứng dụng, sử dụng Firestore để hiển thị, chỉnh sửa và xóa thông tin người dùng. Các phương thức trong lớp này đảm bảo rằng giao diện người dùng được cập nhật theo thời gian thực với dữ liệu từ Firestore.

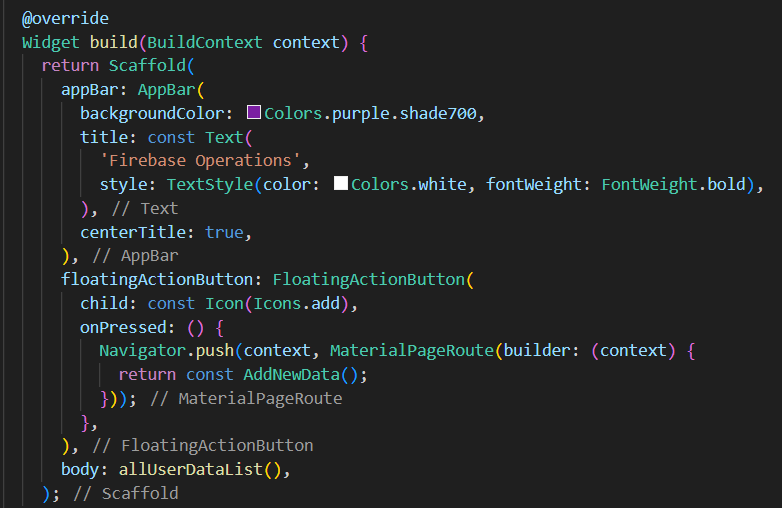


**class \_HomeScreenState**: Đây là lớp trạng thái cho widget HomeScreen. Quản lý tất cả các trạng thái và logic liên quan đến giao diện người dùng của HomeScreen.

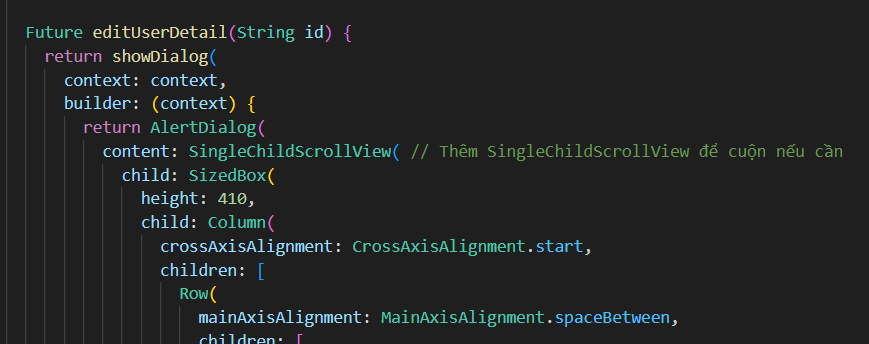


**Widget allUserDataList()**: Phương thức này trả về một widget hiển thị danh sách tất cả người dùng từ Firestore.

**StreamBuilder<QuerySnapshot>**: Sử dụng để xây dựng giao diện dựa trên luồng dữ liệu từ userStream. Nó sẽ tự động cập nhật giao diện khi dữ liệu thay đổi.



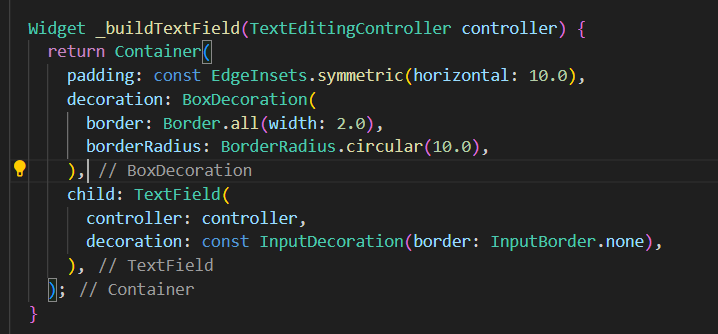
**Widget build(BuildContext context)**: Phương thức này xây dựng giao diện cho HomeScreen.



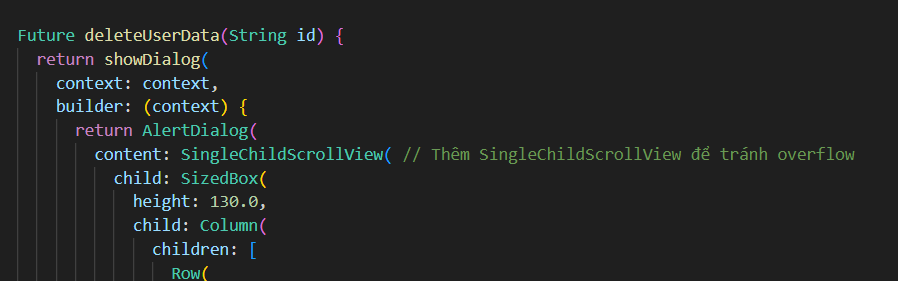
**Future editUserDetail(String id)**: Phương thức này hiển thị hộp thoại cho phép người dùng chỉnh sửa thông tin của một người dùng cụ thể.

**showDialog**: Hiển thị một hộp thoại tùy chỉnh với các trường nhập liệu để chỉnh sửa thông tin người dùng.

**Khi nhấn nút cập nhật**: Dữ liệu được cập nhật trong Firestore và hiển thị thông báo thành công.



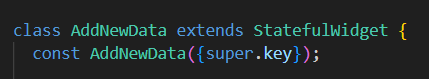
**Widget \_buildTextField(TextEditingController controller)**: Phương thức trả về một trường nhập liệu tùy chỉnh. Nó nhận một controller để quản lý nội dung của trường.



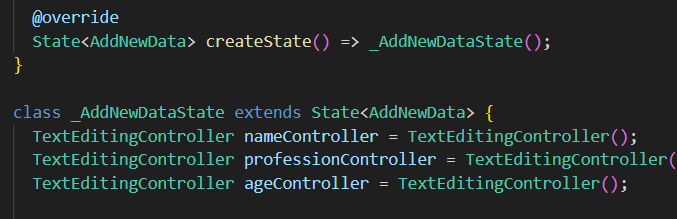
**Future deleteUserData(String id)**: Phương thức này hiển thị hộp thoại xác nhận trước khi xóa thông tin của người dùng.

**showDialog**: Hiển thị hộp thoại để hỏi người dùng có chắc chắn muốn xóa không.

* **File add\_new\_data.dart:** File này định nghĩa một màn hình thêm người dùng mới trong ứng dụng Flutter, với khả năng nhập dữ liệu người dùng và lưu trữ chúng vào cơ sở dữ liệu Firestore. Các thành phần như kiểm tra dữ liệu và thông báo giúp cải thiện trải nghiệm người dùng.



AddNewData: là lớp chính của màn hình để thêm người dùng mới. Nó kế thừa từ StatefulWidget, cho phép màn hình duy trì trạng thái của nó khi người dùng tương tác với nó.



\_AddNewDataState: chứa trạng thái của màn hình AddNewData, bao gồm các phương thức và thuộc tính để quản lý giao diện người dùng và logic.

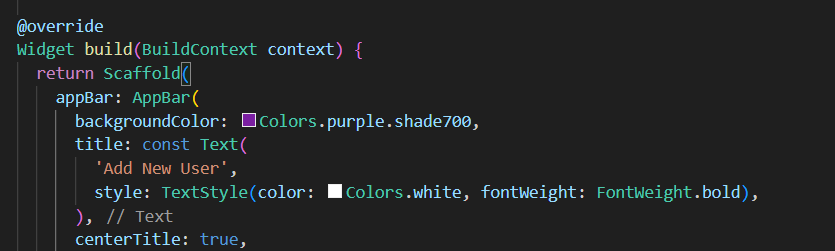
 **Thuộc tính**:

* TextEditingController nameController: Quản lý nội dung của trường nhập tên.
* TextEditingController professionController: Quản lý nội dung của trường nhập nghề nghiệp.
* TextEditingController ageController: Quản lý nội dung của trường nhập tuổi.

 **Phương thức build(BuildContext context)**:

* Xây dựng giao diện người dùng cho màn hình thêm người dùng mới.
* Sử dụng Scaffold để tạo cấu trúc cơ bản với AppBar, một khu vực thân (body) với các trường nhập và nút thêm người dùng.

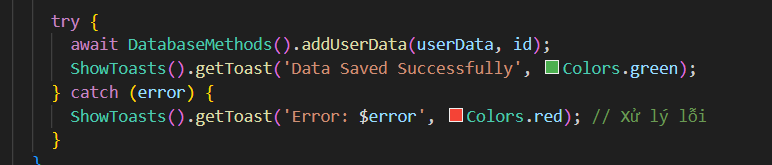
**Các phần tử giao diện**

****

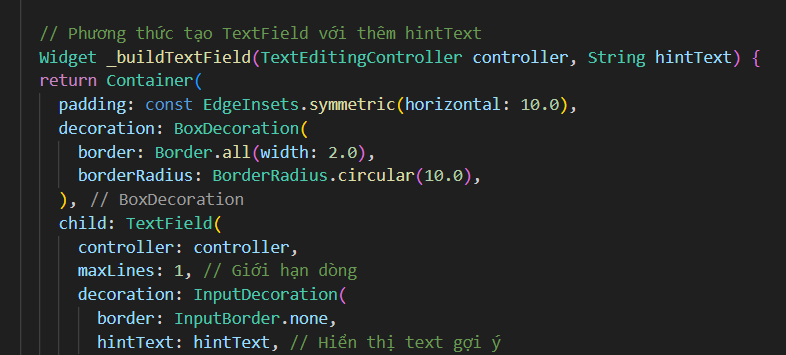
* **AppBar**: Hiển thị tiêu đề "Add New User" trên thanh công cụ.
* **SingleChildScrollView**: Cho phép người dùng cuộn nếu nội dung quá dài.



* **Các trường nhập (TextField)**: Được tạo thông qua phương thức \_buildTextField, cho phép người dùng nhập tên, nghề nghiệp và tuổi.
* **ElevatedButton**: Nút "Add User" để người dùng thêm thông tin sau khi đã nhập đầy đủ dữ liệu.
*  **Tạo ID người dùng**: Sử dụng randomAlphaNumeric(8) từ package random\_string để tạo ID ngẫu nhiên cho người dùng mới.
*  **Tạo bản đồ dữ liệu**: userData chứa thông tin người dùng với các trường "Name", "Profession", "Id" và "Age".
*  **Thêm dữ liệu**: Gọi phương thức addUserData từ lớp DatabaseMethods để thêm người dùng mới vào cơ sở dữ liệu.

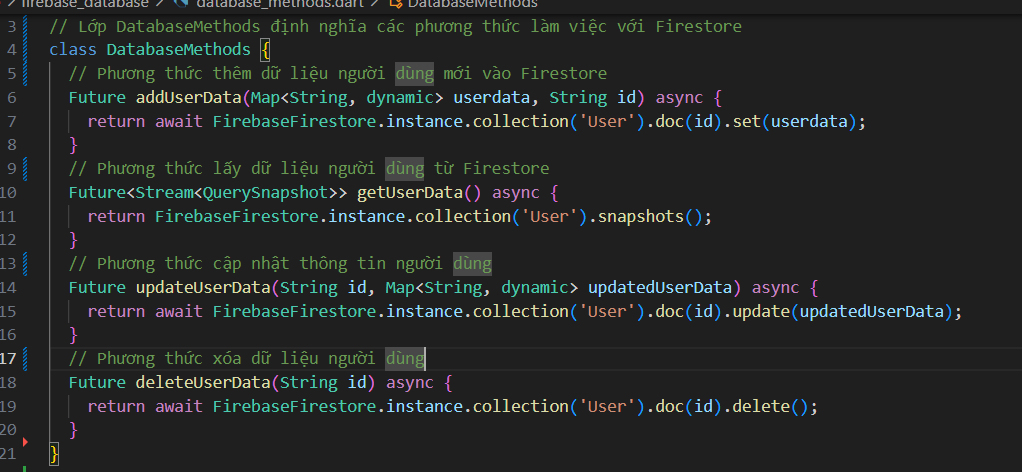


*  **Hiển thị thông báo**: Sử dụng ShowToasts để hiển thị thông báo khi dữ liệu được lưu thành công hoặc gặp lỗi.

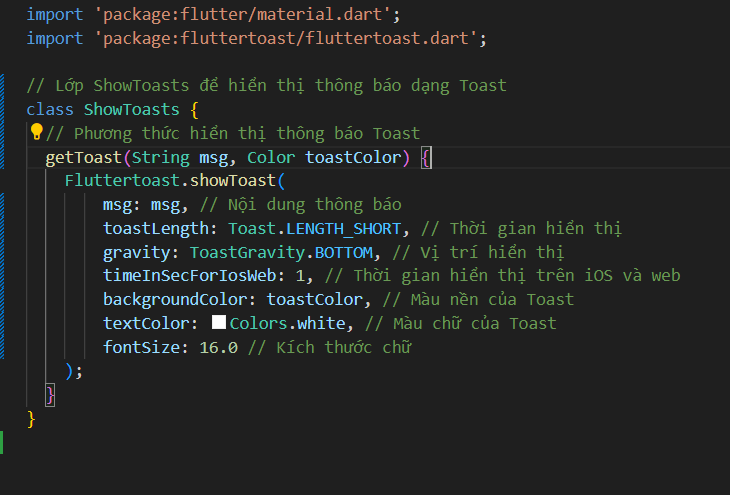


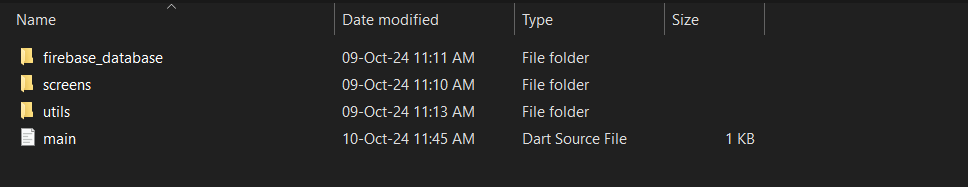
Tạo một widget TextField với các thuộc tính cụ thể, như có một hintText (nội dung gợi ý) và một số kiểu dáng.

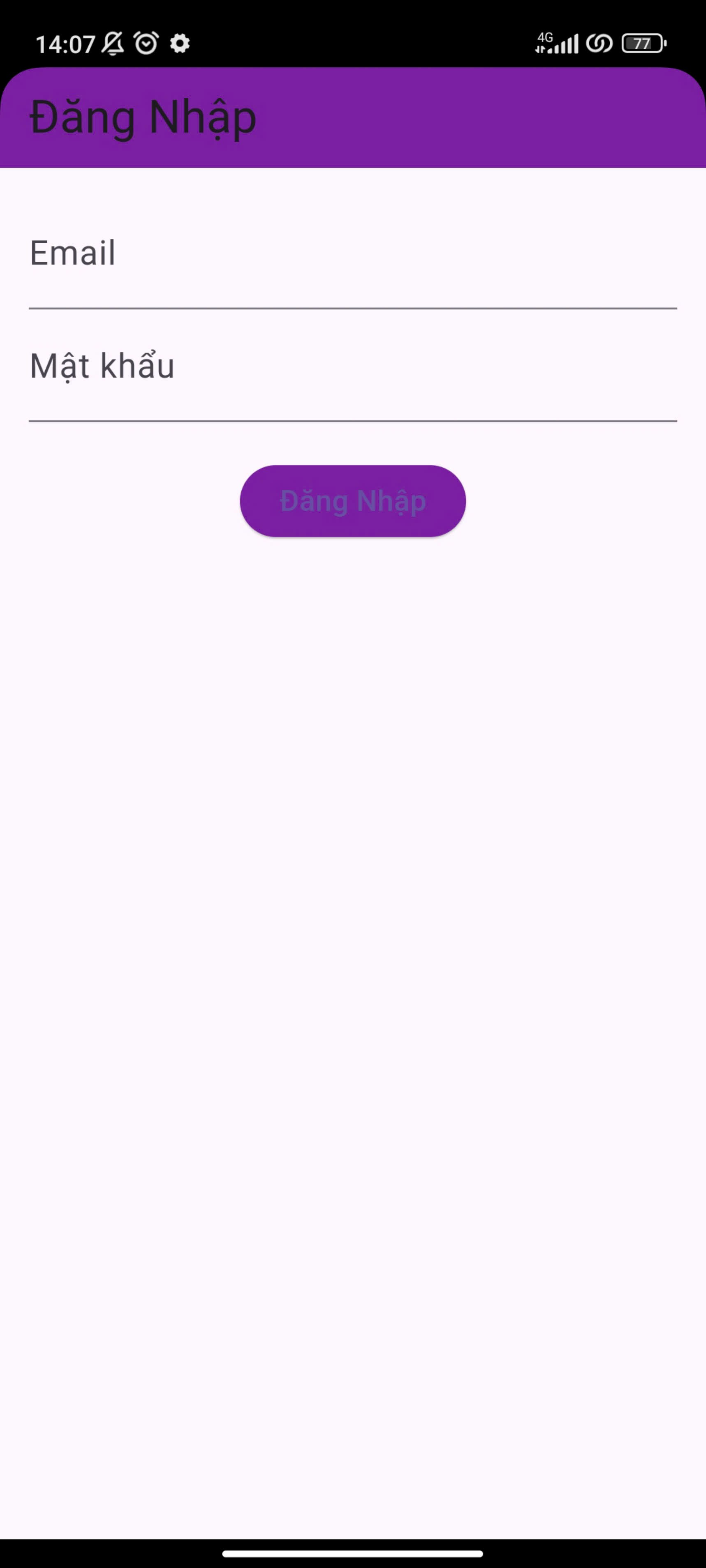
* File **database\_methods.dart:** cung cấp các phương thức để thêm, lấy, cập nhật và xóa người dùng trong collection User của Firestore. Các phương thức này giúp thực hiện các thao tác dữ liệu cơ bản cần thiết trong ứng dụng của bạn.



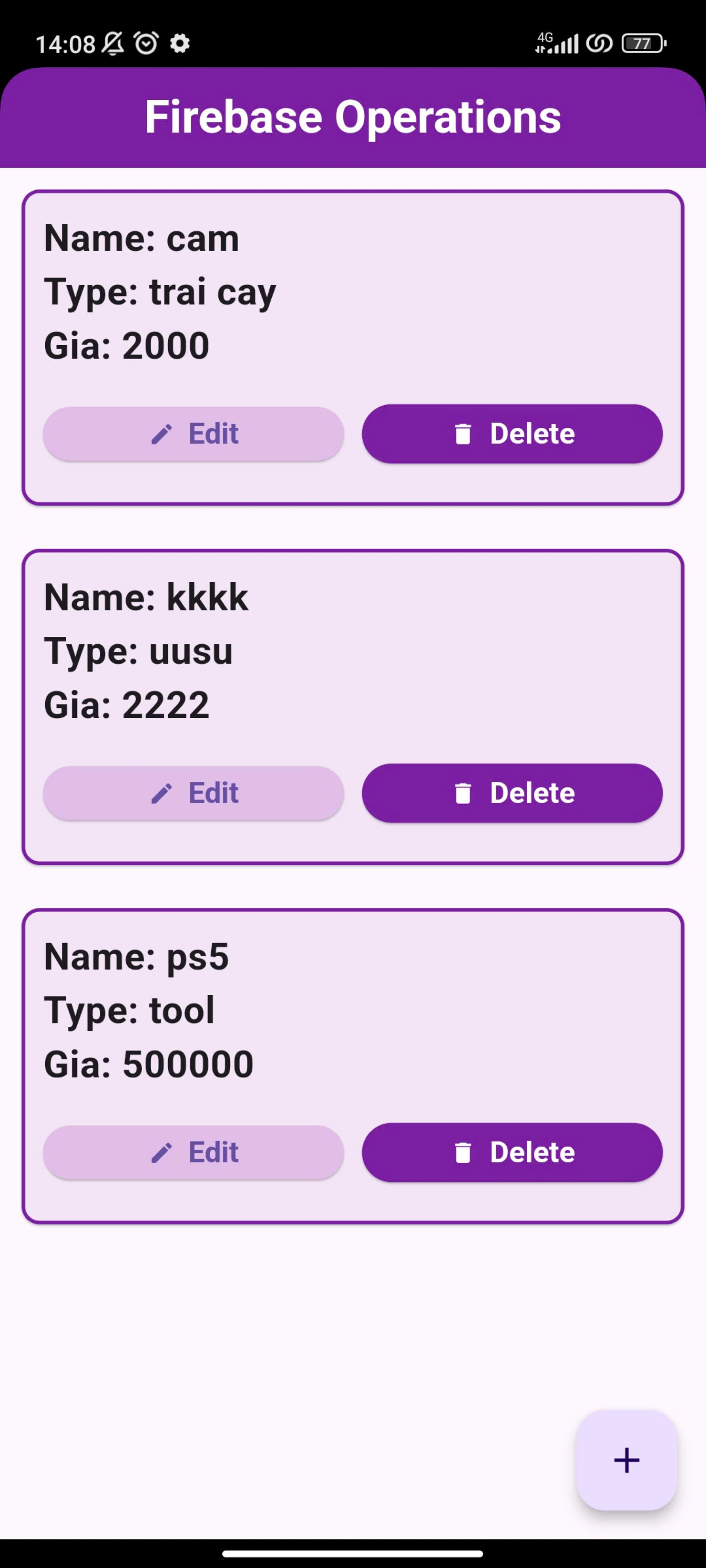
* File **show\_toasts.dart:** cung cấp một cách đơn giản để hiển thị thông báo Toast trong ứng dụng Flutter, giúp người dùng biết về trạng thái các hành động họ thực hiện.



* Thời gian tạo dự án:   
  
* Giao diện Đăng nhập



* Giao diện chính



* Giao diện thêm sản phẩm

