## Chat Realtime with Firebase

**Đề bài**

Xây dựng ứng dụng chat realtime sử dụng firebase.

**Mục tiêu**

Sau bài học này các bạn có thể hiểu thêm về cách cài đặt kết nối firebase, kiến thức về NoSQL dạng cây, và quá trình để xây dựng một app chat cơ bản.

**Những gì bạn sẽ học**

* Cách tạo project firebase bằng tài khoản gmail
* Kết nối project Flutter với tài khoản firebase
* Hiểu mô hình NoSQL của firebase.
* Hiểu cơ bản về thao tác giữa FrontEnd và BackEnd(Xem Firebase như BackEnd).
* Cách sử dụng SQL của Firebase trong Dart.
* Và một số kiến thức bổ ích cho ứng dụng sau này

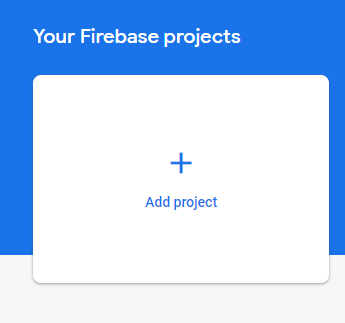
**Thực hành :**

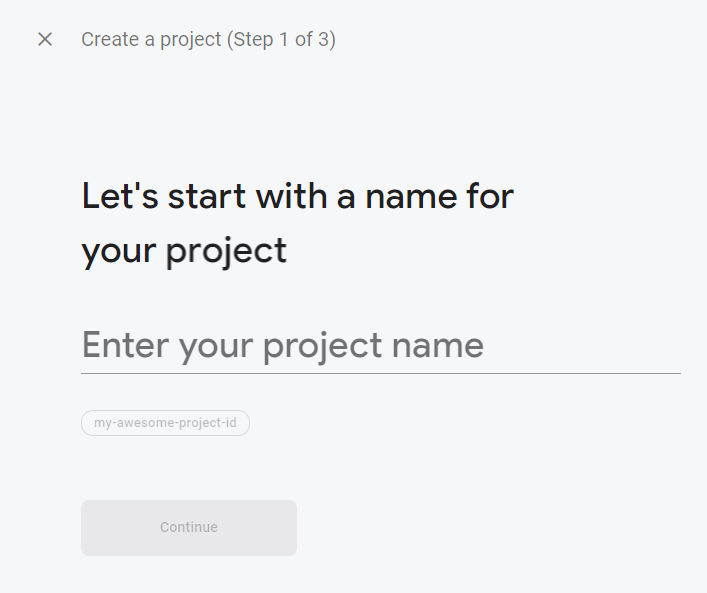
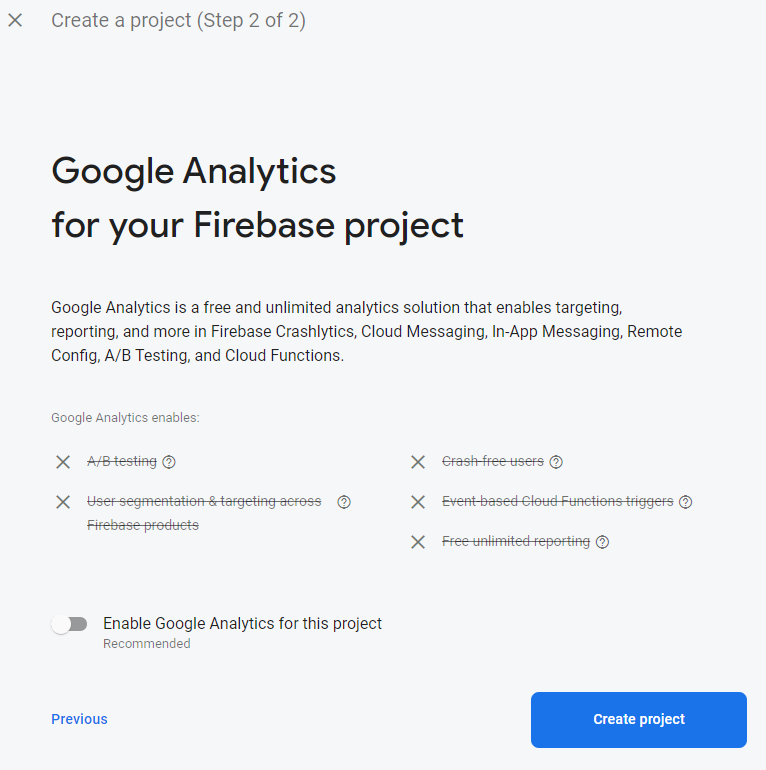
1. **Tạo project firebase**

* Truy cập vào trang <https://firebase.google.com/> và chọn go to console để vào bảng điều khiển của firebase.

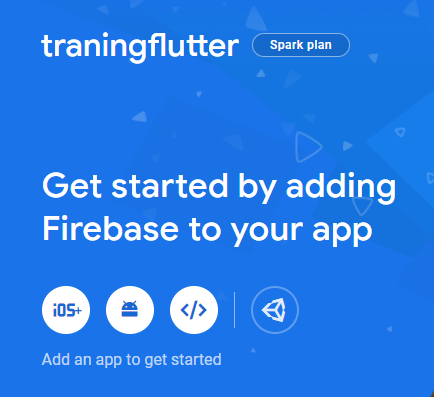
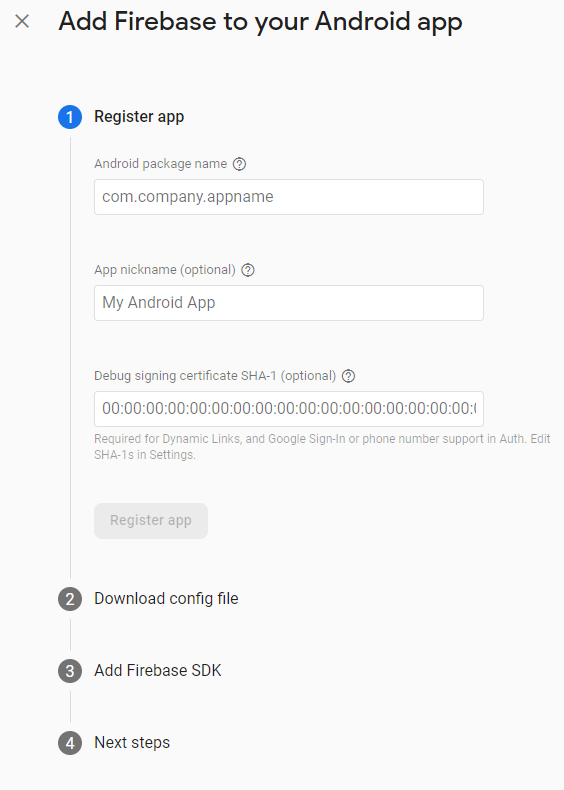
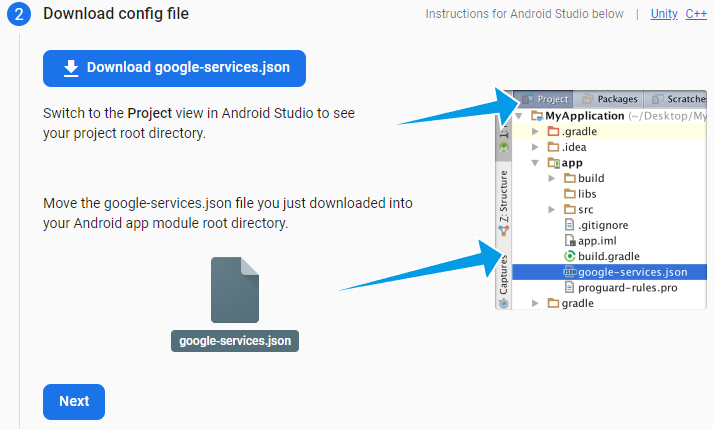
****

* Trong bảng điều khiển **Firebase**, nhấp vào **Thêm dự án (Add project)**



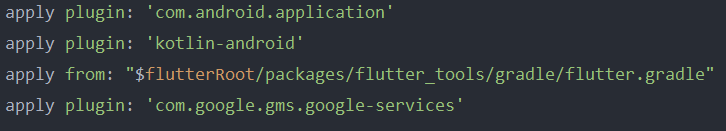
* Chúng ta lần lượt đi qua các bước của việc tạo project

1. **Kết nối project Flutter với project Firebase**

* Sử dụng lệnh **flutter create** để tạo ứng dụng mới:
* **flutter create chat\_firebase**
* **cd chat\_firebase**
* Nhấp vào biểu tượng Android để tạo quy trình thiết lập.
* Chúng ta tìm package name ở thư mục **android/build.gradle** và thêm đúng tên vào ô Android package name
* 
* Tải tệp **google-service.json** về máy, trước khi next thì cần thực hiện các bước tiếp.
* Di chuyển tệp cấu hình của bạn vào thư mục **android**/**app** của ứng dụng **Flutter** của bạn.
* Để bật dịch vụ **Firebase** trong ứng dụng **Android** của bạn, hãy thêm **plugin** **google-services** vào tệp **Gradle** của bạn. Trong tệp **Gradle** cấp gốc (cấp dự án) của bạn ( **android/build.gradle** ), hãy thêm các quy tắc để bao gồm **plugin** **Google** **Services** **Gradle**. Kiểm tra xem bạn cũng có kho lưu trữ **Maven** của **Google**.



* Trong tệp **Gradle** mô-đun (cấp ứng dụng) của bạn (thường là **android/app/build.gradle** ), hãy áp dụng plugin Google Services Gradle.



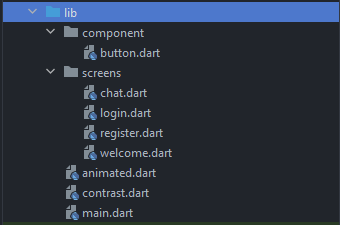
* Chạy **flutter packages get** .
* Quay lại quy trình thiết lập bảng điều khiển **Firebase**, nhấp vào **Tiếp theo** để bỏ qua các bước còn lại.
* Đảm bảo rằng ứng dụng của bạn hiện không chạy trong trình mô phỏng hoặc trên thiết bị của bạn.
* Từ thư mục gốc của ứng dụng **Flutter**, hãy mở tệp **pubspec**.**yaml** của bạn.
* Thêm plugin **FlutterFire** cho SDK Firebase Core Flutter
* Chạy **flutter packages get**



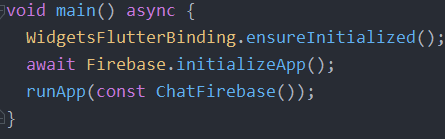
1. **Lập trình**

Các bạn tiến hành một số setup cơ bản:

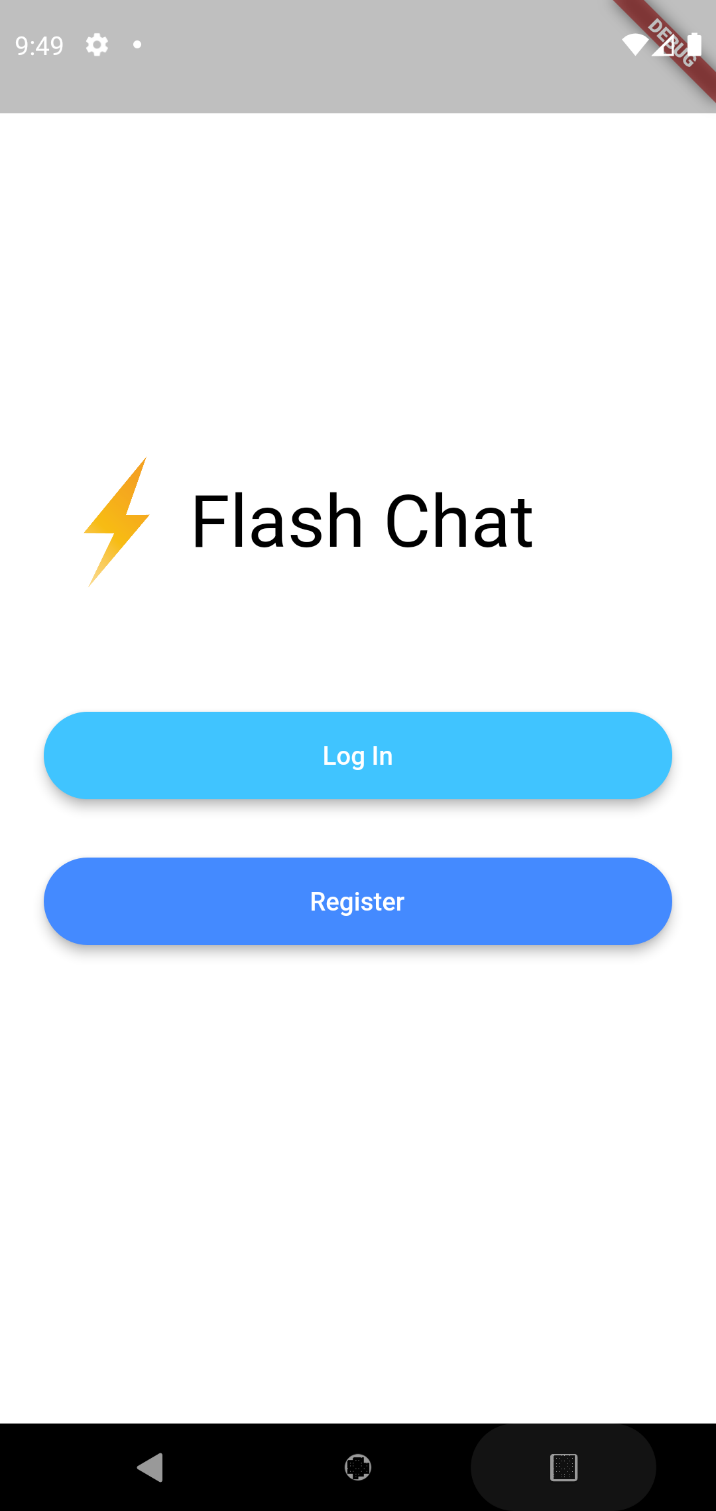
* Đầu tiên chúng ta sẽ tạo các file có thể có trong project trước như sau:



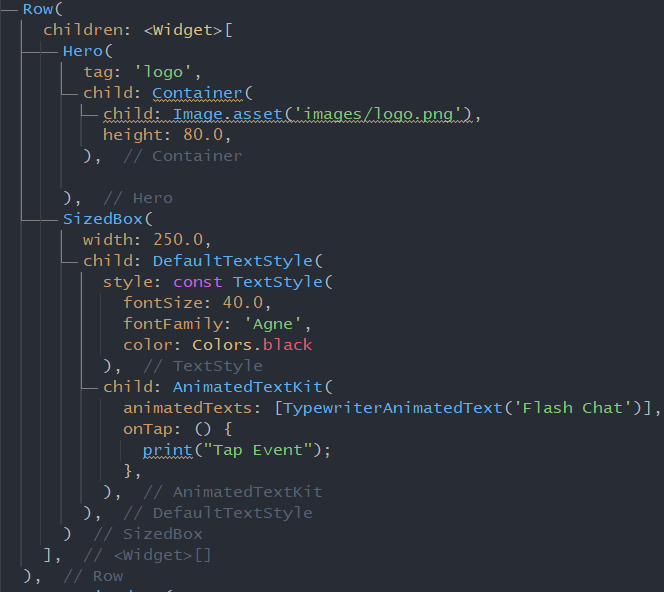
* Trong function main chúng ta thêm vào như sau để kết nối với **firebase** khi chạy ứng dụng



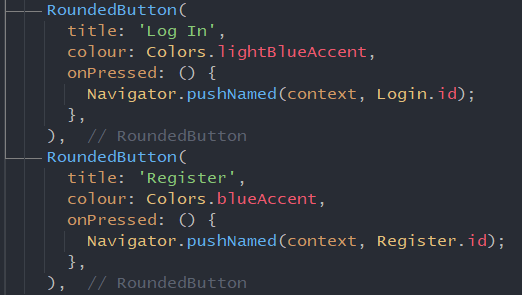
* Về cơ bản chúng ta sẽ có 4 màn hình. Đi qua từng màn hình. Đầu tiên là màn hình ban đầu hay ở đây là **Welcome**.

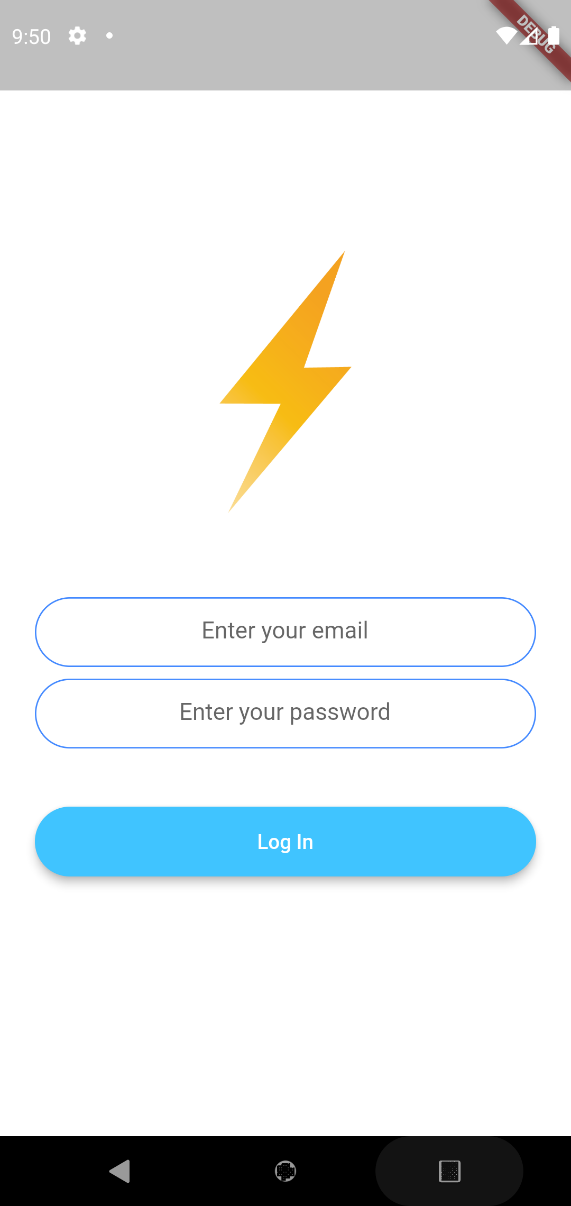
****

* Ở màn hình này chúng ta cần tạo ra 3 thành phần chính gồm 1 Text và 2 Button
* Text: Chúng ta thấy Text sẽ bao gồm Text và Icon

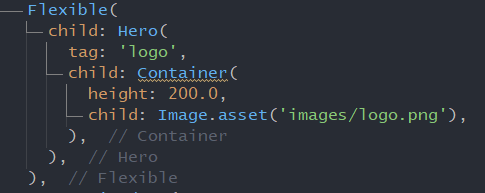


* 2 Button sẽ lần lượt là :

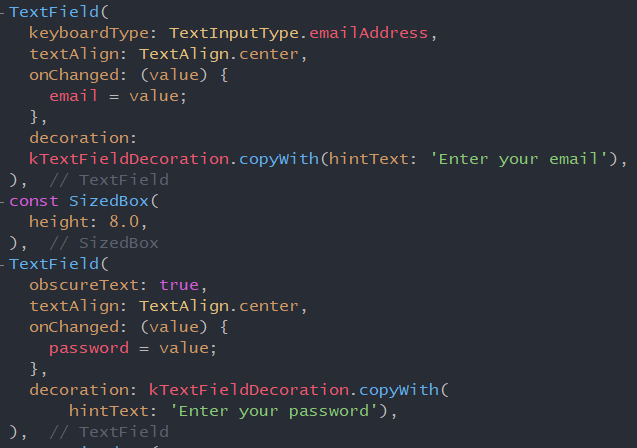


****

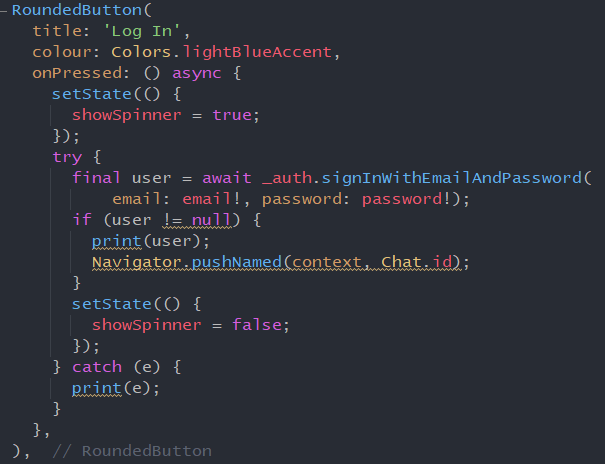
* Ở màn hình Log in chúng ta thấy được bố cục gồm 4 thành phần 1 Icon , 2 TextField và 1 Button
* Icon:

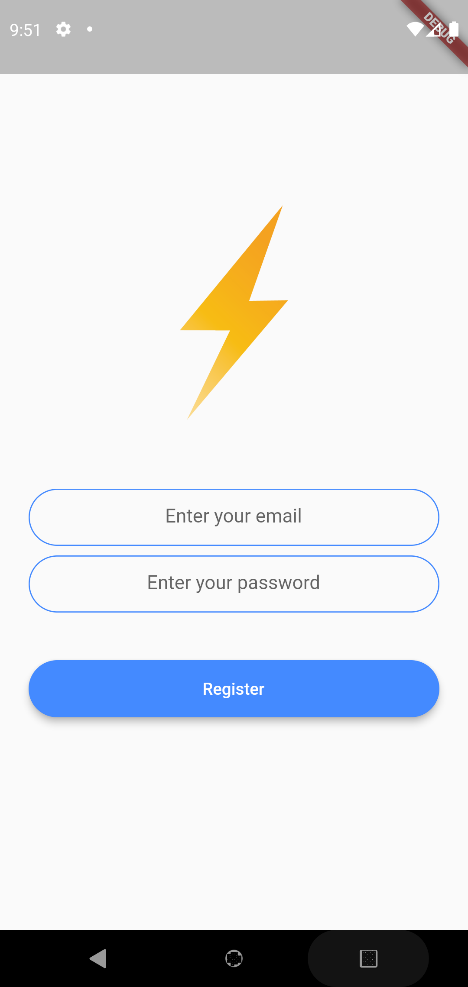
****

* TextField

****

* Button



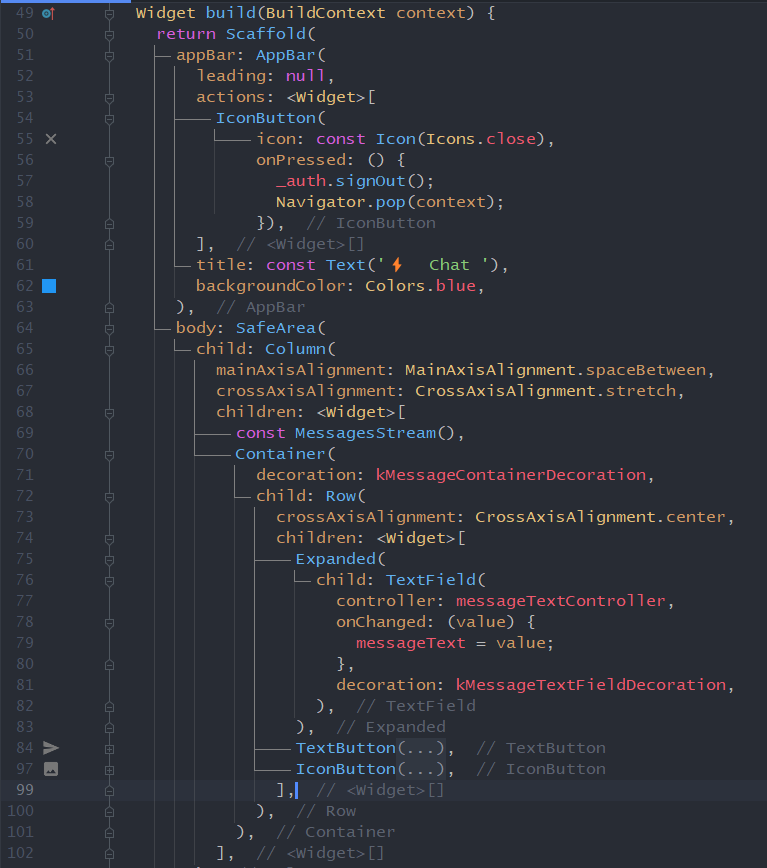
****

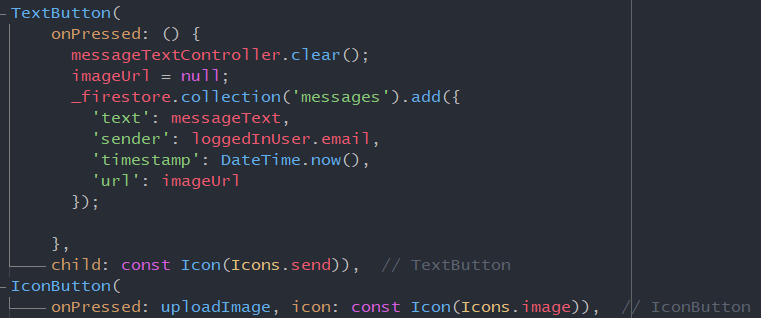
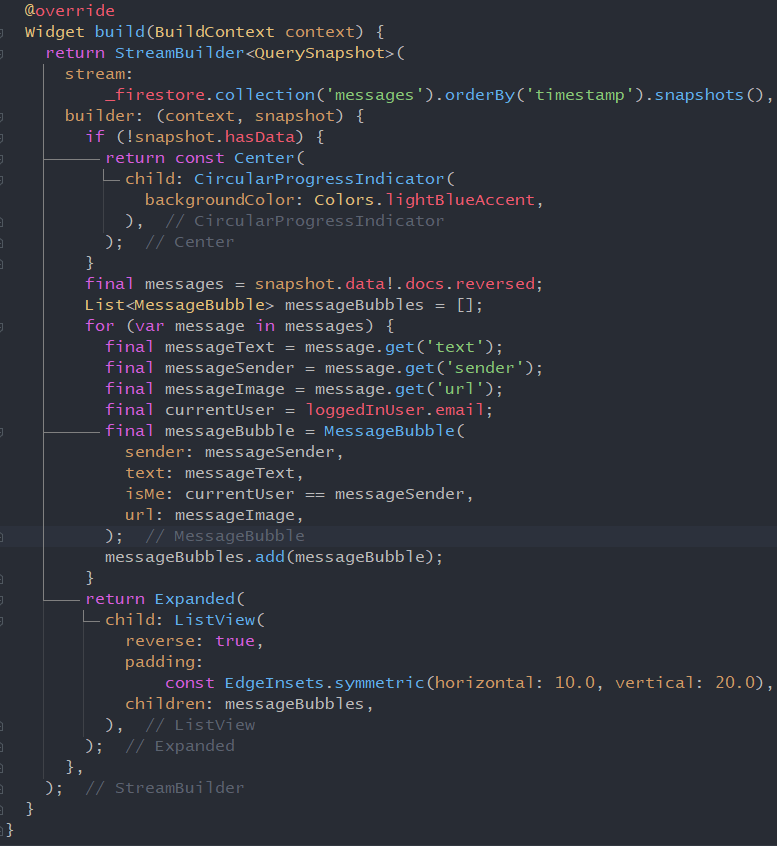
* Ở phần Resgister chúng ta thấy về cơ bản nó giống với Log in nhưng sẽ khác ở phần xử lý của Button

****

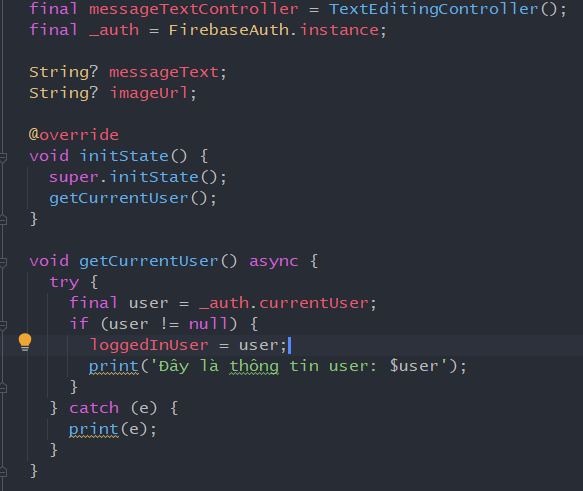
****

* Đến phần quan trọng nhất của bài này, chính là phần **chat**. Đầu tiên chúng ta sẽ thực hiện tạo giao diện trước.

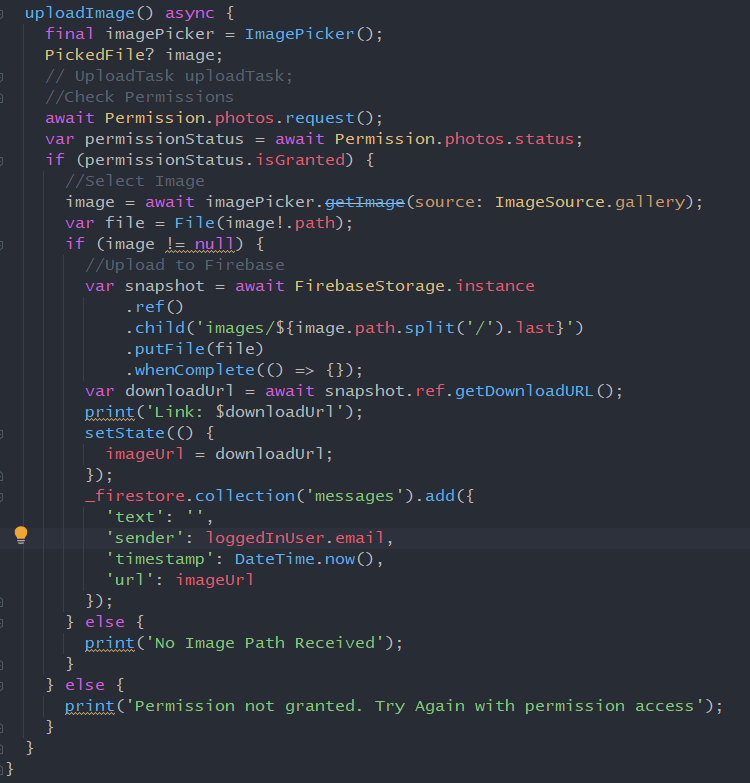


******-**

* Để kiểm tra xem có lấy được thông tin người dùng đăng nhập hay không ta dùng function **getCurrentUser**() và kiểm tra **log**. Đây cũng để **set** **up** thông tin người dùng trên màn hình chat.

****

* Function **uploadImage**() dùng để gửi hình ảnh trong chức năng chat. Để dùng được chức năng này chúng ta phải cài đặt thư viện **image**\_**picker** và **permission**\_**handler**

****