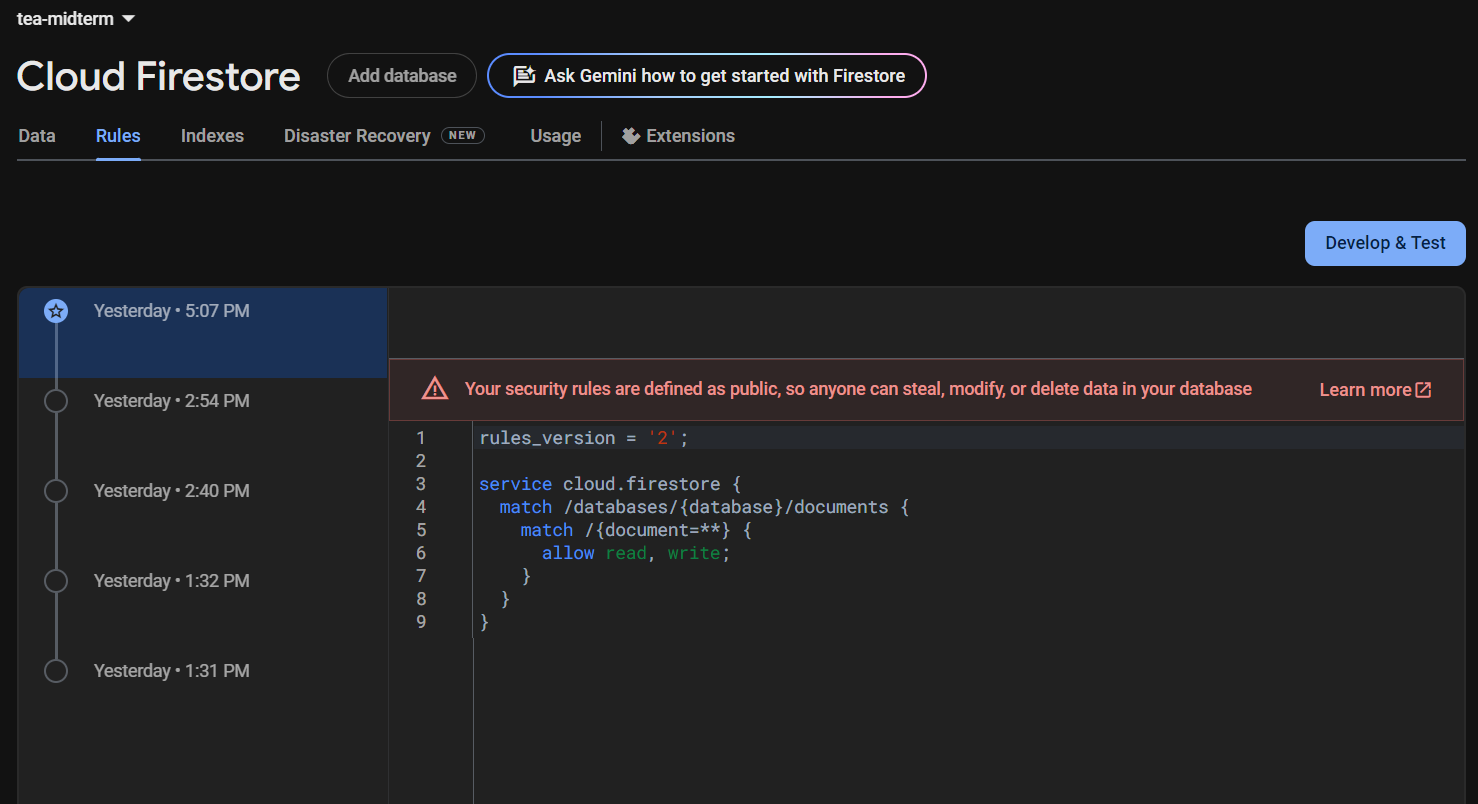
**Tên: Trịnh Công Toàn  
Mã Sinh viên: 21IT388**

Link source code: <https://github.com/KhadaTea/TeaMTFl>

Các bước thực hiện bài giữa kì:

Bước 1: Add firebase

- Lên web chọn firebase database, chỉnh ở phần rule thành như ảnh:



- login firebase trên terminal bằng command: firebase login

- chạy pub global activate flutterfire\_cli trên terminal của admin

- thêm framework flutter core: flutter pub add firebase\_core

- flutterfire configure để tạo file database trên firebase

- flutter run để test project

- sửa hàm main trong file main.dart thành:

Future<void> main() async {

  WidgetsFlutterBinding.ensureInitialized();

  await Firebase.initializeApp(

*options*: DefaultFirebaseOptions.currentPlatform,

  );

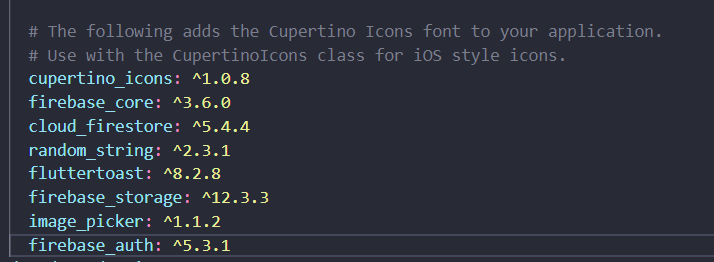
  runApp(const MyApp());

}

=> đã kết nối thành công

+ Import các thư viện và dependencies cần thiết cho dự án như(firebasefirestore, cloud\_firebase,…)

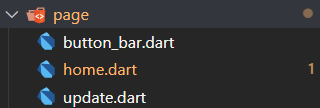




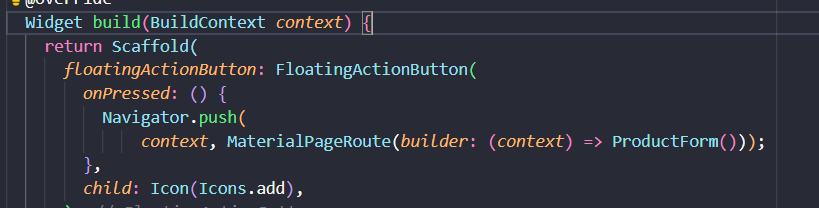
Bước 2:

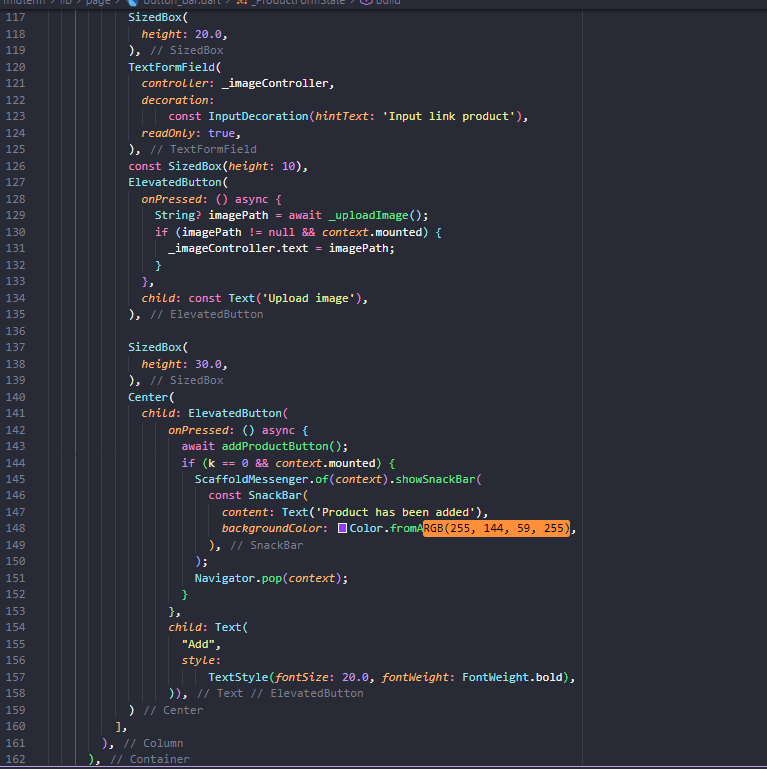
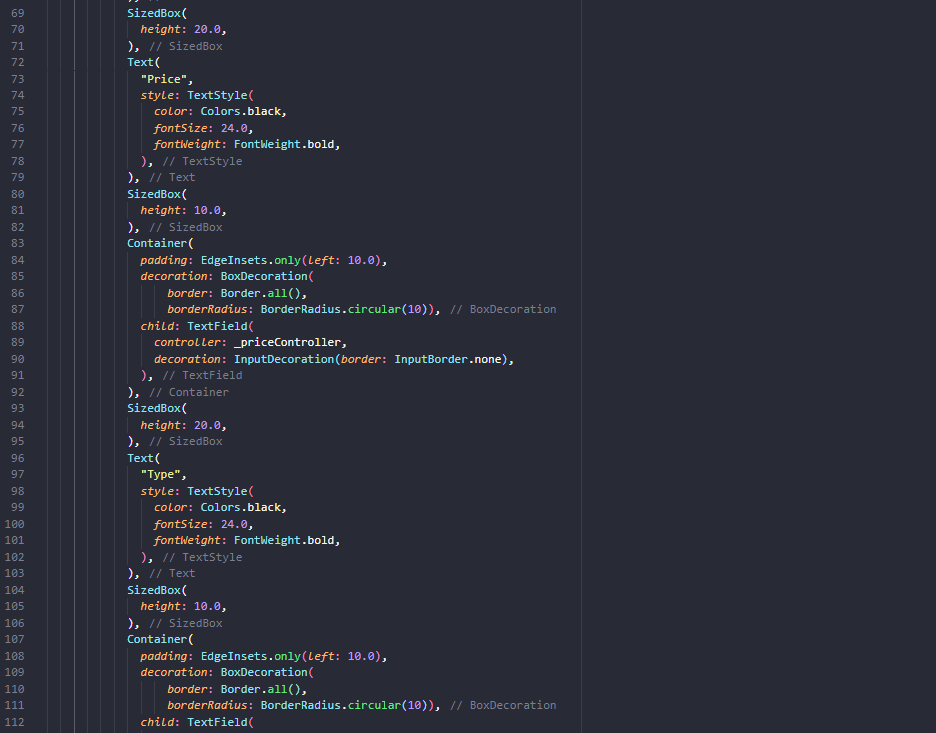
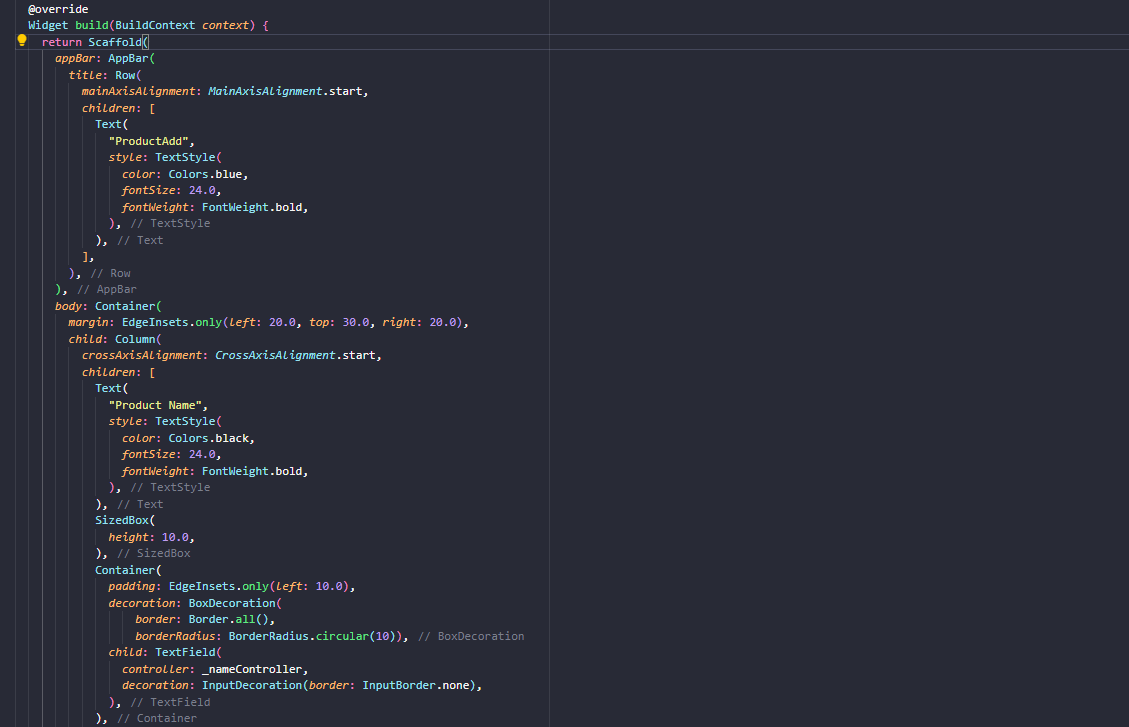
-Tạo folder page trong lib: lib/page

+Tạo home.dart trong đó chứa trang chính khi hoàn thành đăng nhập

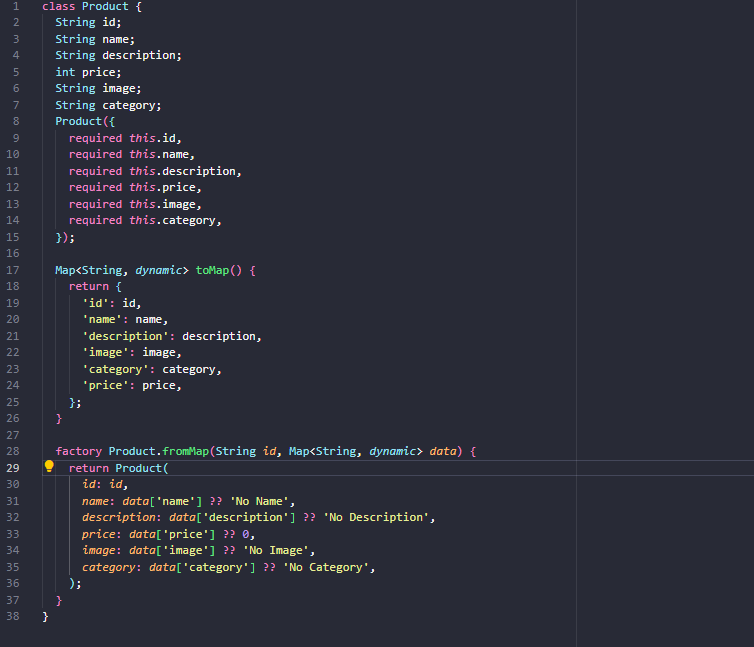


+ Tận dụng code đã có sẵn mẫu khi tạo project, add sự kiện vào nút góc phải ở dưới màn hình -> trỏ tới button\_bar(sau khi đã tạo button\_bar)

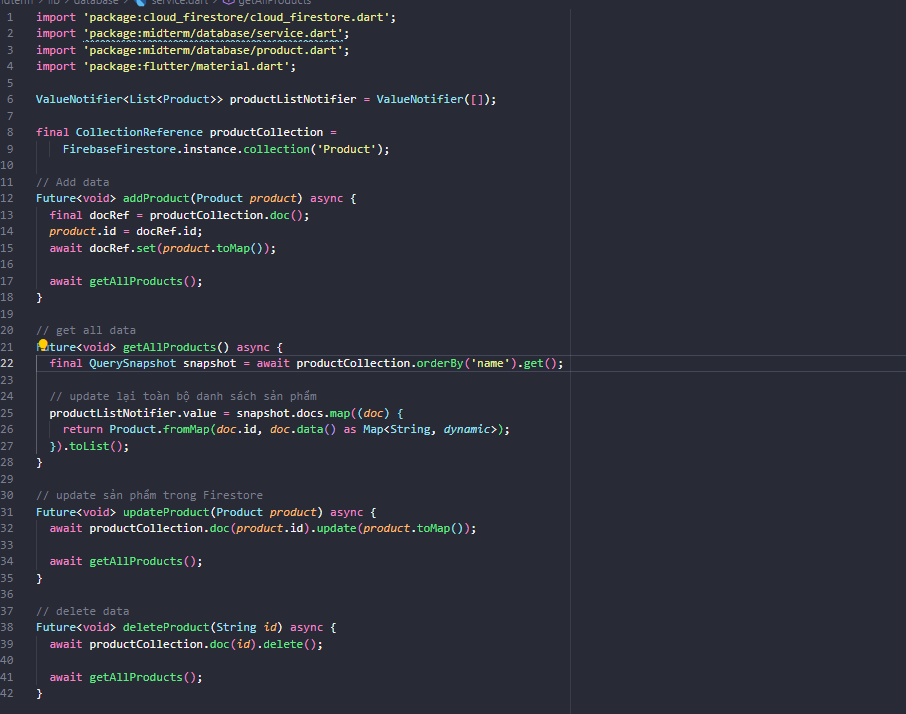


+ Tạo button\_bar.dart, thiết kế form nhập dữ liệu vào và chỉnh sửa giao diện lại. 

+ Tạo product.dart, class này hiện kiểu dữ liệu của các trường

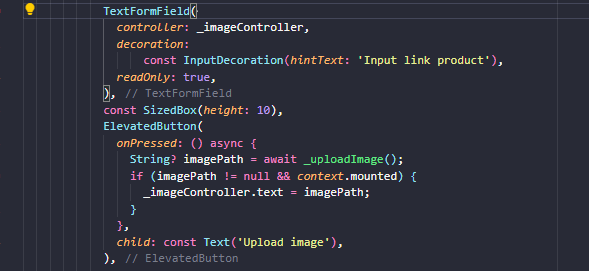


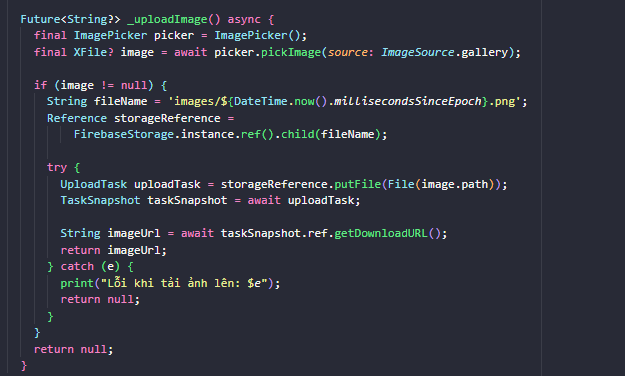
+ Tạo service.dart, class này thể hiện các phần tử được lấy ra khi thao tác với event



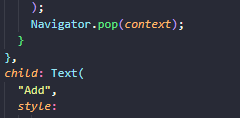
+ Quay về lại button\_bar.dart, sau khi đã có service.dart và product.dart, bắt đầu get kiểu dữ liệu và phần tử có thể được xuất ra để làm event cho các nút khi nhập dữ liệu vào và try catch để bắt 1 số lỗi “đã gặp”

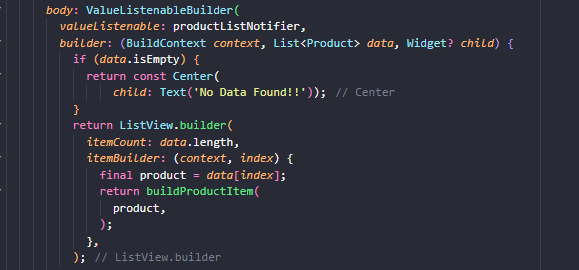




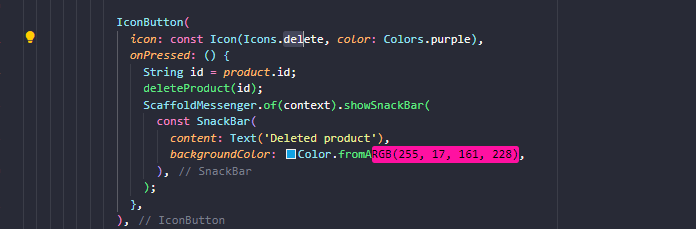


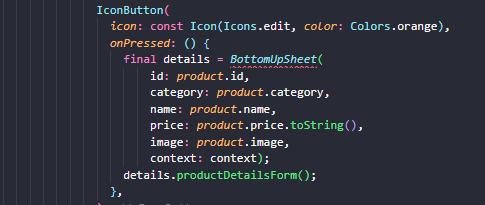
+ Sau khi thêm xong, quay về home.dart code UI và hiện thị thông tin vừa nhập để in ra màn hình.( đã thêm Navigator.pop để tự động quay về trang trước sau khi nhấn Add )



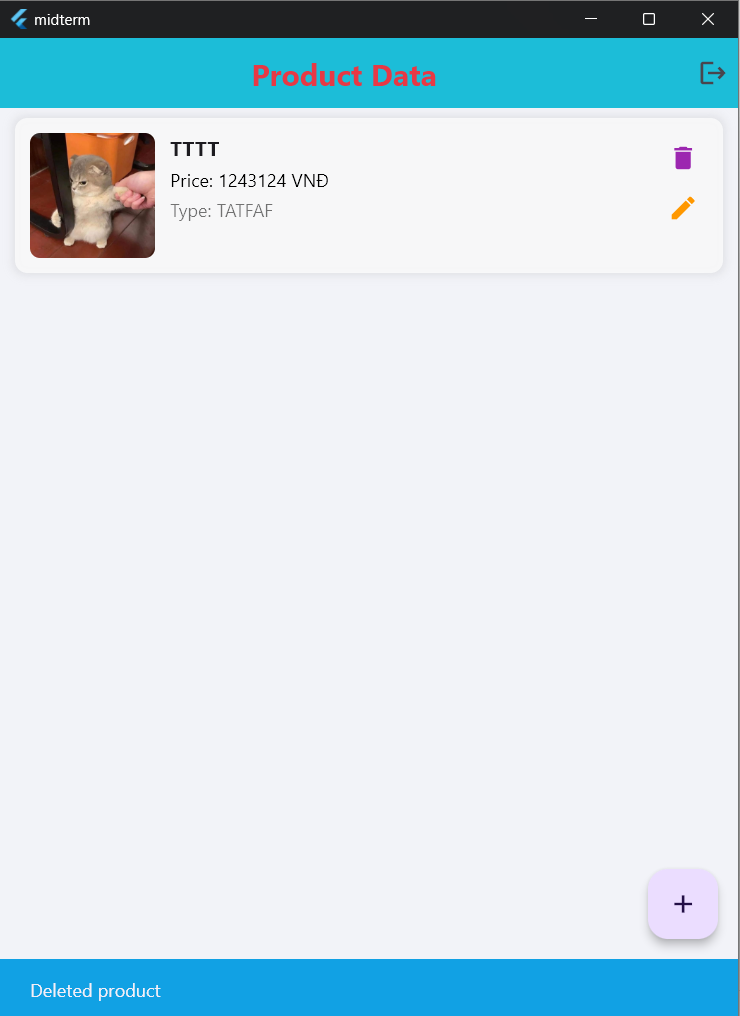


+ Ở home.dart lúc này có icon delete và update, cũng như bước trên, quay về lại service và code các phần tử có thể được xuất ra khi gọi tới và ở product lấy các kiểu dữ liệu của trường.





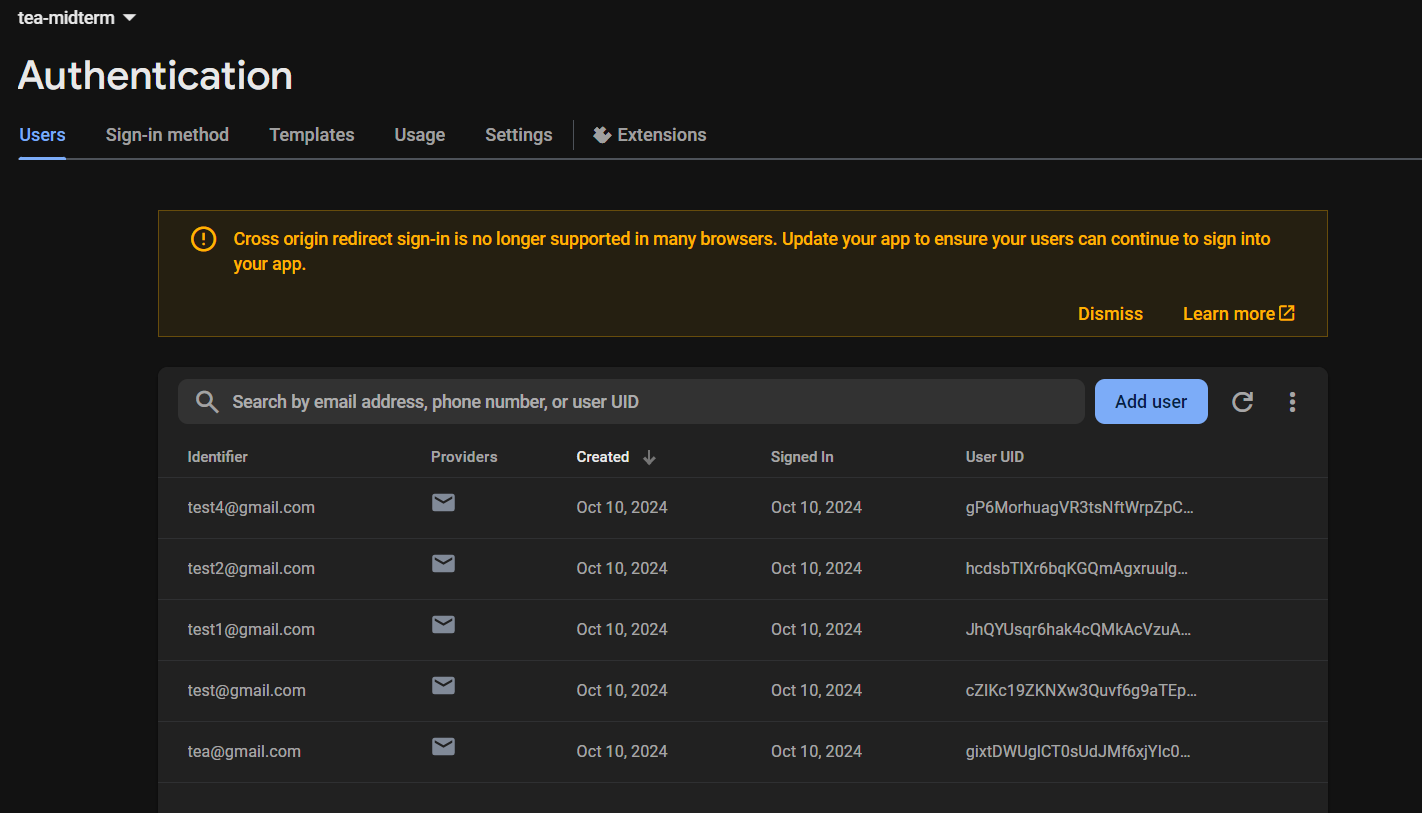
+ Khi delete, sản phẩm bị xóa, và tự điền vào phần trống của ô vừa xóa, thông báo hiện ra dưới góc màn hình: “Deleted Product!!”



+ Khi update, hiện lên 1 widget mới và hiển thị lại các thông tin của ô mình muốn update -> nhập lại dữ liệu mình muốn -> bấm submit -> dữ liệu đã được nhập và tự động refresh lại trang.

Bước 3: Đăng Nhập, Đăng ký, Authentication

- Lên firebase chọn “Authentication” nhấn “Get Started” để khởi tạo phương thức đăng nhập -> Chọn Email/Password -> đã xong phần setup authentication.



- Tạo 3 class:

+ Tạo loginPage.dart và Tạo signup.dart cho vào 1 folder tên là loginform



+ Tạo auth.dart cho vào folder “database” đã có sẵn.



+ Truyền phương thức cho auth.dart trước sau đó qua loginPage.dart và signup.dart để code sự kiện và UI cho cả 2.

+ Sau khi hoàn tất các class trên chạy lên thử xem đã hoàn tất và không gặp lỗi hay chưa, nếu lỗi thì fix, còn không thì hoàn thành.

+ Chạy lại để kiểm tra xem realtime của ảnh,…

Bước 4:

Kiểm tra lại trong firebase trong code để push lên git và hoàn tất bài làm.