**What is Flutter?**

Flutter is a free and open-source mobile UI framework created by Google and released in May 2017. In a few words, it allows you to create a native mobile application with only one codebase. This means that you can use one programming language and one codebase to create two different apps (for IOS and Android).

Flutter consists of two important parts:

* An SDK (Software Development Kit): A collection of tools that are going to help you develop your applications. This includes tools to compile your code into native machine code (code for IOS and Android).
* A Framework (UI Library based on widgets): A collection of reusable UI elements (buttons, text inputs, sliders, and so on) that you can personalize for your own needs.

**Why Flutter?**

1. Fast Development

### 2. **Expressive and Flexible UI**

### **3. Native Performance**

### 4. **Dart Language**

### **5. Important Flutter Tools**

### What is Flutter feature?

### **1. Same UI and Business Logic in All Platforms**

### **2. Reduced Code Development Time**

### **3. Increased Time-to-Market Speed**

### **4. Similar to Native App Performance**

### **5. Custom, Animated UI of Any Complexity Available**

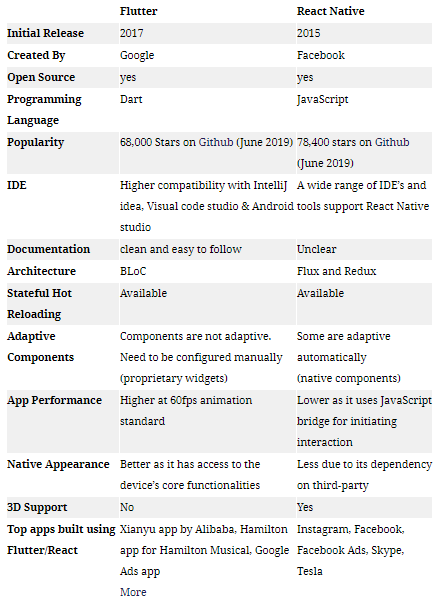
### **6. Own Rendering Engine**

### **7. Simple Platform-Specific Logic Implementation**

### **8. The Potential Ability to Go Beyond Mobile**

### Flutter VS React native

### 

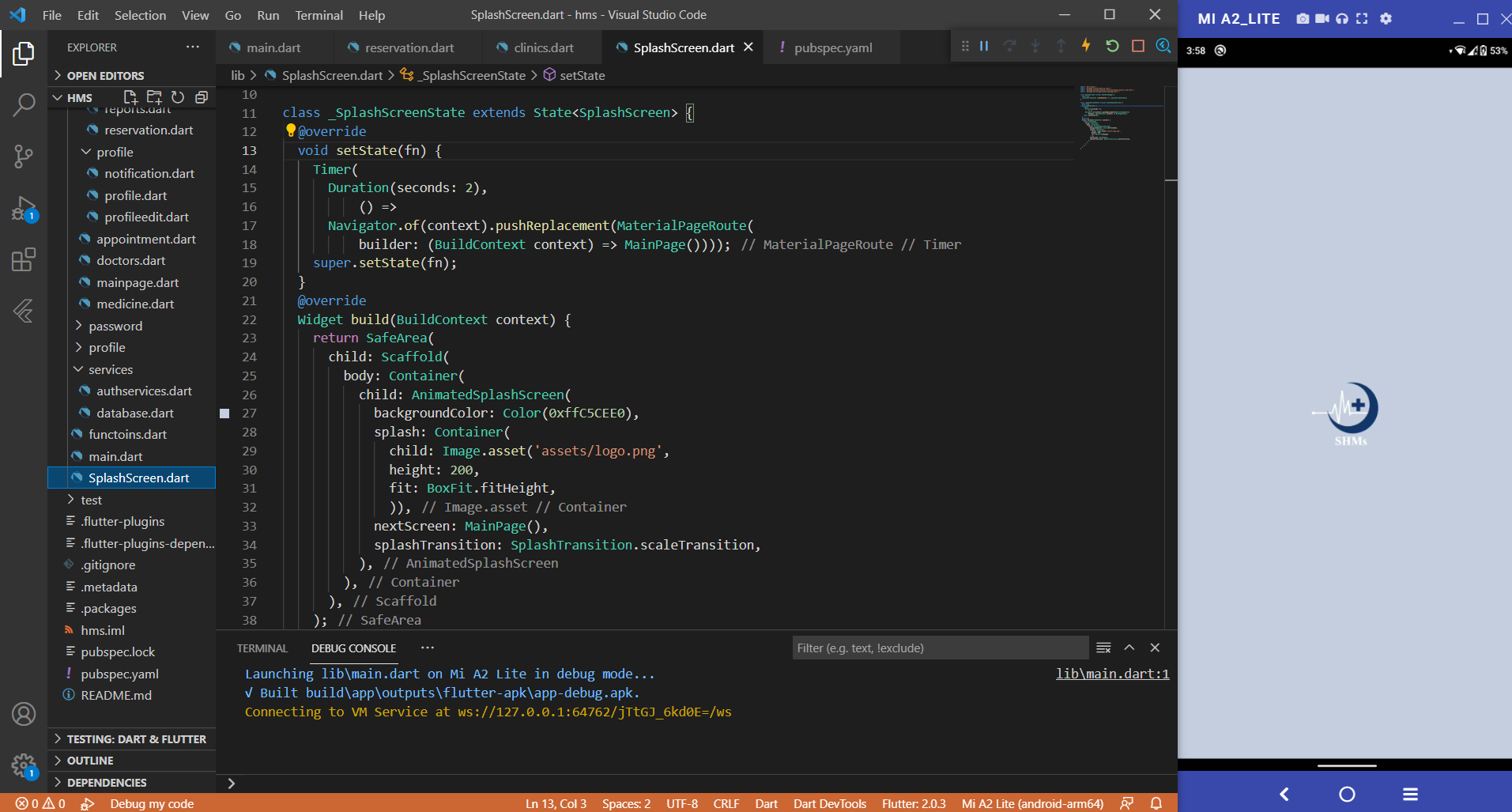


**Library we use in the Project**

* **Animated Splash Screen**

We use it to make welcome page

dependencies: animated\_splash\_screen: ^1.1.0



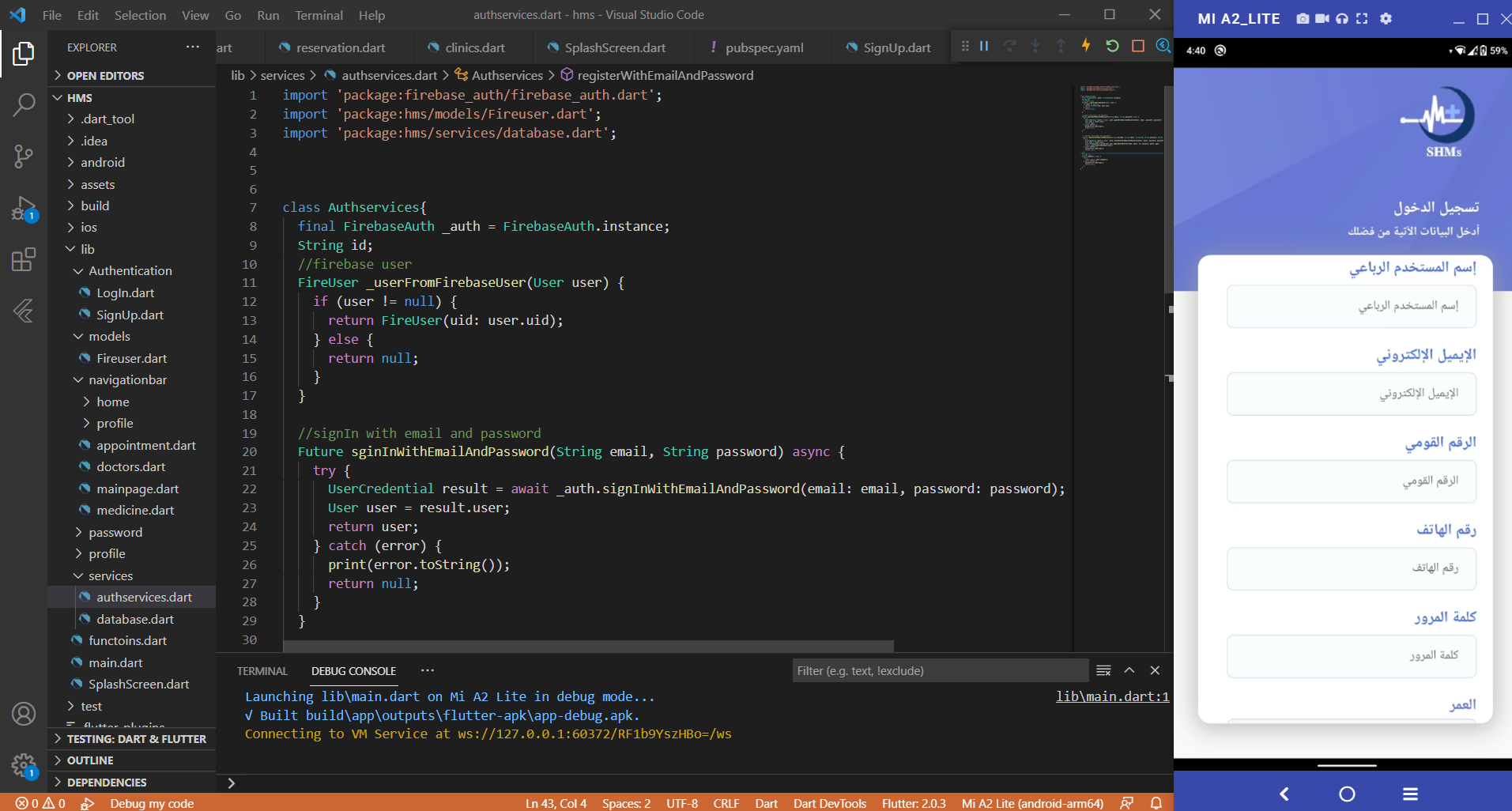
* **Firebase Authentication**

## Authentication state

Firebase Authentication provides many methods and utilities for enabling you to integrate secure authentication into your new or existing Flutter application. In many cases, you will need to know about the authentication state of your user, such as whether they're logged in or logged out.

We use it in the Project to authentication the sign in and register an account

dependencies: firebase\_auth: ^1.4.1

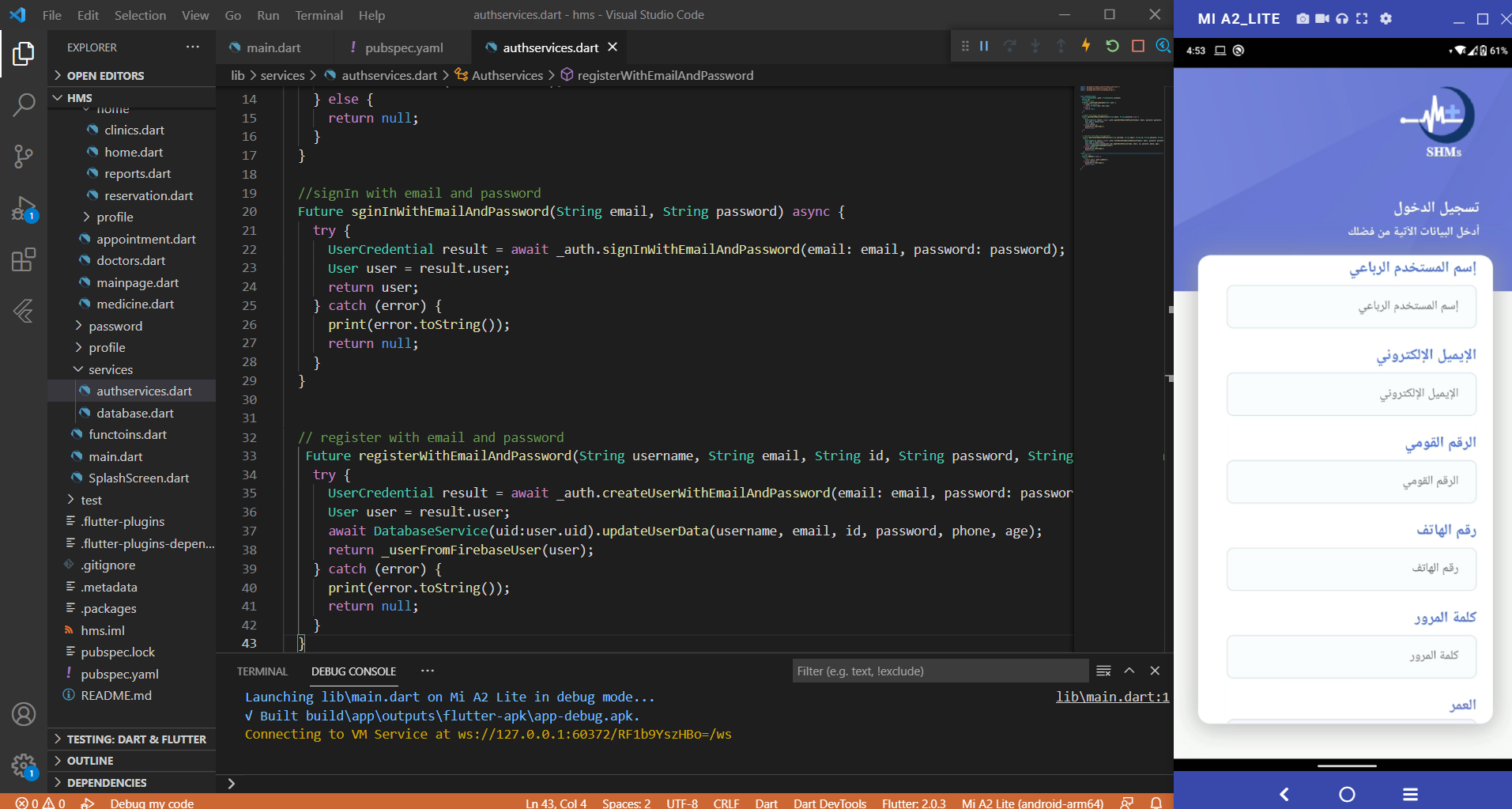


* **Cloud Firestore**

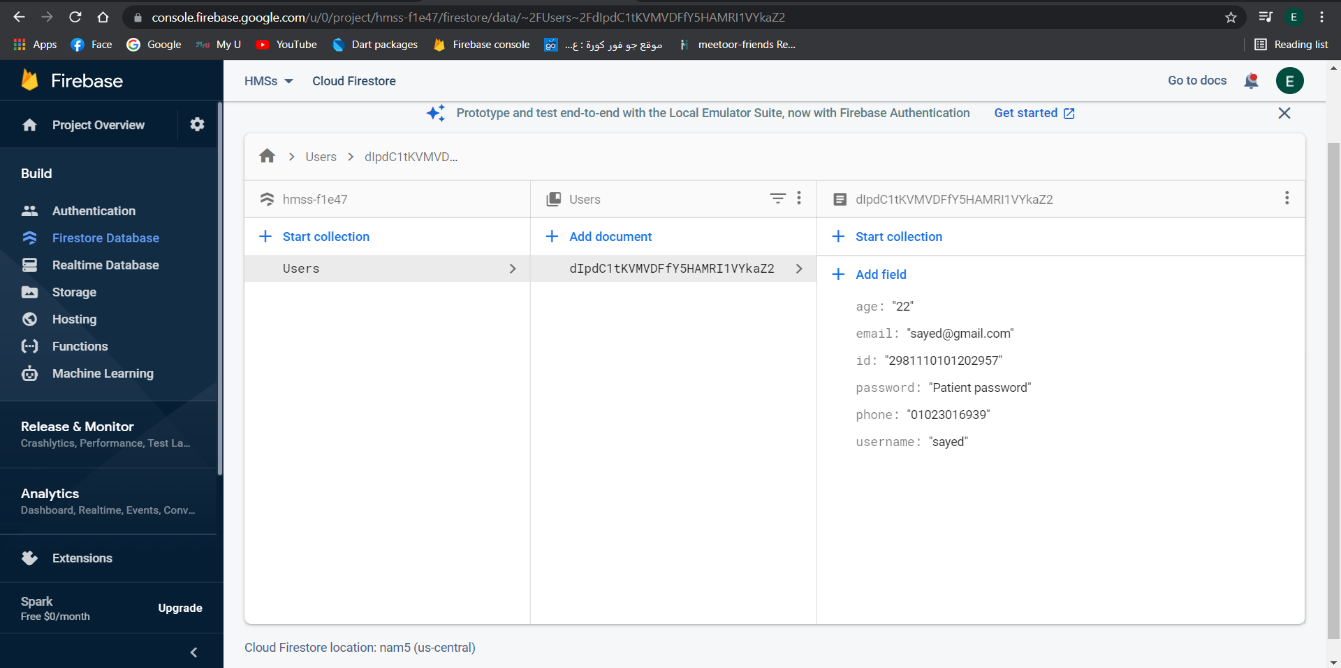
## Collections & Documents

## Firestore stores data within "documents", which are contained within "collections". Documents can also contain nested collections. For example, our users would each have their own "document" stored inside the "Users" collection. The collection method allows us to reference a collection within our code.

dependencies: cloud\_firestore: ^2.2.2



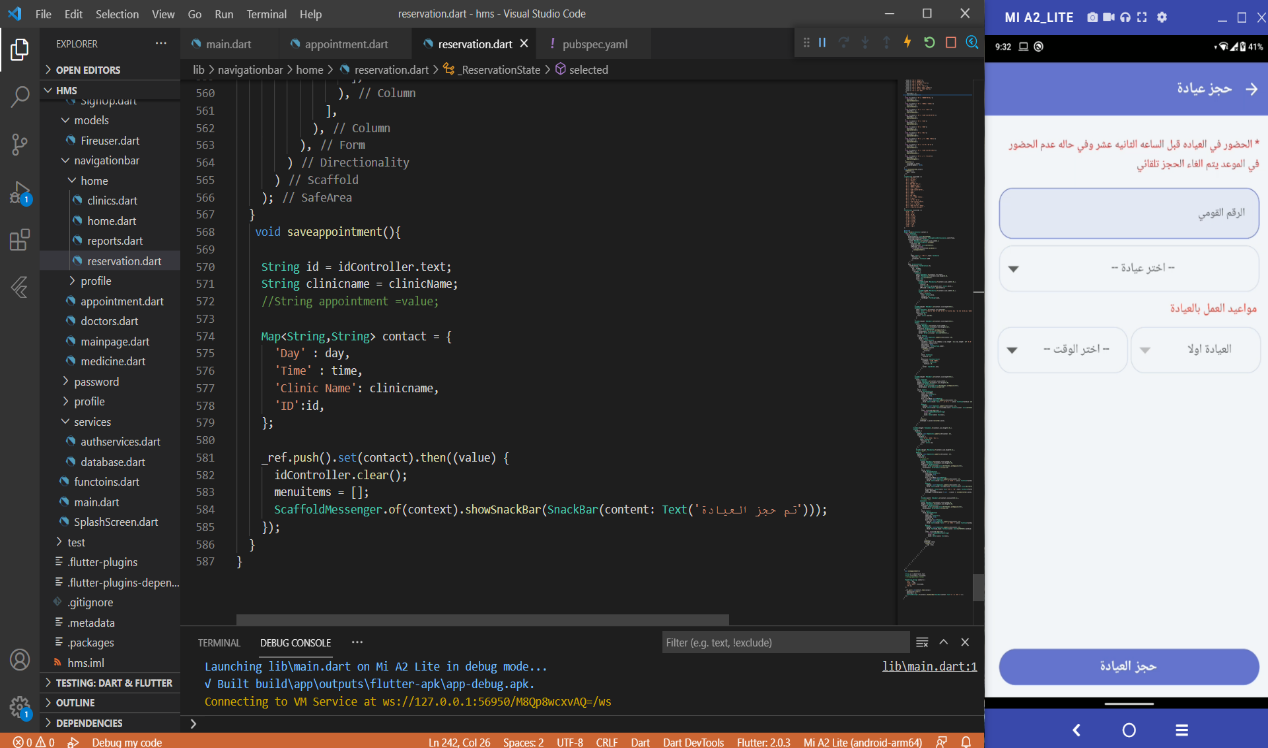
here we use cloud\_firestore to store the input the user enter and there is an example for the registration output in firebase console



* Firebase database

We use this library in the project to store the reservation inputs into database to show them into an appointment page

dependencies: firebase\_database: ^7.1.1



All of inputs ID, clinic name, and appointment store in firebase database

And the output

