

# Firestore Initialize - New Clinic

After you created a new clinic, on the manage page in the management project you can edit the clinic properties. In the form that will open, if there is no firebase config already entered into the clinic, you will see a message with the steps to follow (this guide will do the same with more details).

Firestore Properties:

Firestore Project Init Guide [^](#)

It seems like you didn't set any Firestore project for this clinic yet.

Open the Pdf Guide or follow the next steps:

[Download Pdf Guide](#)

Please go to [Firestore Console](#) and create a project.

After that create web-app and copy the config to the field 'Firestore Props' below.

Create a **Realtime database** and set the rules (**read and write**) to True. (You can follow this [video](#))

In the Firestore project look for **Authetication**, setup our Firestore Authetication method to be Email/Password and Enable the first option.

Don't forget to check the 'Firestore Props Changed' checkbox!

Firestore Config

Copy the firestore config propertis from your firestore app settings.

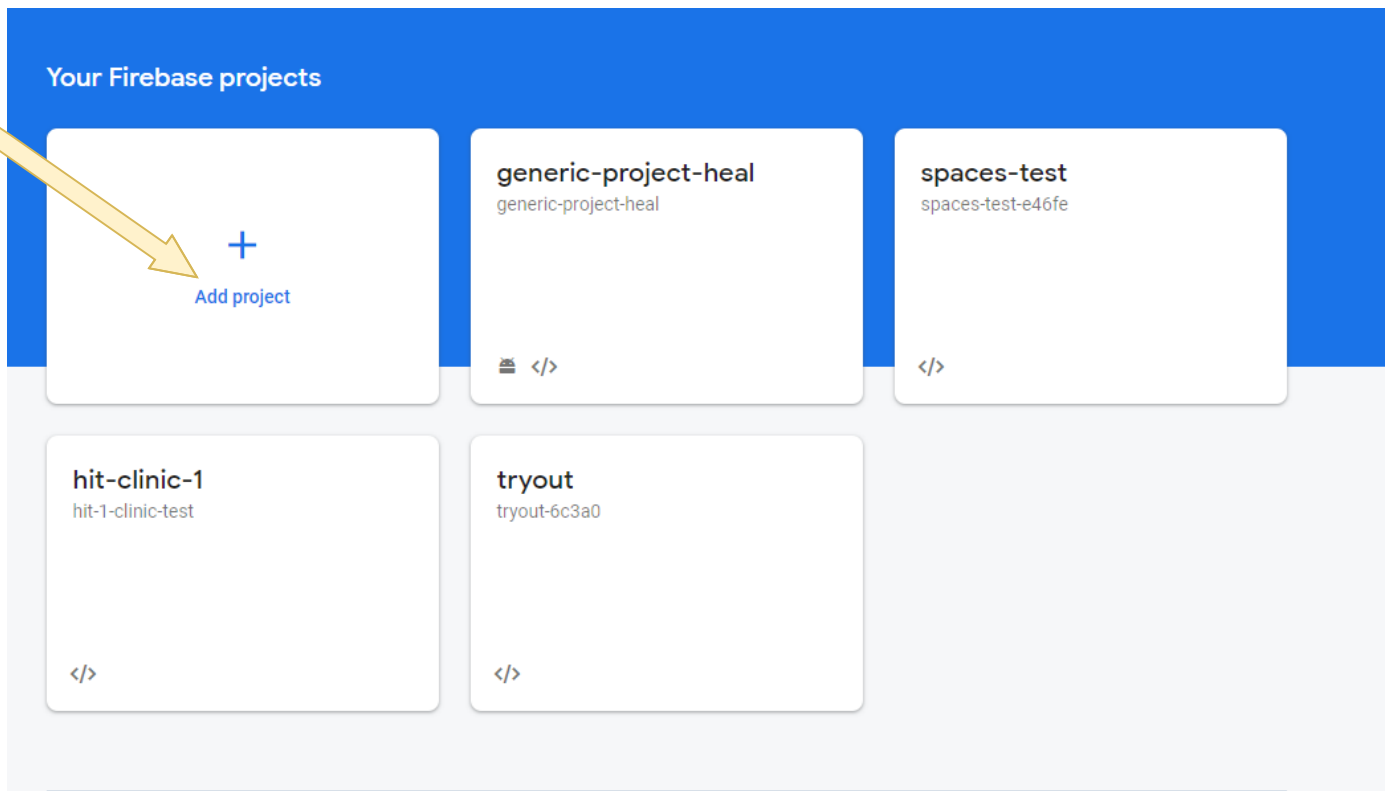
## Step 1: Generate a new firebase project

Go to the firebase console and press add project.

Follow the steps: enter the project name (you can use the clinic name), choose the user that will control the firebase, and let the project generate.

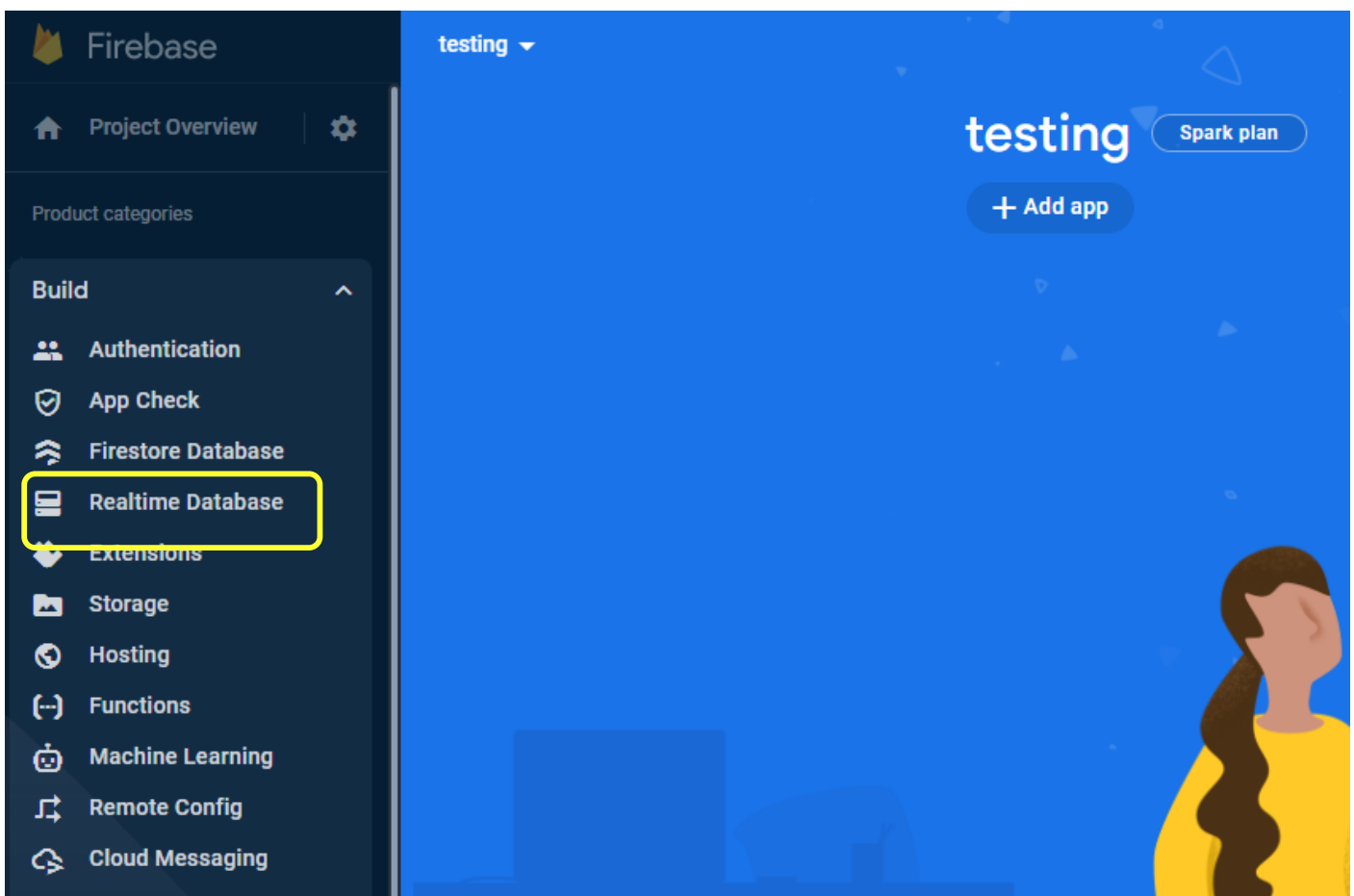
After that firebase will redirect you to the project console.

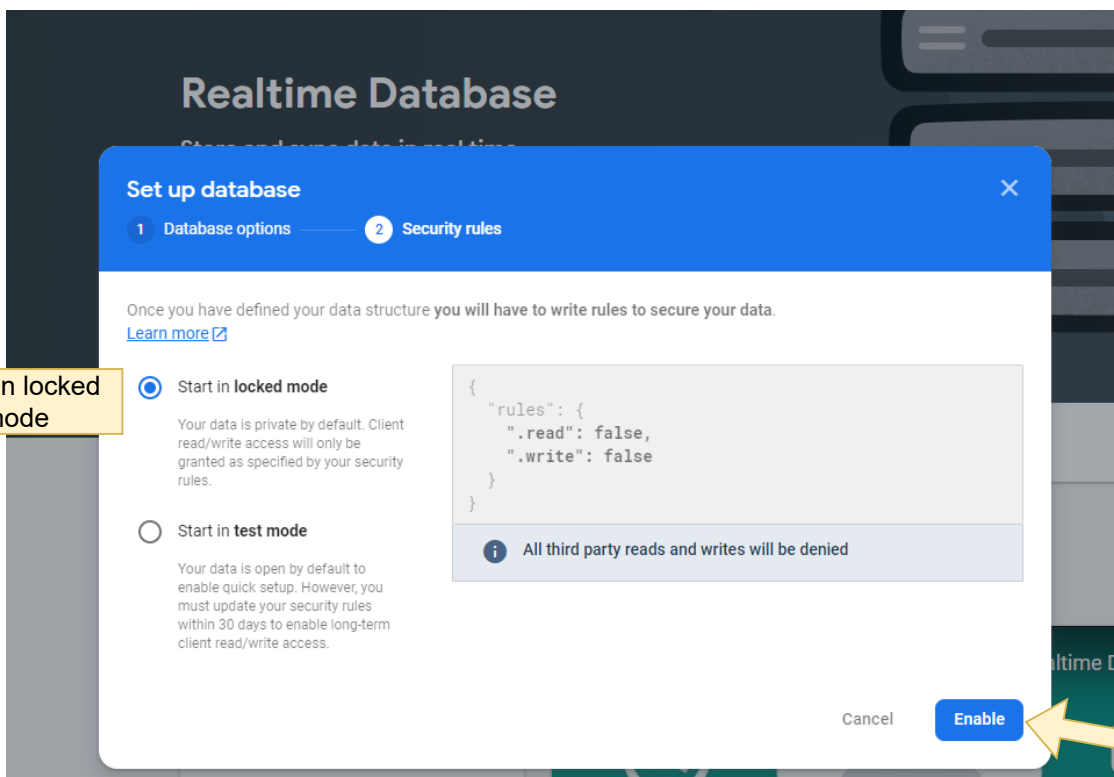
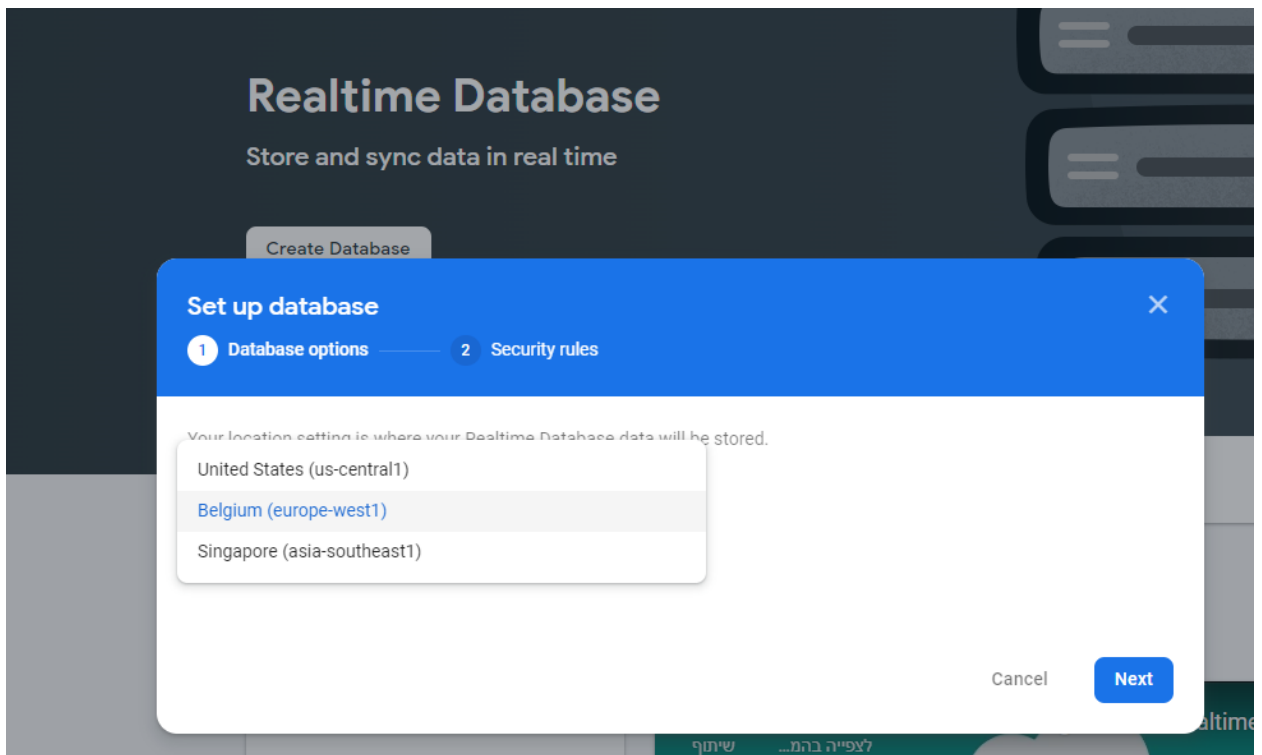
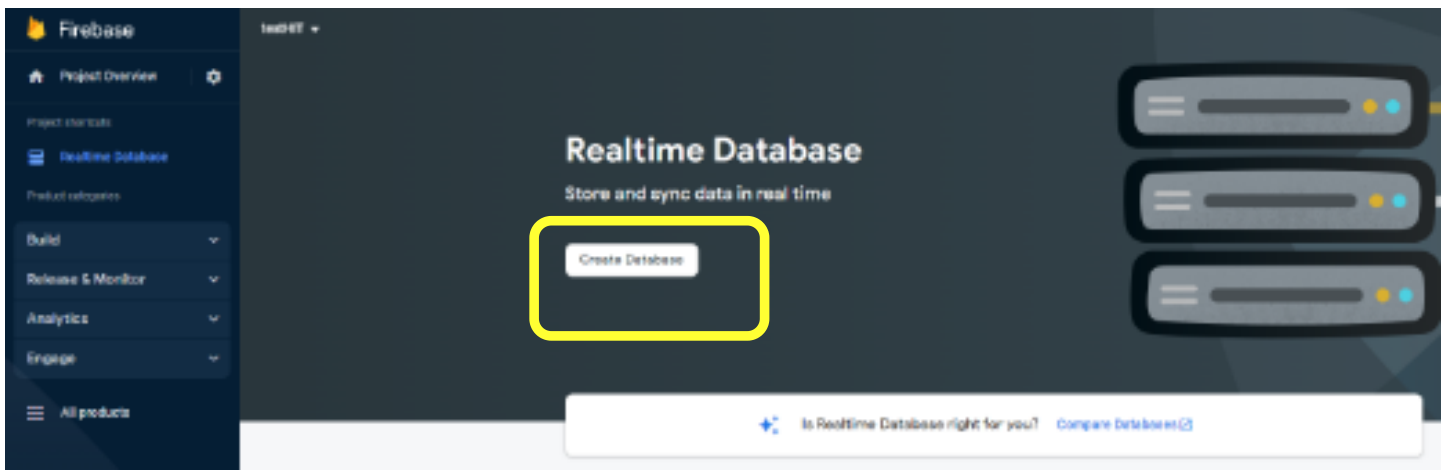
[Firestore console link.](#)



## Step 2: Creating Realtime Database

Let's create our new Realtime database.  
Under the Build section, we can find Realtime Database.  
Go to the section and press create a database, choose your database location (Europe for us - Belgium, Ireland and etc)





Start in locked mode

And enable the database

We are not done yet.  
We need to change the Rules of our new database.  
In the Realtime database section now you can see our database.  
You can see you have an Rules option on the top bar.

## Realtime Database

The screenshot shows the 'Rules' tab in the Realtime Database console. The 'Default security rules are locked from access' message is visible. The current rules configuration is:

```
1 {  
2   "rules": {  
3     ".read": false,  
4     ".write": false  
5   }  
6 }
```

A yellow callout box on the right contains the text: "We will change this config into this:" followed by the new configuration:

```
{  
  "rules": {  
    ".read": "true",  
    ".write": "true",  
    "Users": {  
      ".indexOn": ["first_name", "last_name", "role"]  
    }  
  }  
}
```

Buttons for 'Edit rules', 'Monitor rules', 'Rules playground', 'Learn more', and 'Dismiss' are also visible.

## Realtime Database

The screenshot shows the 'Rules' tab with the updated configuration. A yellow callout box above the 'Publish' button says "Now we press Publish". An arrow points from the callout to the 'Publish' button. The configuration is:

```
1 {  
2   "rules": {  
3     ".read": true,  
4     ".write": true,  
5     "Users": {  
6       ".indexOn": ["first_name", "last_name", "role"]  
7     }  
8   }  
9 }
```

Buttons for 'Edit rules', 'Monitor rules', 'unpublished changes', 'Publish', and 'Discard' are visible.

After that our Realtime database should look like this:

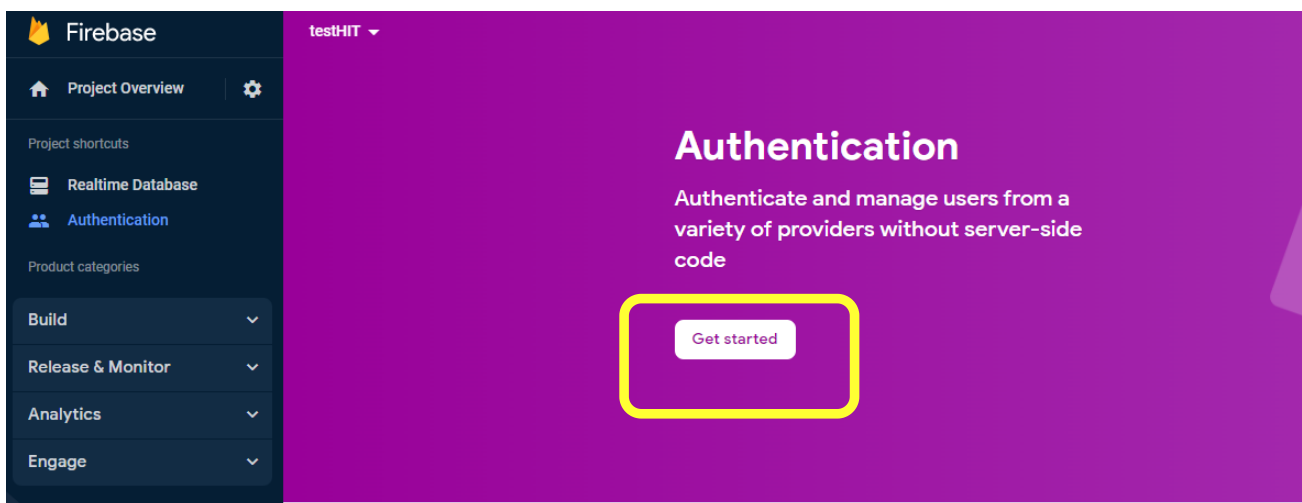
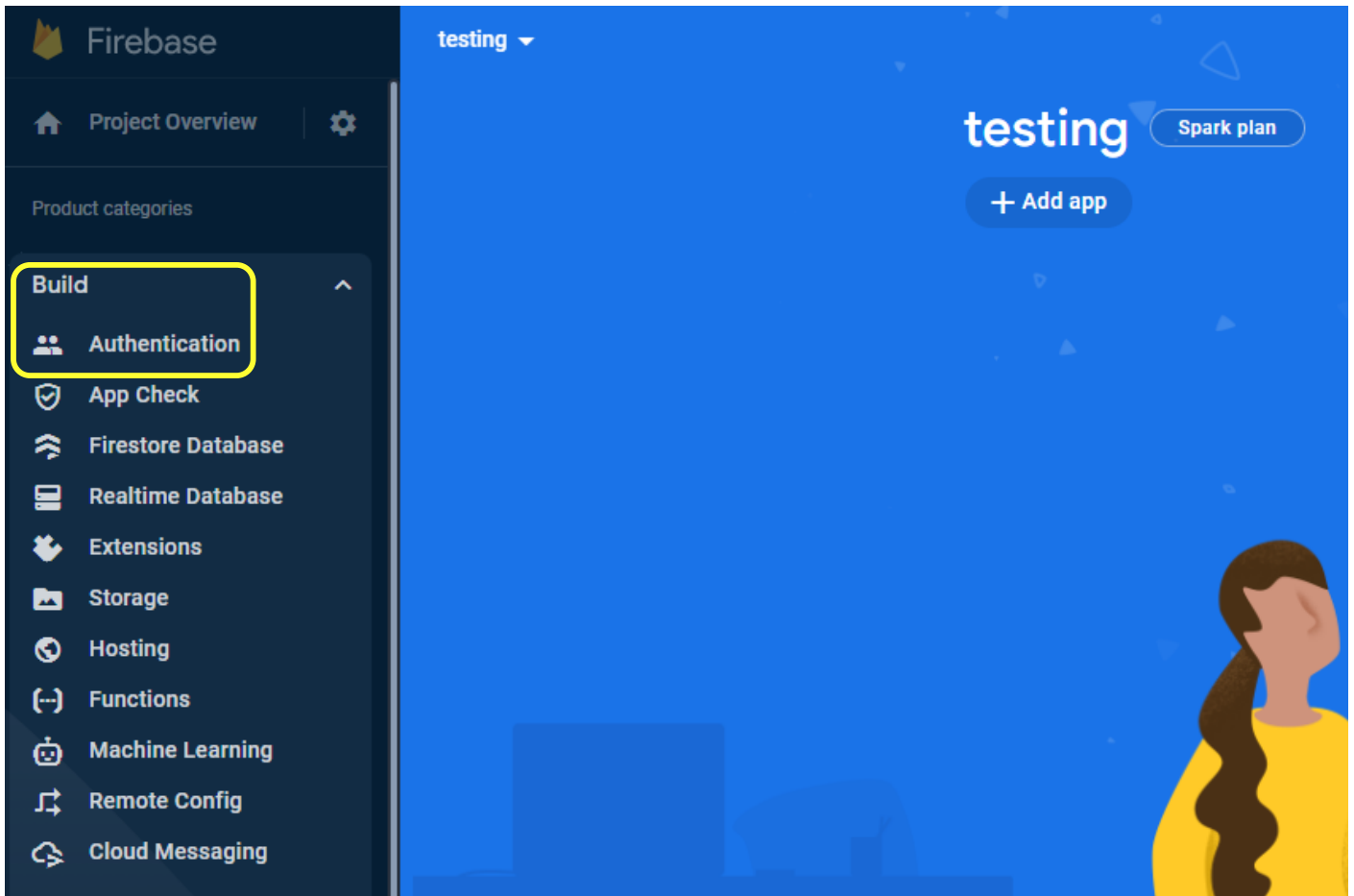
The screenshot shows the final state of the Realtime Database console. The 'Rules' tab is selected, and the configuration is visible in the playground area:

```
https://testhit-a46cc-default-rtdb.europe-west1.firebaseio.com  
  
https://testhit-a46cc-default-rtdb.europe-west1.firebaseio.com/:null
```

The left sidebar shows the Firebase navigation menu with options like 'Project Overview', 'Realtime Database', 'Authentication', 'Build', 'Release & Monitor', 'Analytics', and 'Engage'.

## Step 3: Setup Authentication

Under the Build section on the sidebar, you can find 'Authentication' press on it.



testing ▾

# Authentication

Users Sign-in method Templates Usage Settings

Get started with Firebase Auth by adding your first sign-in method

Native providers

- Email/Password
- Phone
- Anonymous

Additional providers

- Google
- Game Center
- Microsoft
- Facebook
- Apple
- Twitter
- Play Games
- GitHub
- Yahoo

Custom providers

- OpenID Connect
- SAML

Advanced

- SMS Multi-factor Authentication

Here we want to choose Email/Password as our authentication method

Sign-in providers

Mark the first option to be Enable, and save.

Email/Password  Enable

Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives. [Learn more](#)

Email link (passwordless sign-in)  Enable

Cancel **Save**

After that our Authentication page should look like this:  
With Add user button

testHIT ▾

# Authentication

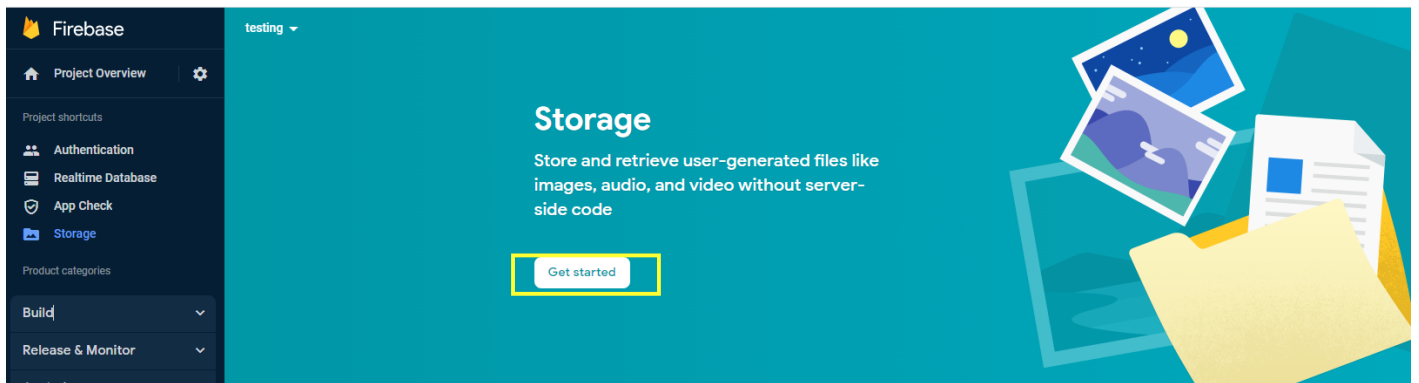
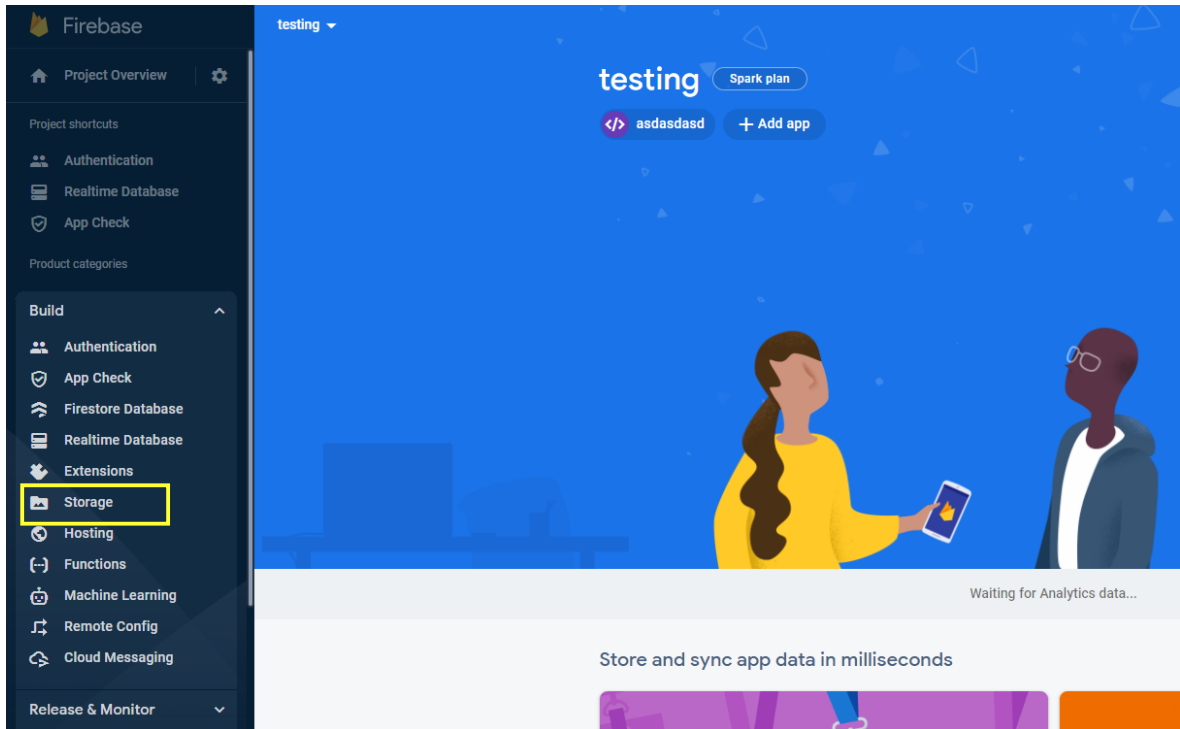
Users Sign-in method Templates Usage Settings

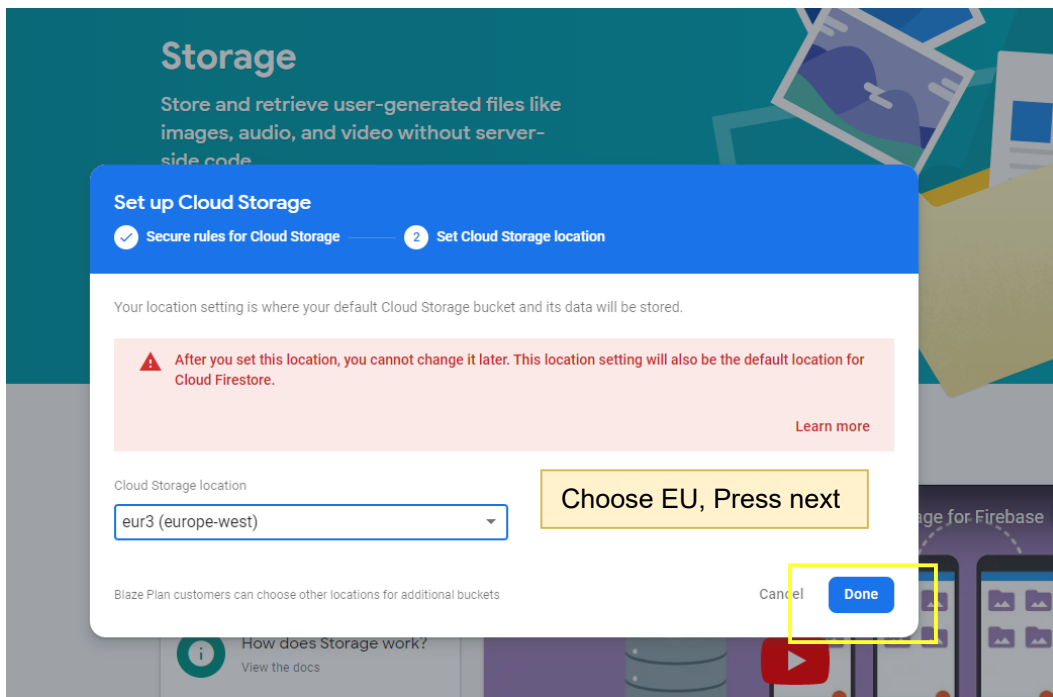
**Add user** ↻ ⋮

Identifier	Providers	Created ↓	Signed In	User UID
No users for this project yet				

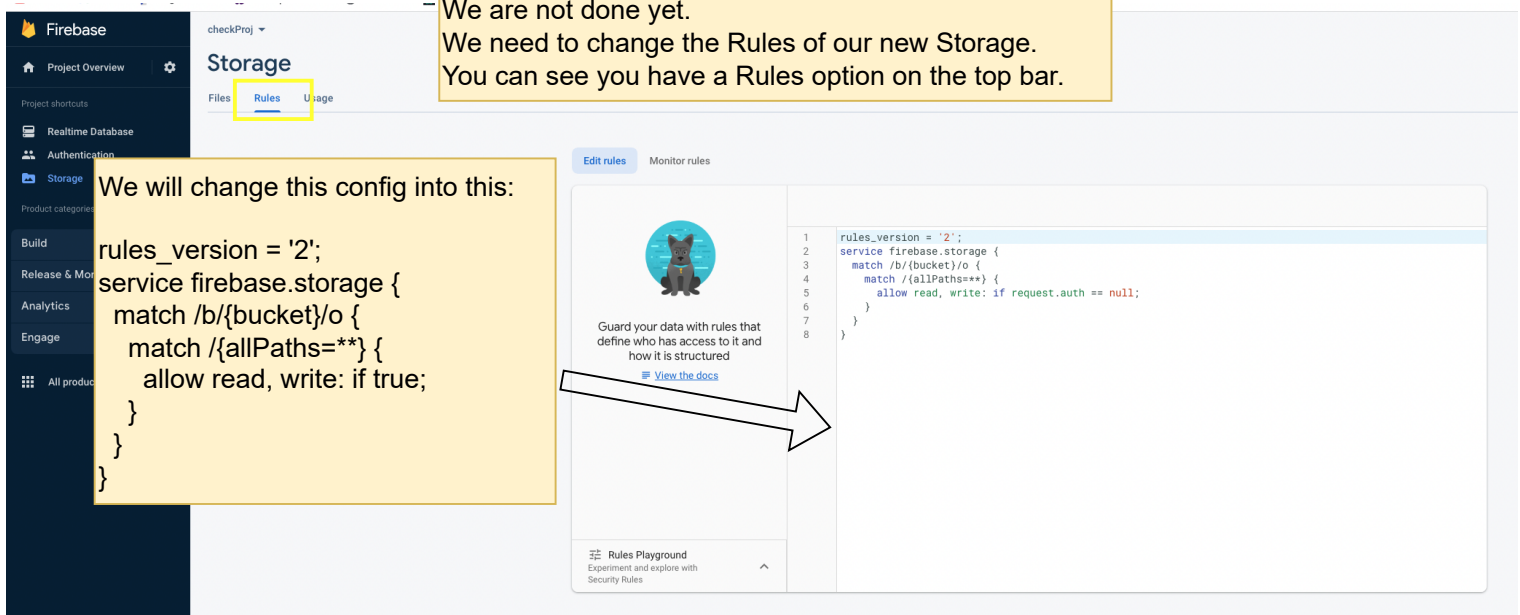
## Step 4: Steup storage

Under the Build section on the sidebar, you can find 'Storage' press on it.





We are not done yet.  
We need to change the Rules of our new Storage.  
You can see you have a Rules option on the top bar.

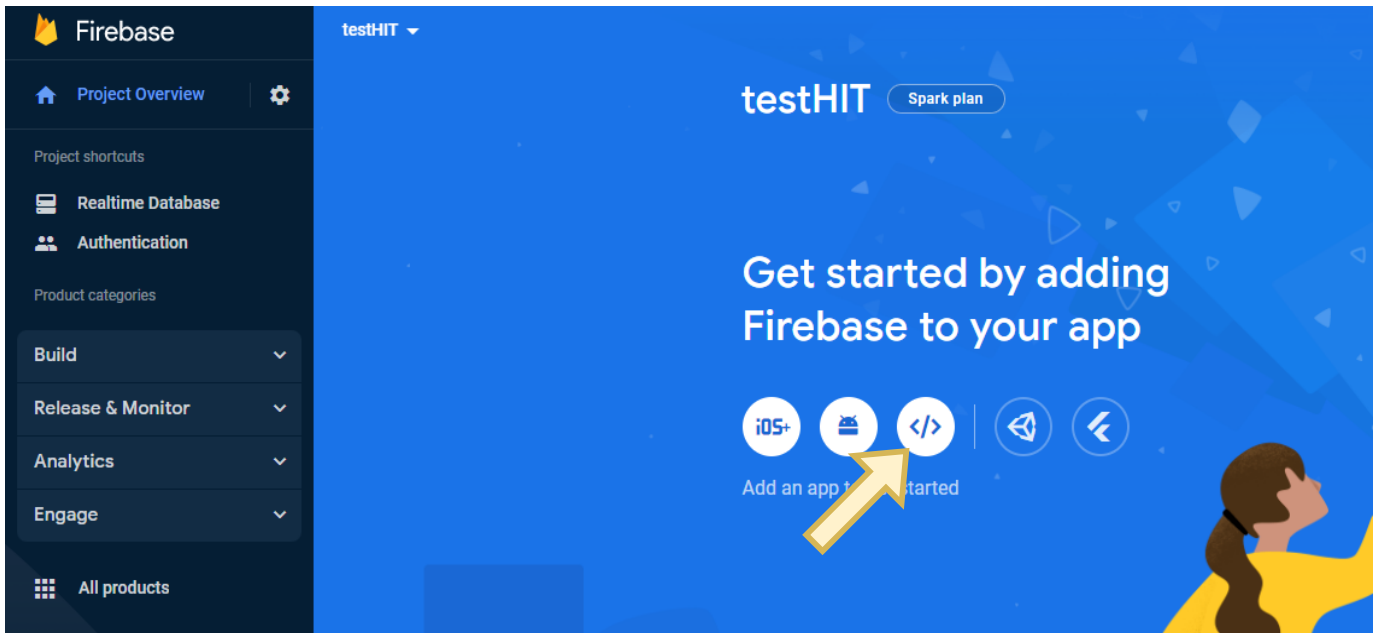


## Step 5: Create a web app for the new project

Now you are in the new project console, we want to create a new web app.

Press on the firebase Logo in the top left corner will redirect to the project console page.

Now can choose to create our new web app.



### 2 Add Firebase SDK

Use npm [?](#)  Use a <script> tag [?](#)

If you're already using [npm](#) and a module bundler such as [webpack](#) or [Rollup](#), you can run the following command to install the latest SDK:

```
$ npm install firebase
```

Create new web app, enter name and generate the app.  
In the end you will get this config file.

Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyAkKU3tIrfkNEsqf_puOpNADyW40juaLpg",
  authDomain: "testhit-a46cc.firebaseio.com",
  databaseURL: "https://testhit-a46cc-default-rtbd.firebaseio.com",
  projectId: "testhit-a46cc",
  storageBucket: "testhit-a46cc.appspot.com",
  messagingSenderId: "692378054216",
  appId: "1:692378054216:web:e0744ebed31f1c6f4cb27d",
  measurementId: "G-M5X52G2YFQ"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

We will copy this section without the brackets !  
We should make sure we have the next keys:  
**apiKey, authDomain, databaseURL, storageBucket**

**Note:** This option uses the [modular JavaScript SDK](#), which provides reduced SDK size.

Learn more about Firebase for web: [Get Started](#), [Web SDK API Reference](#), [Samples](#)

We will copy this config into our clinic edit form and paste it

The screenshot shows the 'uilib' configuration form in the Firebase console. The form includes fields for 'Email Address' (uilib@test.com) and 'Clinic Name' (uilib). Under 'Firebase Properties:', there is a 'Firebase Project Init Guide' dropdown and a 'Firebase Config' field containing the following JSON:

```
rttdb.europe-west1.firebaseio.com",  
projectId: "generic-management-hit",  
storageBucket: "generic-management-hit.appspot.com",  
messagingSenderId: "110025314903",  
appId: "1:110025314903:web:77264c5402735b14670912",  
measurementId: "G-5QGGTD28PV"
```

Below the config field is an 'Api key post' field. Under 'Chat Firebase Properties:', there is a 'Firebase Chat Config' field with instructions to copy chat firebase config properties from the app settings.

## Step 6: Create second Firebase for Chat

Now we will create another firebase as mentioned. This time we will create a web app and an android app.

Create new android app

The screenshot shows the Firebase console interface for the 'generic management hit' project. A yellow arrow points to the '+ Add app' button. A yellow text box with the text 'Press add app' is overlaid on the arrow.



After the RealtimeDatabase is finished and we setup the new rules we can create a new web app, all the steps mentioned above.

After creating the web app we can copy the new config and paste it into the Firebase chat config in the clinic update form.

The screenshot shows a web application interface for managing clinics. On the left, there is a table with columns for Clinic Name, Clinic Email, and Account status. A modal window is open in the center, titled 'Chat Firebase Properties:'. It contains a 'Firebase Chat Config' section with a text area containing the following JSON configuration:

```
firebaseConfig: {
  databaseURL: "https://generic-management-nit-default-rtdb.europe-west1.firebaseio.com",
  projectId: "generic-management-hit",
  storageBucket: "generic-management-hit.appspot.com",
  messagingSenderId: "110025314903",
  appId: "1:110025314903:web:77264c5402735b14670912",
  measurementId: "G-5QGGTD28PV"
}
```

Below the config is an 'Api key post' field and a section for 'Apps' with checkboxes for 'accounts', 'chat', 'dashboard', 'documents\_share', 'measurements', 'medications', and 'questionnaire'. A yellow arrow points to the 'Firebase Chat Config' text area.

Now we will generate cloud messaging API key  
Under project settings we can find Cloud Messaging

The screenshot shows the 'Project settings' page in the Firebase console for the project 'generic management hit'. The 'Cloud Messaging' tab is selected and highlighted with a yellow arrow. The page displays the following project information:

Property	Value
Project name	generic management hit
Project ID	generic-management-hit
Project number	110025314903
Default app instance ID	Not yet selected

generic management hit

## Project settings

General **Cloud Messaging** Integrations Service accounts Data privacy Users and permissions App Check

**Firebase Cloud Messaging API (V1)** Enabled  
 Recommended for most use cases. [Learn more](#)

Sender ID	Service Account
110025314903	<a href="#">Manage Service Accounts</a>

**Cloud Messaging API (Legacy)** Disabled  
 If you are newly integrating messaging into your app, use the latest Firebase Cloud Messaging API (V1). If you are an existing user of Cloud Messaging API (Legacy), consider migrating to the latest Firebase Cloud Messaging API (V1). [Learn more](#)

[Manage API in Google Cloud Console](#)

Now we want to enable cloud messaging. In the right side of the Cloud Messaging API we will have 3 dots we will press on them and then on the button that will be shown

Google Cloud generic management hit

## Cloud Messaging

[Google](#)

Cross-platform messaging solution that lets you reliably deliver messages at no cost.

**ENABLE** Press ENABLE

[OVERVIEW](#) [DOCUMENTATION](#) [SUPPORT](#)

Now we have cloud messaging key and we will copy it to the clinic form

generic management hit

## Project settings

General **Cloud Messaging** Integrations Service accounts Data privacy Users and permissions App Check

**Firebase Cloud Messaging API (V1)** Enabled  
 Recommended for most use cases. [Learn more](#)

Sender ID	Service Account
110025314903	<a href="#">Manage Service Accounts</a>

**Cloud Messaging API (Legacy)** Enabled  
 If you are newly integrating messaging into your app, use the latest Firebase Cloud Messaging API (V1). If you are an existing user of Cloud Messaging API (Legacy), consider migrating to the latest Firebase Cloud Messaging API (V1). [Learn more](#)

Key	Token
Server key	AAAAGZ4FElc-APA91bE3Rs99Y3lnbBYdgEXnM_PlkzkaasMNWjmdkd0v1vzdfM053Z_E-8TWQ-goI2h8f53pU1uL-PRObXHRDHVvgmJJ0CExpQm190MB2JpW9n6cjVVTc0mb3ON2p3s-yFY3Je4M5-uc
Sender ID	
110025314903	

[Add server key](#)

Admin Panel Users Manage Applications Manage Measurements

Enter clinic name...

Clinic Name	Clinic Email	Accounts
clinic1	clinic1@example.com	✓
riri	riri@example.com	✓
clinic3	clinic3@example.com	✓
samuel	samuel@example.com	✓
uilib	uilib@test.com	✓

It seems like you don't yet have a Firebase project for this clinic yet.

Open the pdf guide and follow the steps:

[Download Pdf Guide](#)

Firestore Config

```
databaseURL: "https://generic-management-nit-default-rtdb.europe-west1.firebaseio.com",
projectId: "generic-management-hit",
storageBucket: "generic-management-hit.appspot.com",
messagingSenderId: "110025314903",
appId: "1:110025314903:web:77264c5402735b14670912",
measurementId: "G-5QGGTD28PV"
```

Api key post

```
AAAAAGZ4FEIc:APA91bE3Rs99Y3lnbBYdgEXnM_PkzkaasMNWjmdkd0vV1vZdfMQ53Z_E-BTWQ-gol2h8f53pUP1uLPRObXHRDHVvGmJj0CEXpQm190MB2J
```

Chat Firebase Properties:

Firestore Chat Config

```
databaseURL: "https://generic-management-nit-default-rtdb.europe-west1.firebaseio.com",
projectId: "generic-management-hit",
storageBucket: "generic-management-hit.appspot.com",
messagingSenderId: "110025314903",
appId: "1:110025314903:web:77264c5402735b14670912",
measurementId: "G-5QGGTD28PV"
```

Apps:

Search Q

Accounts	Medications	Questionnaire	Edit
✓	X	X	
✓	X	X	
✓	X	X	
✓	X	X	

Now can choose the applications and the measurements for this project.

Now we can submit, make sure to press on "Save And generate" so the management project will generate a new Django project and will register the clinic into the DB.

storageBucket: "chat-test-hit.appspot.com",  
messagingSenderId: "493286566214",  
appId: "1:493286566214:web:6235cc0679c49f62d447",  
measurementId: "G-WFK4RQJSSC"

Proxy Props:

Public Server URL  
(The url the application will be redirected to)  
https://my.public.domain/

Private Server URL  
(The url this server will call when proxying the login request)  
https://my.local.domain/

Apps:

accounts  chat  dashboard  documents\_share  
 measurements  medications  new\_report  questionnaire

Measurements:

Add Custom Measurement:  
Measurement name  
Measurement value type  
Save Measurement

[Save](#) [Save And Generate](#)

## Clinics

### Generating Project . . .

Clinic Name	Clinic Email	Accounts	Dashboard	Measurements	Medications	Questionnaire	Chat	Edit
react123	test@example.com	✓	✓	✓	✓	X	X	
ui-lib	uilib@test.com	✓	✓	✓	X	X	✓	
mvt	memberview@test.com	✓	✓	✓	✓	✓	✓	
spaces	spaces@test.com	✓	✓	X	X	✓	✓	
owners	owners@test.com	✓	✓	X	X	X	✓	
samuel	samuel@example.com	✓	✓	X	X	X	X	

At the end of the process you will see this success message appears. In this message you can see the password for the clinic to login to the new project the username will be the clinic email.

# Clinics

Finished to create your new project!

The password for clinic firebase login:

'+kz/F8BhC

Password

Close

Clinic Name	Clinic Email	Accounts	Dashboard	Measurements	Medications	Questionnaire	Chat	Edit
react123	test@example.com	✓	✓	✓	✓	X	X	
ui-lib	uilib@test.com	✓	✓	✓	X	X	✓	
mvt	memberview@test.com	✓	✓	✓	✓	✓	✓	
spaces	spaces@test.com	✓	✓	X	X	✓	✓	
owners	owners@test.com	✓	✓	X	X	X	✓	
samuel	samuel@exmaple.com	✓	✓	X	✓	✓	X	

Username

Finally the new clinic DB should look like this:

The screenshot shows the Firebase Realtime Database console for project 'testHIT'. The 'Data' tab is selected, displaying the database structure. The root node is 'https://testhit-a46cc-default-rtdb.europe-west1.firebaseio.com'. Underneath, there are three nodes: 'Features', 'Users', and 'checkMes'. A sidebar on the left shows the project overview and navigation options.

And the Authentication:

The screenshot shows the Firebase Authentication console for project 'testHIT'. The 'Users' tab is selected, displaying a table of users. The table has columns for Identifier, Providers, Created, Signed In, and User UID. One user is listed with the identifier 'samuel@exmaple.com', provider 'Email', created on 'Sep 3, 2022', signed in on 'Sep 3, 2022', and user UID 'a7azaAF40Mc5dMcUW6cju3Elb5...'. A search bar and 'Add user' button are visible at the top.

And the folder created under the clinic email name

Name	Date modified	Type	Size
samuel@exmaple.com	13/08/2022 20:34	File folder	