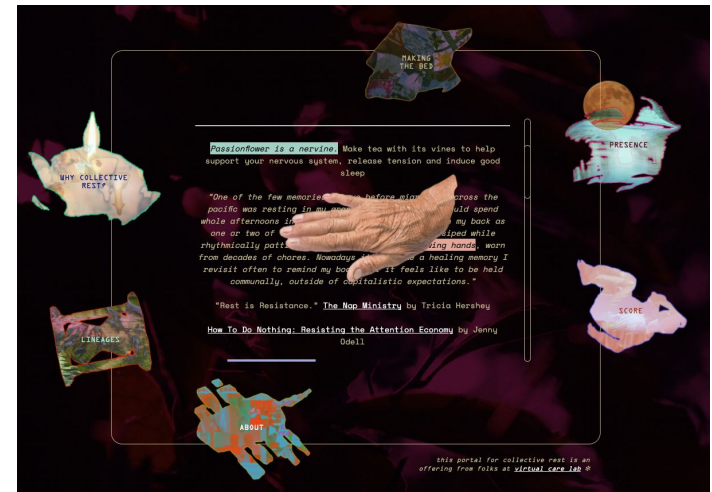


Firestore Demo

Some *inspirations*



<https://rest.virtualcarelab.com/>

Memorypedia

memory dream



color



creepy



killer doll



potato



far west horse



a very big lake A very

small fish



light-ness regret



RETURN TO THE HOMEPAGE

Project link: <https://memorypedia.herokuapp.com/>

Code repo: <https://github.com/drawwithcode/2020-cgals>

Embed JavaScript to your website examples:

Firebase + plain JavaScript



Live Site: <https://qianqian-ye.com/iml300-fall2025/firebase-text-input/>
[Code link](#)

Embed JavaScript to your website examples:

Firebase + p5.js



Live Site: <https://qianqian-ye.com/im1300-fall2025/firebase-p5-canvas/>
Code link

In-class Firebase Workshop

1. Remix from this project

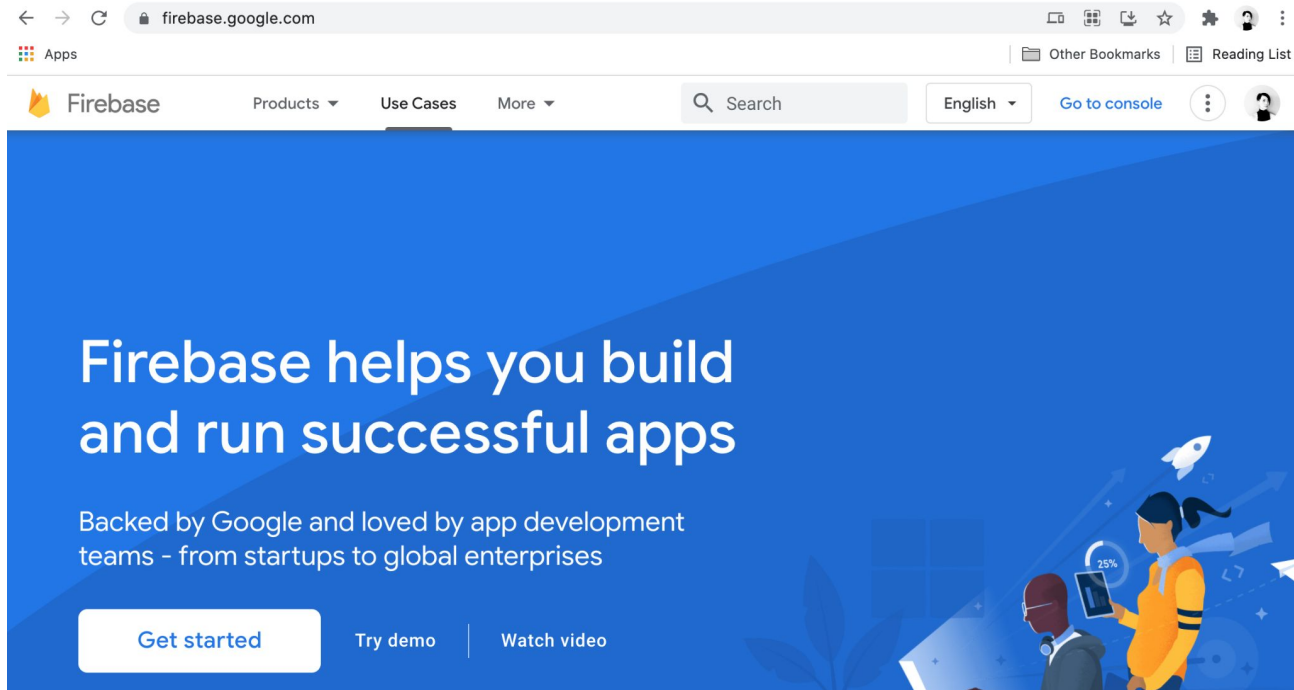
[Code link](#)



2. Go to Firebase & Click 'Get started'

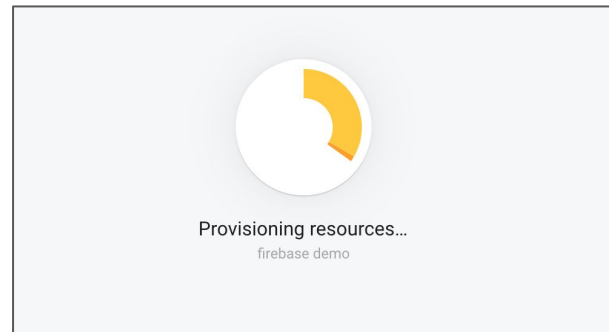
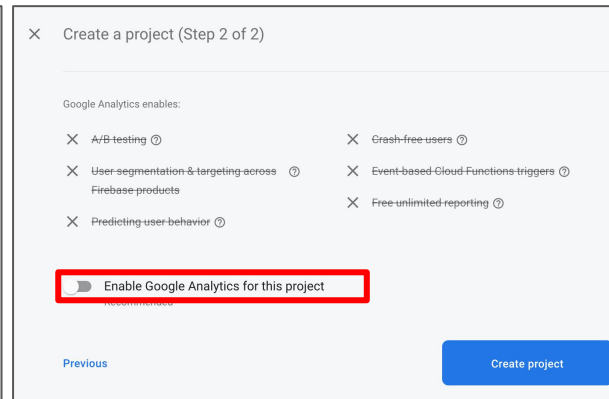
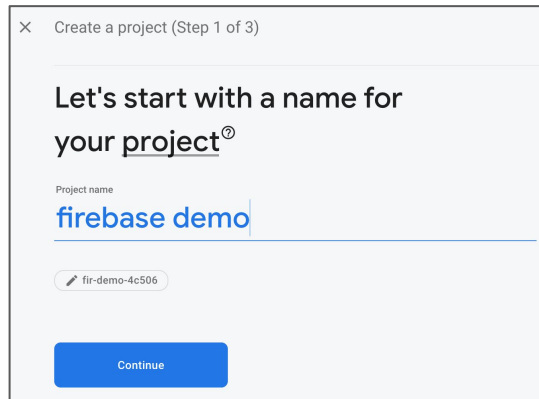
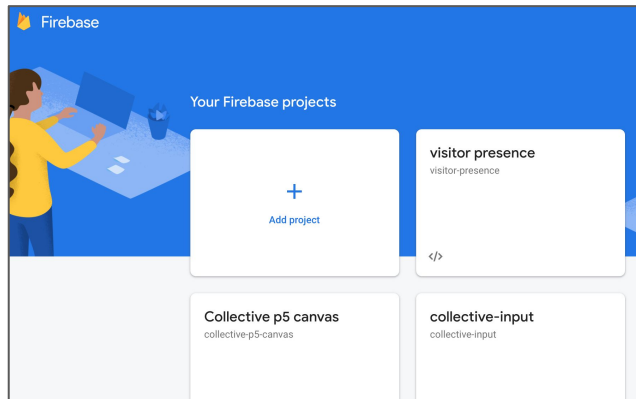
<https://firebase.google.com/>

If you have issues signing up with your usc.edu email address, try your other email address.

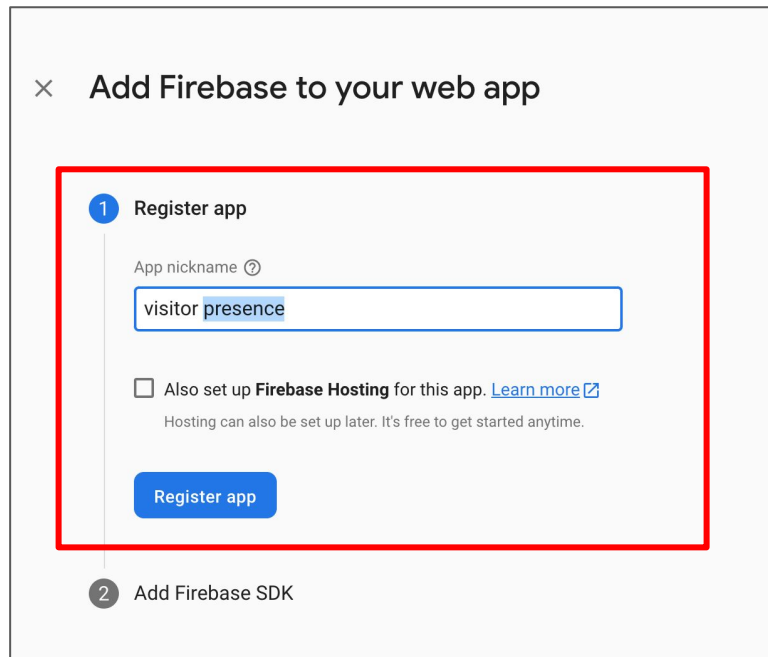
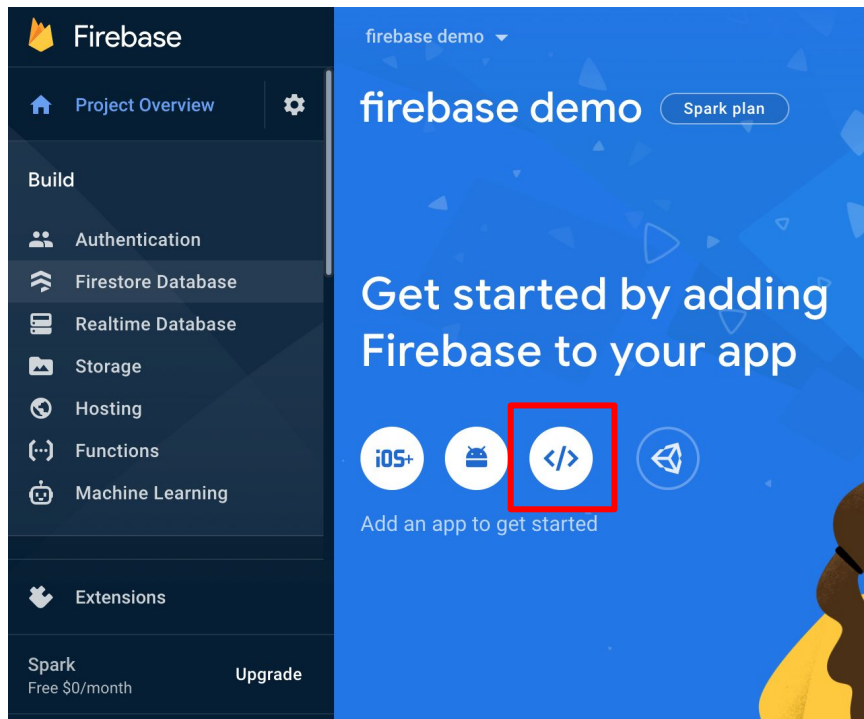


3. Add your Firebase project

<https://console.firebase.google.com/u/0/>



4. Adding firebase to your web app



5. Adding firebase configuration to your script on Glitch

× Add Firebase to your web app

✓ Register app

2 Add Firebase SDK

☐ Use npm ☒ Use a <script> tag

Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

```
<script type="module">
  // Import the functions you need from the SDKs you need
  import { initializeApp } from "https://www.gstatic.com/firebasejs/9.4.1/firebase-app.js";
  import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.4.1/firebase-analytics.js";
  // 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
  const firebaseConfig = {
    apiKey: "AIzaSyCaA-523AeXCLxc0hpQWnhWj2TCzQXPOM",
    authDomain: "visitor-presence.firebaseio.com",
    databaseURL: "https://visitor-presence-default-rtbd.firebaseio.com",
    projectId: "visitor-presence",
    storageBucket: "visitor-presence.appspot.com",
    messagingSenderId: "691771698376",
    appId: "1:691771698376:web:6cd1a50309fe475dbf49fd",
    measurementId: "G-GN7C7SWSKX"
  };

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

firebase-demo-2 Show script.js

Format This File

New File

assets
.gitignore
LICENSE
index.html
script.js
style.css

```
// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyCaA-523AeXCLxc0hpQWnhWj2TCzQXPOM",
  authDomain: "visitor-presence.firebaseio.com",
  databaseURL: "https://visitor-presence-default-rtbd.firebaseio.com",
  projectId: "visitor-presence",
  storageBucket: "visitor-presence.appspot.com",
  messagingSenderId: "691771698376",
  appId: "1:691771698376:web:6cd1a50309fe475dbf49fd",
  measurementId: "G-GN7C7SWSKX"
};

firebase.initializeApp(firebaseConfig);
database = firebase.database();
let ref = database.ref("words");
```

6.1 Set up the Realtime Database

The image shows the Firebase console interface. On the left, the 'Build' menu is open, and 'Realtime Database' is highlighted with a red rectangle. In the center, the 'Realtime Database' section is visible, with a 'Create Database' button highlighted by a red rectangle. On the right, a 'Set up database' dialog box is open, showing the 'Database options' step. The 'Realtime Database location' dropdown is set to 'United States (us-central1)' and is highlighted with a red rectangle. The dialog box also shows a 'Next' button.

Realtime Database

Store and sync data in real time

Create Database

Set up database

1 Database options 2 Security rules

Your location setting is where your Realtime Database data will be stored.

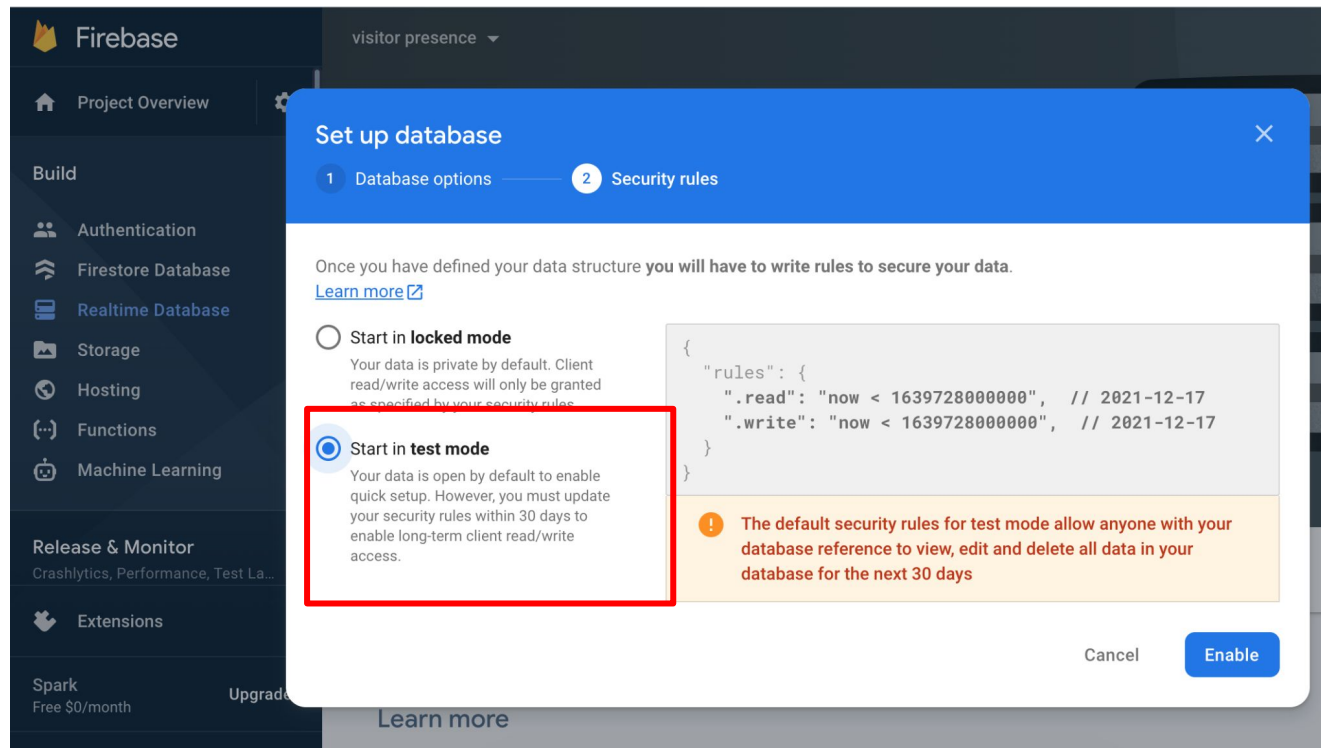
Realtime Database location

United States (us-central1)

Cancel Next

6.2 Set up the Realtime Database

You can start with the 'test mode'



Set up database

1 Database options — 2 Security rules

Once you have defined your data structure **you will have to write rules to secure your data.**
[Learn more](#)

☐ Start in **locked mode**
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

☒ Start in **test mode**
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

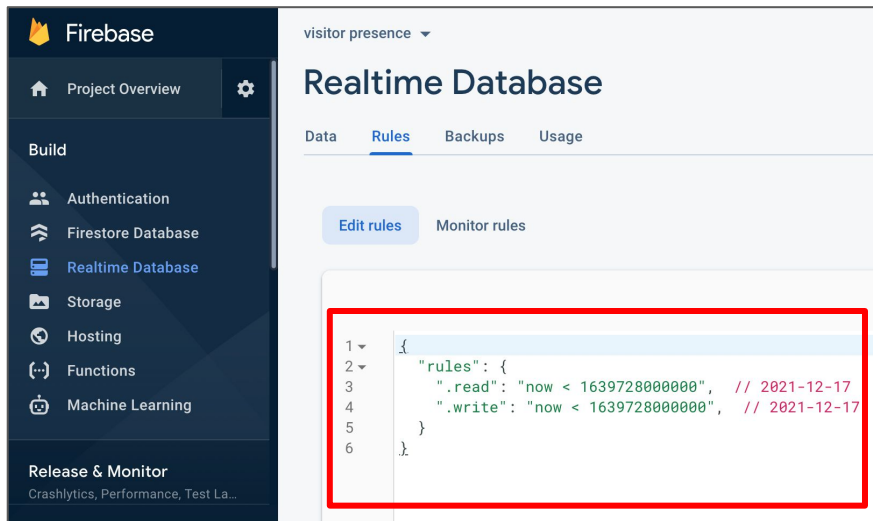
```
{
  "rules": {
    ".read": "now < 1639728000000", // 2021-12-17
    ".write": "now < 1639728000000", // 2021-12-17
  }
}
```

! The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Cancel **Enable**

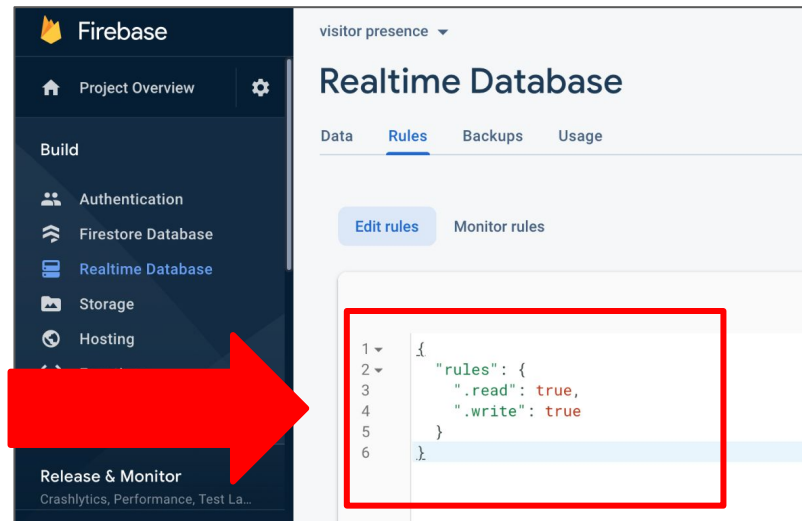
6.3 Set up the Realtime Database

You can update the rules later



This screenshot shows the Firebase Realtime Database Rules editor. The left sidebar contains the 'Build' menu with options like Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, and Machine Learning. The main panel is titled 'Realtime Database' and has tabs for 'Data', 'Rules', 'Backups', and 'Usage'. The 'Rules' tab is selected, showing 'Edit rules' and 'Monitor rules' buttons. A red box highlights the initial ruleset:

```
1 {  
2   "rules": {  
3     ".read": "now < 1639728000000", // 2021-12-17  
4     ".write": "now < 1639728000000", // 2021-12-17  
5   }  
6 }
```



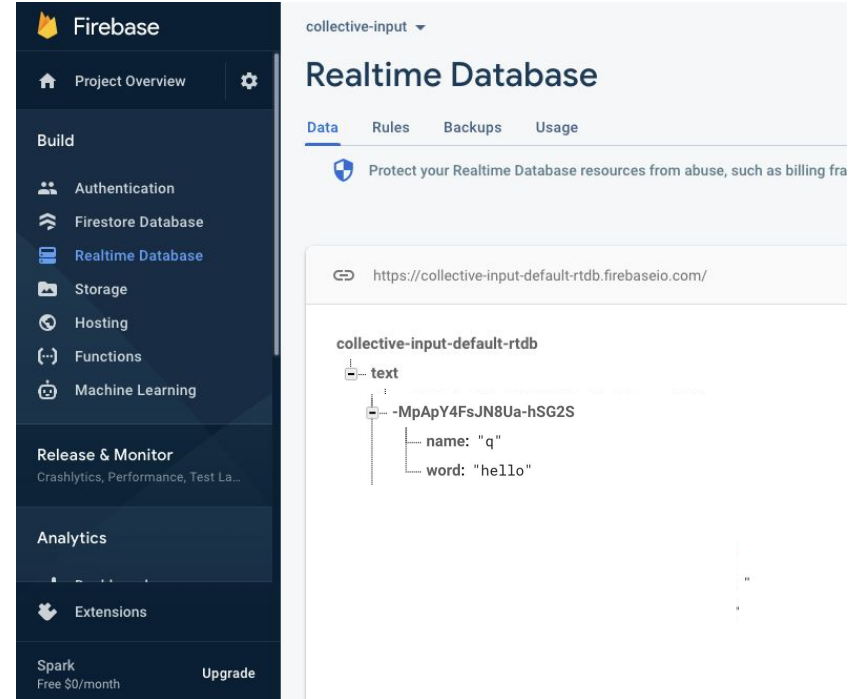
This screenshot shows the same Firebase Realtime Database Rules editor, but with updated, more permissive rules. A large red arrow points from the previous screenshot to this one, indicating the update. The ruleset is now:

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

7.1 Check out the database

Uncomment the line 22-27 and refresh the page, then see the changes in Firebase Database
Then remove this entry from Firebase

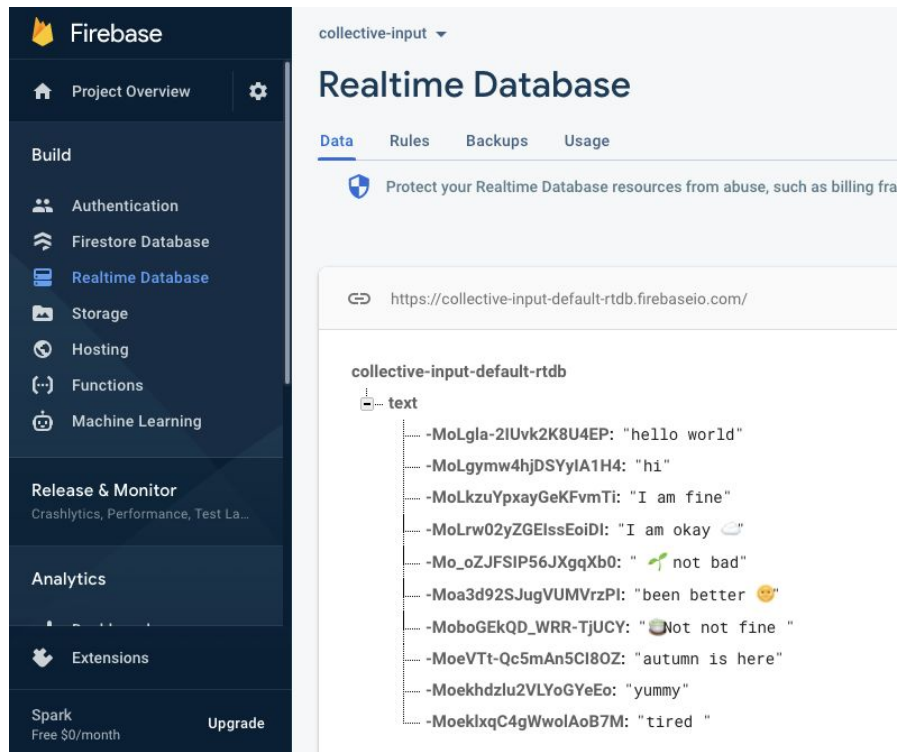
```
script.js > ...
1 // Import the functions you need from the SDKs you need
2 import { initializeApp } from "firebase/app";
3 // TODO: Add SDKs for Firebase products that you want to
4 // https://firebase.google.com/docs/web/setup#available-
5
6 // Your web app's Firebase configuration
7 const firebaseConfig = {
8   apiKey: "AIzaSyDJLptGH0IFRu06S2-90-VciqZcKcDFuik",
9   authDomain: "iml300-fall25.firebaseio.com",
10  projectId: "iml300-fall25",
11  storageBucket: "iml300-fall25.firebaseio.com",
12  messagingSenderId: "399849667098",
13  appId: "1:399849667098:web:81ff8c6a7af40d321cefa3",
14 };
15
16 // Initialize Firebase
17 const app = initializeApp(firebaseConfig);
18
19 const db = firebase.database();
20 let dbRef = db.ref("text");
21
22 // var data = {
23 //   name: "q",
24 //   word: "hello"
25 // }
26
27 // dbRef.push(data);
28
```



The screenshot displays the Firebase console interface. On the left, a sidebar menu lists services: Project Overview, Authentication, Firestore Database, Realtime Database (selected), Storage, Hosting, Functions, Machine Learning, Release & Monitor, Analytics, and Extensions. The main content area is titled 'Realtime Database' and shows the 'collective-input' database. A tree view on the right shows a 'text' node with a single entry: `-MpApY4FsJN8Ua-hSG2S` containing `name: "q"` and `word: "hello"`. The URL bar shows `https://collective-input-default-rtdb.firebaseio.com/`.

7.2 Check out the database

Add text input in your Glitch project and see the changes in Firebase Database
You can remove any of the entries from Firebase



The screenshot shows the Firebase console interface. On the left is a dark sidebar with navigation options: Project Overview, Build (Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning), Release & Monitor, Analytics, and Extensions. The main area is titled 'Realtime Database' and shows the 'Data' tab selected. A URL bar displays 'https://collective-input-default-rtdb.firebaseio.com/'. Below this, a tree view shows a node named 'collective-input-default-rtdb' containing a 'text' node. This 'text' node contains a list of 10 entries, each with a unique key and a string value.

Key	Value
-MoLgla-2IUvk2K8U4EP:	"hello world"
-MoLgymw4hjDSYlA1H4:	"h1"
-MoLkzuYpxayGeKFvmTi:	"I am fine"
-MoLrw02yZGEIssEoiDI:	"I am okay 🌀"
-Mo_oZJFSIP56JXgqXb0:	"🌱 not bad"
-Moa3d92SJugVUMVrzPI:	"been better 🍕"
-MoboGEkQD_WRR-TjUCY:	"🍷 Not not fine "
-MoeVTt-Qc5mAn5CI80Z:	"autumn is here"
-Moekhdzlu2VLYoGYeEo:	"yummy"
-MoeklxqC4gWwolAoB7M:	"tired "