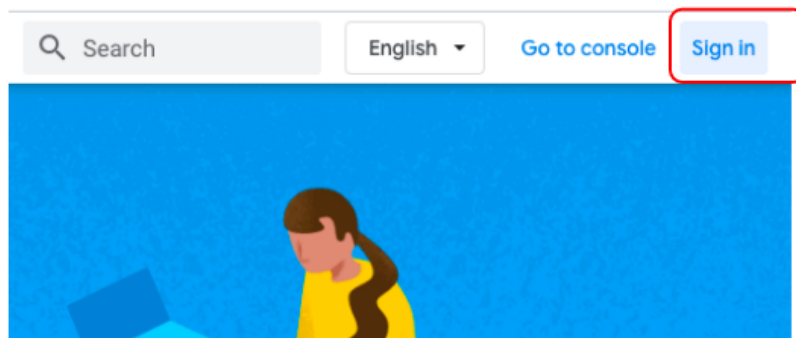


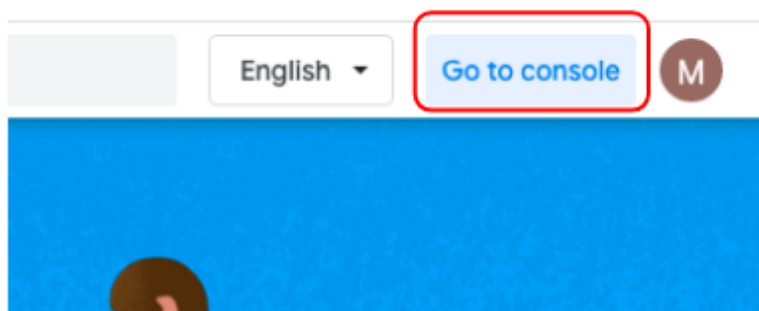
## Steps to create database

Open <https://firebase.google.com/> it is a firebase page.

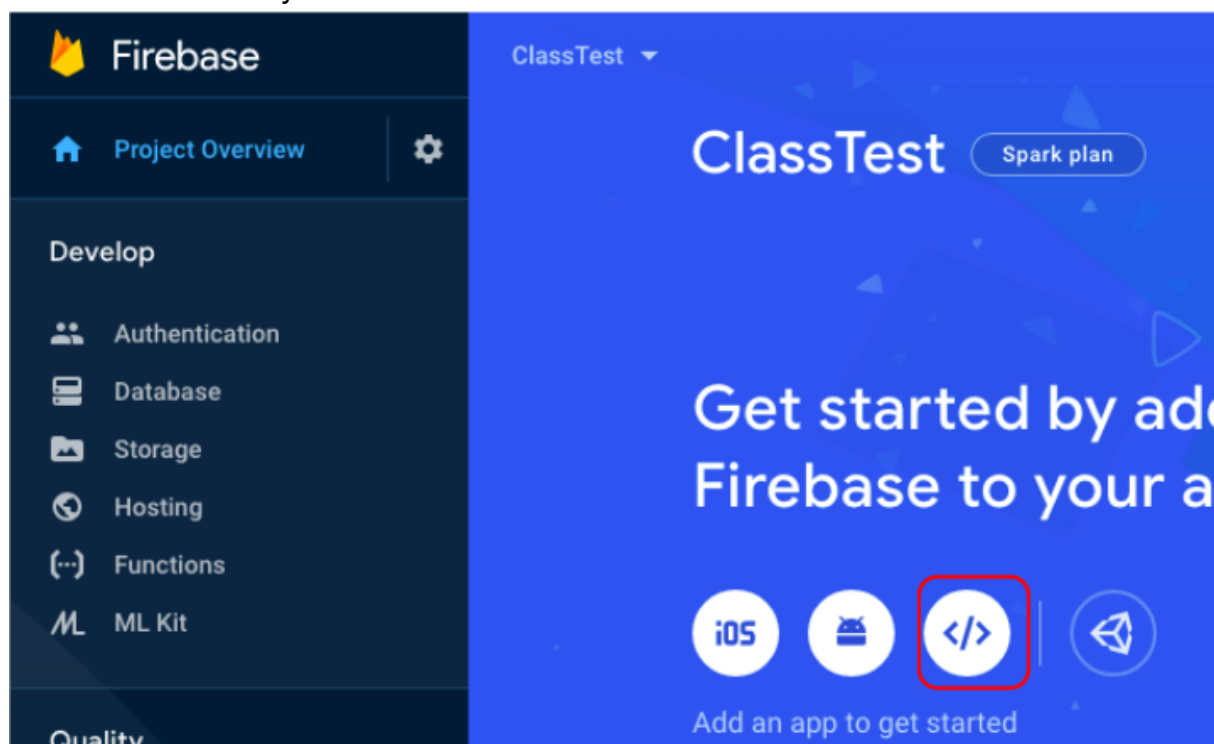
1. Login in your Gmail account if you haven't logged in.



2. Then after login click on **Go to console** on the top right corner.



3. So now select the project which you have created in the previous class, you will be redirected to the main page of your project there click on the web symbol.



4. Now give the name to the app, and press the **Register app** button.

1

Register app

App nickname ?

ClassTest

☐

Also set up **Firebase Hosting** for this app. [Learn more](#) ↗

Hosting can also be set up later. It's free to get started anytime.

Register app

2

Add Firebase SDK

5. Now it will generate firebase links. Once done, click on the “**Continue to Console**” button.

✓ Register 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
```



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";
// 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: "AIzaSyBfUjRkGPWpjrqERGDfRFJaXWWKC_QzybI",
  authDomain: "kwitternew-faedb.firebaseio.com",
  projectId: "kwitternew-faedb",
  storageBucket: "kwitternew-faedb.appspot.com",
  messagingSenderId: "523332773835",
  appId: "1:523332773835:web:f47e467904295d3807b84e"
};

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

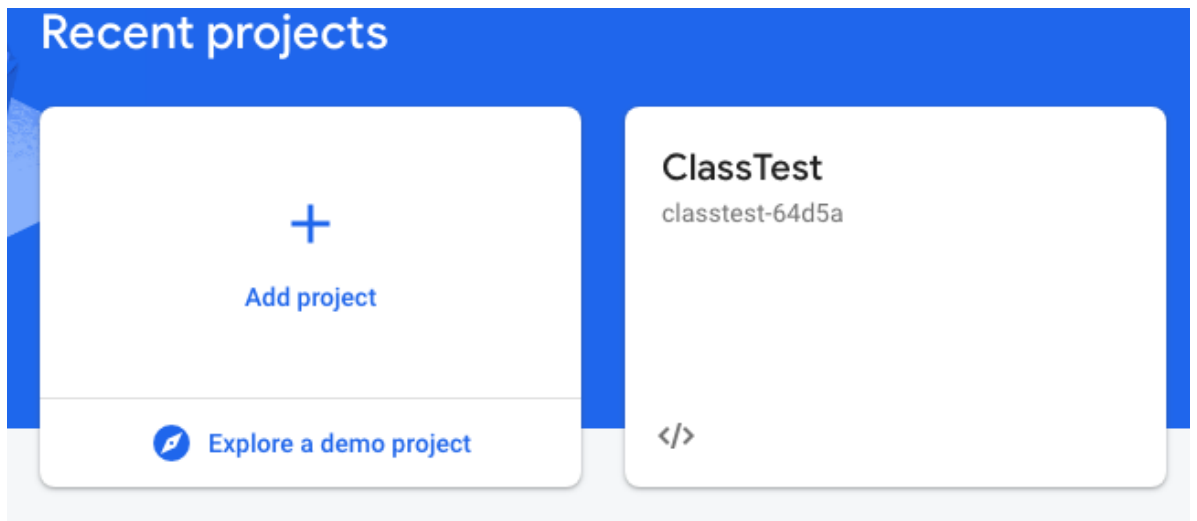


**Note:** This option uses the [modular JavaScript SDK](#), which provides reduced SDK size.

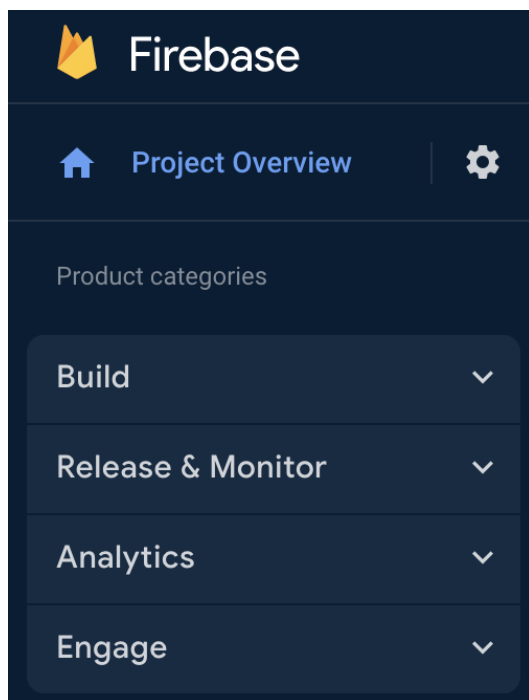
Learn more about Firebase for web: [Get Started](#), [Web SDK API Reference](#), [Samples](#)

Continue to console

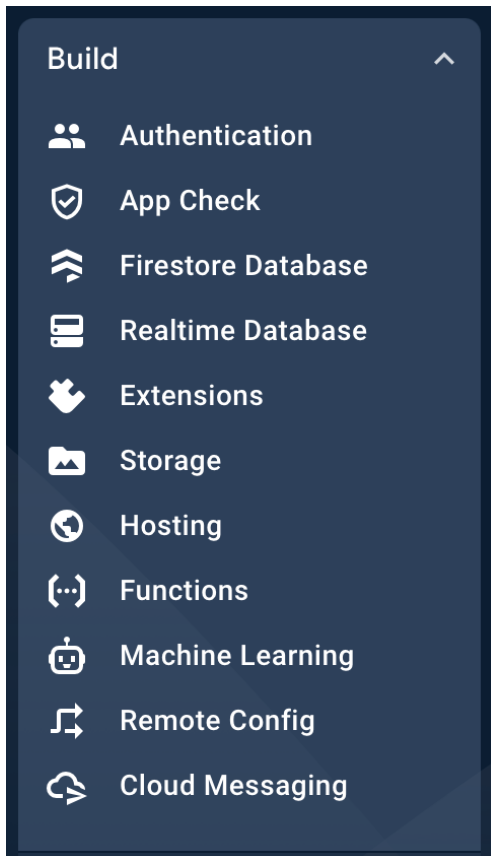
6. On the console select the project name which you have created in the previous class.



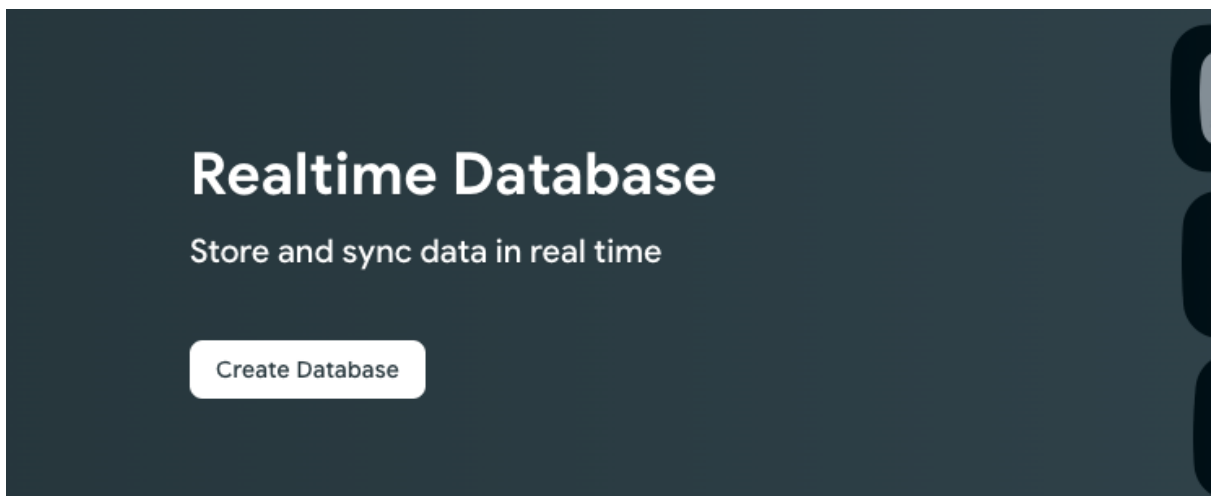
7. Then on left hand side click on **Build**:



8. Inside the “**Build**” product category click on **Realtime Database**:



9. Then click on **Create Database**:



10. Then select the location of your Realtime Database as “**United States**” and click on the “**Next**” button.

**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**

11. Then select test mode and press the **enable** button.

**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.

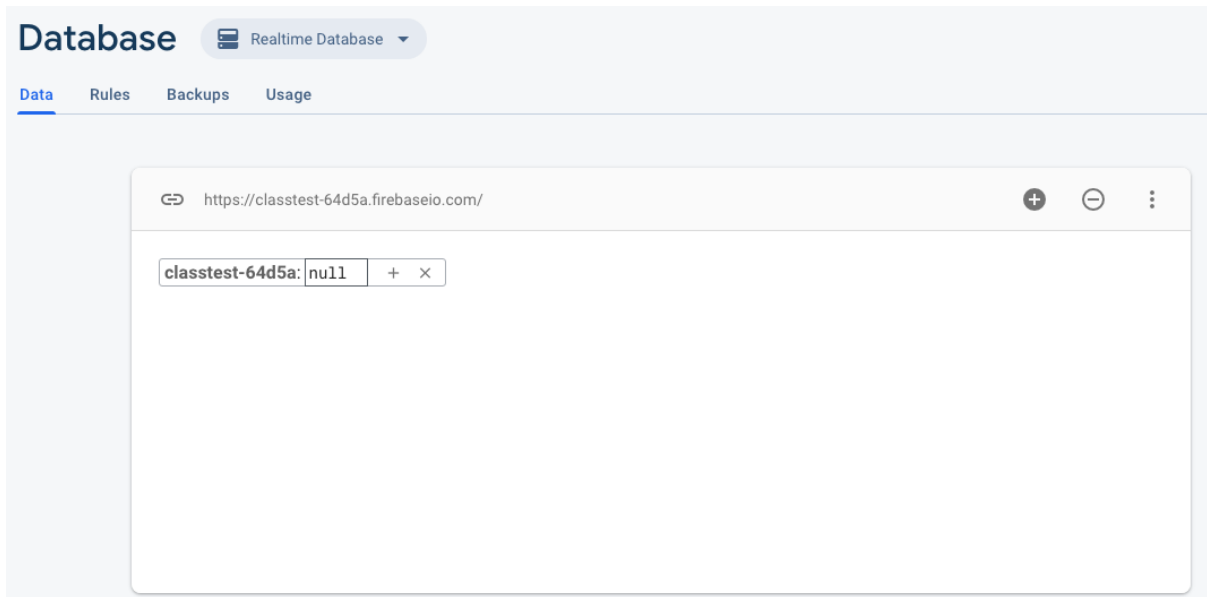
☒ **Start in test mode**  
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
{
  "rules": {
    ".read": "now < 1660069800000", // 2022-8-10
    ".write": "now < 1660069800000", // 2022-8-10
  }
}
```

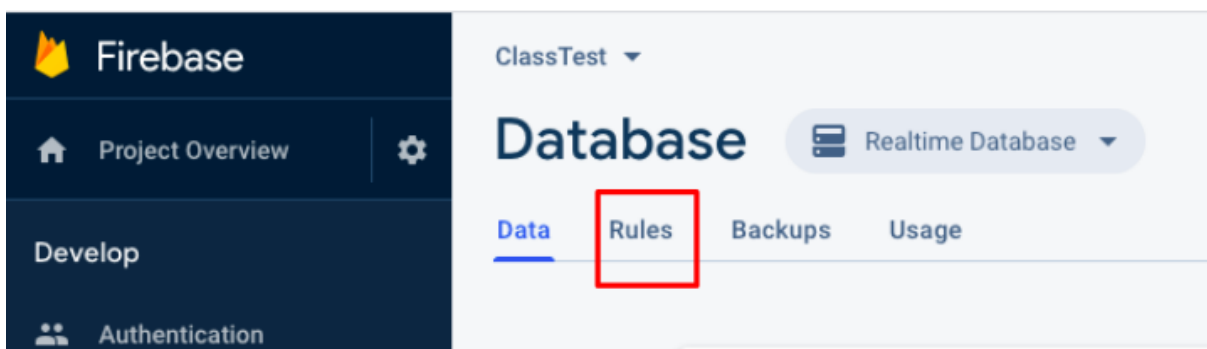
**! The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days**

Cancel **Enable**

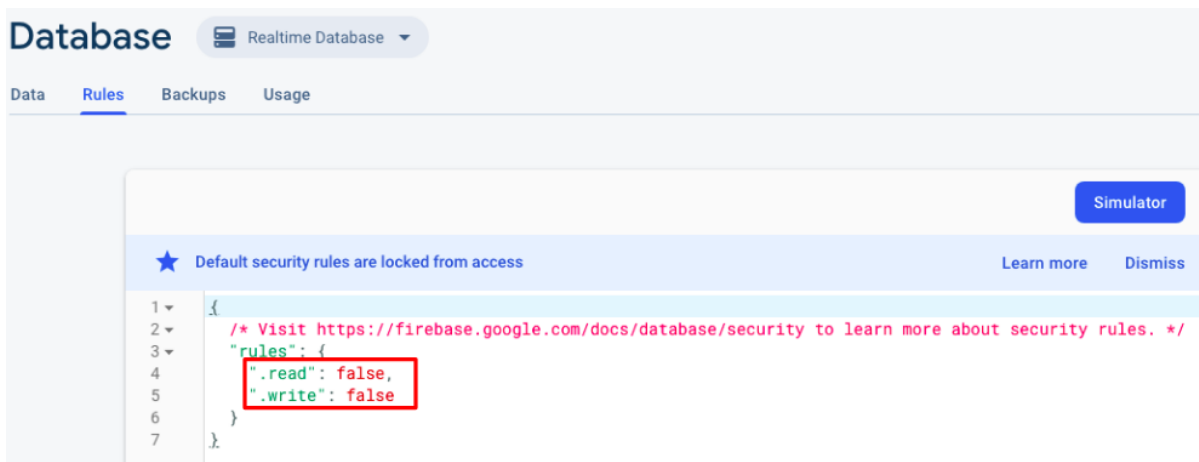
Great, we have made our database:



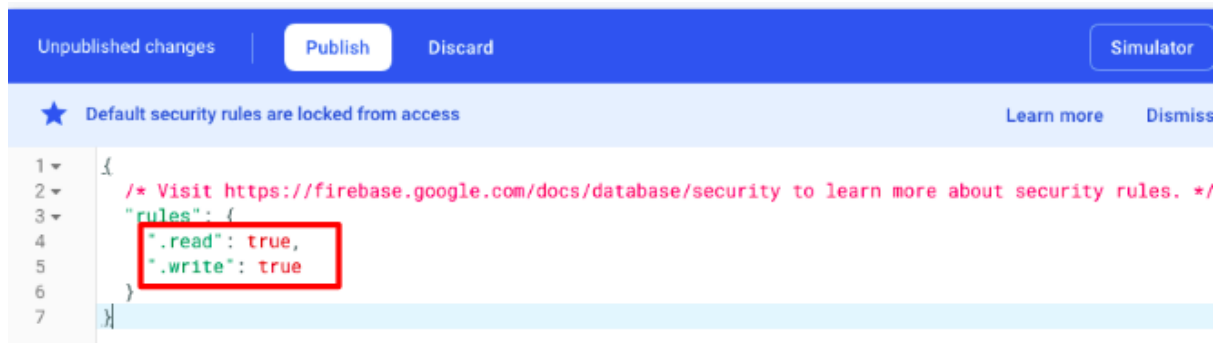
12. Now we want to read and write data in firebase, so just check the rules of reading and writing data in firebase, for that rule. Click **Rules**:



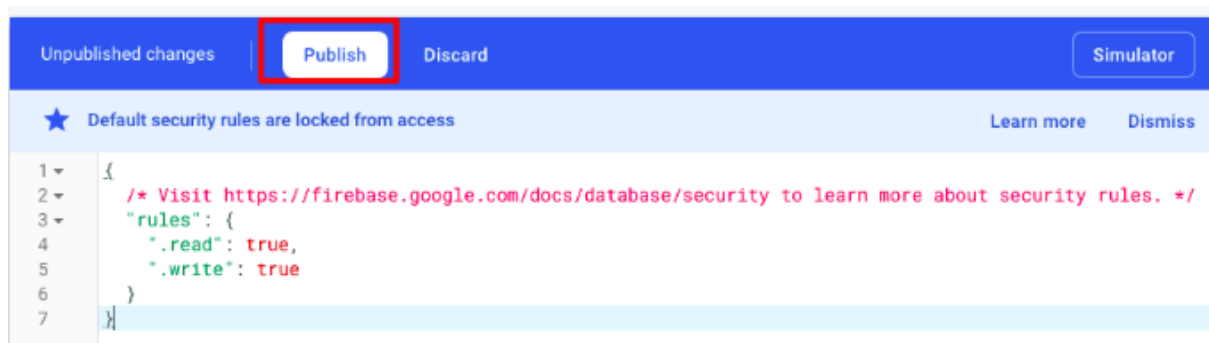
- If the rules are **false** like this:



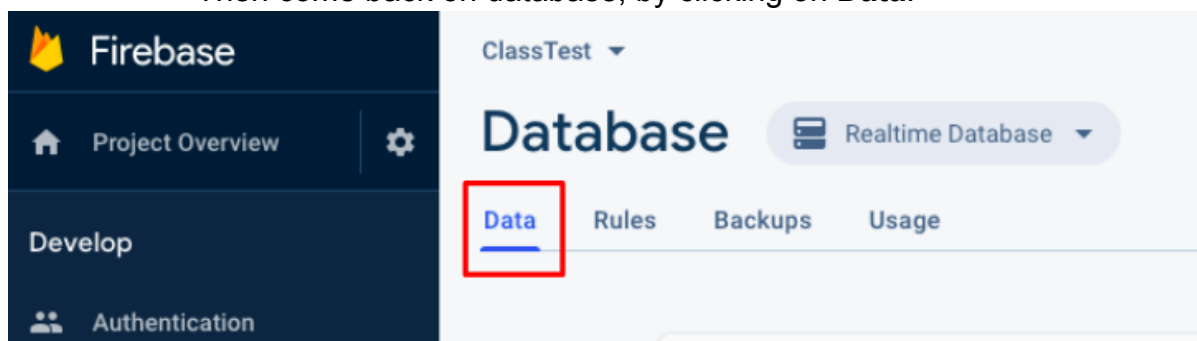
- Then double click on the **false**, and change it to **true**, like this:



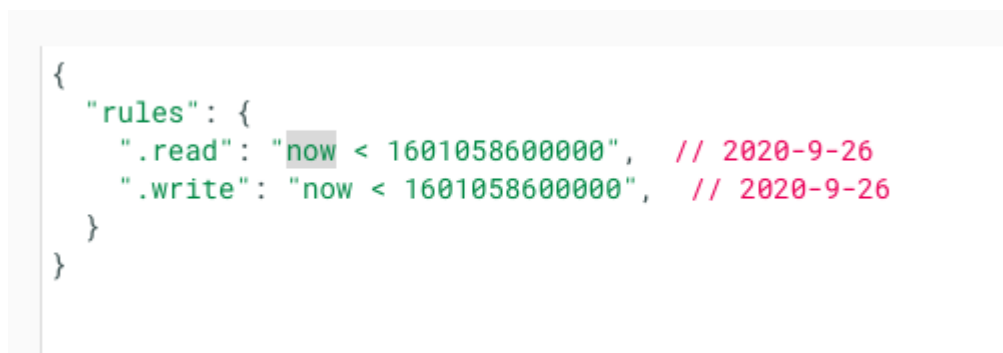
- Then click on **Publish**:



- Then come back on database, by clicking on **Data**:

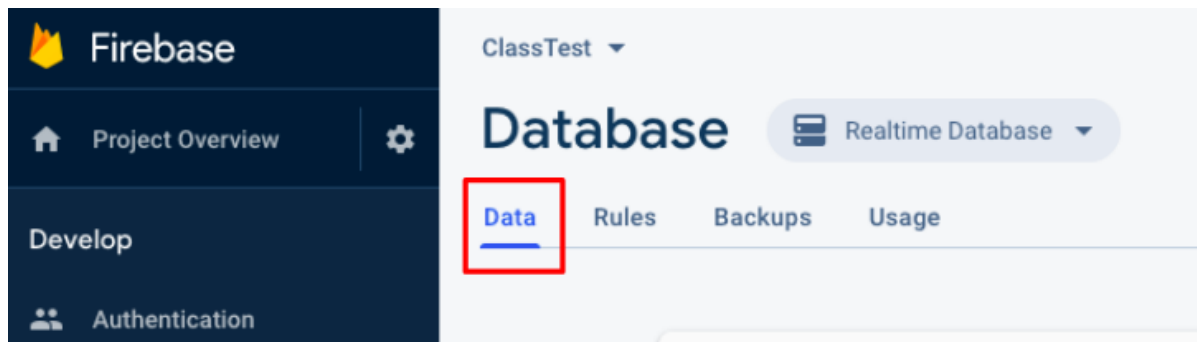


- If the rules are **not false** like this:



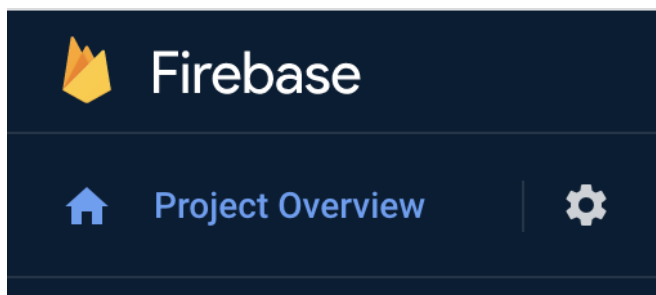
having **now** keyword and some number, then let it be and come back on database, by clicking on **Data**:



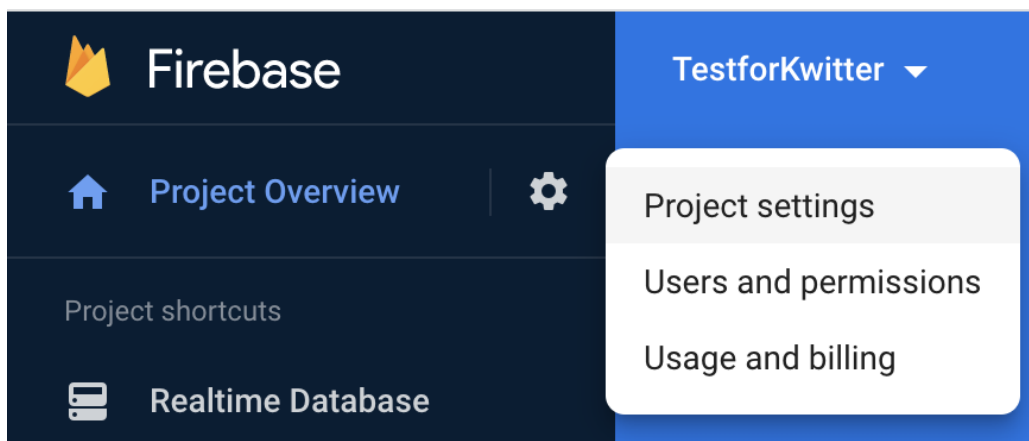


Now to get the updated database link with databaseURL to perform the following steps.

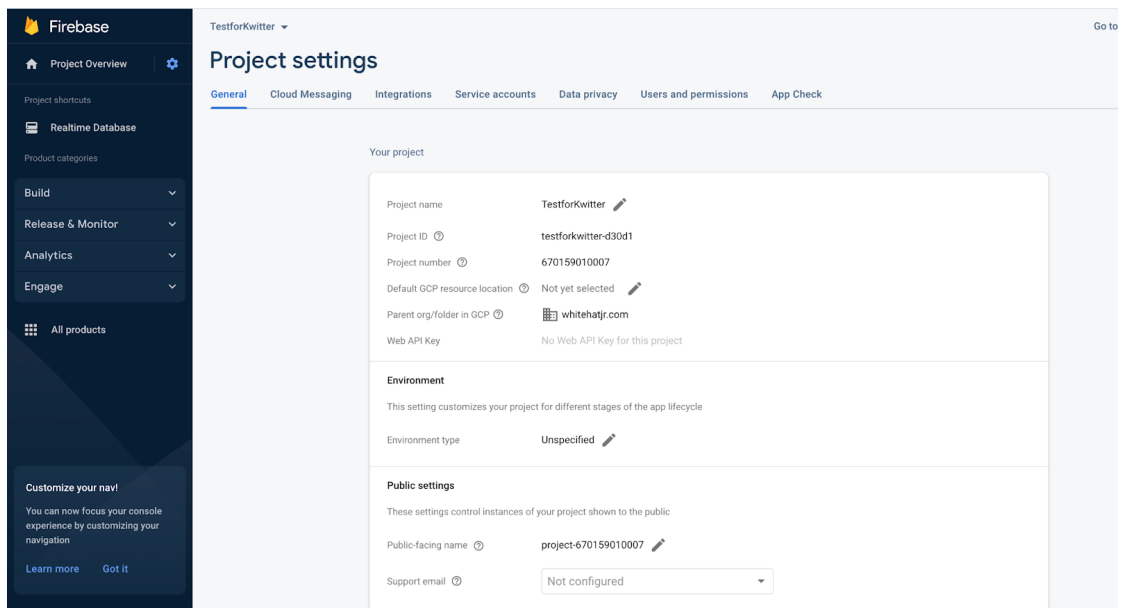
1. On the left slide of the console click on the Gear Icon of the **Project Overview** tab.



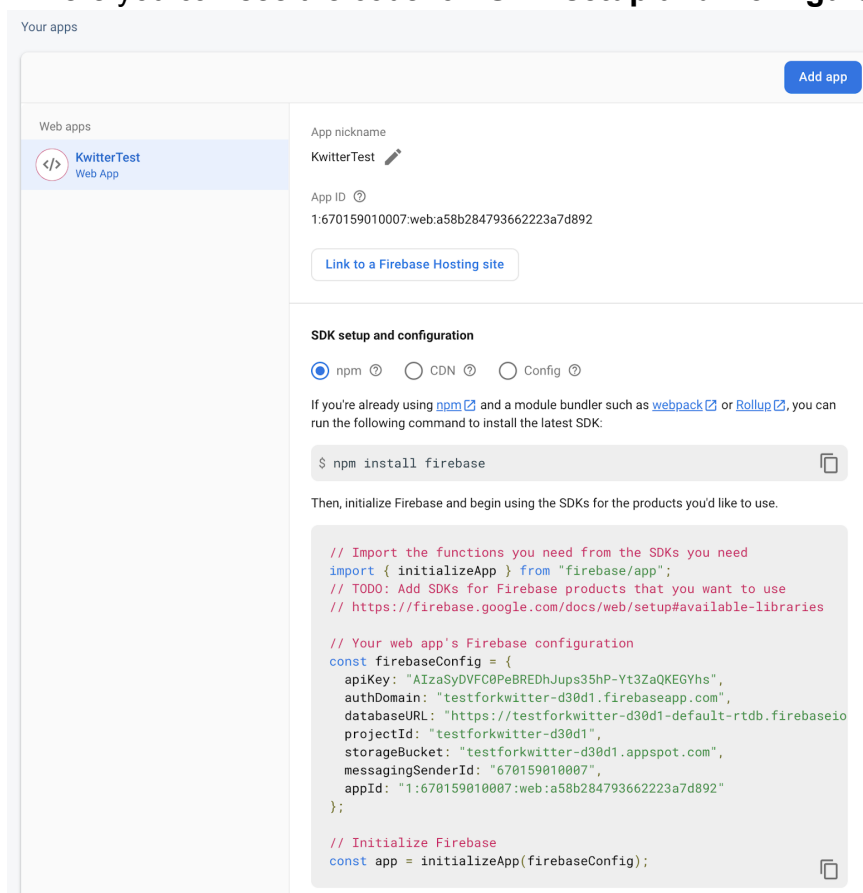
2. Then click on the “**Project Setting**”



3. On the Project setting page, scroll down to “**You apps**”



4. Here you can see the code for “**SDK setup and Configuration**”



5. You need to copy only the marked part of the firebase links and paste it into `twitter_room.js` file at the starting of the file like the below image.

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";
// 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: "AIzaSyDVFC0PeBREDhJups35hP-Yt3ZaQKEGYhs",
  authDomain: "testforkwitter-d30d1.firebaseio.com",
  databaseURL: "https://testforkwitter-d30d1-default-rtdb.firebaseio.com",
  projectId: "testforkwitter-d30d1",
  storageBucket: "testforkwitter-d30d1.appspot.com",
  messagingSenderId: "670159010007",
  appId: "1:670159010007:web:a58b284793662223a7d892"
};

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

Your code will look like this in the **kwitter\_room.js** file, it is present in the kwitter folder, this folder has been downloaded by the student.

```
// Your web app's Firebase configuration
var firebaseConfig = {
  apiKey: "AIzaSyBFnaGQeR0xOrIDmSTDucLsrFvhupMRE60",
  authDomain: "classtest-64d5a.firebaseio.com",
  databaseURL: "https://classtest-64d5a.firebaseio.com",
  projectId: "classtest-64d5a",
  storageBucket: "classtest-64d5a.appspot.com",
  messagingSenderId: "33276442296",
  appId: "1:33276442296:web:85f5e95e9a2a53e515264d"
};
// Initialize Firebase
firebase.initializeApp(firebaseConfig);

function getData() {
  firebase.database().ref("/").on('value', function(snapshot) {
    document.getElementById("output").innerHTML +=
      Group_name = childKey;
    //Start code
    //End code
  });
}
getData();
```

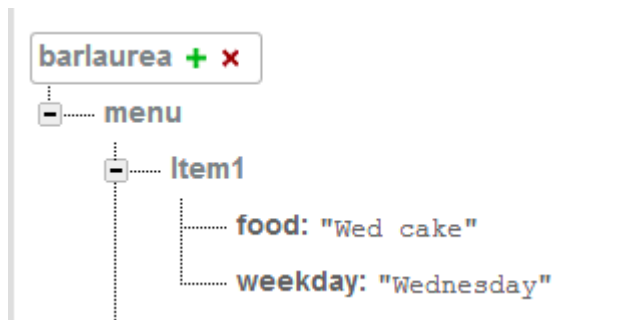
Like this

**Don't touch** his part of the code, This part of the code is already given in firebase guide book, we will discuss this in next class

**kwitter\_room.js**

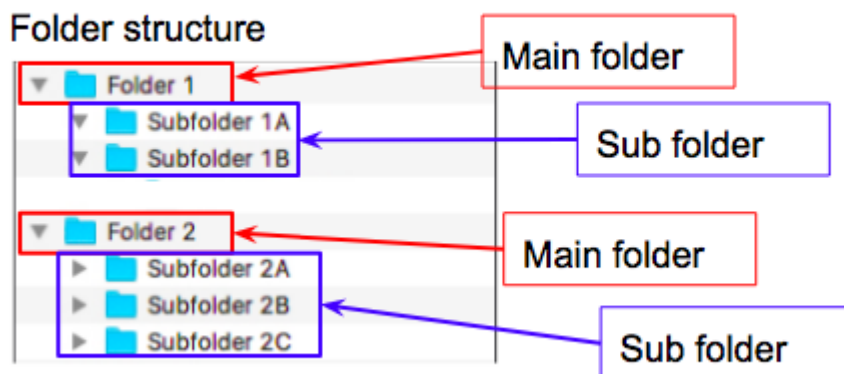
## Explaining firebase database structure

Example of firebase database structure.

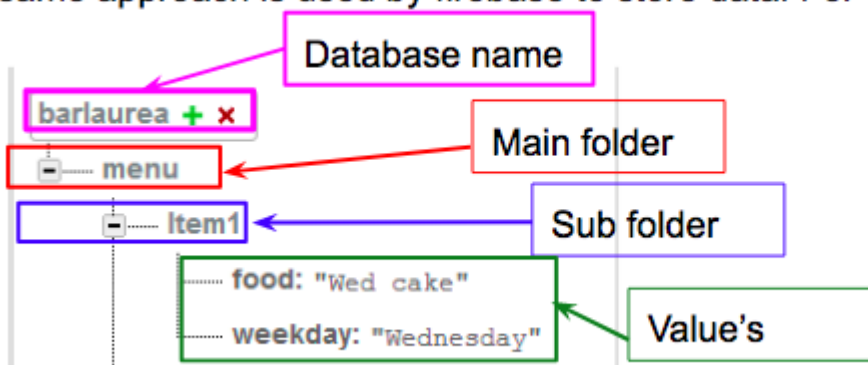


The explanation of the above data structure is given in the below image-

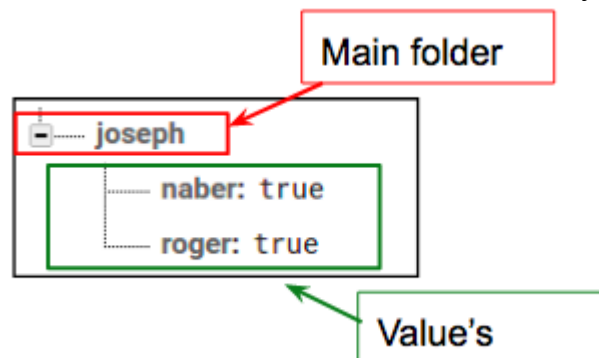
**The structure of firebase base database is same like the folder structure**



So there are folder and inside this folder there are sub folders, this same approach is used by firebase to store data. For eg -

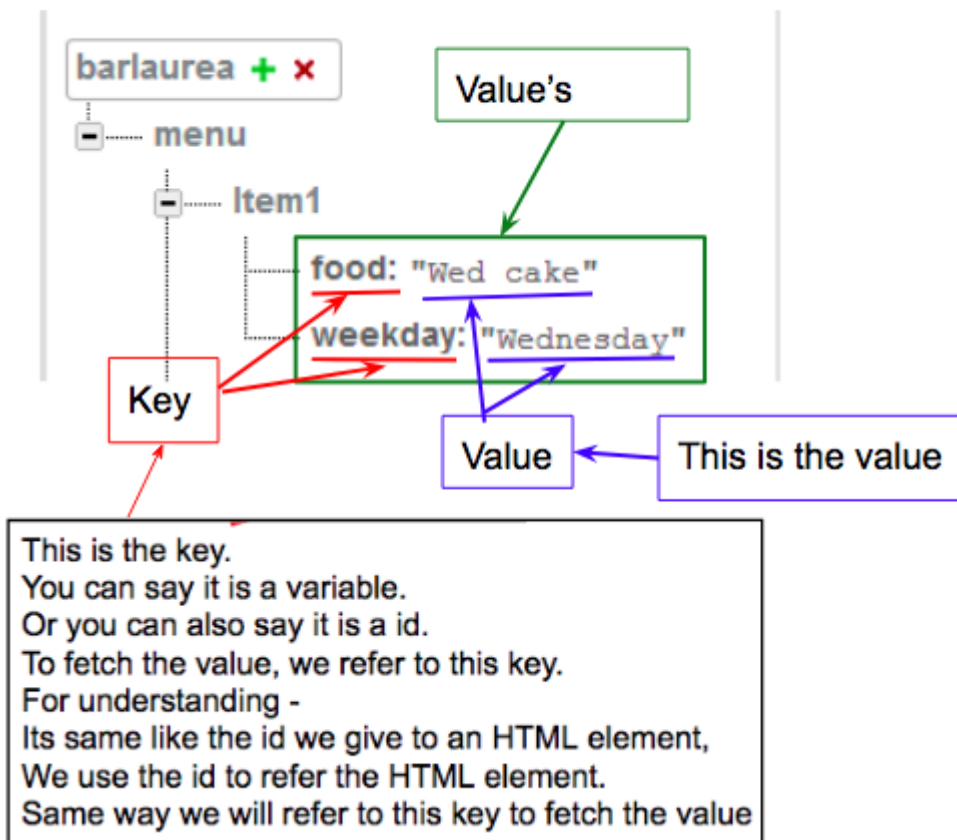


It is not compulsory that the **main folder** should have a **sub folder** and then the **sub folder** have the values like the above image. The main folder can have the values directly, like this -



Explaining the values

## Explaining the values



The following code is already given to the student in the student-activity  
**practice.html**

```

<html>
<head>
  <title>Practice</title>

  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

  <script src="https://www.gstatic.com/firebasejs/7.6.2/firebase-app.js"></script>
  <script src="https://www.gstatic.com/firebasejs/7.6.2/firebase-firestore.js"></script>
  <script src="https://www.gstatic.com/firebasejs/live/3.1/firebase.js"></script>

  <script src="practice.js"></script>
</head>
<body>
<div class="container">

  <h1>Practice Activity</h1>

  <div class="form-group">
    <label>User Name:</label>
  </div>

  <button class="btn btn-success">Add User Name</button>

</div>

</body>
</html>

```

**Bootstrap Links**

**Firebase Links**

**Our JS file link**

**Complete code for practice.html**

```

<html>
<head>
  <title>Practice</title>

  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

  <script src="https://www.gstatic.com/firebasejs/7.6.2/firebase-app.js"></script>
  <script src="https://www.gstatic.com/firebasejs/7.6.2/firebase-firestore.js"></script>
  <script src="https://www.gstatic.com/firebasejs/live/3.1/firebase.js"></script>

  <script src="practice.js"></script>
</head>
<body>
<div class="container">

  <h1>Practice Activity</h1>

  <div class="form-group">
    <label>User Name:</label>
    <input type="text" id="user_name" class="form-control" placeholder="User Name">
  </div>

  <button class="btn btn-success" onclick="addUser()">Add User Name</button>

</div>

</body>
</html>

```

Complete the code for practice.js

```

// PASTE YOUR FIREBASE LINK HERE

function addUser()
{
  user_name = document.getElementById("user_name").value;
  firebase.database().ref("/").child(user_name).update({
    purpose : "adding user"
  });
}

```