SOCIAL WEBSITE

What is our GOAL for this MODULE?

We created a firebase database. We understood the structure of the database. We also learned how to add data to the database.

What did we ACHIEVE in the class TODAY?

- Created a database.
- Added data to our firebase database.

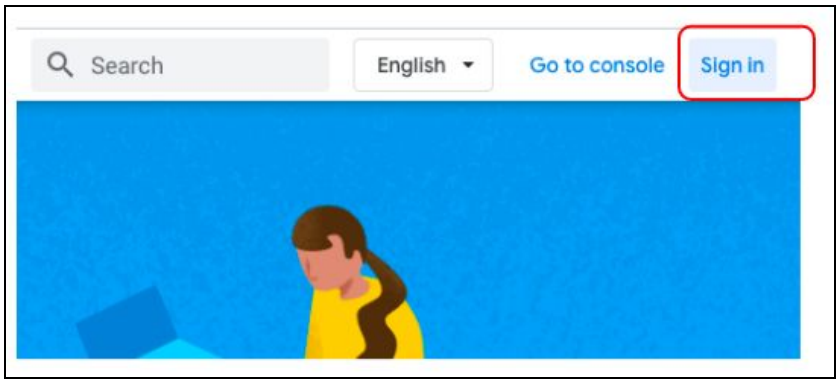
Which CONCEPTS/ CODING did we cover today?

- Completed the HTML code for the practice activity.
- Understood the structure of the database.
- Learned how to add data to the database.

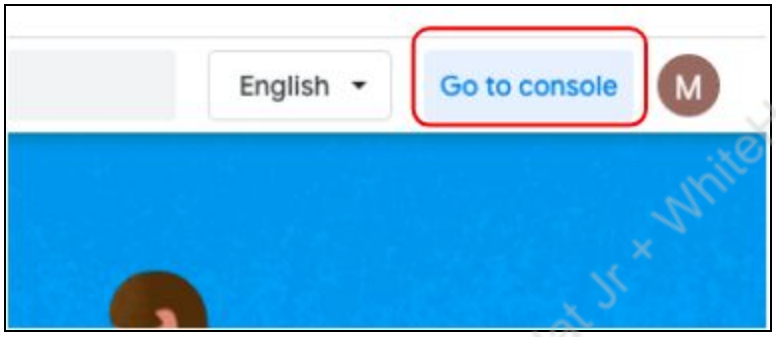
How did we DO the activities?

Steps to create a database.

1. Open https://firebase.google.com and login in your gmail account if you haven't logged in.

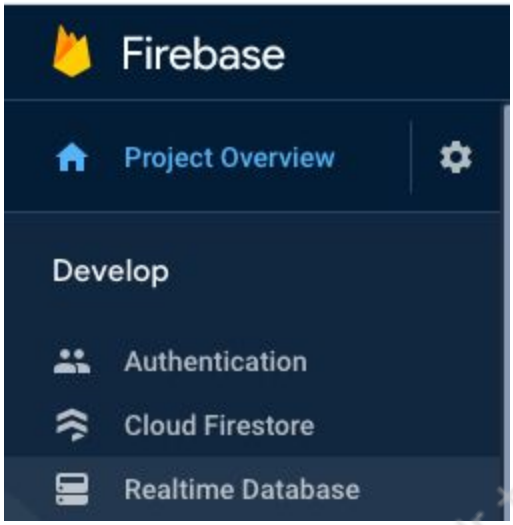


2. Then after login click on **Go to console** on the top right corner.

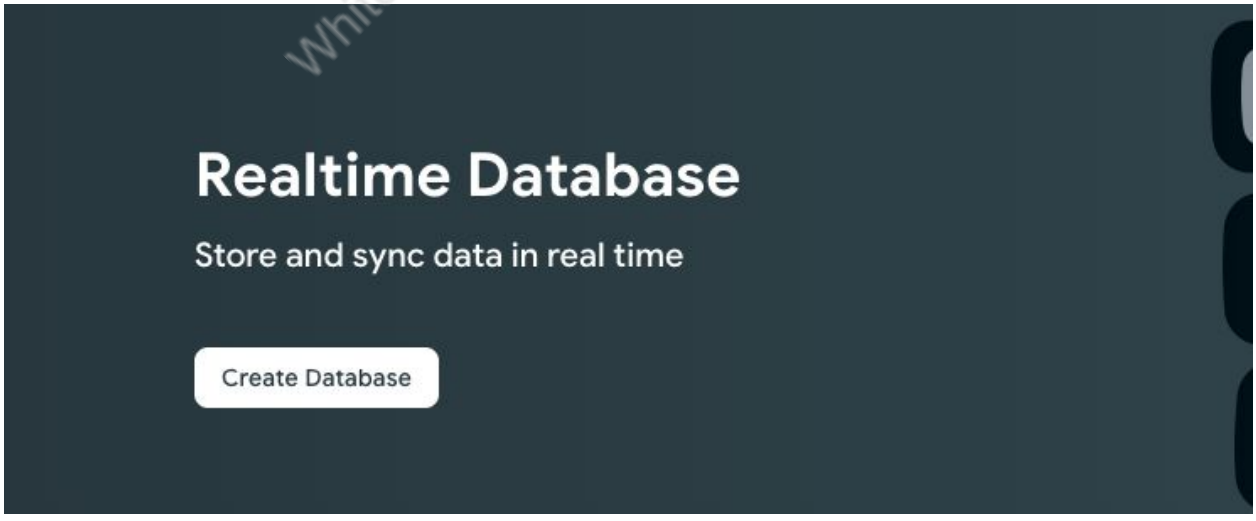


3. Then select the project which you created in the last class.

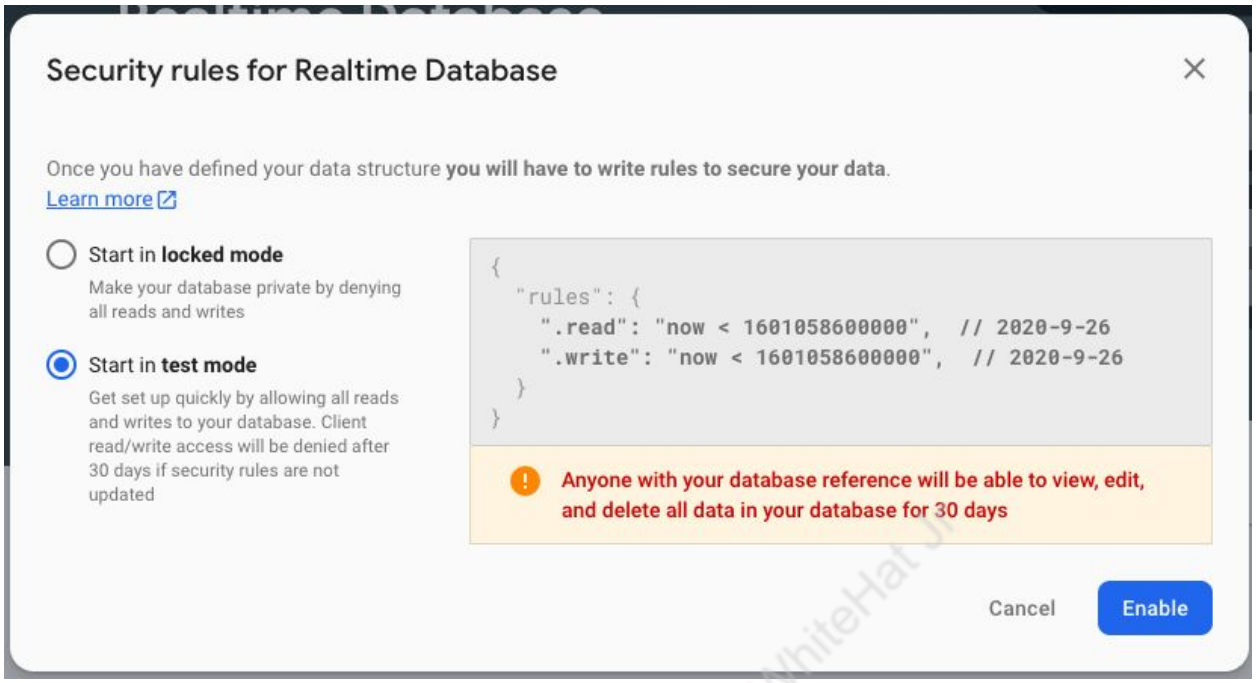
4. Then on left hand side click on **Realtime Database**:



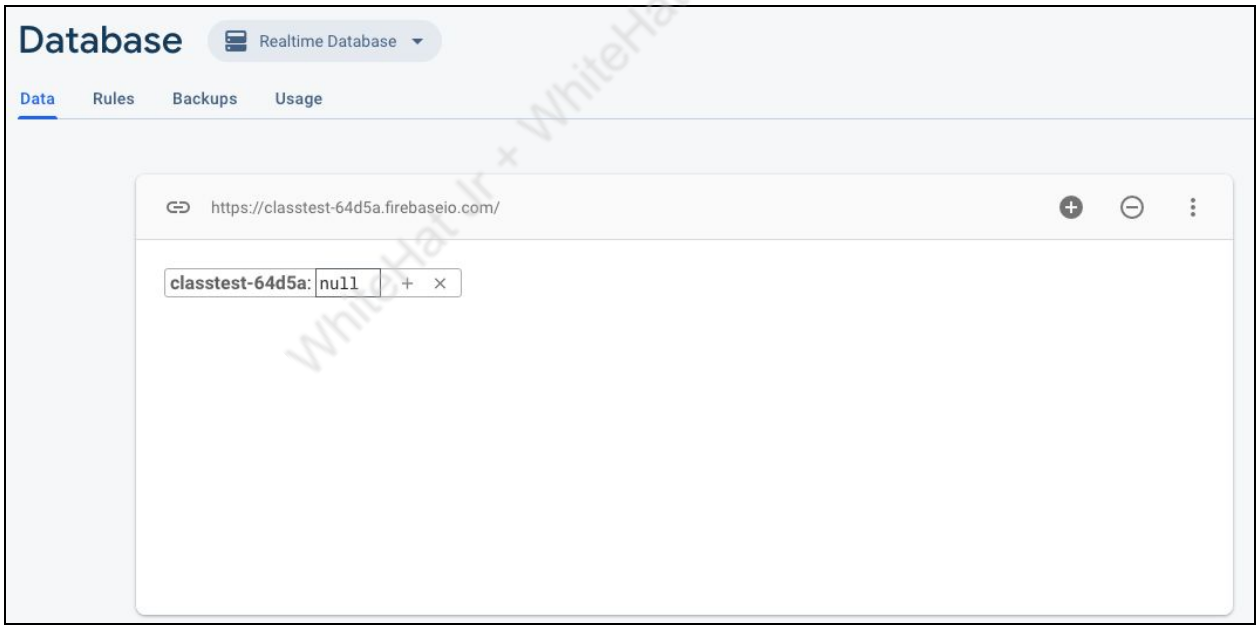
5. Then click on **Create Database**:



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

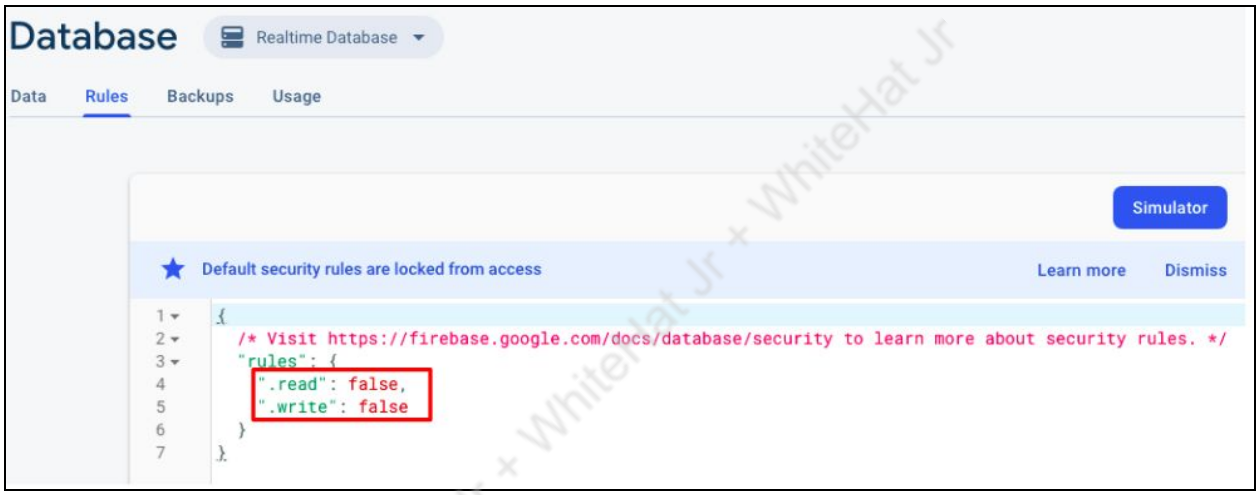


Great, we have made our database:

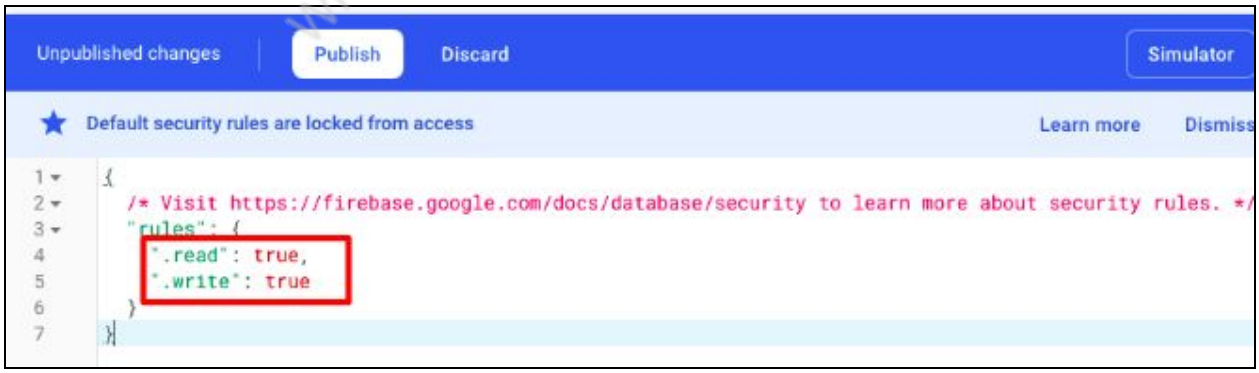


7. Now we want to read and write data in firebase, so just check the rules of reading and writing data in firebase, for that rule. Click **Rules**:

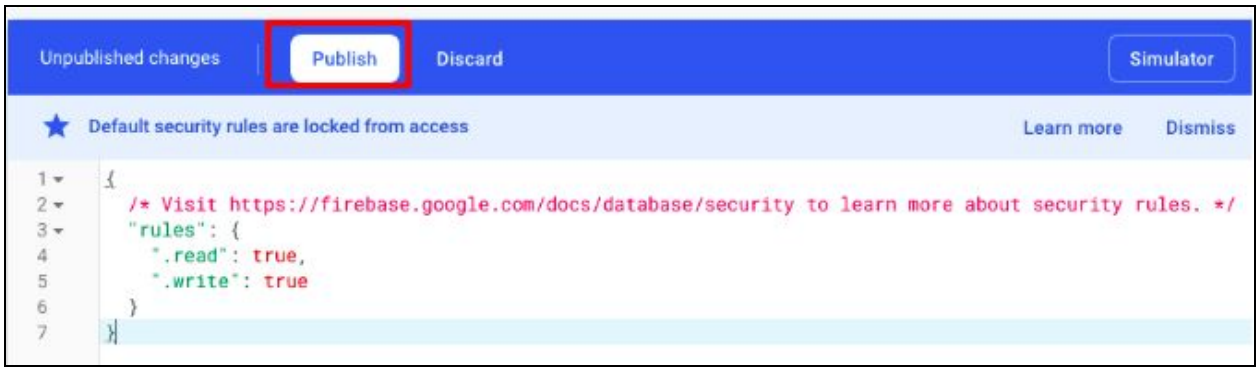
- If the rules are **false** like this:



  - 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**:

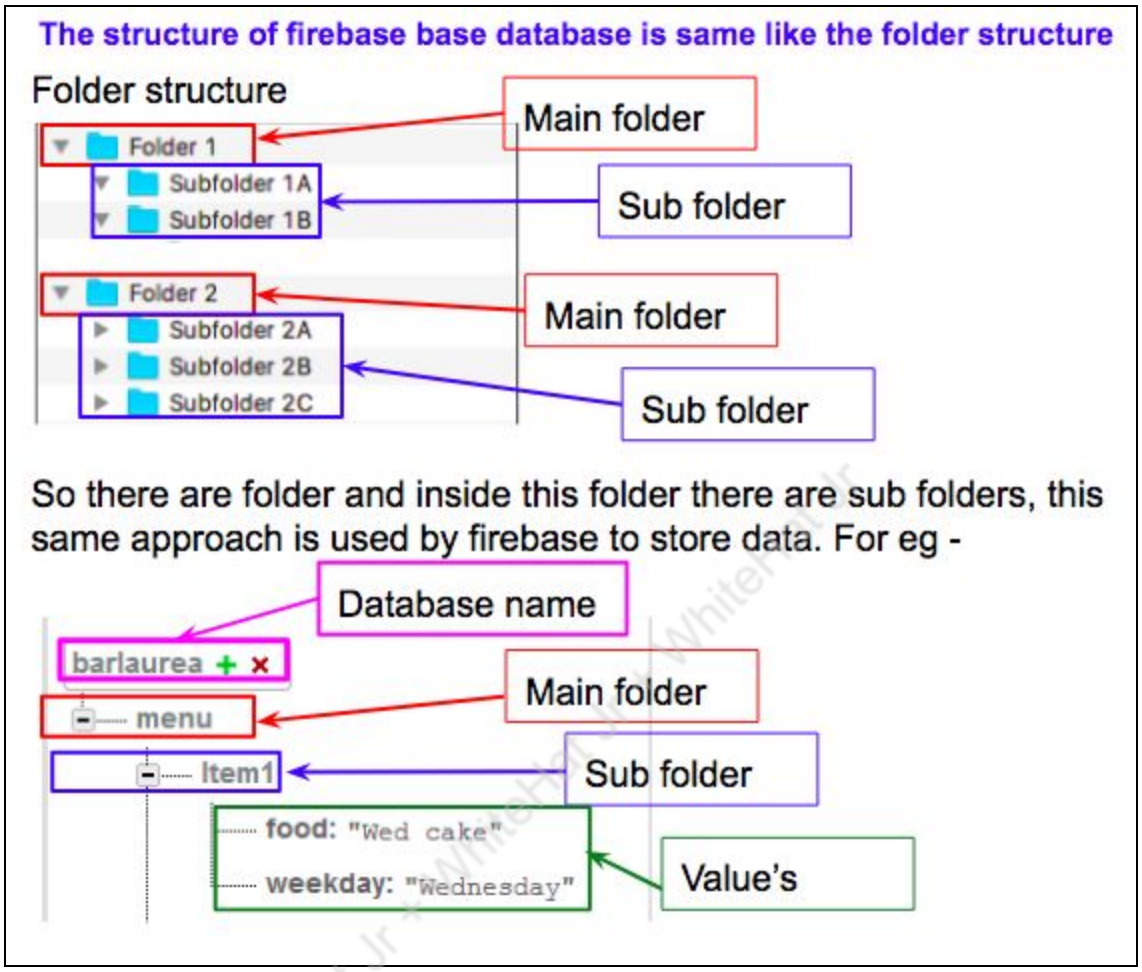
- If the rules are **not false** like this:

```
{
  "rules": {
    ".read": "now < 1601058600000",   // 2020-9-26
    ".write": "now < 1601058600000",   // 2020-9-26
  }
}
```

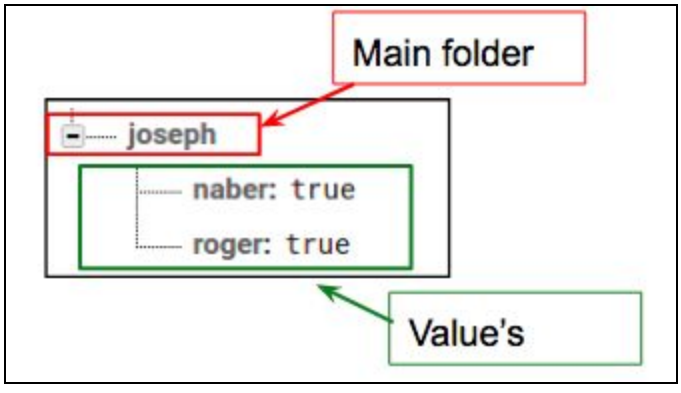
having **now** keyword and some number, then let it be and come back on database, by clicking on **Data:**

Now understand the structure of how data is stored in the firebase.
Example of firebase database structure:

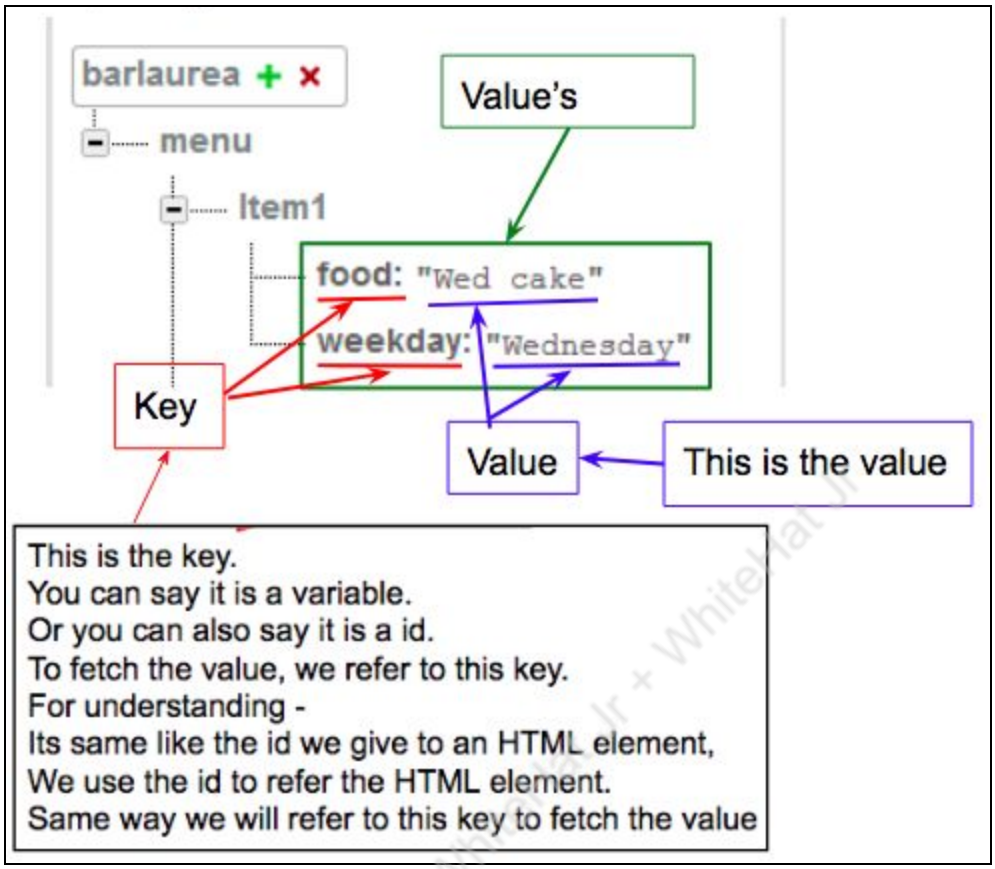
The explanation for the preceding data structure is given in the following diagram:



- It is not compulsory that the **main folder** should have a **sub folder** and the **sub folder** should have the values like in the preceding diagram.
- The main folder can have the values directly, like this:



Explaining the values:



You have already downloaded **C94_practice** folder in today's class.
This **C94_practice** folder has:

- **practice.html** - It has some pre HTML code.
- **practice.js** - It is a blank JS file.

The following code is already given to you.

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>
```

The HTML code has:

- **Bootstrap** links
- **Firebase** links
- **JS** file link
- And some HTML code which you already know.

1. Now add some HTML elements like, **input** element and **onclick** functions.

```
<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">
```

Input box code explanation:

- **id** - we have given id to identify this input box
- **class="form-control"** - this is a bootstrap class, that adds padding, margin and border to the input box.
- **placeholder** - It is like a hint to the input box.
- Output:

2. Add the onclick **addUser()** function to the add user button. This **addUser()** function will add the user name in the firebase database.

```
<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>
```

Button code explanation:

- **id** - we have given id to identify this button
- **btn** - this bootstrap class will add the padding and remove the default border and color of the button.
- **btn-success** - this bootstrap class will add the light green border and the background color to the **onclick**="**addUser()**" button.

Now write the JS code in **practice.js** file, which has been downloaded.

You are required to copy and paste the firebase links in **practice.js,** the same way you did in **kwitter_room.js**.

1. Copy the links from there and paste it in **practice.js** file. Like this:

```
// Your web app's Firebase configuration
var firebaseConfig = {
  apiKey: "AIzaSyBFnaGQeR0xOrIDm5TDucLsrFvhupMRE60",
  authDomain: "classtest-64d5a.firebaseapp.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);
```

2. Now code the **addUser()** function:

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

Explaining the code:

- Define the function.

```
function addUser()
```

- Take the user name from the input box, using the input box id. And store it inside a variable.

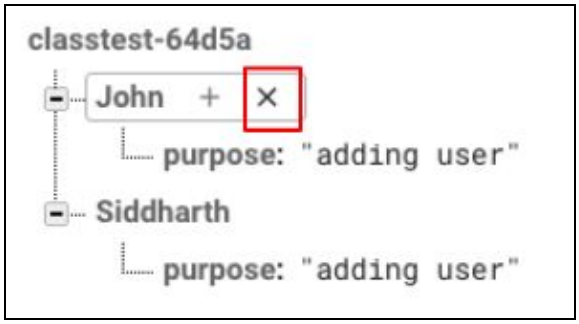
```
function addUser()
{
  user_name = document.getElementById("user_name").value;
```

- Now add this value to the firebase:

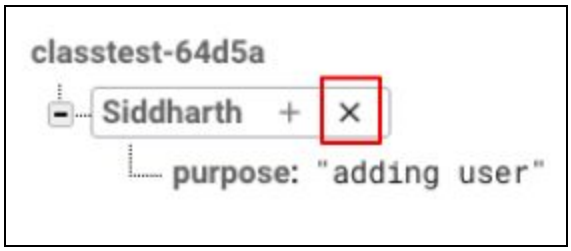
```
firebase.database().ref("/").child(user_name).update({
  purpose : "adding user"
});
```

- **firebase** - is used to set the reference for adding data to the database.
- **database()** - means we want to add data to the database.
- **ref("/")** - **ref()** means the reference, where we want to add a user name in the database.
- **"/"** - this means we want to add the user name in the root as the main folder.
- **child(user_name)** - **child()** function is used to give the name to the main folder. **user_name** is the name of the main folder.
- **update** - is the firebase function used to update the database with the values.
  - The main folder, **user_name**, alone can't be added, so we need to add some values.
- **purpose** is the key word for the value "**adding user**".
  - You can give any value you want. As the purpose of this activity was to add users that's why we have given the value as "**adding user**".

After completing the practice activity, delete all the data which has been added in their firebase database. To delete the data, hover over the main folder and click on the "**x**" button like this:



Then the other one:



Do this for all the **user_name** until no **user_name** is left in the database. We are doing this because we will use the same database for Kwitter.

What's NEXT?

We will learn about a function that will get all the room names from firebase. We will also learn to write a function that redirects to rooms.

EXTEND YOUR KNOWLEDGE

Here are some Best References we've compiled together to enhance your knowledge and understanding of the concepts we learned today in the class. This will help you become a pro at coding and creating industry-grade tech products!

Short Videos: Watch these Short Videos to understand the application of the concepts learned in class in real-world applications.

1. Firebase cloud functions: https://www.youtube.com/watch?v=29BNaJiqWB4
2. Firebase Realtime Database on the Web: https://www.youtube.com/watch?v=noB98K6A0TY
3. JavaScript LocalStorage Tutorial: https://www.youtube.com/watch?v=cELVC0LMC5w

Coding Playground: Try out these code examples to get more practice in making Websites and Playstore ready apps.

1. 

: https://firebase.google.com/docs/database/web/structure-data

2. 

: https://firebase.google.com/docs/database/web/read-and-write