SOCIAL WEBSITE

What is our GOAL for this MODULE?

We learned to create a function to add the user name in the local storage. We also learned to create a function to add a room in the local storage.

What did we ACHIEVE in the class TODAY?

- Wrote the HTML code for the kwitter login page, and kwitter room page.
- Wrote JS code for twitter login page.
- Created a firebase project.

Which CONCEPTS/ CODING did we cover today?

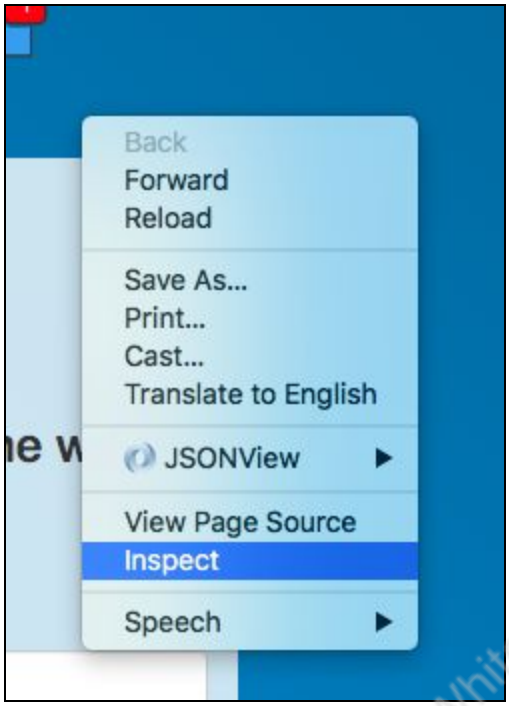
- Completed kwitter login page by adding some HTML elements, bootstrap classes and onclick **adduser()** function to the button.
- Completed the kwitter room page by adding some HTML elements, bootstrap classes, onclick **logout()** function, and onclick **addRoom()** function to the buttons.

How did we DO the activities?

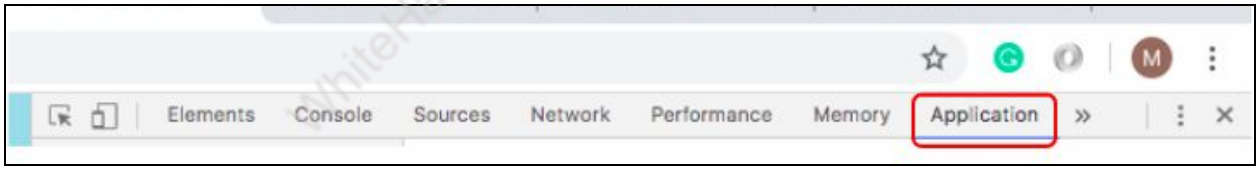
Steps to view local storage:

1. Open https://jynyhy9vu5r8xz5zy0utww-on.drv.tw/www.Kwitter.com/kwitter.html.

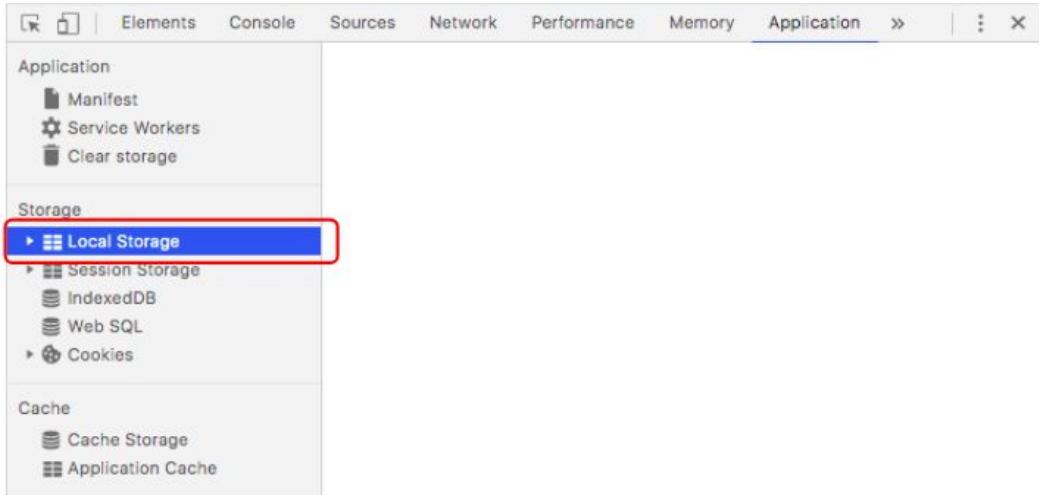
2. Right click on the browser and select Inspect.



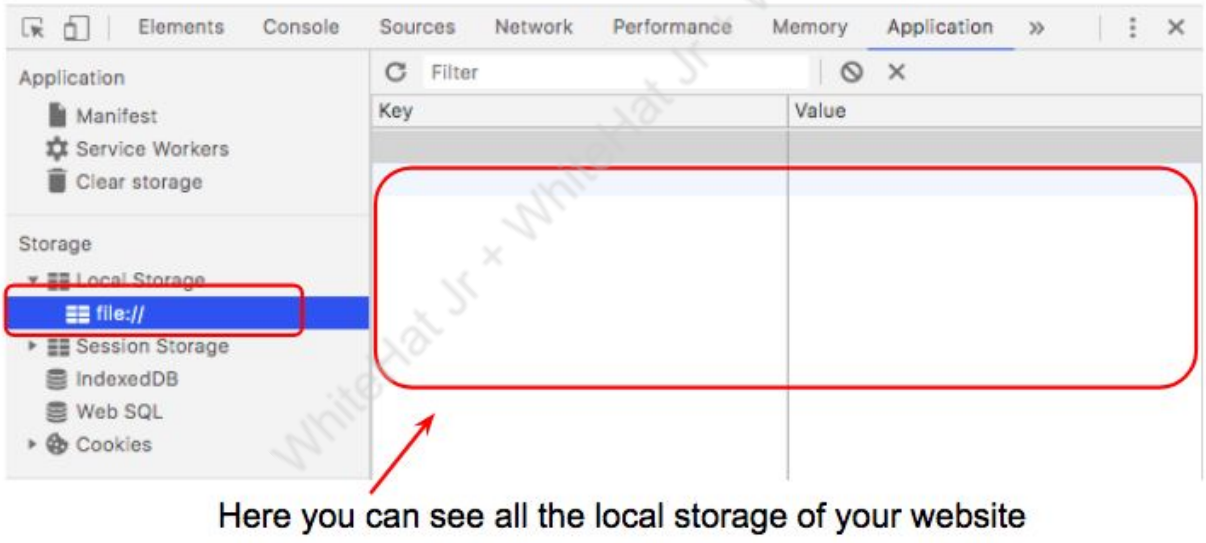
3. And click on the application tab.



4. Then double click on the local storage.



5. Then click on the file, and then on the right hand side there will be a window where you can see all your localstorage.

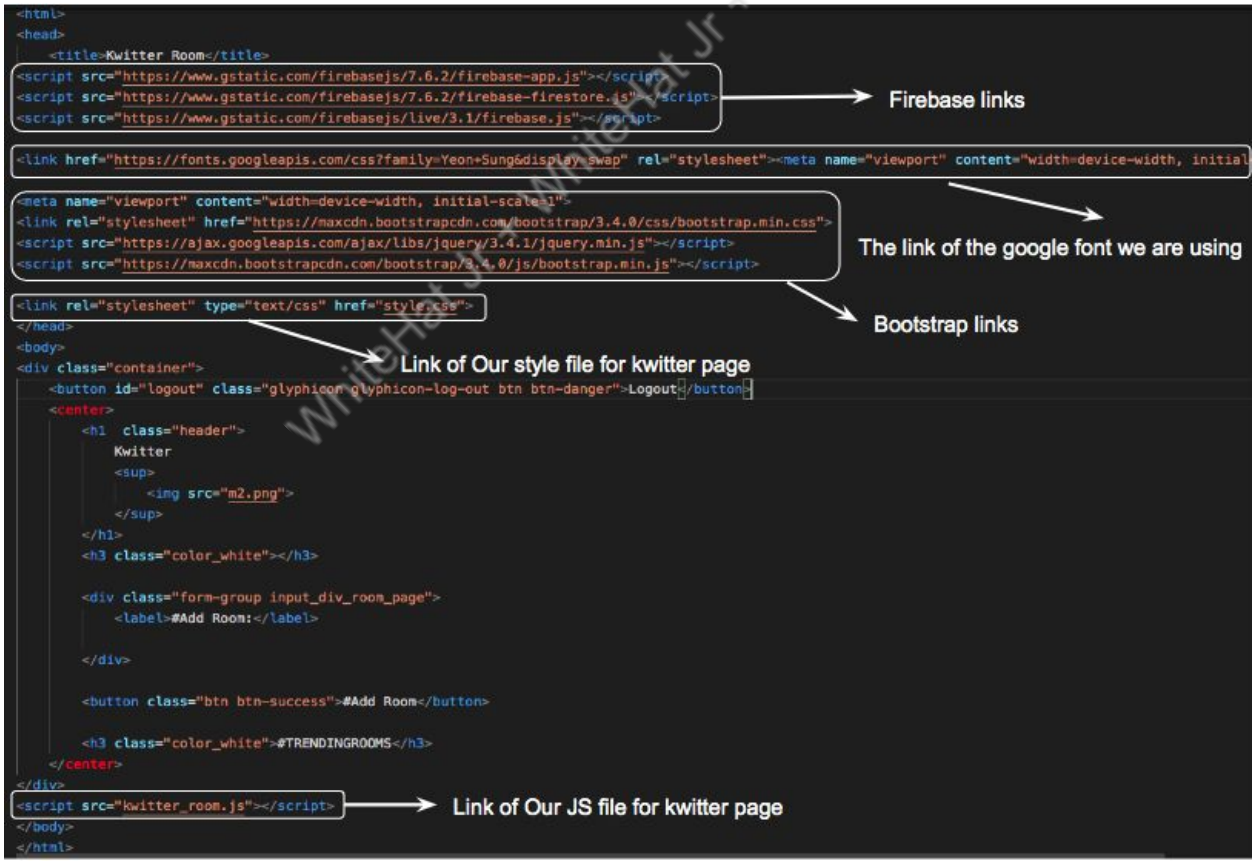


You had downloaded the **Kwitter** folder in today's class.
This **Kwitter** folder has:

- **kwitter.html** - It has some pre HTML code.
- **kwitter.js** - It is a blank JS file.
- **kwitter_room.html** - It has some pre HTML code.
- **kwitter_room.js** - It has some JS pre written code, **DON'T TOUCH** that code, we will discuss it in next class.
- **style.css** - It has all the css, so no need to change anything in this file.

The following code is already given to you:

kwitter.html:



This HTML code has:

- **Firebase links** - these links are there so that we can send and receive data from firebase.
- The link to the google font that we are using.
- **Bootstrap** links
- Our **style.css** file link.
- Our **kwitter_room.js** file link.
- And it has some HTML elements that we already know.

Now add some HTML elements, onclick functions, and ids in this same HTML file.

1. Add onclick **logout()** function to the logout button. This **logout()** function will logout the user from the kwitter app and redirect to the kwitter login page.

```
<body>
<div class="container">
    <button id="logout" onclick="logout();" class="glyphicon glyphicon-log-out btn btn-danger">Logout</button>
```

- **Button** code explanation:
- **id** - we have given id to identify this button.
- **glyphicon glyphicon-log-out** - this bootstrap class will add the logout icon inside the logout button.

- **onclick**="**logout()**" - we will define this function next class.

2. Now add **id="user_name"** to the **h3** tag who has **class="color_white**", this **h3** tag will be used to hold the user name. Like this:


```
<body>
<div class="container">
<button id="logout" onclick="logout();" class="glyphicon glyphicon-log-out btn btn-danger">Logout</button>
    <center>
        <h1  class="header">
            Kwitter
            <sup>
                <img src="m2.png">
            </sup>
        </h1>
        <h3 id="user_name" class="color_white"></h3>
```

- We are giving **id** to the **h3** tag, so that we can refer to this element using this id and update this HTML element with **“Welcome” + user_name +”!”** from the JS code.

3. Now add an input box for accepting the room name:

```
<div class="form-group input_div_room_page">
    <label>#Add Room:</label>
    <input type="text" id="room_name" class="form-control" placeholder="#Room Name">
</div>
```

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 the Hint to the input box.
- Output -

4. Now add the onclick **addRoom()** function to the add room button. This **addRoom()** function will add the room name in the local storage and the firebase database and then redirect the user to **kwitter_page**.

```
<div class="form-group input_div_room_page">
    <label>#Add Room:</label>
    <input type="text" id="room_name" class="form-control" placeholder="#Room Name">
</div>

<button onclick="addRoom();" class="btn btn-success">#Add Room</button>
```

Button code explanation:

- **id** - we have given id identify this button
- **btn** - this bootstrap class will add padding and will remove the default border and color of the button.

- **btn-success** - this bootstrap class will add a light green border and a background color to the button.
- **onclick**="**addRoom()**" - we will define this function in the next class.

5. Now add a div with **id="output"**, this div tag will be used to hold all the room names.

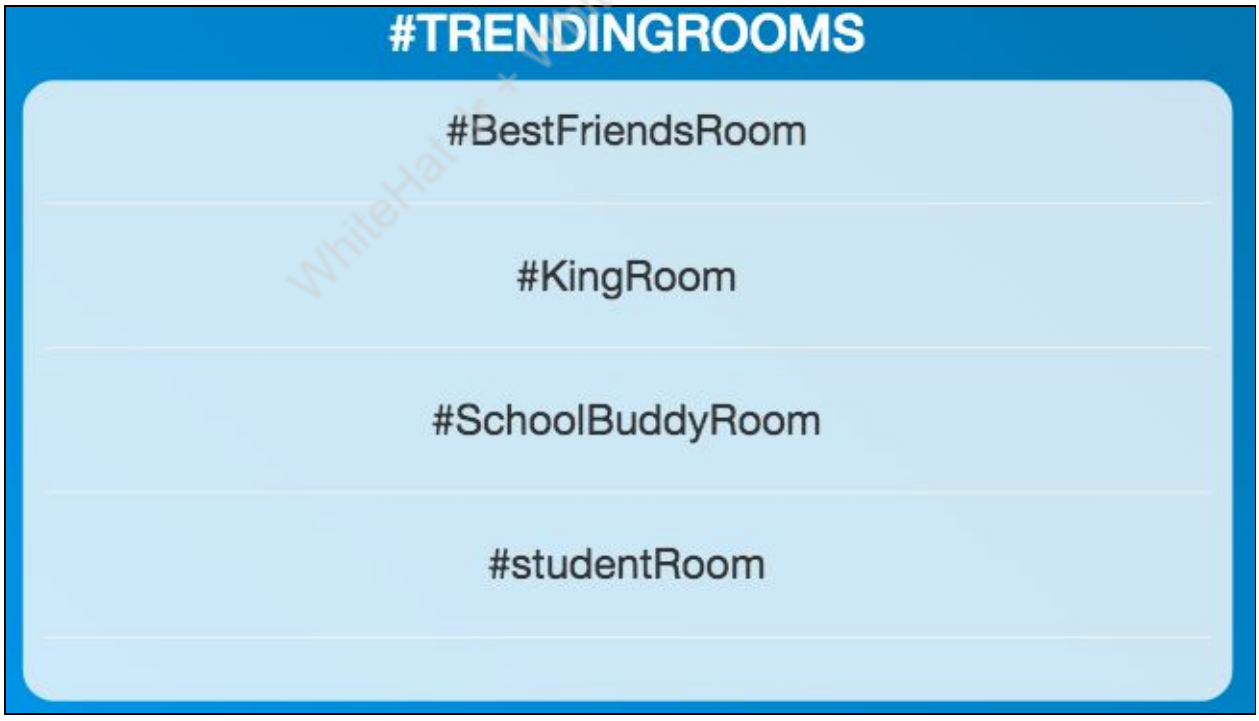
```
<button onclick="addRoom();" class="btn btn-success">#Add Room</button>

<h3 class="color_white">#TRENDINGROOMS</h3>

<div id="output"></div>
```

- We are giving **id** to the **div** tag, so that we can refer to this element using this id and update this HTML element all the room names.

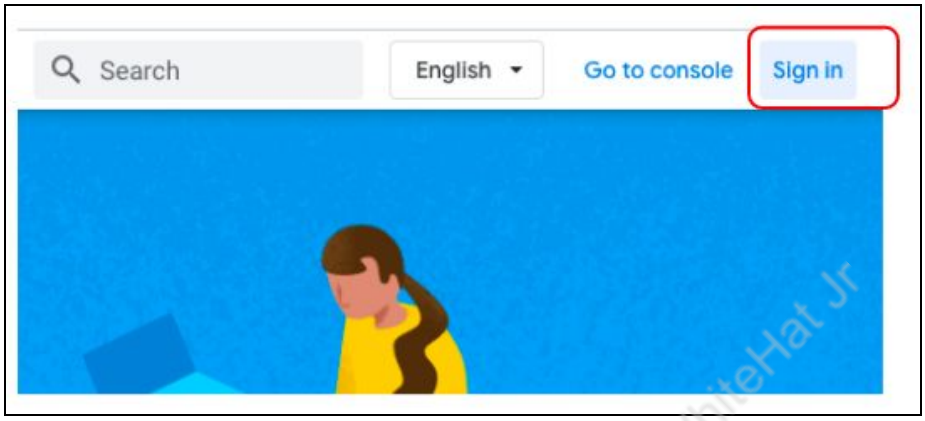
- Sample output:



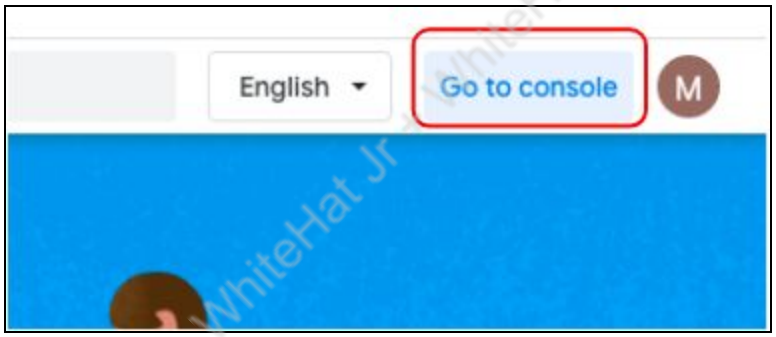
Steps to create a firebase project.

Open https://firebase.google.com. If you have not logged in with Gmail then do follow from step 1, else start following from step 3 -

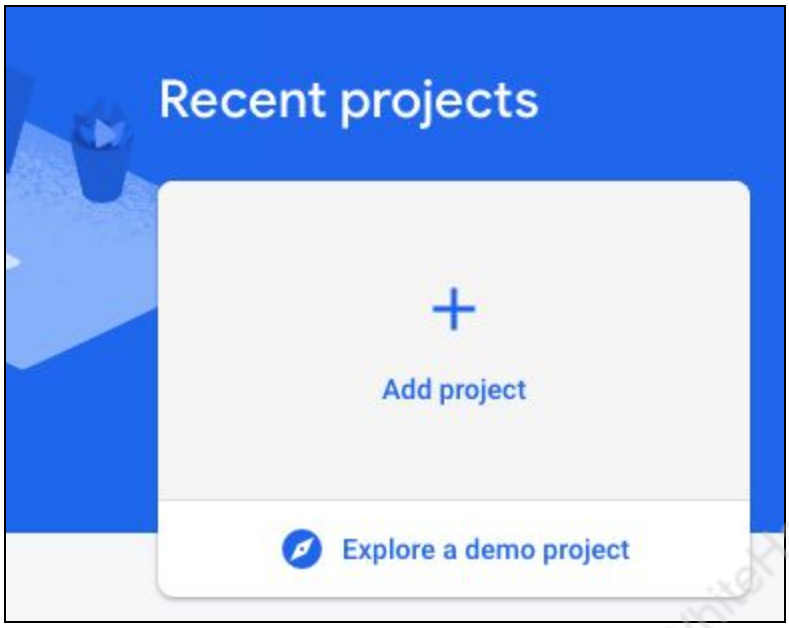
1. Click on the **Sign In** button on the top right corner.



2. Then select the google account for **Sign In**.
3. Click on **Go to console** on the top right corner.

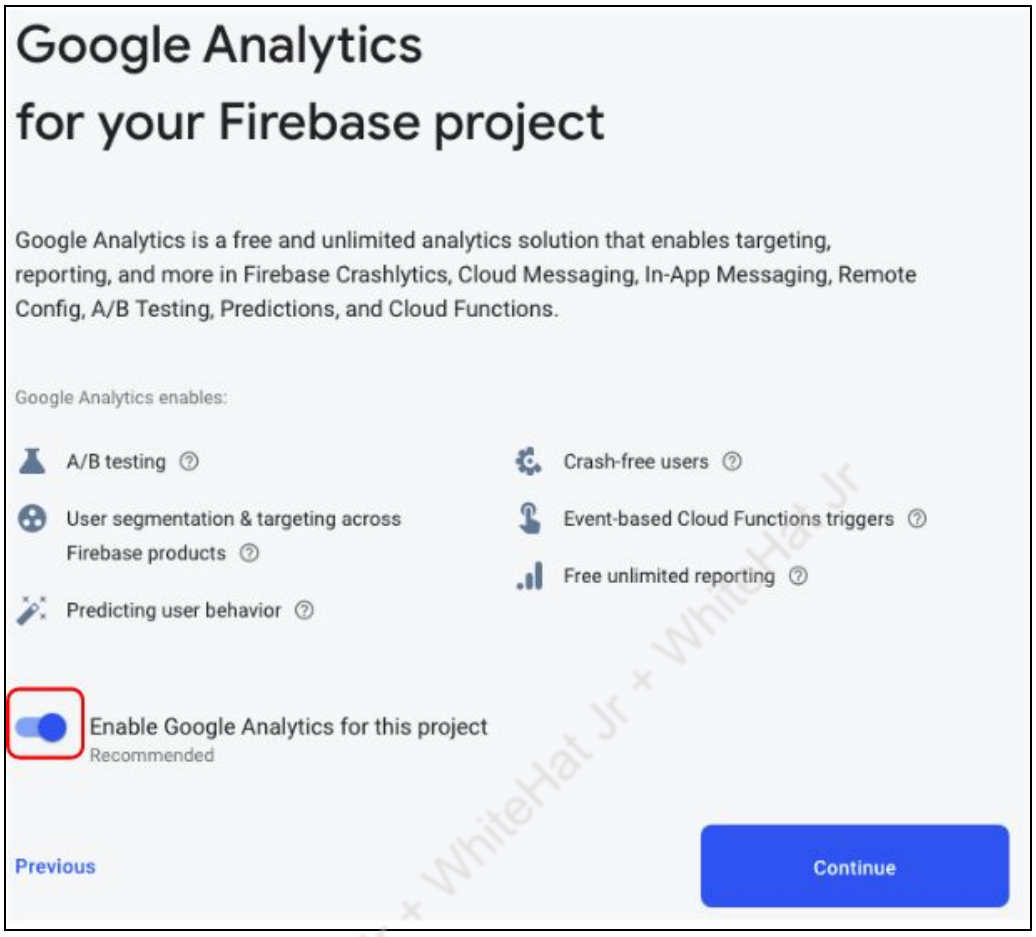


4. Then click on "**Add project**".



5. Then give a name to your project and press the **Continue** button.

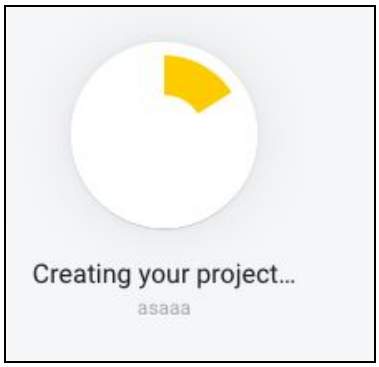
6. Then turn **Google Analytics** off.



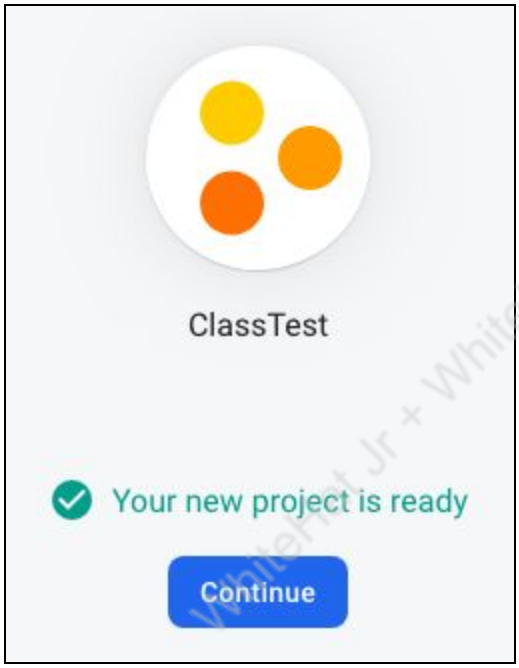
- Like this:

7. And press the create project button.

8. Then it will load like this:

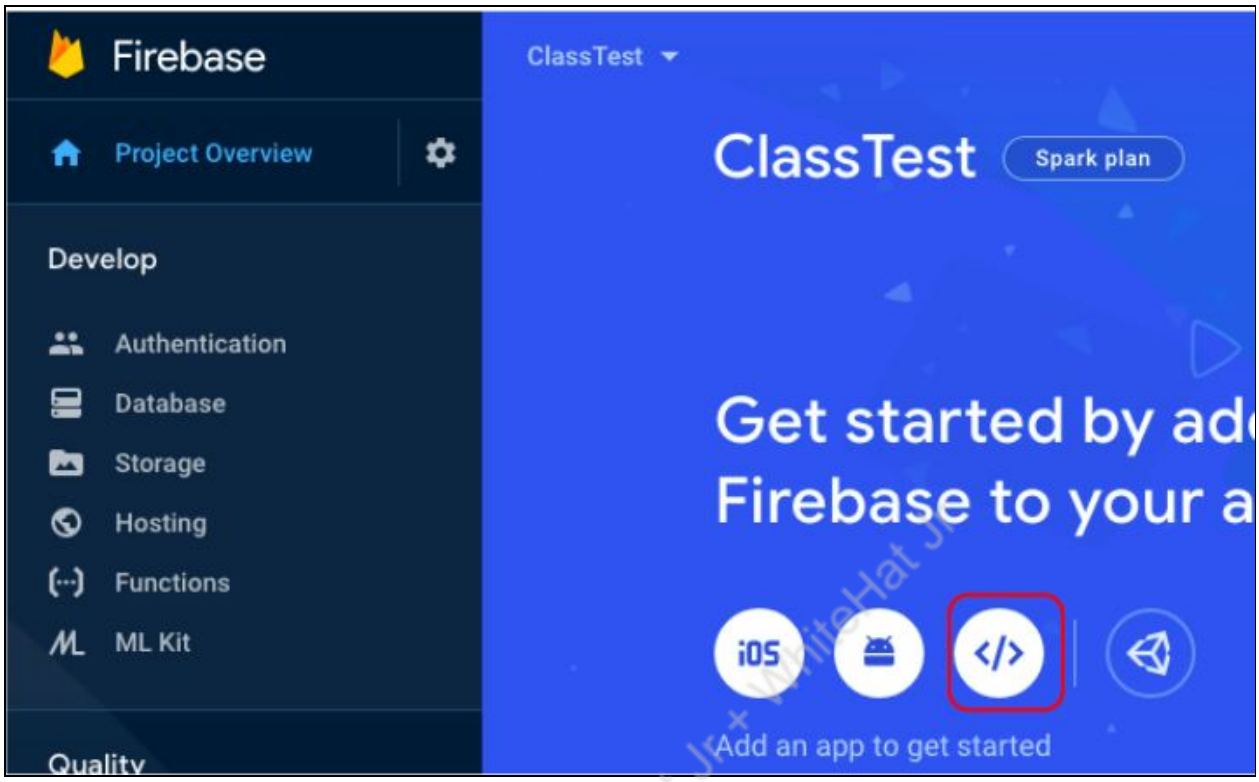


9. After creating the project, click on the continue button:

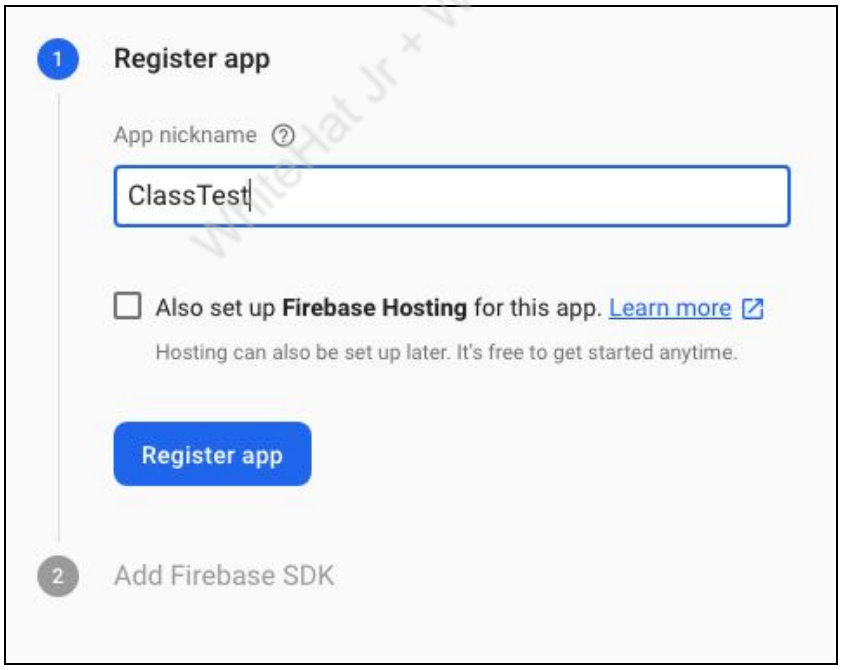


After the project is created, we need to generate our firebase configuration links, these links will be our database links, and it will help for sending and receiving data from the firebase database.

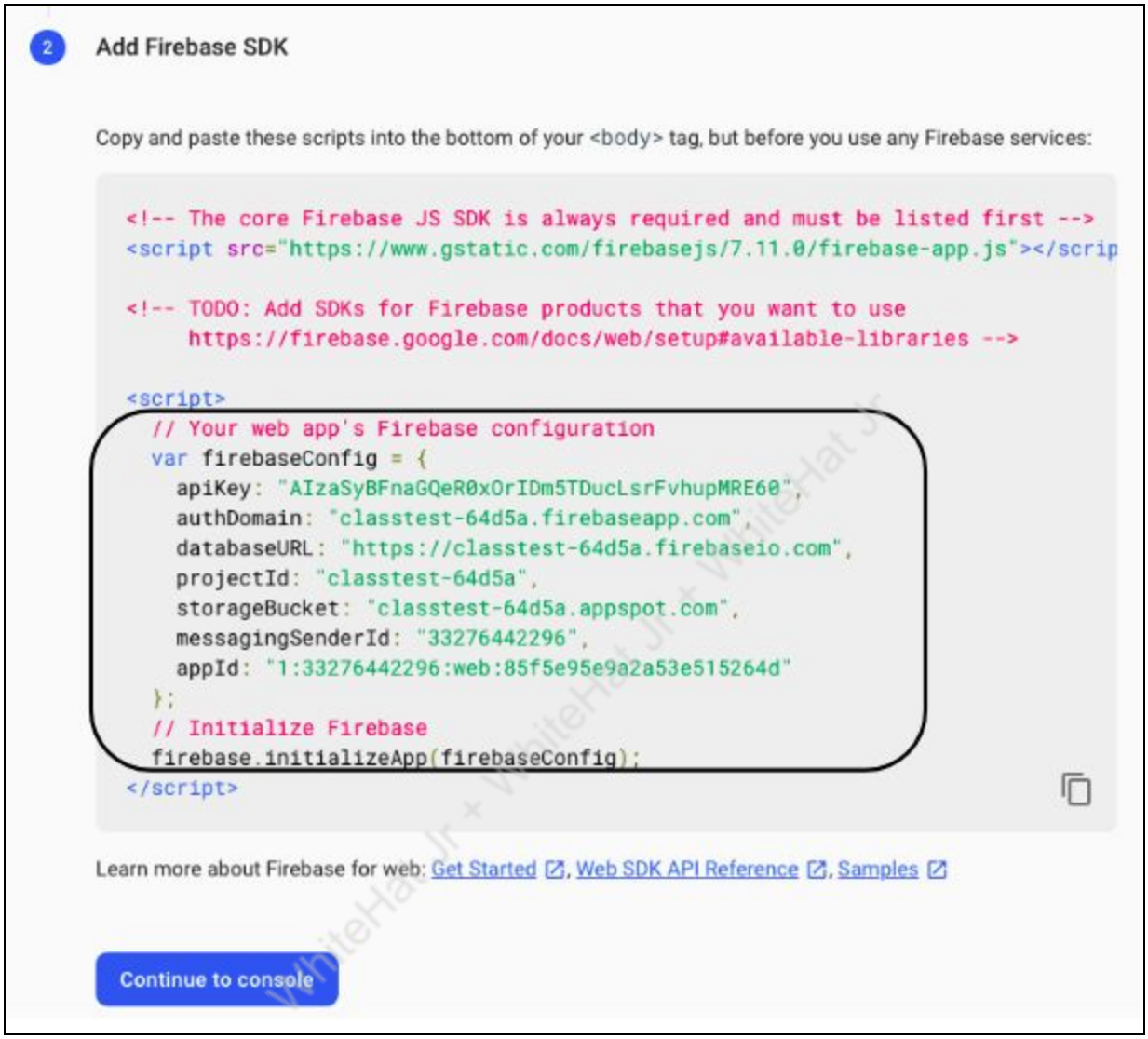
10. So now after you click on continue, you will be redirected to the main page of your project there click on the web symbol.



11. Now give the name for the app, press the **Register app** button.



12. Now it will generate the firebase links.



***Note: You are required to copy only the marked part of the firebase links and paste it in kwitter_room.js file in the starting of the file like the below image. This file is present in the kwitter folder, this folder has been downloaded by you in today's class.**

kwitter_room.js:

What's NEXT?

We will create a firebase database. We will understand the structure of the database. We will also learn how to add data to the database.

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. How to change background image using JS: https://www.youtube.com/watch?v=Gm3znBN0SCM
2. Storing object with local storage in JS: https://www.youtube.com/watch?v=AUOzvFzdIk4
3. Creating a firebase project: https://www.youtube.com/watch?v=6juww5Lmvgo

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

1. 

: https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/onclick

2. 

: https://developer.mozilla.org/en-US/docs/Web/API/Window/location

3. 

: https://firebase.google.com/docs/web/setup