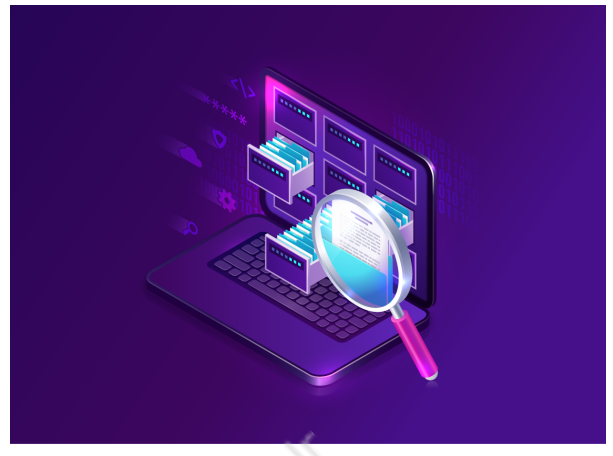


## CONNECTING A-Frame & DATABASE



### What is our GOAL for this MODULE?

We learned to connect A-Frame with the database and to read data from the database in the AR scene.

### What did we ACHIEVE in the class TODAY?

- Learned to read data from database in A-Frame.
- Learned to host images/models online and read values from the database.

### Which CONCEPTS/CODING BLOCKS did we cover today?

- Firebase as database.
- `<a-marker>`, `<a-entity>` , `<a-assets>` tags.

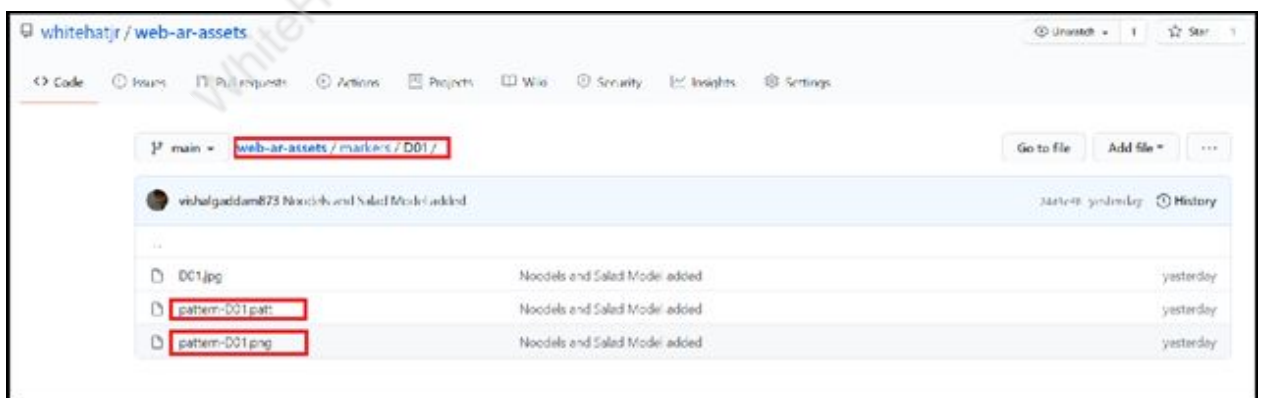
### How did we DO the activities?

1. Upload all the files associated with the 3D models, including textures, in a GitHub repository, so that we can add the (raw) URL in the database.

- Add .gltf file:

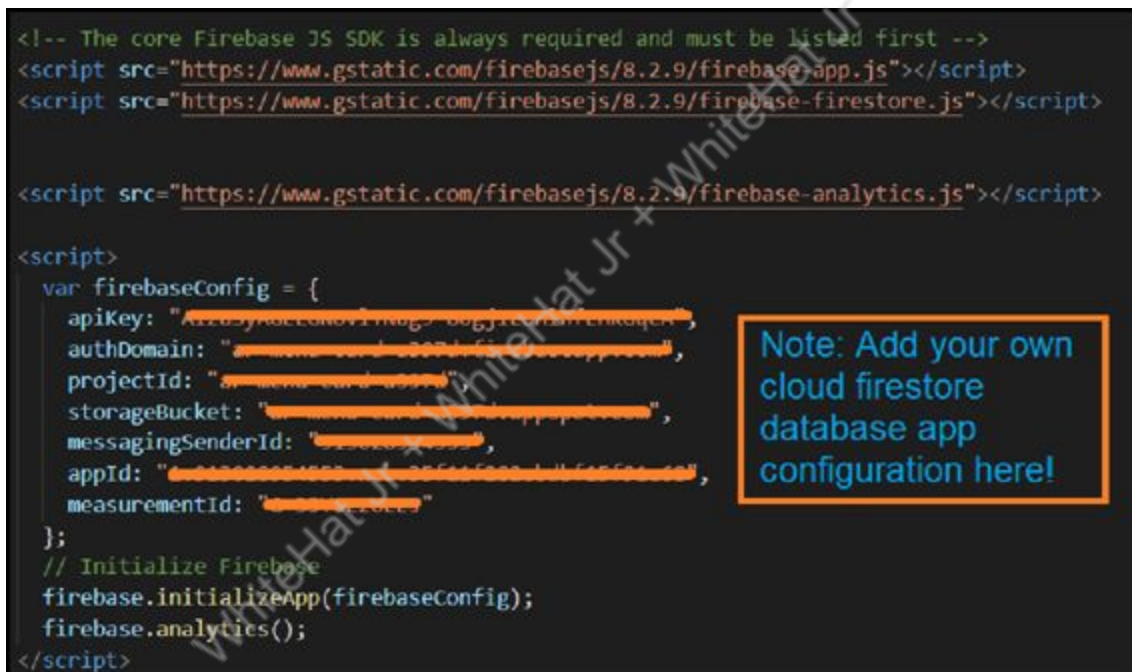


- Add marker .patt and .png files.





2. Add the Firebase configuration in the index.html.



3. Write the create-marker component to set the marker and **getDishes()** function to get the values of the dishes collection from the firestore database and call the function inside the .init() method.

```
AFRAME.registerComponent("create-markers", {  
  init: async function() {  
    var mainScene = document.querySelector("#main-scene");  
  
    //get the dishes collection from firestore database  
    var dishes = await this.getDishes();  
  
    dishes.map(dish => {  
      });  
  },  
  //function to get the dishes collection from firestore database  
  getDishes: async function() {  
    return await firebase  
      .firestore()  
      .collection("dishes")  
      .get()  
      .then(snap => {  
        return snap.docs.map(doc => doc.data());  
      });  
  }  
});
```

4. Add marker entity.

```
dishes.map(dish => {  
  var marker = document.createElement("a-marker");  
  marker.setAttribute("id", dish.id);  
  marker.setAttribute("type", "pattern");  
  marker.setAttribute("url", dish.marker_pattern_url);  
  marker.setAttribute("cursor", {  
    rayOrigin: "mouse"  
  });  
});  
  
//set the markerhandler component  
marker.setAttribute("markerhandler", {});  
mainScene.appendChild(marker);
```

5. Add the model entity.

```
// Adding 3D model to scene
var model = document.createElement("a-entity");

model.setAttribute("id", `model-${dish.id}`);
model.setAttribute("position", dish.model_geometry.position);
model.setAttribute("rotation", dish.model_geometry.rotation);
model.setAttribute("scale", dish.model_geometry.scale);
model.setAttribute("gltf-model", `url(${dish.model_url})`);
model.setAttribute("gesture-handler", {});
marker.appendChild(model);
```

6. Add the plane entity.

```
// Ingredients Container
var mainPlane = document.createElement("a-plane");
mainPlane.setAttribute("id", `main-plane-${dish.id}`);
mainPlane.setAttribute("position", { x: 0, y: 0, z: 0 });
mainPlane.setAttribute("rotation", { x: -90, y: 0, z: 0 });
mainPlane.setAttribute("width", 1.7);
mainPlane.setAttribute("height", 1.5);
marker.appendChild(mainPlane);

// Dish title background plane
var titlePlane = document.createElement("a-plane");
titlePlane.setAttribute("id", `title-plane-${dish.id}`);
titlePlane.setAttribute("position", { x: 0, y: 0.89, z: 0.02 });
titlePlane.setAttribute("rotation", { x: 0, y: 0, z: 0 });
titlePlane.setAttribute("width", 1.69);
titlePlane.setAttribute("height", 0.3);
titlePlane.setAttribute("material", { color: "#F0C30F" });
mainPlane.appendChild(titlePlane);
```

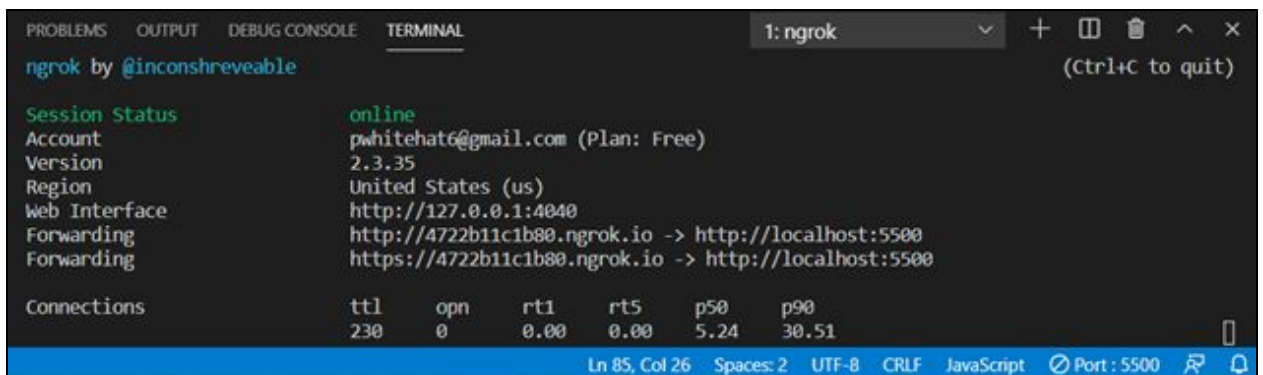


7. Add title and ingredients list entity.

```
// Dish title
var dishTitle = document.createElement("a-entity");
dishTitle.setAttribute("id", `dish-title-${dish.id}`);
dishTitle.setAttribute("position", { x: 0, y: 0, z: 0.1 });
dishTitle.setAttribute("rotation", { x: 0, y: 0, z: 0 });
dishTitle.setAttribute("text", {
  font: "monoid",
  color: "black",
  width: 1.8,
  height: 1,
  align: "center",
  value: dish.dish_name.toUpperCase()
});
titlePlane.appendChild(dishTitle);

// Ingredients List
var ingredients = document.createElement("a-entity");
ingredients.setAttribute("id", `ingredients-${dish.id}`);
ingredients.setAttribute("position", { x: 0.3, y: 0, z: 0.1 });
ingredients.setAttribute("rotation", { x: 0, y: 0, z: 0 });
ingredients.setAttribute("text", {
  font: "monoid",
  color: "black",
  width: 2,
  align: "left",
  value: `${dish.ingredients.join("\n\n")}`
});
mainPlane.appendChild(ingredients);
```

8. Run the program using ngrok.



The screenshot shows a terminal window with the following content:

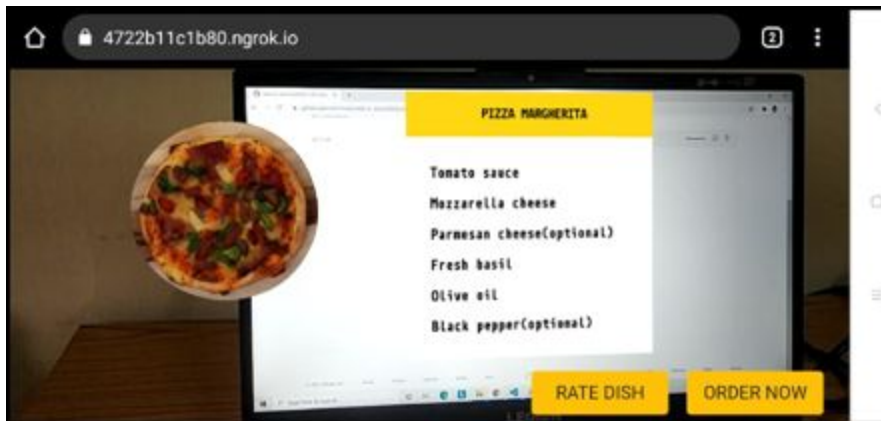
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: ngrok
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status online
Account pwhitehat6@gmail.com (Plan: Free)
Version 2.3.35
Region United States (us)
Web Interface http://127.0.0.1:4040
Forwarding http://4722b11c1b80.ngrok.io -> http://localhost:5500
Forwarding https://4722b11c1b80.ngrok.io -> http://localhost:5500

Connections
```

	ttl	opn	rt1	rt5	p50	p90
	230	0	0.00	0.00	5.24	30.51

Ln 85, Col 26 Spaces: 2 UTF-8 CRLF JavaScript Port: 5500



### What's NEXT?

In the next class, we will learn to add the order details in the AR scene.

### EXTEND YOUR KNOWLEDGE:

- You can refer to the link below to explore more about A-Frame  
[A-Frame](#)
- You can refer to the link below to explore more about AR.js  
[AR.js](#)