

ADDING ORDER DETAILS



What is our GOAL for this MODULE?

We learned to write data in the database in the AR scene and also to take input through alert and add that data into the firestore database.

What did we ACHIEVE in the class TODAY?

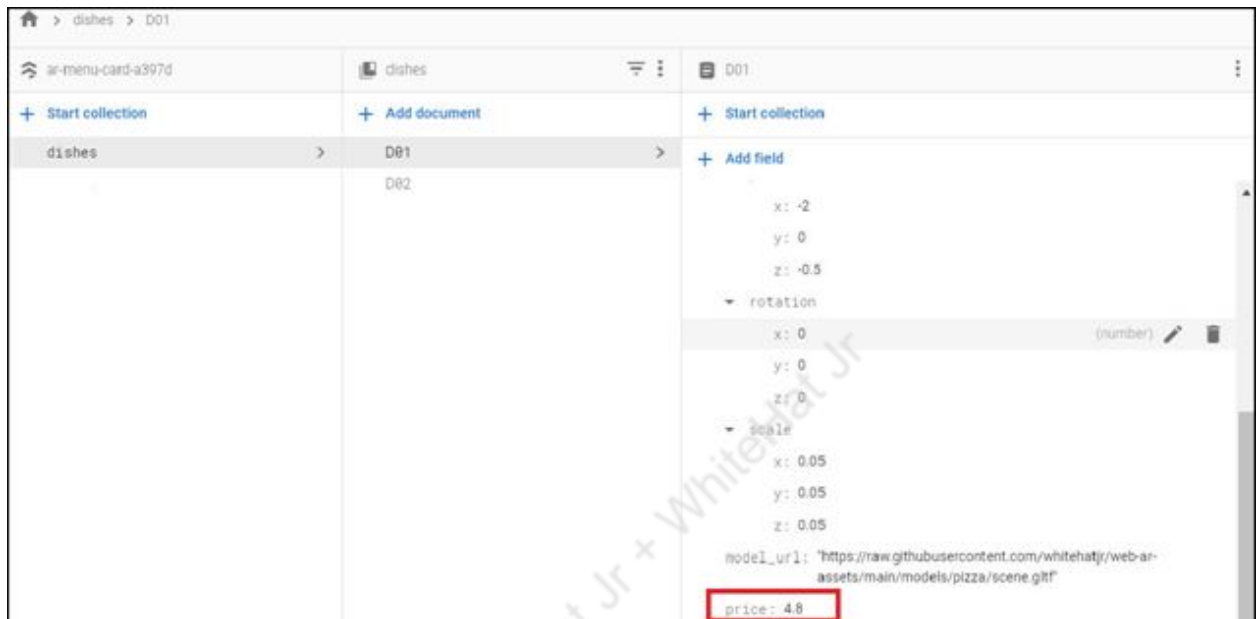
- Learned to write data in the database in A-Frame.
- Learn to take input values through alert and add data fields in the database.

Which CONCEPTS/CODING BLOCKS did we cover today?

- Firebase as database.
- The swal() function.
- closeOnClickOutside, content.
- <a-marker>,<a-entity> , <a-assets> tags.
- ngrok to run the application.

How did we DO the activities?

1. Add the price field in the database for all the dishes present in the database.

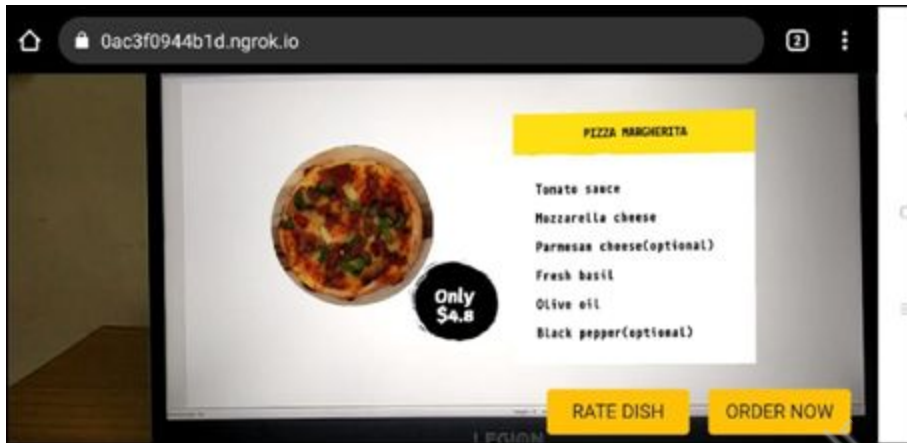


2. Update the **addMarker.js** to add the price in the AR scene UI objects.

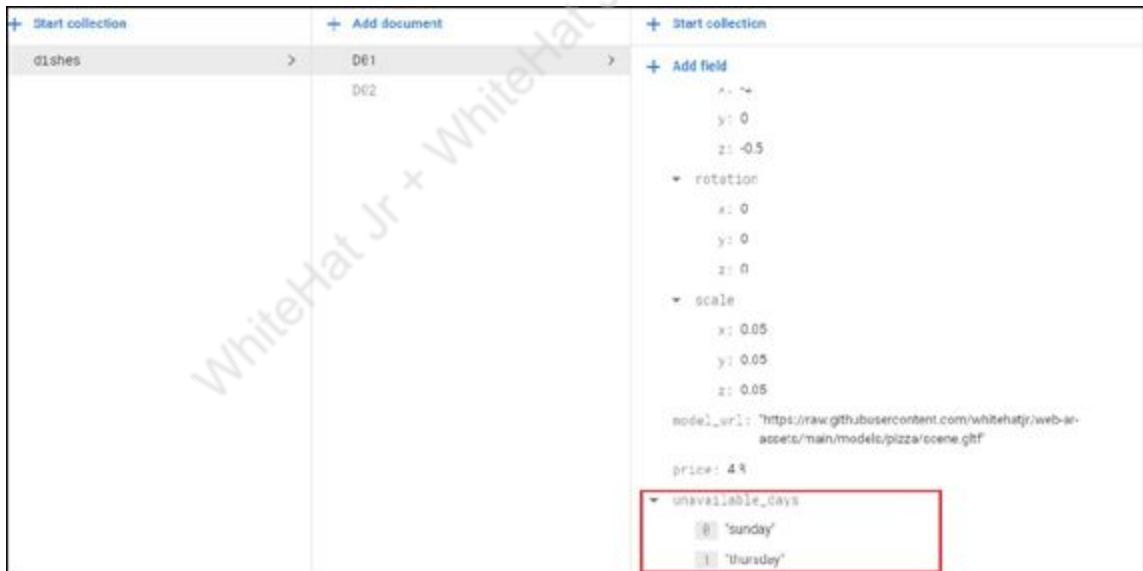
```
//Plane to show the price of the dish
var pricePlane = document.createElement("a-image");
pricePlane.setAttribute("id", "price-plane-${dish.id}");
pricePlane.setAttribute(
  "src",
  "https://raw.githubusercontent.com/whitehatjr/menu-card-app/main/black-circle.png"
);
pricePlane.setAttribute("width", 0.8);
pricePlane.setAttribute("height", 0.8);
pricePlane.setAttribute("position", { x: -1.3, y: 0, z: 0.3 });
pricePlane.setAttribute("rotation", { x: -90, y: 0, z: 0 });

//Price of the dish
var price = document.createElement("a-entity");
price.setAttribute("id", "price-${dish.id}");
price.setAttribute("position", { x: 0.03, y: 0.05, z: 0.1 });
price.setAttribute("rotation", { x: 0, y: 0, z: 0 });
price.setAttribute("text", {
  font: "mozillavr",
  color: "white",
  width: 3,
  align: "center",
  value: `Only\n${dish.price}`
});

pricePlane.appendChild(price);
marker.appendChild(pricePlane);
```



3. Add the `unavailable_days` field of type array in the database for all the dishes present in the database.



4. Add the functionality in `addMarker.js` & `markerHandler.js` to check whether the dish is available on a particular weekday or not and test the output.

```
handleMarkerFound: function (dishes, markerId) {  
  // Getting today's day  
  var todaysDate = new Date();  
  var todaysDay = todaysDate.getDay();  
  // Sunday - Saturday : 0 - 6  
  var days = [  
    "sunday",  
    "monday",  
    "tuesday",  
    "wednesday",  
    "thursday",  
    "friday",  
    "saturday"  
  ];  
  
  var dish = dishes.filter(dish => dish.id === markerId)[0];  
  
  if (dish.unavailable_days.includes(days[todaysDay])) {  
    swal({  
      icon: "warning",  
      title: dish.dish_name.toUpperCase(),  
      text: "This dish is not available today!!!",  
      timer: 2500,  
      buttons: false  
    });  
  } else {  
    // Changing Model scale to initial scale  
  
    // Changing button div visibility  
  
    // Handling Click Events  
  }  
},
```

```

AFRAME.registerComponent("create-markers", {
  init: async function () {
    var mainScene = document.querySelector("#main-scene");
    var dishes = await this.getDishes();
    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"
      });
      marker.setAttribute("markerhandler", {});
      mainScene.appendChild(marker);

      // Getting today's day
      var todaysDate = new Date();
      var todaysDay = todaysDate.getDay();
      // Sunday - Saturday : 0 - 6
      var days = [
        "sunday",
        "monday",
        "tuesday",
        "wednesday",
        "thursday",
        "friday",
        "saturday"
      ];

      if (!dish.unavailable_days.includes(days[todaysDay])) {

        // Adding 3D model to scene

        // Ingredients Container

        // Dish title background plane

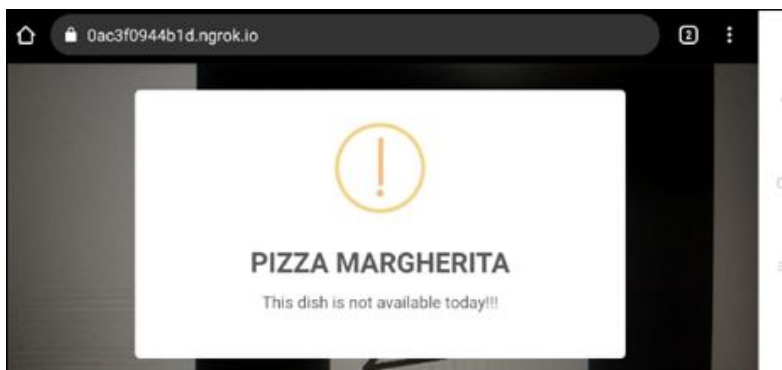
        // Dish title

        // Ingredients List

        //Plane to show the price of the dish

      }
    });
  },
});

```



5. Write a function **askTableNumber()** which will add an input alert to take table number and test the output.

```
var tableNumber = null;

AFRAME.registerComponent("markerhandler", {
  init: async function () {

    if (tableNumber === null) {
      this.askTableNumber();
    }

    //Get the dishes collection
    var dishes = await this.getDishes();

    //markerFound Event
    this.el.addEventListener("markerFound", () => {

      var markerId = this.el.id;
      this.handleMarkerFound(dishes, markerId);

    });

    //markerLost Event
    this.el.addEventListener("markerLost", () => {
      this.handleMarkerLost();
    });
  },

  askTableNumber: function () {
    var iconUrl = "https://raw.githubusercontent.com/whitehatjr/menu-card-app/main/hunger.png";

    swal({
      title: "Welcome to Hunger!!",
      icon: iconUrl,
      content: {
        element: "input",
        attributes: {
          placeholder: "Type your table number",
          type: "number",
          min: 1
        }
      },
      closeOnClickOutside: false,
    }).then(inputValue => {
      tableNumber = inputValue;
    });
  },
});
```



6. Update the dish model and ingredients plane visibility to false in addMarker.js and change it to true in markerHandler.js

- Update **addMarker.js**

```
// 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", {});
model.setAttribute("visible", false);
marker.appendChild(model);
```

```
// 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);
mainPlane.setAttribute("visible", false);
marker.appendChild(mainPlane);
```

- Update **markerHandler.js**

```
//markerFound Event
this.el.addEventListener("markerFound", () => {
  if (tableNumber !== null) {
    var markerId = this.el.id;
    this.handleMarkerFound(dishes, markerId);
  }
});
```

```

handleMarkerFound: function (dishes, markerId) {
  // Getting today's day
  var todaysDate = new Date();
  var todaysDay = todaysDate.getDay();
  // Sunday - Saturday : 0 - 6
  var days = [
    "sunday",
    "monday",
    "tuesday",
    "wednesday",
    "thursday",
    "friday",
    "saturday"
  ];

  //Get the dish based on ID
  var dish = dishes.filter(dish => dish.id === markerId)[0];

  //Update UI content VISIBILITY of AR scene(MODEL , INGREDIENTS & PRICE)
  var model = document.querySelector(`#model-${dish.id}`);
  model.setAttribute("visible", true);

  var ingredientsContainer = document.querySelector(`#main-plane-${dish.id}`);
  ingredientsContainer.setAttribute("visible", true);

  var priceplane = document.querySelector(`#price-plane-${dish.id}`);
  priceplane.setAttribute("visible", true)
}

```

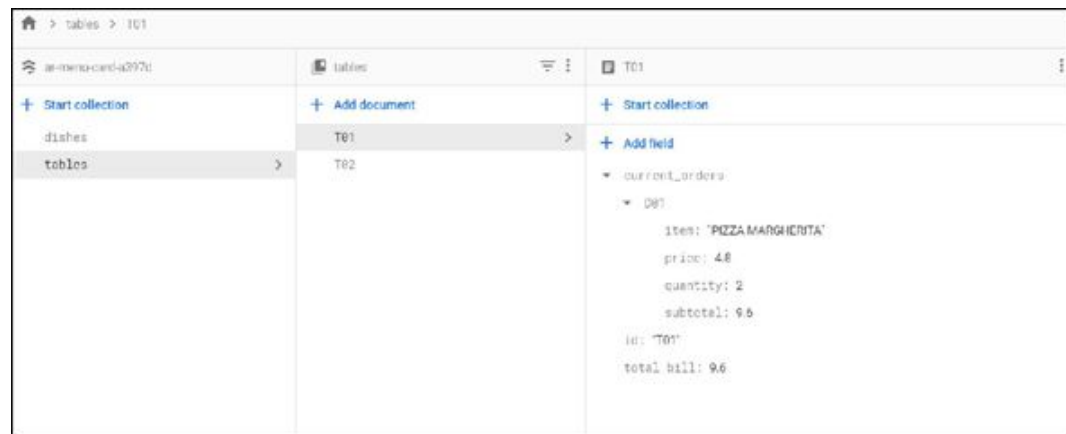
7. Add the tables collection with respective documents and field details to the database.

<div>+ Start collection</div> <div>dishes</div> <div>tables ></div>	<div>+ Add document</div> <div>T01 ></div>	<div>+ Start collection</div> <div>+ Add field</div> <div>current_orders</div> <div>id: 'T01'</div> <div>total_bill: 0</div>
--	---	--

8. Write the function **handleOrder()** and call it inside the order now button click event.

```
handleOrder: function(tNumber, dish) {  
  // Reading current table order details  
  firebase  
    .firestore()  
    .collection("tables")  
    .doc(tNumber)  
    .get()  
    .then(doc => {  
      var details = doc.data();  
  
      if (details["current_orders"][dish.id]) {  
        // Increasing Current Quantity  
        details["current_orders"][dish.id]["quantity"] += 1;  
  
        //Calculating Subtotal of item  
        var currentQuantity = details["current_orders"][dish.id]["quantity"];  
  
        details["current_orders"][dish.id]["subtotal"] =  
          currentQuantity * dish.price;  
      } else {  
        details["current_orders"][dish.id] = {  
          item: dish.dish_name,  
          price: dish.price,  
          quantity: 1,  
          subtotal: dish.price * 1  
        };  
      }  
  
      details.total_bill += dish.price;  
  
      // Updating Db  
      firebase  
        .firestore()  
        .collection("tables")  
        .doc(doc.id)  
        .update(details);  
    });  
},
```

9. Test the output using ngrok:
- Open application using HTTPS URL.
 - Enter table number.
 - Point to the pattern marker of the dish you want to order.
 - Click on "ORDER NOW".
 - Check the database.



We learned how to connect the A-Frame and the firebase database in AR to write data into the database.

What's NEXT?

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

EXTEND YOUR KNOWLEDGE:

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