PRO-C172

ORDER SUMMARY

What is our GOAL for this MODULE?

We learned to read data from the database to show the order summary by adding HTML elements in the AR scene and updating the order details in the firestore database.

What did we ACHIEVE in the class TODAY?

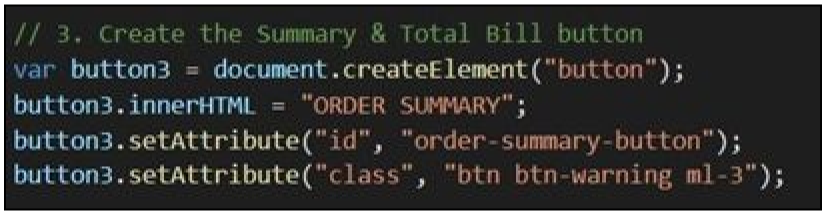
- Learned to read data from the database in A-Frame.
- Learned to add HTML elements in the A-Frame AR scene to show order summary.
- Learned to update order details to confirm order.

Which CONCEPTS/CODING BLOCKS did we cover today?

- Firebase as database
- HTML Modal Box,<table>,<tr>,<th>,<td> tags
- .innerHTML, .createElement('strong') etc
- <a-marker>,<a-entity> , <a-assets> tags
- ngrok to run the application

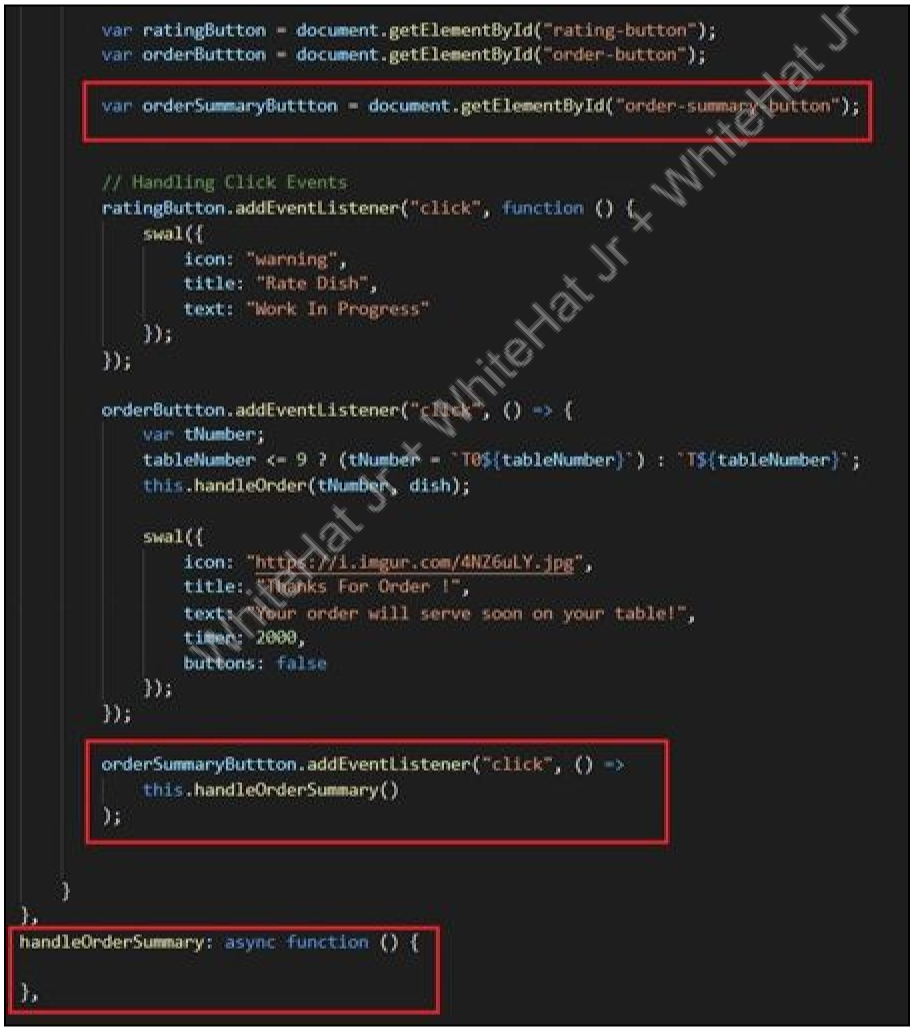
How did we DO the activities?

1. Create the button element and append it to the button div in the scene.



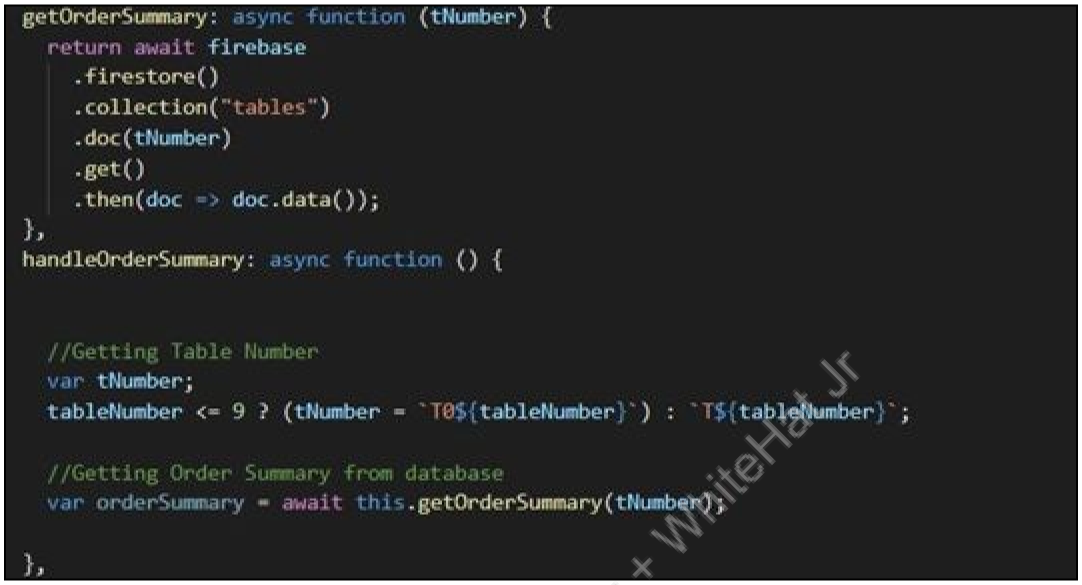
```
// 3. Create the Summary & Total Bill button
var button3 = document.createElement("button");
button3.innerHTML = "ORDER SUMMARY";
button3.setAttribute("id", "order-summary-button");
button3.setAttribute("class", "btn btn-warning ml-3");
```

2. Add a click event listener to it in the **markerhandler** component.



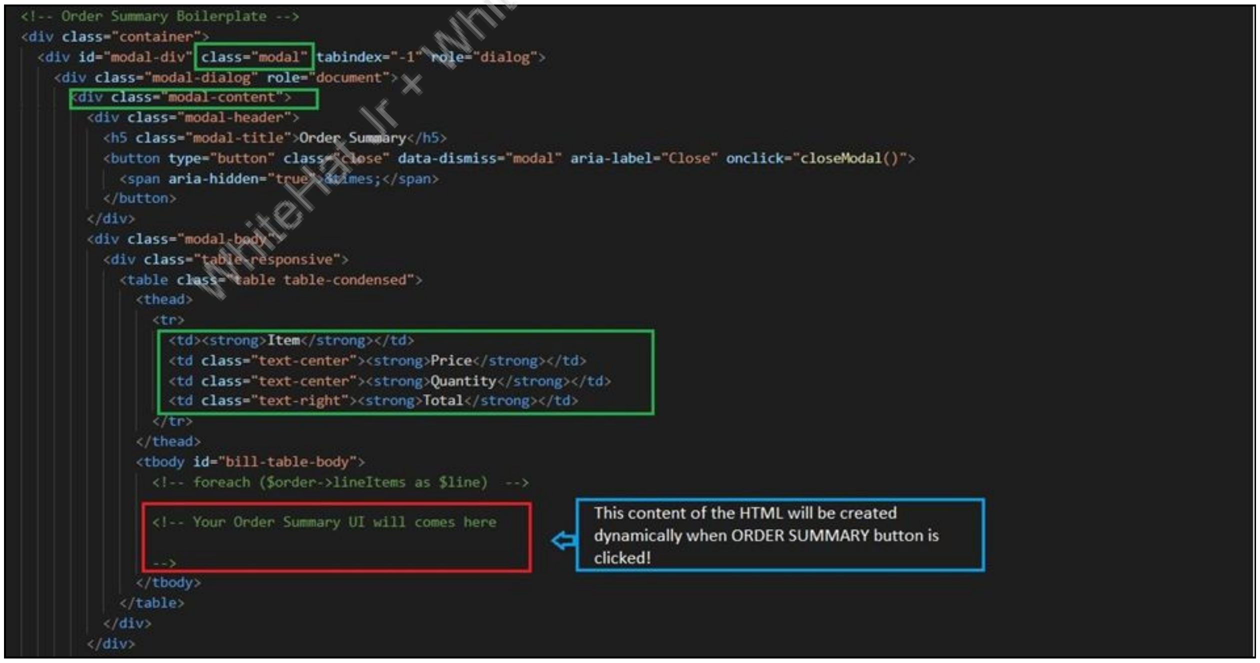
```
      var ratingButton = document.getElementById("rating-button");
      var orderButtton = document.getElementById("order-button");

      var orderSummaryButtton = document.getElementById("order-summary-button");

      // Handling Click Events
      ratingButton.addEventListener("click", function () {
        swal({
          icon: "warning",
          title: "Rate Dish",
          text: "Work In Progress"
        });
      });

      orderButtton.addEventListener("click", () => {
        var tNumber;
        tableNumber <= 9 ? (tNumber = `T0${tableNumber}`) : `T${tableNumber}`;
        this.handleOrder(tNumber, dish);

        swal({
          icon: "https://i.imgur.com/4NZ6uLY.jpg",
          title: "Thanks For Order !",
          text: "Your order will serve soon on your table!",
          timer: 2000,
          buttons: false
        });
      });

      orderSummaryButtton.addEventListener("click", () =>
        this.handleOrderSummary()
      );

    }
  },
  handleOrderSummary: async function () {

  },
```

3. Write a function **getOrderSummary()** to get the tables collection from the database and call it inside **handleOrderSummary()**.



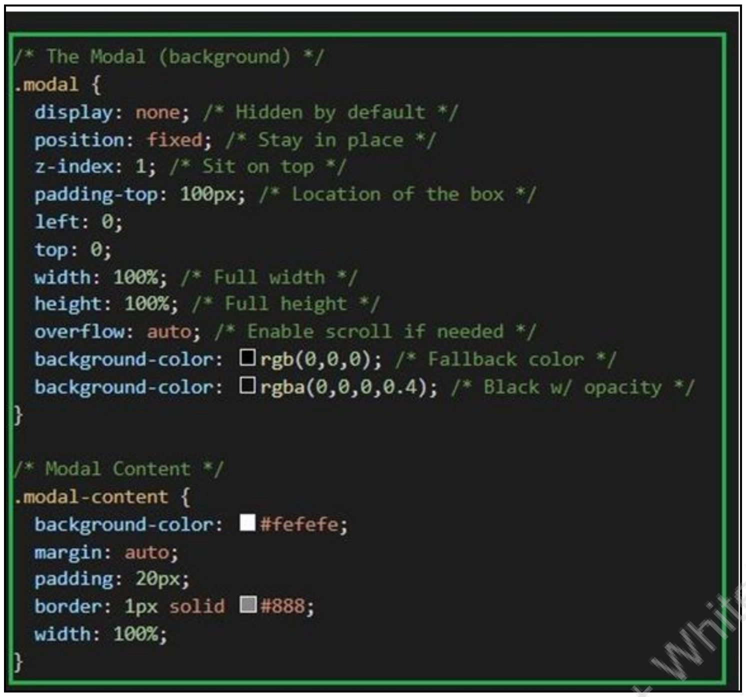
```
getOrderSummary: async function (tNumber) {
  return await firebase
    .firestore()
    .collection("tables")
    .doc(tNumber)
    .get()
    .then(doc => doc.data());
},
handleOrderSummary: async function () {


  //Getting Table Number
  var tNumber;
  tableNumber <= 9 ? (tNumber = `T0${tableNumber}`) : `T${tableNumber}`;

  //Getting Order Summary from database
  var orderSummary = await this.getOrderSummary(tNumber);

},
```

4. Use HTML Modal Box to show the details and add the styling.



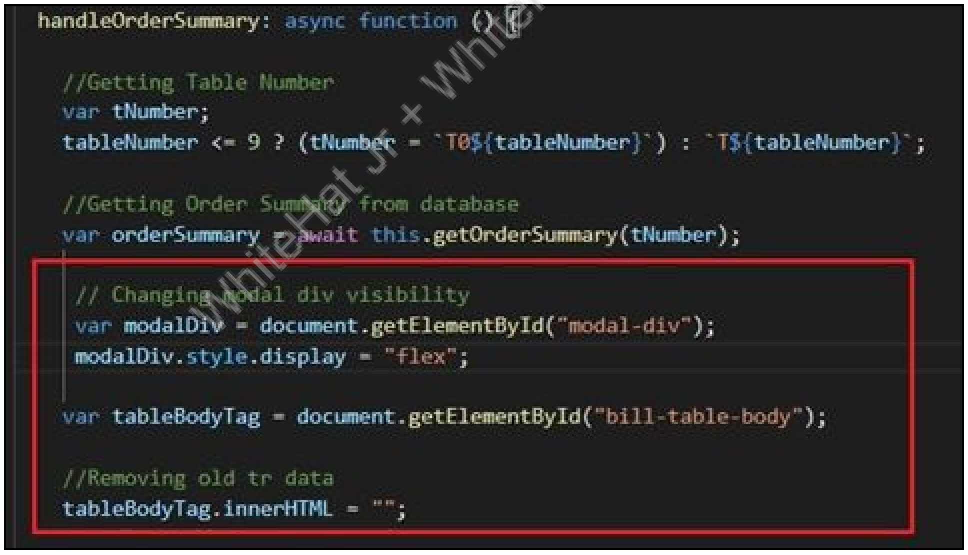
```
<!-- Order Summary Boilerplate -->
<div class="container">
  <div id="modal-div" class="modal" tabindex="-1" role="dialog">
    <div class="modal-dialog" role="document">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title">Order Summary</h5>
          <button type="button" class="close" data-dismiss="modal" aria-label="Close" onclick="closeModal()">
            <span aria-hidden="true">&times;</span>
          </button>
        </div>
        <div class="modal-body">
          <div class="table-responsive">
            <table class="table table-condensed">
              <thead>
                <tr>
                  <td><strong>Item</strong></td>
                  <td class="text-center"><strong>Price</strong></td>
                  <td class="text-center"><strong>Quantity</strong></td>
                  <td class="text-right"><strong>Total</strong></td>
                </tr>
              </thead>
              <tbody id="bill-table-body">
                <!-- foreach ($order->lineItems as $line)  -->

               <!-- Your Order Summary UI will comes here

               -->
              </tbody>
            </table>
          </div>
        </div>
```

This content of the HTML will be created dynamically when ORDER SUMMARY button is clicked!

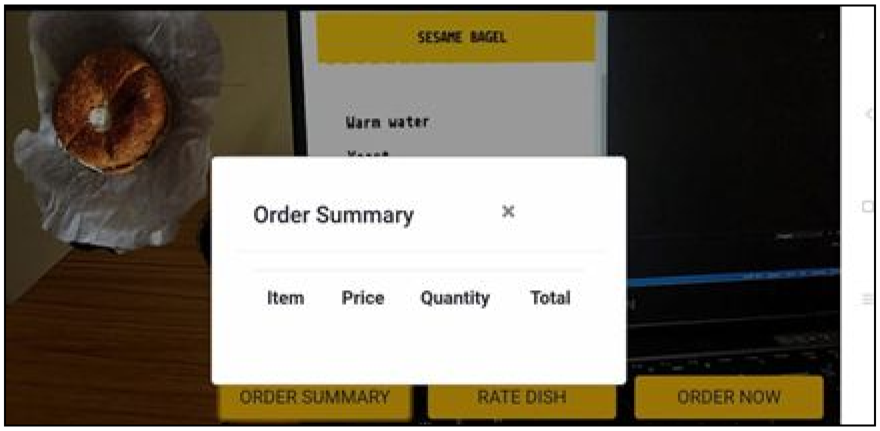


```
/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  padding-top: 100px; /* Location of the box */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
}

/* Modal Content */
.modal-content {
  background-color: #fefefe;
  margin: auto;
  padding: 20px;
  border: 1px solid #888;
  width: 100%;
}
```

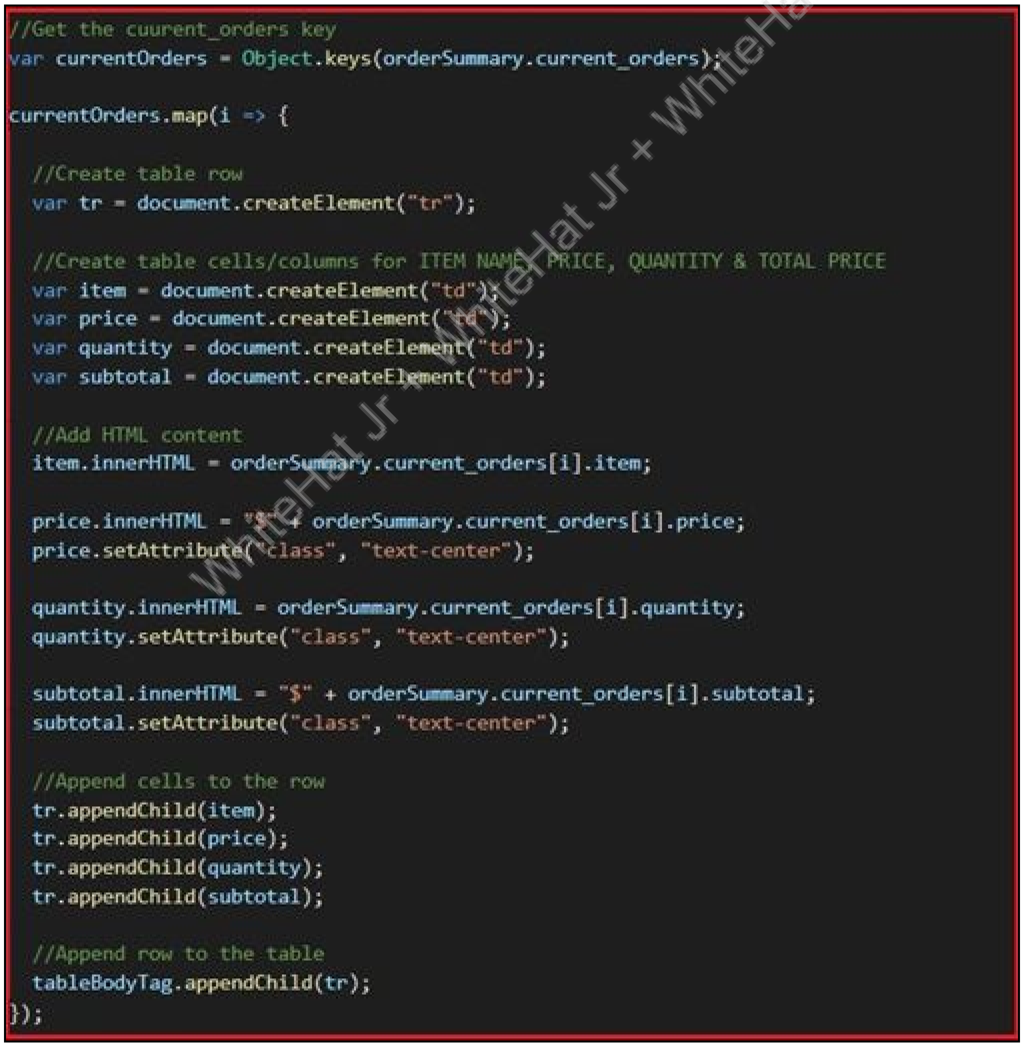
5. Update the modal display property, get the table element and show the output.



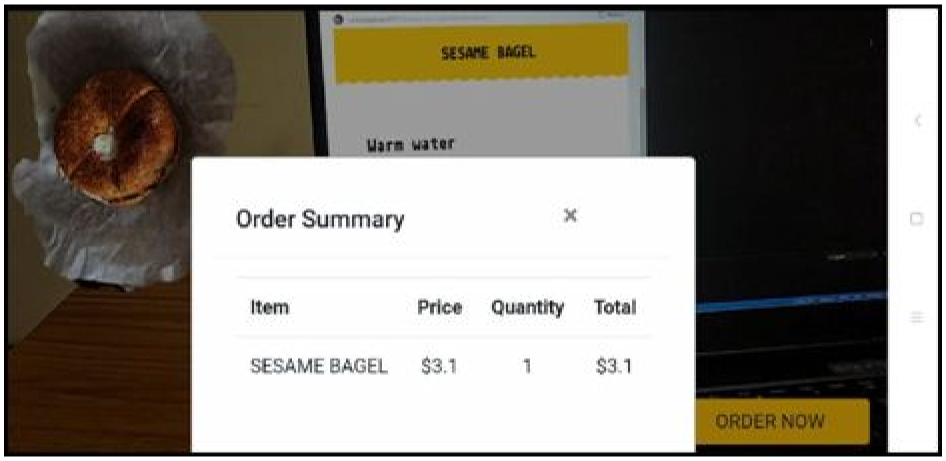
```
handleOrderSummary: async function () {

  //Getting Table Number
  var tNumber;
  tableNumber <= 9 ? (tNumber = `T0${tableNumber}`) : `T${tableNumber}`;

  //Getting Order Summary from database
  var orderSummary = await this.getOrderSummary(tNumber);

   // Changing modal div visibility
   var modalDiv = document.getElementById("modal-div");
   modalDiv.style.display = "flex";

  var tableBodyTag = document.getElementById("bill-table-body");

  //Removing old tr data
  tableBodyTag.innerHTML = "";
```



6. Get the **current_order** field from the database and display the details.



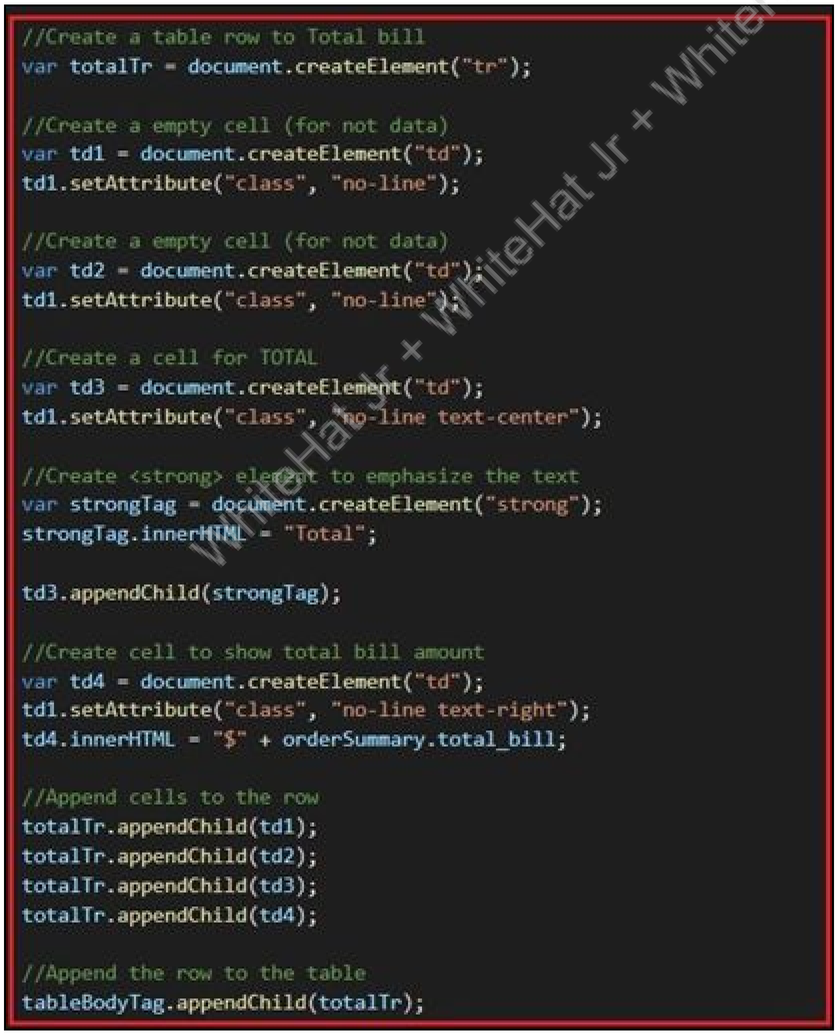
```
//Get the cuurent_orders key
var currentOrders = Object.keys(orderSummary.current_orders);

currentOrders.map(i => {

  //Create table row
  var tr = document.createElement("tr");

  //Create table cells/columns for ITEM NAME, PRICE, QUANTITY & TOTAL PRICE
  var item = document.createElement("td");
  var price = document.createElement("td");
  var quantity = document.createElement("td");
  var subtotal = document.createElement("td");

  //Add HTML content
  item.innerHTML = orderSummary.current_orders[i].item;

  price.innerHTML = "$" + orderSummary.current_orders[i].price;
  price.setAttribute("class", "text-center");

  quantity.innerHTML = orderSummary.current_orders[i].quantity;
  quantity.setAttribute("class", "text-center");

  subtotal.innerHTML = "$" + orderSummary.current_orders[i].subtotal;
  subtotal.setAttribute("class", "text-center");

  //Append cells to the row
  tr.appendChild(item);
  tr.appendChild(price);
  tr.appendChild(quantity);
  tr.appendChild(subtotal);

  //Append row to the table
  tableBodyTag.appendChild(tr);
});
```



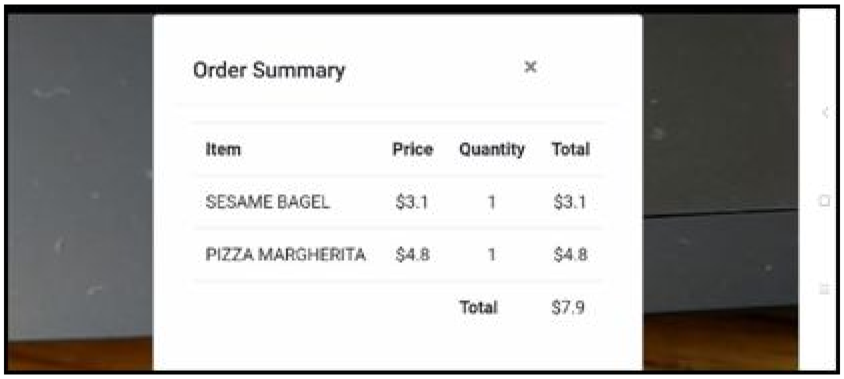
Order Summary ×

Item	Price	Quantity	Total
SESAME BAGEL	$3.1	1	$3.1

7. Add a table row to show the total bill in the order summary.



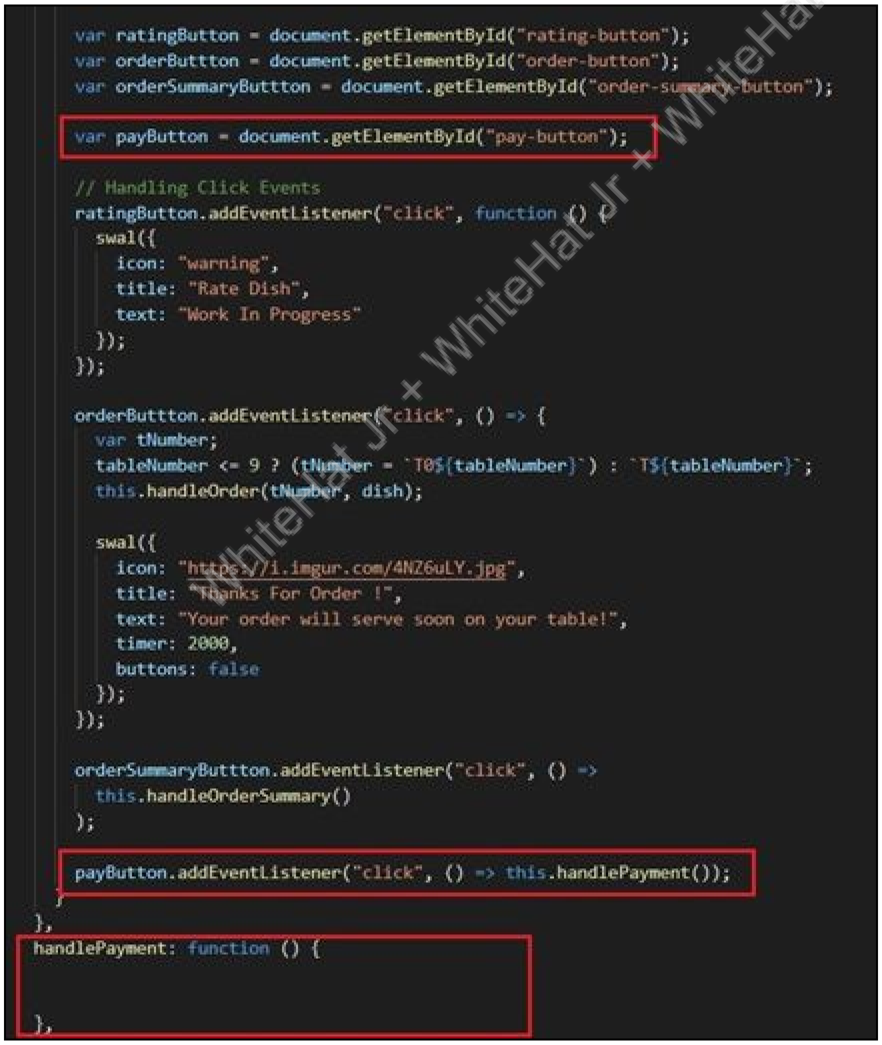
```
//Create a table row to Total bill
var totalTr = document.createElement("tr");

//Create a empty cell (for not data)
var td1 = document.createElement("td");
td1.setAttribute("class", "no-line");

//Create a empty cell (for not data)
var td2 = document.createElement("td");
td1.setAttribute("class", "no-line");

//Create a cell for TOTAL
var td3 = document.createElement("td");
td1.setAttribute("class", "no-line text-center");

//Create <strong> element to emphasize the text
var strongTag = document.createElement("strong");
strongTag.innerHTML = "Total";

td3.appendChild(strongTag);

//Create cell to show total bill amount
var td4 = document.createElement("td");
td1.setAttribute("class", "no-line text-right");
td4.innerHTML = "$" + orderSummary.total_bill;

//Append cells to the row
totalTr.appendChild(td1);
totalTr.appendChild(td2);
totalTr.appendChild(td3);
totalTr.appendChild(td4);

//Append the row to the table
tableBodyTag.appendChild(totalTr);
```



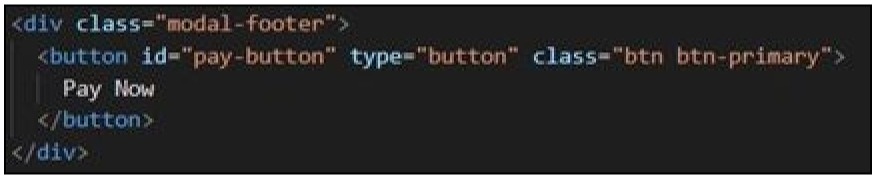
Order Summary

Item	Price	Quantity	Total
SESAME BAGEL	$3.1	1	$3.1
PIZZA MARGHERITA	$4.8	1	$4.8
		Total	$7.9

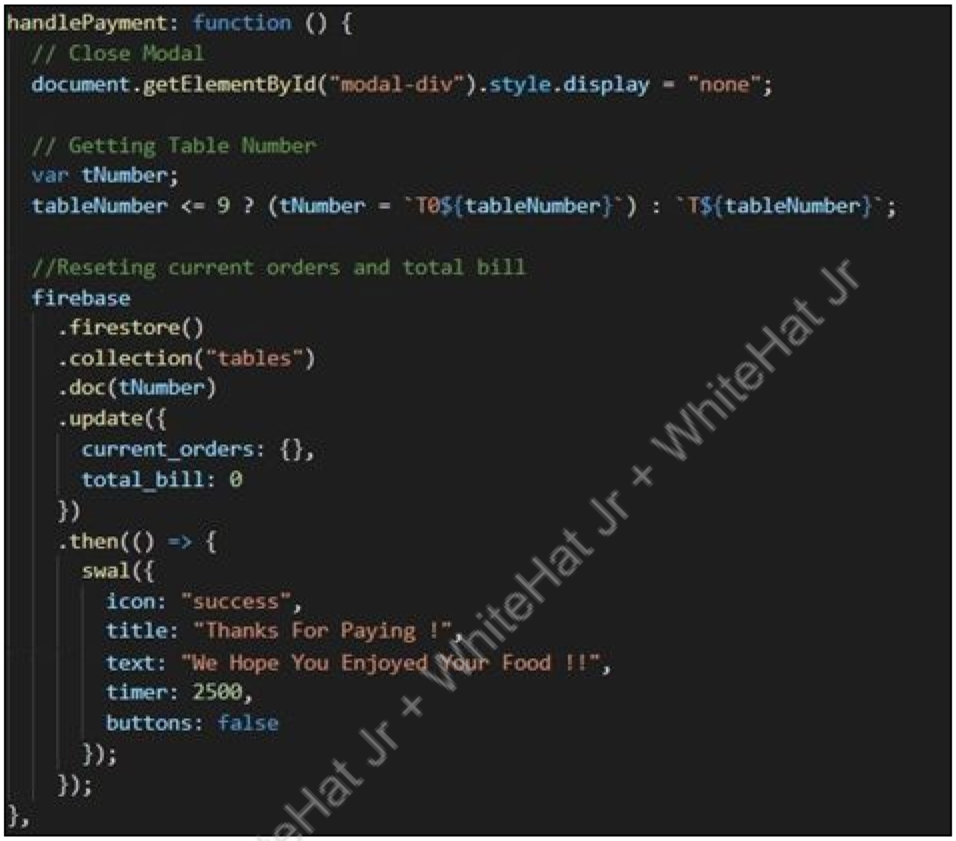
8. Get the "**Pay Now**" button element in **markerhandler** component add a click event listener to call the **handlePayment()** function.



```
    var ratingButton = document.getElementById("rating-button");
    var orderButtton = document.getElementById("order-button");
    var orderSummaryButtton = document.getElementById("order-summary-button");

    var payButton = document.getElementById("pay-button");

    // Handling Click Events
    ratingButton.addEventListener("click", function () {
      swal({
        icon: "warning",
        title: "Rate Dish",
        text: "Work In Progress"
      });
    });

    orderButtton.addEventListener("click", () => {
      var tNumber;
      tableNumber <= 9 ? (tNumber = `T0${tableNumber}`) : `T${tableNumber}`;
      this.handleOrder(tNumber, dish);

      swal({
        icon: "https://i.imgur.com/4NZ6uLY.jpg",
        title: "Thanks For Order !",
        text: "Your order will serve soon on your table!",
        timer: 2000,
        buttons: false
      });
    });

    orderSummaryButtton.addEventListener("click", () =>
      this.handleOrderSummary()
    );

    payButton.addEventListener("click", () => this.handlePayment());
  }
},
handlePayment: function () {


},
```

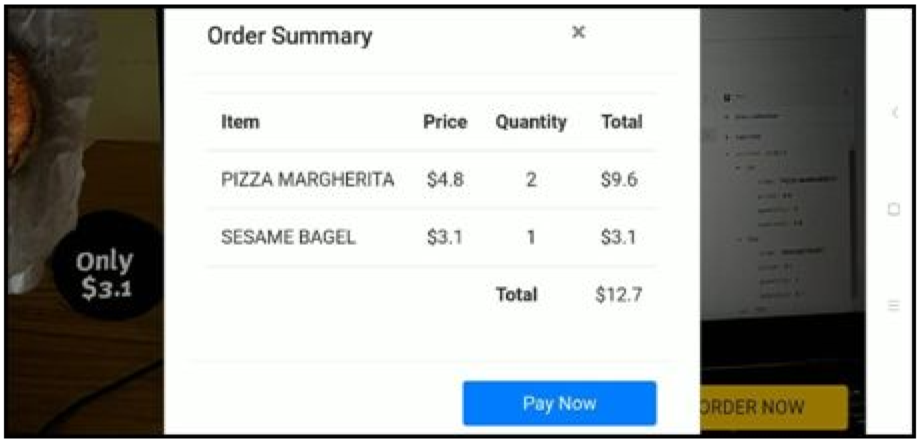


```
<div class="modal-footer">
  <button id="pay-button" type="button" class="btn btn-primary">
    Pay Now
  </button>
</div>
```



```
handlePayment: function () {
  // Close Modal
  document.getElementById("modal-div").style.display = "none";

  // Getting Table Number
  var tNumber;
  tableNumber <= 9 ? (tNumber = `T0${tableNumber}`) : `T${tableNumber}`;

  //Reseting current orders and total bill
  firebase
    .firestore()
    .collection("tables")
    .doc(tNumber)
    .update({
      current_orders: {},
      total_bill: 0
    })
    .then(() => {
      swal({
        icon: "success",
        title: "Thanks For Paying !",
        text: "We Hope You Enjoyed Your Food !!",
        timer: 2500,
        buttons: false
      });
    });
},
```

9. Test the output using ngrok:



Order Summary

Item	Price	Quantity	Total
PIZZA MARGHERITA	$4.8	2	$9.6
SESAME BAGEL	$3.1	1	$3.1
		Total	$12.7

We learned to add HTML elements in the A-Frame to show content in Augmented reality and to read and write into the database.

What's NEXT?

In the next class, we will learn to add star rating functionality for a particular item that was ordered 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 HTML modal
  HTML modal