V&V Content:
Maui Chicken Website

Prepared By
TadBerry

Larry Delgado
Luke Sunaoka
Annie Tran
Nima Fathali Siah

Order Queue: Backpage.js

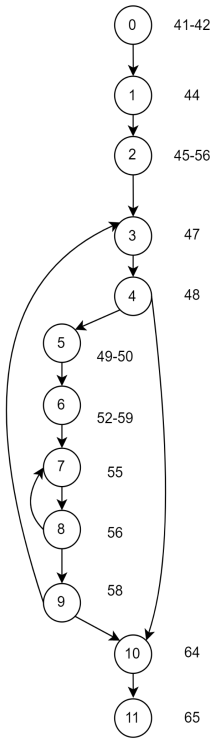
getOrder()

Solving cyclomatic complexity:

- V(G) = 14 edges - 12 nodes + 2 = 4

Paths:

- Path 1: 0-1-2-3-4-5-6-7-8-9-10-11
- Path 2: 0-1-2-3-4-10-11
- Path 3: 0-1-2-3-4-5-6-7-8-7-8-9-10-11
- Path 4: 0-1-2-3-4-5-6-7-8-7-8-9-3-4-10-11




```
function getOrders(loc){ //loc is location
    const q = collection(db, loc); // q is Query
    var hiddenBackPage = document.getElementById("hiddenbackpage");
    //const unsubscribe =
    orderQueueHTMLArray.push(`<h2 id="hi">Orders</h2>`);
    onSnapshot(q, (querySnapshot) => {
        orderQueueHTMLArray = [];
    querySnapshot.forEach((doc) => {
    if(doc.id != "orderCount" && doc){
        let itemsInOrder = doc.data()["cart"];
        console.log(itemsInOrder[0]);
        //update orderQueue HTML to fit new order
        orderQueueHTMLArray.push(`<div id=${doc.id}>`);//Start of Div
        orderQueueHTMLArray.push(`<h3>Order #:  ${doc.id}, Name: ${doc.data()["cart"][0].name}</h3><p>Notes ${doc.data()["cart"][0].notes} Total: ${doc.data()["cart"][0].total}</p><butt
        ordersInQueue.push(doc.id);//each index is an order in queue
        for(let i = 1; i < doc.data()["cart"].length; i++){
            orderQueueHTMLArray.push(`<p>${doc.data()["cart"][i].title}</p><p>Quantity: ${doc.data()["cart"][i].quantity}</p>`);
        }
        orderQueueHTMLArray.push(`</div>`);//End of Div
            }
        });
        // for(let i = 0; i < orderQueueHTMLArray.length-1; i++){
        //     orderQueueHTML += orderQueueHTMLArray[i]; //fills html elements with orders
        // }
        hiddenBackPage.innerHTML = orderQueueHTMLArray;
        removeBtnListeners(ordersInQueue,locate);
    });
    }
```

Test Case Table

Test Case: n is the amount of items returned

Path	P1	P2		Expected Output	Actual Output
1	1	2	3	n	n
2	1	2	3	n	n
3	1	2	3	n	n
4	1	2	3	n	n

Process Order: ShoppingCart.js

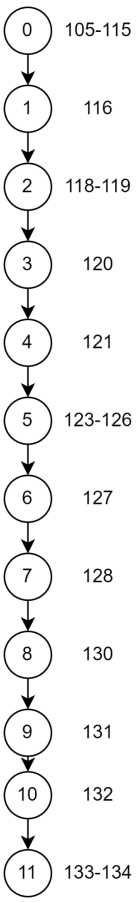
addData()

Solving cyclomatic complexity:

- V(G) = 11 edges - 12 nodes + 2 = 1

Paths:

- Path 1: 0-1-2-3-4-5-6-7-8-9-10-11




```
103  //Use setTimeout to continue to add the order to database
104  async function addData(db){ //Add Data to the database to add to Order Queue
105    const notes = document.getElementById("notes").value;
106    const name = document.getElementById("name").value;
107    const locate = document.getElementById("locations").value;
108    //cart.name = name; //Adding Name of person order to cart
109    //cart.notes = notes;
110    let orderNum = 0;
111    const getOrderNum = async() => {
112        let docSnap = await getDoc(doc(db,locate,"orderCount"));
113        orderNum = docSnap.data().count;
114        console.log("The number" + orderNum);
115      };
116    getOrderNum();
117    setTimeout(()=>{
118      console.log("Order # " + orderNum);//Rest of code goes here because of asyncy getOrderNum
119      let date = new Date();
120      date.setTime(date.getTime() + (30*24*60*60*1000));//30 days,24hours,60 minutes
121      date = date.toUTCString();
122
123      const updateOrderCount = async() => {await setDoc(doc(db,locate,"orderCount"),{
124        count:orderNum+1,
125        date:date
126      })};
127      updateOrderCount(); //Increment OrderNumber after we got it.
128      cart.unshift({name: name,notes: notes,total:total});//Adding order details to beginning of cart
129      //WORKS but be careful
130      const dataToAdd = async() => {await setDoc(doc(db, locate, orderNum.toString()), {cart}) }; //order is an object array {[]}
131      dataToAdd(); // need to call
132      loadReceipt(orderNum,cart);
133      eraseCookie(); //erase cookie afterward
134    }, 1000);//must wait 1sec to get the OrderNumber
135  }
```

Test Case Table
Test case:

n is an incremented value that increases after parameter is passed

Path	P1	Expected Output	Actual Output
1	1	n+1	n+1

Process Order: ShoppingCart.js

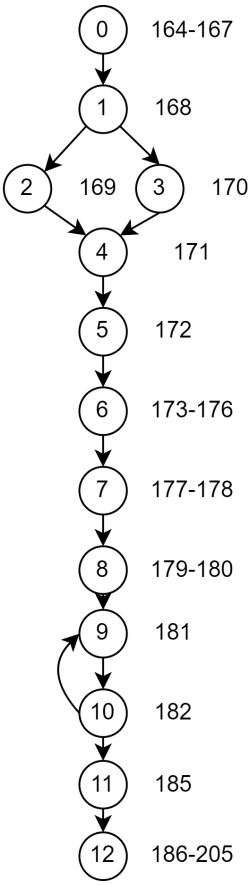
loadReceipt()

Solving cyclomatic complexity:

- V(G) = 13 edges - 11 nodes + 2 = 4

Paths:

- Path 1: 0-1-2-4-5-6-7-8-9-10-11-12
- Path 2: 0-1-3-4-5-6-7-8-9-10-11-12
- Path 3: 0-1-3-4-5-6-7-8-9-10-9-10-11-12
- Path 4: 0-1-2-4-5-6-7-8-9-10-9-10-11-12




```
160  // Receipt
161  //window.onload = function () {
162  function loadReceipt(orderNum,cartt){
163    //document.hidden = false;
164    document.getElementById("receiptDiv").hidden = false;
165    const notes = document.getElementById("notes").value;
166    const name = document.getElementById("name").value;
167    let locat = document.getElementById("locations").value;
168    if (locat == "location1") locat = "2100 Redondo Beach Blvd. #A Torrance, CA 90504";
169    else if (locat == "location2") locat ="4509 Sepulveda Blvd. Torrance, CA 90505";
170    else if (locat == "location2") locat ="29217 S. Western Ave. Rancho Palos Verdes, CA 90275";
171    document.getElementById("rOrderNum").innerHTML = "Order #:" + orderNum;
172    let date = new Date();
173    document.getElementById("rDate").innerHTML = date;
174    document.getElementById("rLocation").innerHTML = locat;
175    document.getElementById("rtaxTotal").innerHTML = taxTotal;
176    document.getElementById("rTotal").innerHTML = total;
177    let cartItems = document.getElementsByClassName("list list-unstyled mb-0 text-left");
178    let htmlItems = "";
179    console.log("THE CART")
180    console.log(cartt);
181    for(let i = 1; i < cartt.length; i++){
182      htmlItems +="<li> - " + cartt[i].title + "</li>";
183
184    }
185    cartItems[1].innerHTML = htmlItems;
186    document.getElementsByClassName("my-2")[0].innerHTML = "Name: " + name;
187    document.getElementById("rLocation").innerHTML = locat;
188
189    document.getElementById("download")
190        .addEventListener("click", () => {
191            //const invoice = this.document.getElementById("invoice");
192            const invoice = document.getElementById("invoice");
193            console.log(invoice);
194            console.log(window);
195            var opt = {
196                margin: 1,
197                filename: 'myfile.pdf',
198                image: { type: 'jpeg', quality: 0.98 },
199                html2canvas: { scale: 2 },
200                jsPDF: { unit: 'in', format: 'letter', orientation: 'portrait' }
201            };
202            html2pdf().from(invoice).set(opt).save();
203        })
204  }
```

Test Case Table
Test case:

Parameters 1 and array of [n] output to n amount of items

Path	P1	P2	expected Output	actual Output
1	1	[n]	n	n
2	1	[n]	n	n
3	1	[n]	n	n
4	1	[n]	n	n

Shopping Cart: Menu.js

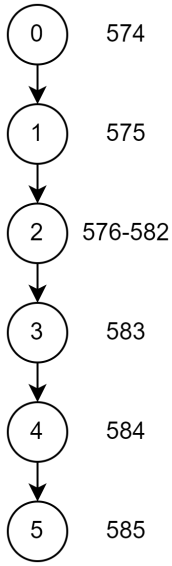
addToCart()

Solving cyclomatic complexity:

- V(G) = 5 edges - 6 nodes + 2 = 1

Paths:

- Path 1: 0-1-2-3-4-5




```
573  function addToCart(itemSelected) {
574    //itemSelected = (menu[i]);
575    let quantitySelected = document.getElementById(itemSelected.id+"quantity").value; // id="${item.id}quantity">
576    console.log(quantitySelected);
577    let itemToAdd = {
578        id: itemSelected.id,
579        title: itemSelected.title,
580        img: itemSelected.img,
581        price: itemSelected.price,
582        quantity: quantitySelected
583    }
584    cart.push(itemToAdd);
585    console.log(cart[0]);
586    setCookie(cart);
587  }
588  const hasListener = (id) =>{
589    document.getElementById(id);
590    return true;
591  };
```

Test Case Table

Test Case: 1 item enters and goes through path to output

Path	P1	Expected Output	Actual Output
1	1	1	1