Firebase: Mini Project Extended -II

Importing Firebase In App

- Use componentDidMount function to import firebase database.
- Import firebase database using firebase.firestore.collection.
- Then we have to methods to fetch data that are:
    1. get() method
    2. Event Listener - onSnapshot()

What is a get() method?

The **GET** method is used to retrieve data from a given server using a given URI. Applications using GET should only return data and should not have any other effect on the data.
This method will retrieve data for us but if there is any change in the database, it will not show it up in the UI as soon as we refresh the page. For this, there comes an event listener called onSnapshot.

What is onSnapshot function?

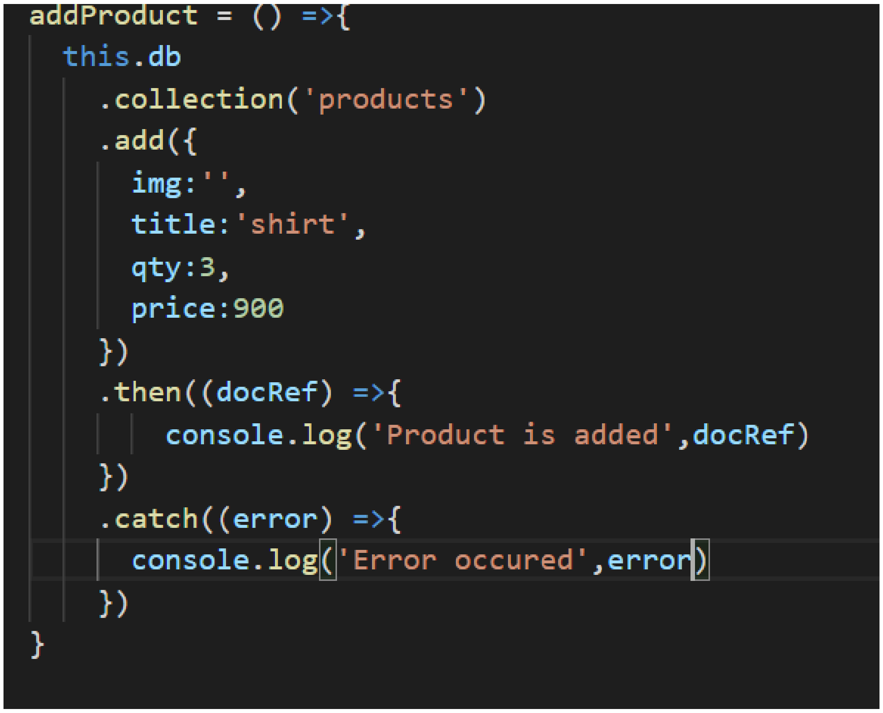
onSnapshot() function is used to fetch data from the firebase document. The initial call using the backup you provided creates a document snapshot with the current contents of the same document. Then, every time the content changes, the second call updates the document snapshot.

Adding product from app to database

For this functionality, we will create an add button that will be linked to an addProduct function where we will call database and will use **.add** function that will contain the object that is to be entered.

.add function: The add function is to add the specified objects with unique values to a database or set objects. The values entered should be **unique.**

Here is the code snippet for your reference:



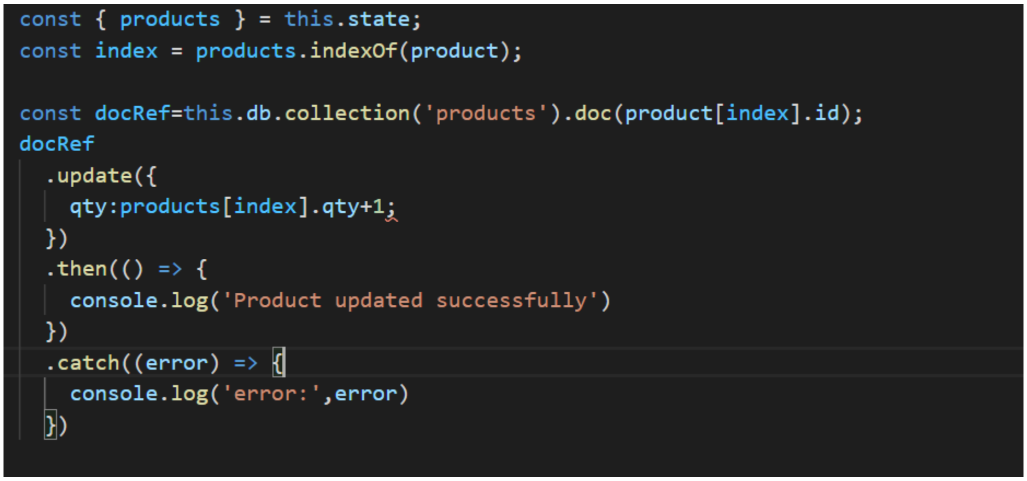
```
addProduct = () =>{
  this.db
    .collection('products')
    .add({
      img:'',
      title:'shirt',
      qty:3,
      price:900
    })
    .then((docRef) =>{
        console.log('Product is added',docRef)
    })
    .catch((error) =>{
      console.log('Error occured',error)
    })
}
```

The add function returns promise to confirm that the product is added. If there is some error then it will be handled by a **catch** statement.

docRef: Document Reference refers to the location of a document in the Firebase database and can be used to write, read or listen to a location. The document may or may not be in the place mentioned. Document reference can also be used to create a collection reference for sub-collection.In simple words, it is a pointer to a specific document in a collection.

Updating Values from App to Firebase

Call the product that is to be updated with its unique product id and then use **.update** function which further returns a promise that the product has been updated. And if by any chance an error occurs then it is handled by a catch statement.
Here is the code snippet for your reference:



```
const { products } = this.state;
const index = products.indexOf(product);

const docRef=this.db.collection('products').doc(product[index].id);
docRef
  .update({
    qty:products[index].qty+1;
  })
  .then(() => {
    console.log('Product updated successfully')
  })
  .catch((error) => {
    console.log('error:',error)
  })
```

Deleting and Querying in Firebase

.delete function: This function uniquely accesses the product by its product id and delete the document for that product in the firebase. Deleting a document doesn’t mean that the whole collection or subcollection has been deleted.
If you want to do so you have to do it manually.

.where function: This function is used under the componentDidMount function where we are fetching the data. This function is to put some restrictions or conditions while rendering the data. For example we want to fetch the products where the **price is 900**
then the **syntax** would be: **.where(‘price’ ‘==’ 900)**

Summarising It

Let’s summarise what we have learnt in this module:

- Learned about get () method and onSnapshot () method.
- Learned about adding, updating and deleting methods in firebase.
- Learned about querying in firebase.

Some References:

1. Cloud Firestore docs
https://firebase.google.com/docs/firestore
2. Queries in Cloud Firestore
https://firebase.google.com/docs/firestore/query-data/queries