Monnig Meteorite Gallery + Native Earth Native Sky

NaMe Interactive Map Learning Tool
Developer Manual

Version 1.0

Revision History

Date	Version	Description	Author
21/10/2022	1.0	Original	Alberto Gaucin

Table of Contents

Developer Manual

1. Manual

This document will provide an overview of how to replicate the functionality in the project. The schema will show up here for the instructor tools and meteorites will show up here, they are not the main focus. To find out more info on the schema, visit the Glossary.

1.1 "API"

Since this project uses Firebase, there is no official API. However, I will walk through how to achieve the functionality of CRUD with the Firebase API.

Create-

```
const addMeteorite = async (meteoriteData) => {


   const db = firestore
   const q = query(collection(db, "Meteorites"));
       try{
           const querySnapshot = await addDoc(q, meteoriteData);
       }catch(err){
              alert(err)
       }


};
```

"Meteorites" is the name of the Collection in Firebase that has all of the Meteorites.

Do the same for instructor tools.

Read - See next section

Update-

```
const editMeteorite = async (e) => {


   e.preventDefault();
   //console.log(new GeoPoint(parseInt(latitude), parseInt(longitude)))
   //let _id = uuid().toString();
   const db = firestore
```

```
   const collectionRef = collection(db, "Meteorites");
   const q = query(collectionRef, where("_id", "==", data._id));
   try{
   const cool = onSnapshot(q, (querySnapshot) => {
   querySnapshot.forEach((doc) =>{

     const coordinates = new GeoPoint(parseFloat(latitude),
parseFloat(longitude))

     updateDoc(doc.ref, {
     description: description,
     name: name,
     nation: nation,
     location: location,
     picture: picture,
     coordinates: coordinates,
     visible: visible,
     type: type,
     clazz : clazz,
     clan: clan,
     group: group
   });


   })


 })
 alert("successfully updated meteorite");
}catch(e){
 alert("something went wrong, try again")
}



 };
```

Do the same for instructor tools, but with its schema

Delete-

```
const deleteMeteorite = async(data) =>{
     const db = firestore
     const collectionRef = collection(db, "Meteorites");
     const q = query(collectionRef, where("_id", "==", data._id));
     const cool = onSnapshot(q, (querySnapshot) => {
     querySnapshot.forEach((doc) =>{
       deleteDoc(doc.ref);
       alert('successfully deleted a meteorite')

    })
  }
  )

 }
```

Do the same for instructor tools.

1.2 Display

To display the content on the main website, just read from the database. Here is an example code snippet:

```
const getMeteorites = async () => {

   const db = firestore

   const dbRef = collection(db, "Meteorites");

   let q;

   let newArray = []

   if(state === 'Oklahoma'){

     q = query(dbRef, where("location", "==", "Oklahoma, USA"));

   }else{

     q = query(dbRef, where("nation", "==", state));

   }

   const querySnapshot = await getDocs(q);
```

```
  querySnapshot.forEach((doc) => {

  let newData = doc.data()

  newArray.push(newData)

}

 );

  setMeteorites(newArray)



  };
```

“Meteorites” is the name of the Collection in Firebase that has all of the Meteorites. After you build the query, then you must wait for the docs to come back from Firebase, where you can read each one by specifying doc.data().

Repeat this for instructor tools.

1.3 Auth

Firebase Auth is very easy

```
const login = (email, password) => {

      return signInWithEmailAndPassword(auth, email, password)

    }
```

2. Deployment

2.1 Backend

The ‘backend’ in our project will be Firebase, which is a BaaS. This is already deployed once you create a project.

2.2 Frontend

The easiest place to deploy a NEXT.js project is Vercel.