EIU
TRƯỜNG ĐẠI HỌC QUỐC TẾ MIỀN ĐÔNG
EASTERN INTERNATIONAL UNIVERSITY

Mobile Programming

Chapter 6: GOOGLE FIREBASE(Cont)

Firebase with react native

Firebase Cloud Firestore

Cloud Firestore is Firebase's new database developed for mobile apps.
It is the successor of Realtime Database with a new and more intuitive data model.
Cloud Firestore is richer, faster, and more scalable than Realtime Database.

Cloud Firestore

Firebase with react native

Cloud Firestore

- Installation

```
npm install @react-native-firebase/app @react-native-firebase/firestore
```

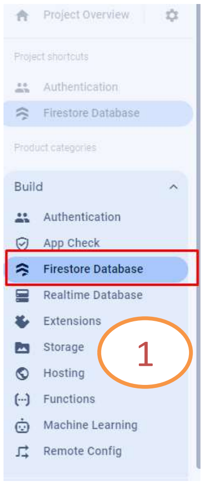
- For IOS:

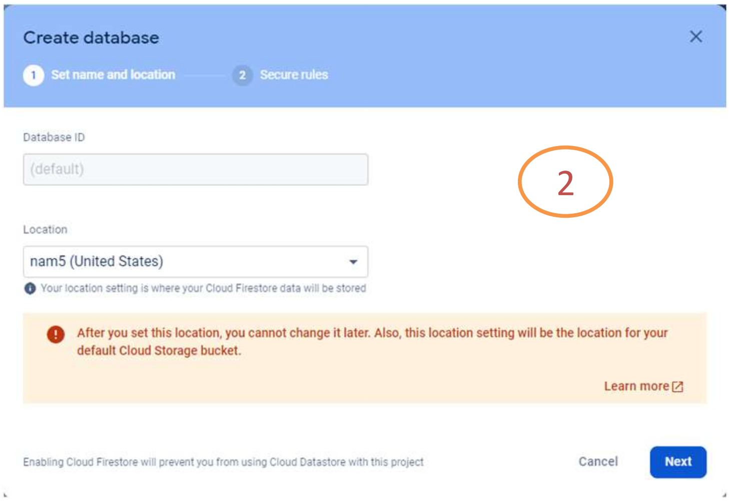
```
cd ios/ && pod install
```

3

Firebase with react native

Create firestore Database





Create database

1 Set name and location — 2 Secure rules

Database ID: (default)

Location: nam5 (United States)

Your location setting is where your Cloud Firestore data will be stored

After you set this location, you cannot change it later. Also, this location setting will be the location for your default Cloud Storage bucket.

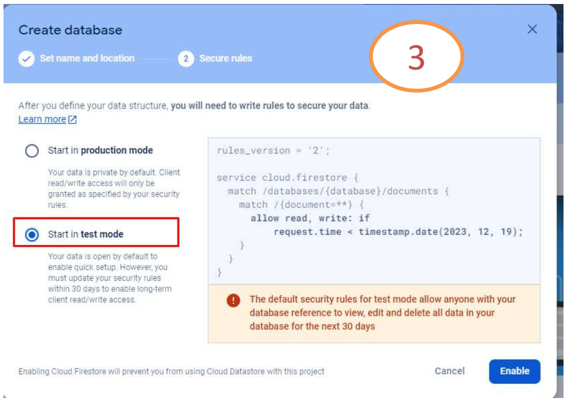
Learn more

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project

4

Firebase with react native

Create firestore Database



After you define your data structure, **you will need to write rules to secure your data.**

○ Start in **production mode**

Your data is private by default. Client read/write access will only be granted as specified by your security rules.

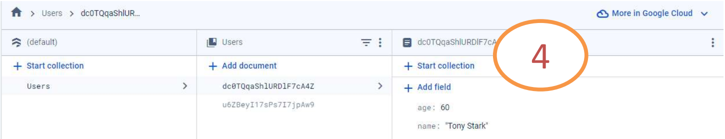
● Start in **test mode**

Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
          request.time < timestamp.date(2023, 12, 19);
    }
  }
}
```

The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project



Users

dc0TQqaShlURDlF7cA4Z

u6ZBeyI17sPs7I7jpAw9

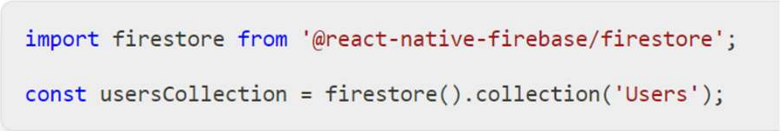
age: 60

name: "Tony Stark"

Firebase with react native

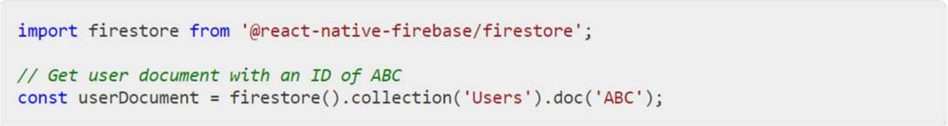
Cloud Firestore

- Collections



```
import firestore from '@react-native-firebase/firestore';

const usersCollection = firestore().collection('Users');
```

- Documents

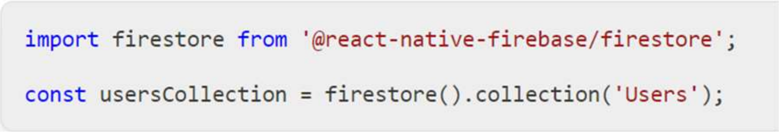


```
import firestore from '@react-native-firebase/firestore';

// Get user document with an ID of ABC
const userDocument = firestore().collection('Users').doc('ABC');
```

Firebase with react native

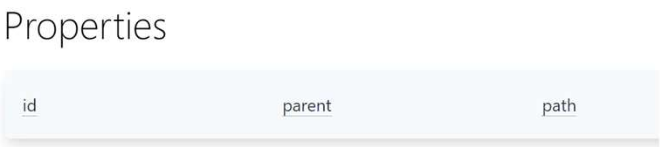
Cloud Firestore

- Collections



```
import firestore from '@react-native-firebase/firestore';

const usersCollection = firestore().collection('Users');
```

- The collection method returns a CollectionReference class.



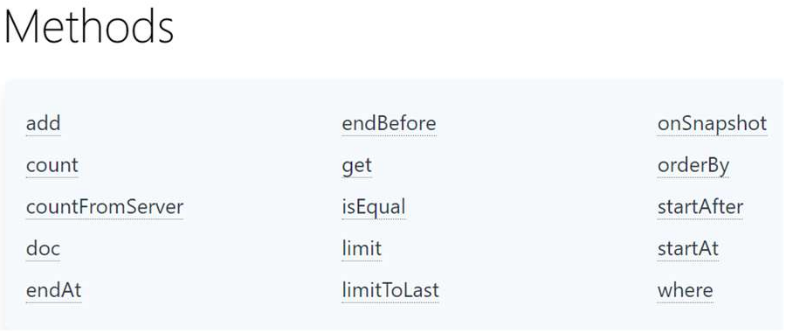
Properties

id	parent	path

Firebase with react native

Cloud Firestore

- The collection method returns a CollectionReference class.



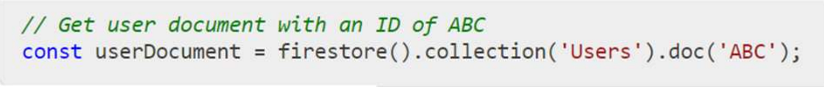
Methods

add	endBefore	onSnapshot
count	get	orderBy
countFromServer	isEqual	startAfter
doc	limit	startAt
endAt	limitToLast	where

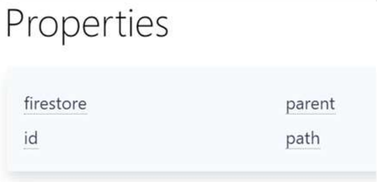
Firebase with react native

Cloud Firestore

- The doc method returns a DocumentReference.

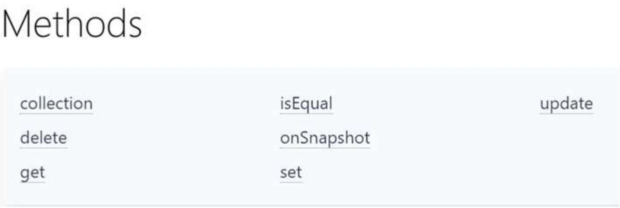


```
// Get user document with an ID of ABC
const userDocument = firestore().collection('Users').doc('ABC');
```



Properties

firestore	parent
id	path



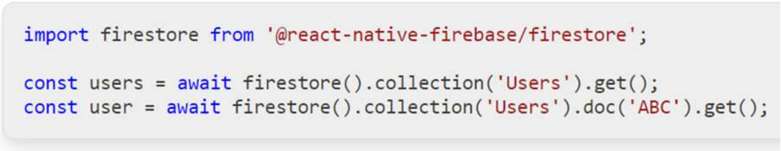
Methods

collection	isEqual	update
delete	onSnapshot	
get	set	

Firebase with react native

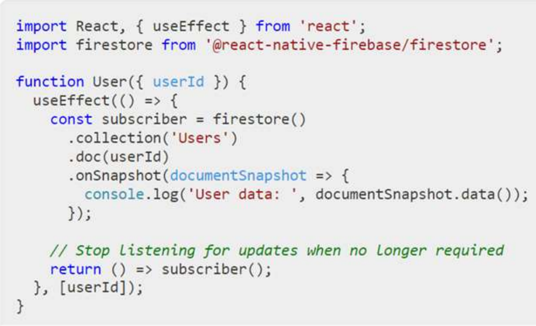
Cloud Firestore

- One-time read



```
import firestore from '@react-native-firebase/firestore';

const users = await firestore().collection('Users').get();
const user = await firestore().collection('Users').doc('ABC').get();
```

- Realtime changes
  - Realtime changes via the **onSnapshot** method

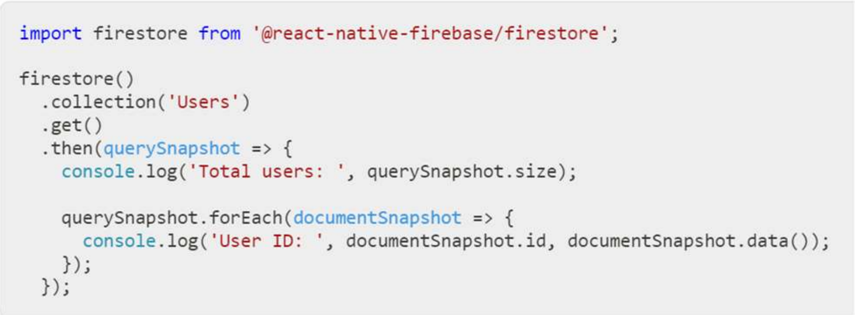


```
import React, { useEffect } from 'react';
import firestore from '@react-native-firebase/firestore';

function User({ userId }) {
  useEffect(() => {
    const subscriber = firestore()
      .collection('Users')
      .doc(userId)
      .onSnapshot(documentSnapshot => {
        console.log('User data: ', documentSnapshot.data());
      });

    // Stop listening for updates when no longer required
    return () => subscriber();
  }, [userId]);
}
```

Firebase with react native

Cloud Firestore

- **QuerySnapshot** is a QueryDocumentSnapshot. which allows you to access specific information about a document



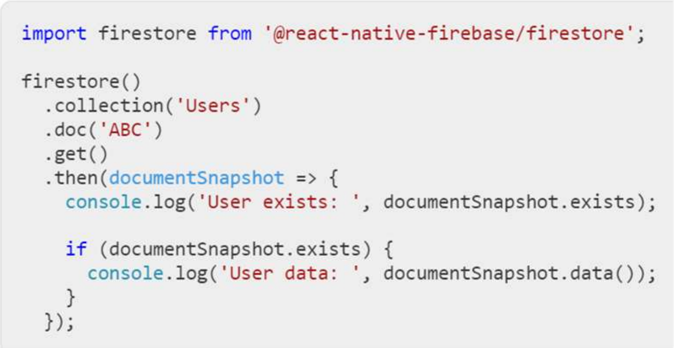
```
import firestore from '@react-native-firebase/firestore';

firestore()
  .collection('Users')
  .get()
  .then(querySnapshot => {
    console.log('Total users: ', querySnapshot.size);

    querySnapshot.forEach(documentSnapshot => {
      console.log('User ID: ', documentSnapshot.id, documentSnapshot.data());
    });
  });
```

11

Firebase with react native

Cloud Firestore

- A DocumentSnapshot is returned from a query to a specific document



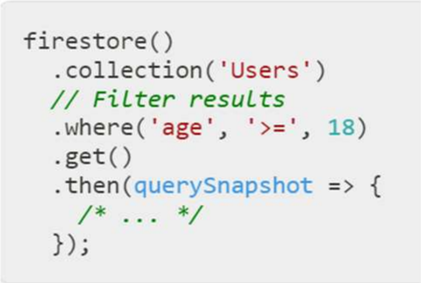
```
import firestore from '@react-native-firebase/firestore';

firestore()
  .collection('Users')
  .doc('ABC')
  .get()
  .then(documentSnapshot => {
    console.log('User exists: ', documentSnapshot.exists);

    if (documentSnapshot.exists) {
      console.log('User data: ', documentSnapshot.data());
    }
  });
```

12

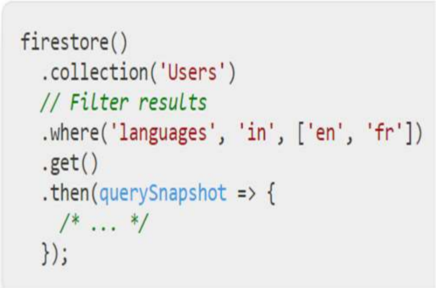
Firebase with react native

Cloud Firestore

- Filtering
  - To filter documents within a collection using where method



```
firestore()
  .collection('Users')
  // Filter results
  .where('age', '>=', 18)
  .get()
  .then(querySnapshot => {
    /* ... */
  });
```



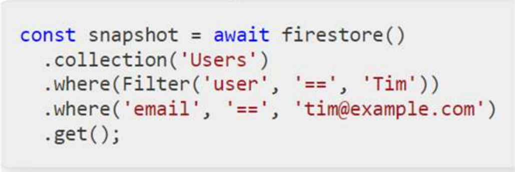
```
firestore()
  .collection('Users')
  // Filter results
  .where('languages', 'in', ['en', 'fr'])
  .get()
  .then(querySnapshot => {
    /* ... */
  });
```

13

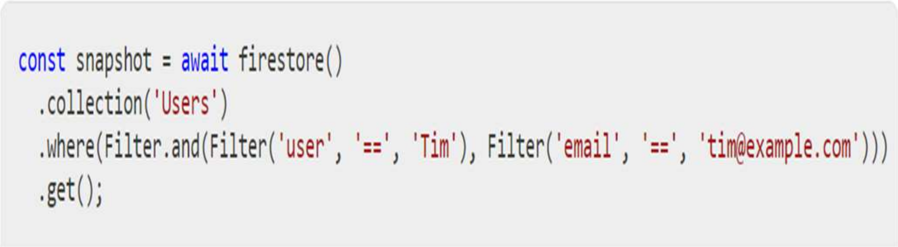
Firebase with react native

Cloud Firestore

- Filter.and()



```
const snapshot = await firestore()
  .collection('Users')
  .where(Filter('user', '==', 'Tim'))
  .where('email', '==', 'tim@example.com')
  .get();
```



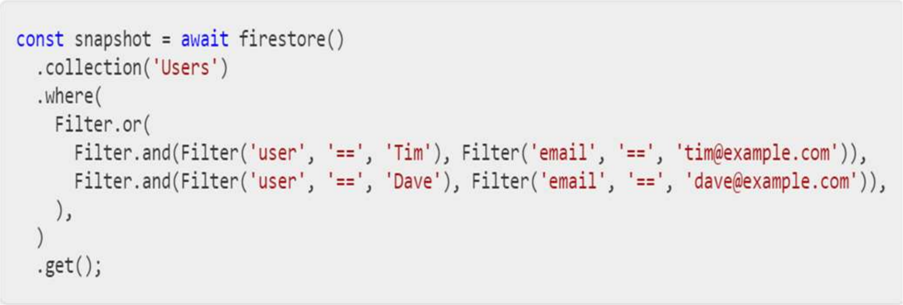
```
const snapshot = await firestore()
  .collection('Users')
  .where(Filter.and(Filter('user', '==', 'Tim'), Filter('email', '==', 'tim@example.com')))
  .get();
```

14

Firebase with react native

Cloud Firestore

- Filter.or()



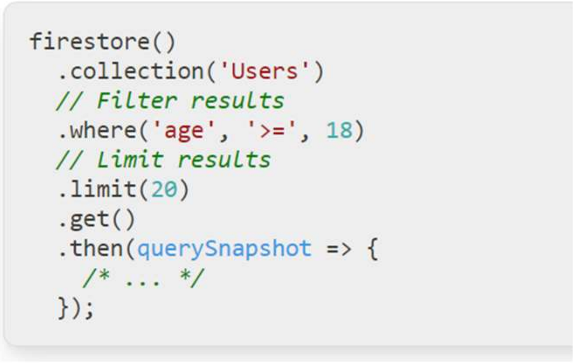
```
const snapshot = await firestore()
  .collection('Users')
  .where(
    Filter.or(
      Filter.and(Filter('user', '==', 'Tim'), Filter('email', '==', 'tim@example.com')),
      Filter.and(Filter('user', '==', 'Dave'), Filter('email', '==', 'dave@example.com')),
    ),
  )
  .get();
```

15

Firebase with react native

Cloud Firestore

- Limiting



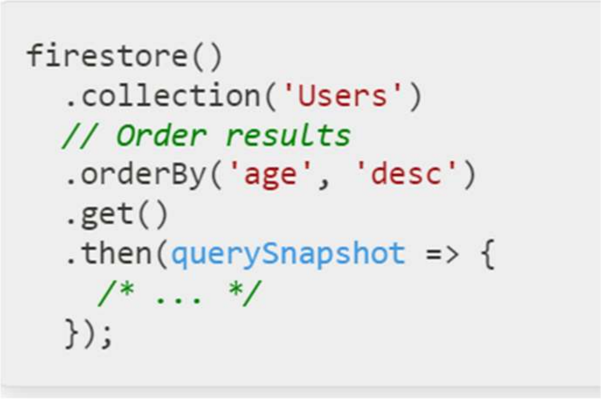
```
firestore()
  .collection('Users')
  // Filter results
  .where('age', '>=', 18)
  // Limit results
  .limit(20)
  .get()
  .then(querySnapshot => {
    /* ... */
  });
```

16

Firebase with react native

Cloud Firestore

- Ordering



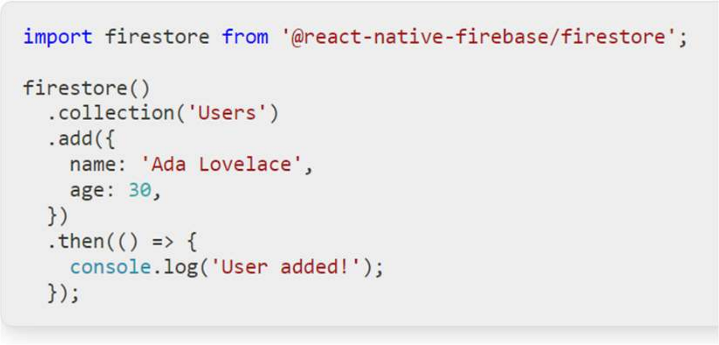
```
firestore()
  .collection('Users')
  // Order results
  .orderBy('age', 'desc')
  .get()
  .then(querySnapshot => {
    /* ... */
  });
```

17

Firebase with react native

Cloud Firestore

- Adding documents



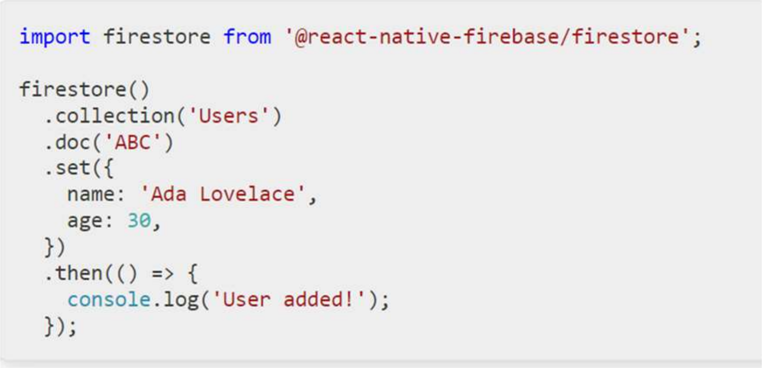
```
import firestore from '@react-native-firebase/firestore';

firestore()
  .collection('Users')
  .add({
    name: 'Ada Lovelace',
    age: 30,
  })
  .then(() => {
    console.log('User added!');
  });
```

18

Firebase with react native

Cloud Firestore

- Adding documents



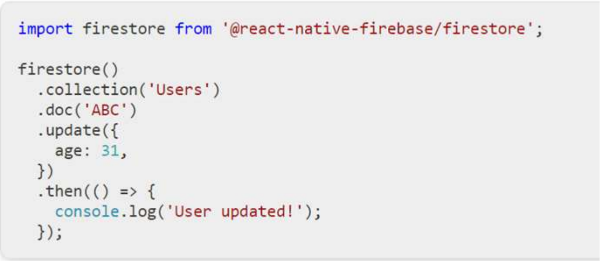
```
import firestore from '@react-native-firebase/firestore';

firestore()
  .collection('Users')
  .doc('ABC')
  .set({
    name: 'Ada Lovelace',
    age: 30,
  })
  .then(() => {
    console.log('User added!');
  });
```

19

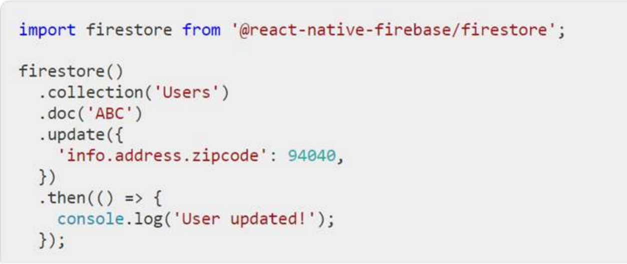
Firebase with react native

Cloud Firestore

- Updating documents



```
import firestore from '@react-native-firebase/firestore';

firestore()
  .collection('Users')
  .doc('ABC')
  .update({
    age: 31,
  })
  .then(() => {
    console.log('User updated!');
  });
```



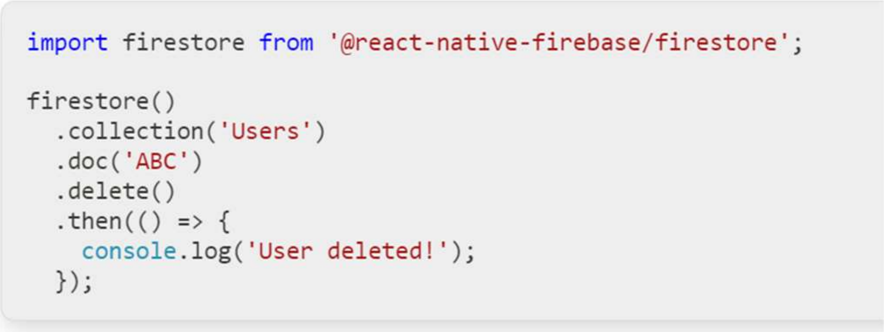
```
import firestore from '@react-native-firebase/firestore';

firestore()
  .collection('Users')
  .doc('ABC')
  .update({
    'info.address.zipcode': 94040,
  })
  .then(() => {
    console.log('User updated!');
  });
```

20

Firebase with react native

Cloud Firestore

- Removing data

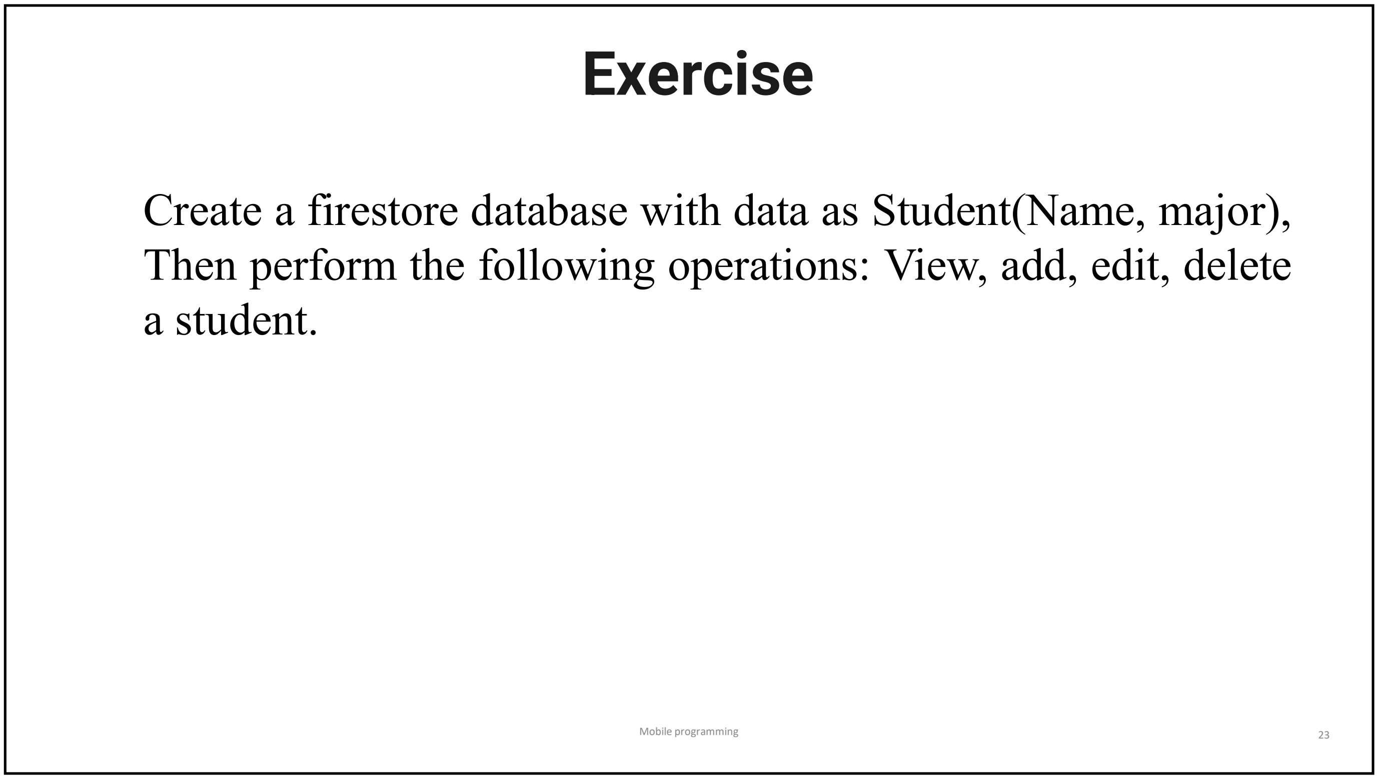


```
import firestore from '@react-native-firebase/firestore';

firestore()
  .collection('Users')
  .doc('ABC')
  .delete()
  .then(() => {
    console.log('User deleted!');
  });
```

Firebase with react native

Demo

https://github.com/taitv86/DemoFireStore



Exercise

Create a firestore database with data as Student(Name, major), Then perform the following operations: View, add, edit, delete a student.

23

Q&A

24