



ICT Department

បញ្ចូល, ទាញយក, កែប្រែ និងលុបចិត្តន័យ ក្នុង Firestore

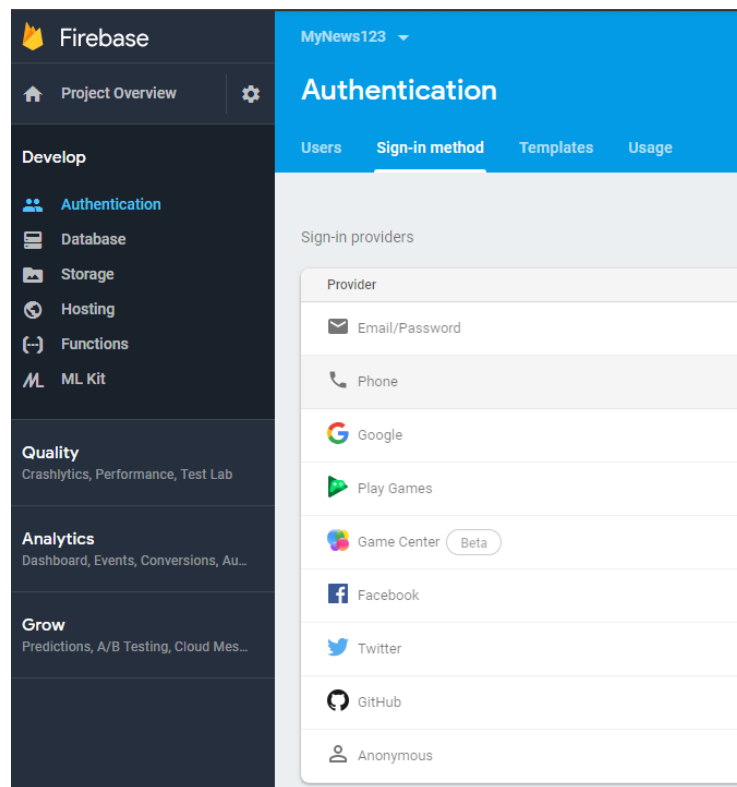
Instructor: Oum Saokosal, *Master of Engineering in Information Systems, South Korea '2010*

Email: oumsaokosal@gmail.com

Phone: 012 252 752 (Telegram)

- Firebase គឺជាផលិតផលរបស់ Google ដែលដំបូងឡើង គឺវាផ្ដោតតែលើ database និង storage ប៉ុណ្ណោះ។ ប៉ុន្តែបន្ទាប់មក វាបានបញ្ចូលនូវ features ល្អៗផ្សេងទៀតដូចជា Authentication, Notification, Hosting, Function, និង Machine Learning។
- ក្នុងមេរៀននេះ យើងនឹងសិក្សាអំពីចំណុចសំខាន់ៗ៣៖

 - Firestore
 - Firebase Authentication
 - Notification



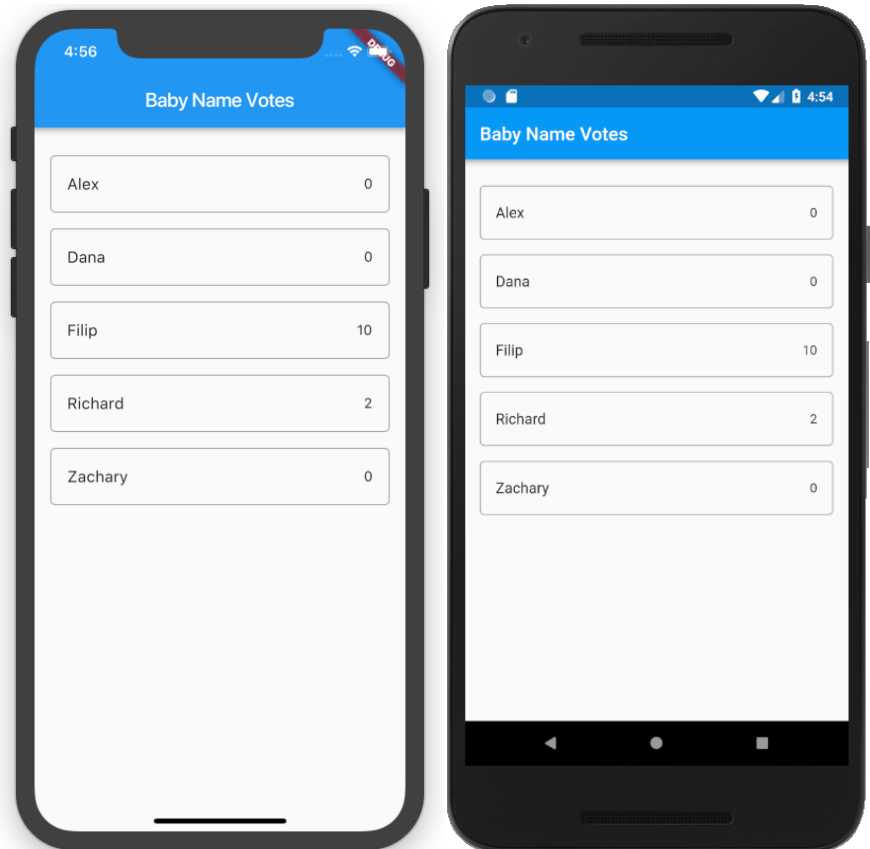
- សំរាប់ការចាប់ផ្តើមដំបូងគឺមិនគិតថ្លៃទេ តែវានឹងគិតថ្លៃ ពេលយើងប្រើលើសកំណត់។

Pricing plans

Start for free,
then pay as you go.

Products	Free Spark Plan <small>Generous limits to get started</small>	Pay as you go Blaze Plan <small>Calculate pricing for apps at scale ✓ Free usage from Spark plan included*</small>
A/B Testing		Free
Analytics		Free
App Distribution		Free
App Indexing		Free
Authentication <small>Phone Auth - US, Canada, and India ?</small>	10k/month	\$0.01/verification

- Firestore គឺជាផលិតផល database ថ្មីជំនួសអោយ realtime firebase database។ Firestore មានចំនុចពិសេសដូចជា៖
 - Real Time Database
 - NoSQL database (JSON)
 - Secure access with Rule
- *slide បន្ទាប់នឹងរៀបរាប់ពី ការ configure Firestore៖*



I. ដាក់ firestore dependency

1. ដាក់ *cloud_firestore* dependency ក្នុង *pubspec.yaml*:

```
dependencies:  
  flutter:  
    sdk: flutter  
  cloud_firestore: ^0.14.0+2
```

អាចឆែកមើល៖ https://pub.dev/packages/cloud_firestore

2. Import firestore ចូលក្នុង project:

```
import 'package:cloud_firestore/cloud_firestore.dart';
```

II. បង្កើត Firestore project


1. ចូលក្នុង <https://console.firebase.google.com/> ហើយចូលប្រើ
តាម Gmail៖
2. រួចហើយ បង្កើត project៖

× Create a project (Step 1 of 3)

Let's start with a name for
your project[?]

Project name

samplebase123

 samplebase123

Continue

3. ចុះឈ្មោះ app ដែលក្នុងនេះវាទាមទារអោយយើងដាក់ឈ្មោះ package អោយដូចជាមួយនឹង package ក្នុង App យើង ([android/app/src/main/AndroidManifest.xml](#))។

× Add Firebase to your Android app

1 Register app

Android package name ⓘ

App nickname (optional) ⓘ

Debug signing certificate SHA-1 (optional) ⓘ

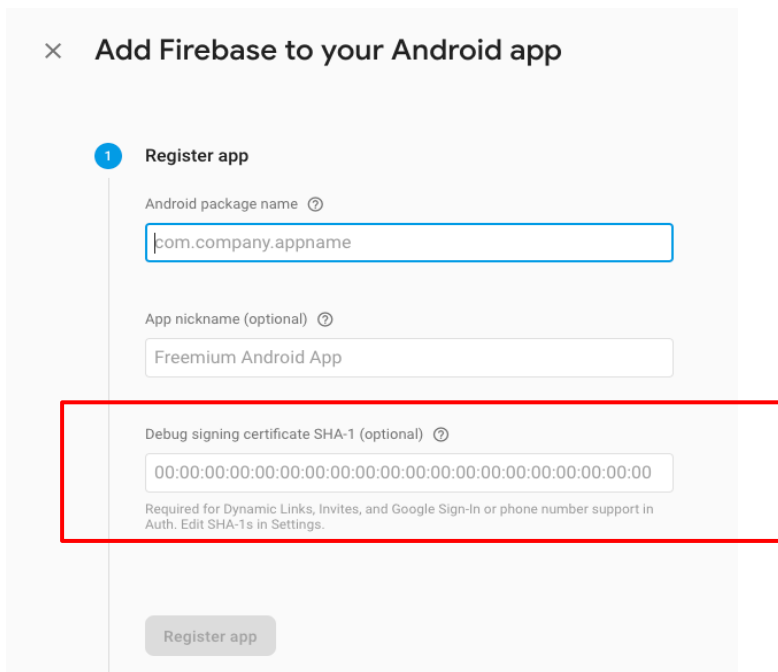
Required for Dynamic Links, Invites, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

Register app

4. ទាញយក SHA-1 ដោយប្រើកូដ៖

```
keytool -list -v -alias androiddebugkey -keystore "%USERPROFILE%\.android\debug.keystore"
```

password: **android**



MAC USER:

```
keytool -list -v -alias androiddebugkey -keystore ~/.android/debug.keystore
```

more details at:

<https://developers.google.com/android/guides/client-auth>

III. បង្កើត Firestore Database

- ទិន្នន័យបញ្ចូលជាលក្ខណៈ JSON។ ដែលដំបូងចាប់ផ្តើមពី Collection បន្ទាប់មក Document និង Field។

Start a collection

✓ Set collection ID

2 Add first document

Document parent path

/baby

Document ID

dana

Field	Type	Value
name	= string	Dana
votes	= number	0

+

Cancel

Save

IV. បង្កើត class សំរាប់តំណាង JSON

- ខាងក្រោមនេះគឺជាកូដនៃការបង្កើត class សំរាប់តំណាងអោយ JSON Object៖

```
import 'package:cloud_firestore/cloud_firestore.dart';

class Record {
  final String name;
  final num votes;
  final DocumentReference reference;
  Record({this.name, this.votes, this.reference});
  Record.fromMap(Map<dynamic, dynamic> map, {this.reference}){
    name = map['name'] ?? "unknown";
    votes = map['votes'] ?? 0;
  }

  Record.fromSnapshot(DocumentSnapshot snapshot)
    : this.fromMap(snapshot.data(), reference: snapshot.reference);

  Map<String, dynamic> toMap() {
    Map<String, dynamic> data = Map<String, dynamic>();
    data['name'] = name;
    data['votes'] = votes;
    return data;
  }

  @override
  String toString() => "Record<$name:$votes>";
}
```

cloud_firestore: ^0.14.0+2

- ចាប់ពី Cloud Firestore version 0.14 គឺមានកែប្រែមួយយ៉ាងធំ គឺមុននឹងយើងអាចប្រើ Firestore បាន គឺយើងត្រូវតែដាក់ក្នុង Firebase អោយវាដើរមុននឹង run app។ គឺដាក់ក្នុង MyApp តែម្តង ក្នុង file main.dart តែម្តង!

```
return FutureBuilder(
  future: Firebase.initializeApp(),
  builder: (context, snapshot) {
    if (snapshot.hasError) {
      return Scaffold(body: Center(child: Text("Error:${snapshot.error}")));
    }

    if (snapshot.connectionState == ConnectionState.done) {
      return MyHome();
    }
    return Center(child: CircularProgressIndicator());
  },
);
```

```
main.dart x
1 import 'package:fbase_aug2020/myhome.dart';|
2 import 'package:firebase_core/firebase_core.dart';
3 import 'package:flutter/material.dart';
4
5 void main() => runApp(MyApp());
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10     return MaterialApp(
11       home: _firebaseInit,
12     );
13   }
14
15   get _firebaseInit {
16     return FutureBuilder(
17       future: Firebase.initializeApp(),
18       builder: (context, snapshot) {
19         if (snapshot.hasError) {
20           return Scaffold(body: Center(child: Text("Error:${snapshot.error}")));
21         }
22
23         if (snapshot.connectionState == ConnectionState.done) {
24           return MyHome();
25         }
26         return Center(child: CircularProgressIndicator());
27       },
28     );
29   }
30 }
```

VI. StreamBuilder សំរាប់ទាញ real time database

StreamBuilder គឺសំរាប់ Stream ទិន្នន័យពី Firestore ជាលក្ខណៈ real time។ មានន័យថាបើយើងដូរទិន្នន័យក្នុង database នោះវាប្តូរក្នុងទូរស័ព្ទយើងដូចគ្នា។

```
StreamBuilder<QuerySnapshot>(
  stream: FirebaseFirestore.instance.collection("baby").snapshots(),
  builder: (index, snapshot) {
    if (snapshot.hasError) {
      return Center(child: Text("Firebase Error: ${snapshot.error}"));
    } else {
      if (snapshot.hasData) return _buildListView(snapshot.data.docs);
      else return Center(child: CircularProgressIndicator());
    }
  },
);
```

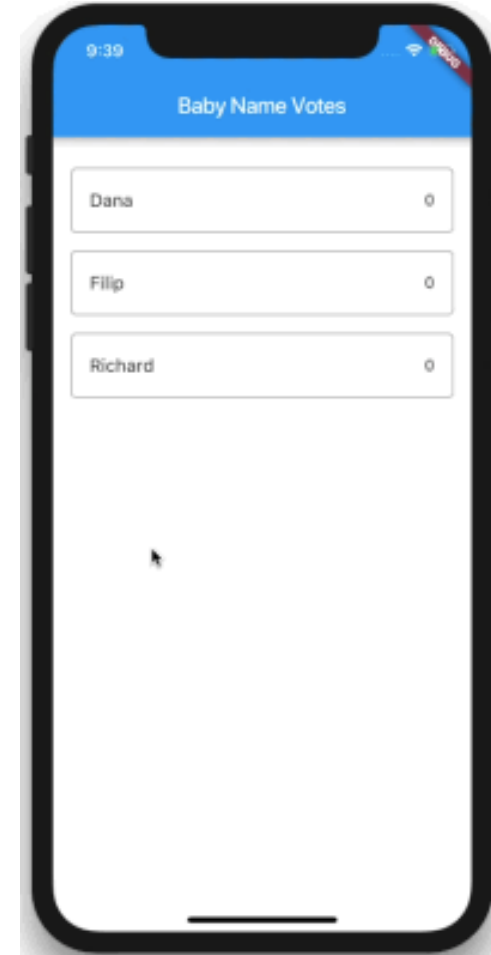
```
_buildListView(List<DocumentSnapshot> documents){
  List<Record> recordList = documents.map((data)=>Record.fromSnapshot(data)).toList();
  return ListView.builder(
    itemCount: recordList.length,
    itemBuilder: (context, index){
      return _buildListViewItem(recordList[index]);
    },
  );
}
```

```
_buildListViewItem(itemAtIndex) => Container(child: itemAtIndex.name);
```

VII. បញ្ជូនទិន្នន័យចូល Firestore

ដើម្បីកែប្រែទិន្នន័យក្នុង Firestore គឺយើងអាចប្រើ Firestore transaction សំរាប់កែប្រែបាន។ ខាងក្រោមនេះគឺជា បន្ថែម ទិន្នន័យចូល collection "baby"។ គួរបញ្ជាក់ផងដែរថា ទិន្នន័យ ដែលបញ្ចូលក៏ត្រូវតែជាទម្រង់ JSON ផងដែរ។

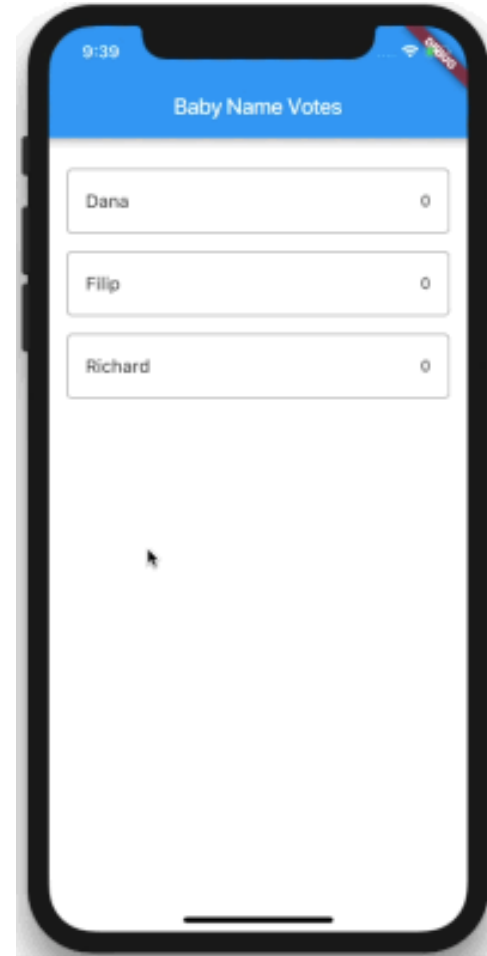
```
insertRecord(Record record){
    FirebaseFirestore.instance.runTransaction((trx) async{
        CollectionReference colRef =
            FirebaseFirestore.instance.collection("baby");
        await colRef.add(record.toMap());
    }).then((x){
        print("data added");
    });
}
```



VIII. កែប្រែទិន្នន័យក្នុង Firestore

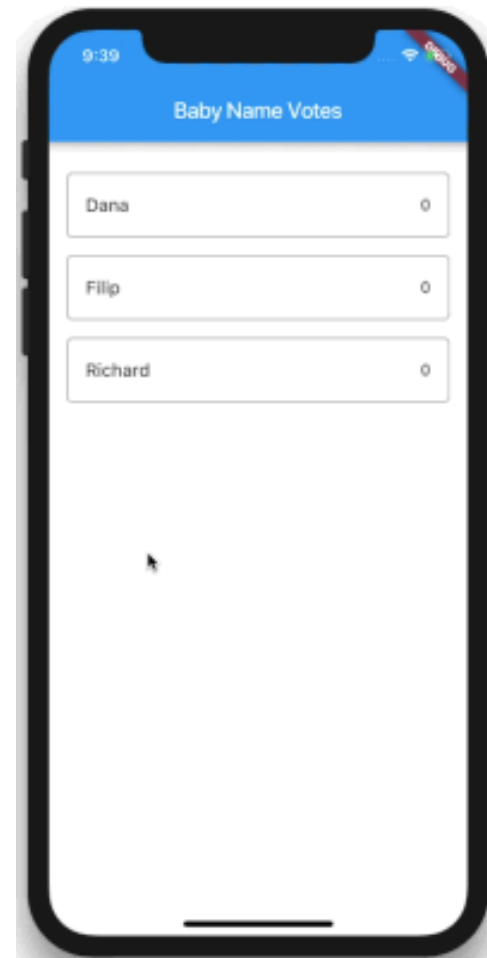
ដើម្បីកែប្រែទិន្នន័យក្នុង field ណាមួយ គឺយើងអាចប្រើ method `update` របស់ `transaction` ហើយបន្ទាប់យើងហៅ field ណាមួយនេះមកកែប្រែបាន។

```
updateRecord(DocumentReference reference, Record record){
    FirebaseFirestore.instance.runTransaction((trx) async{
        await trx.update(reference, record.toMap());
    });
}
```



សំរាប់លុបវិញ គឺងាយគឺគ្រាន់ហៅ method delete របស់ transaction មកប្រើជាការស្រេច។

```
deleteRecord(Record record){
    Firestore.instance.runTransaction((trx) async{
        await trx.delete(record.reference);
    });
}
```



X. តម្រៀបទិន្នន័យ (orderBy)ក្នុង Firestore

- យើងអាចអោយ Firestore តម្រៀបទិន្នន័យអោយយើងបានផងដែរ ដោយប្រើ method orderBy() ហើយដាក់ field ណាមួយអោយវាតម្រៀប។
- យើងក៏អាចអោយវាតម្រៀបពីធំមកតូចបានផងដែរ ដោយប្រើ descending: true។

```
return StreamBuilder<QuerySnapshot>(  
  stream: FirebaseFirestore.instance.collection('baby')  
    .orderBy('name', descending: true)  
    .snapshots(),  
  builder: (context, snapshot) {  
    if (!snapshot.hasData) return LinearProgressIndicator();  
  
    return _buildList(context, snapshot.data.docs);  
  },  
);
```

XI. ស្វែងរកអក្សរក្នុង Firestore

Firestore អនុញ្ញាតិអោយយើងអាចស្វែងរកពាក្យណាមួយបាន ដោយប្រើ method `where()` ដែលយើងអាចកំណត់អោយវារកក្នុង field ណាមួយបាន។

```
return StreamBuilder<QuerySnapshot>(
  stream: FirebaseFirestore.instance.collection('baby')
    .where('name', arrayContains: 'sok')
    .snapshots(),
  builder: (context, snapshot) {
    if (!snapshot.hasData) return CircularProgressIndicator();

    return _buildList(context, snapshot.data.docs);
  },
);
```

