Custom project progress report

COS30017 Software Development for Mobile Devices 2023

Trung Kien Nguyen
104053642

Table of Contents

Overview of project

My project is a cryptocurrency app using Android Development with Kotlin. Like many typical apps in the market, such as CoinGecko, Coinbase, Kraken, etc., my app has some basic functionalities as follows:

- User authentication: The app allows users to create accounts, log in with them, or reset their password in case of forgetting.
- Wallet managing: It also creates and manages cryptocurrency wallets for different cryptocurrencies, e.g. Bitcoin, Etherium, ….
- Viewing transaction history
- Buy and sell cryptocurrencies as will.

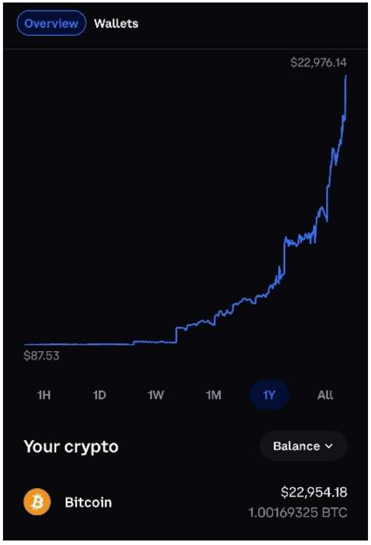
Weekly reports

Week 7

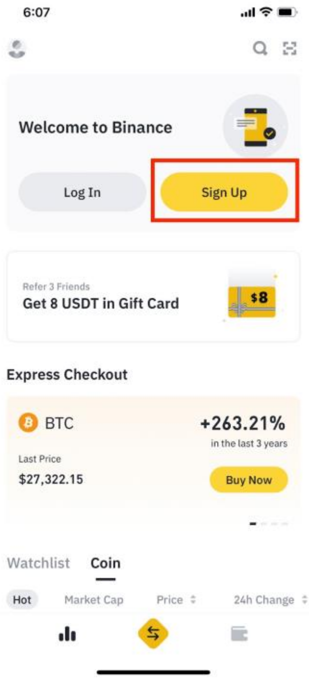
Note: I have completed Core 1, submitted Core 2, but not completed the discussion tasks. However, I decided to continue with the custom project as the resubmission for those discussions would not be available until 6 Oct 2023 (end of week 9)

In this step, I focused mainly in searching and referring to the cryptocurrency applications on the market, especially those available on the Android platform:

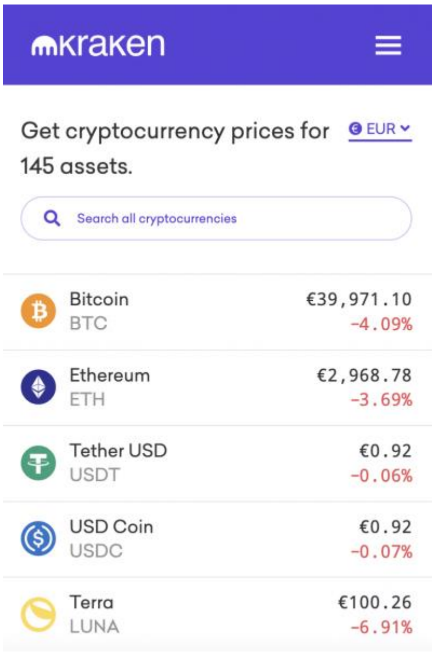
- Coinbase: In my opinion, this app has an user-friendly interface, ideal for beginners. It offers a secure and regulated platform for buying, selling, and managing cryptocurrencies.



- Binance: This offers not only a wide range of cryptocurrencies for trading, but also additional features like advanced charting tools. It's one of the most popular app among experienced traders.



- Kraken: I firmly believe that this app is well-known for its strong security features, providing access to a variety of cryptocurrencies and trading pairs.



Get cryptocurrency prices for 145 assets.

Search all cryptocurrencies

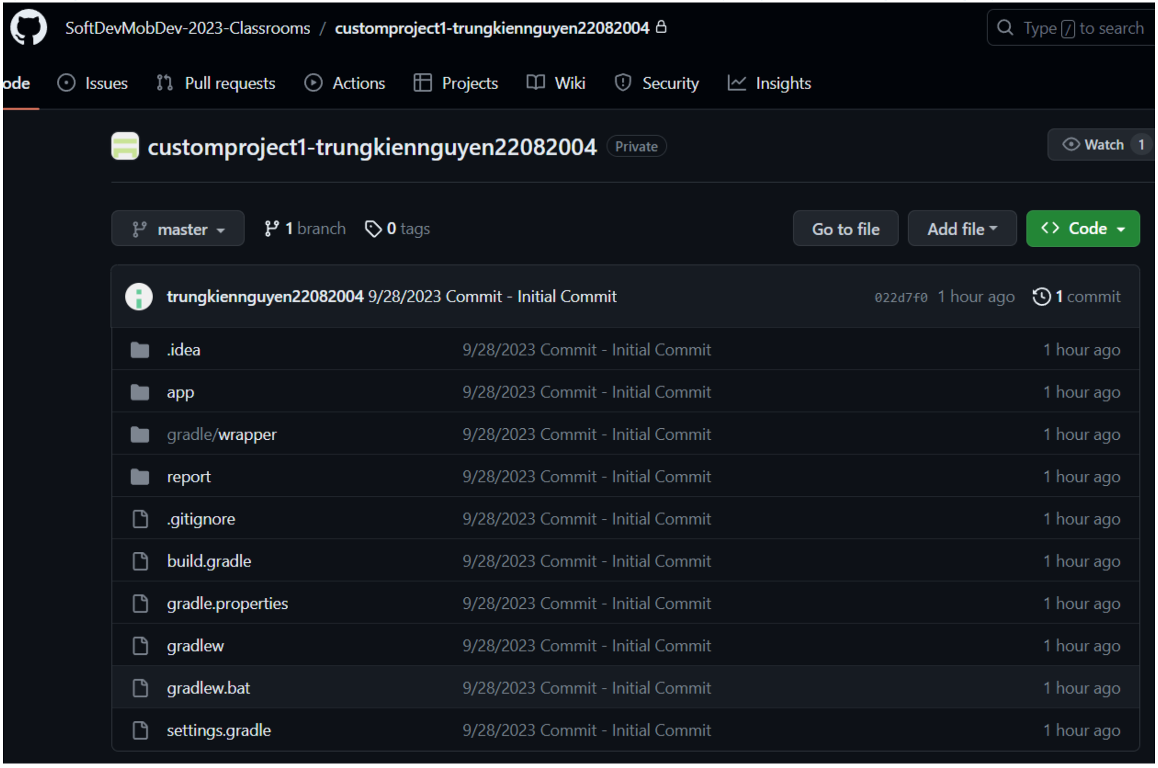
Bitcoin BTC	€39,971.10 -4.09%
Ethereum ETH	€2,968.78 -3.69%
Tether USD USDT	€0.92 -0.06%
USD Coin USDC	€0.92 -0.07%
Terra LUNA	€100.26 -6.91%

In the following week (Week 8), I think I will try to complete the Authentication feature of the app (Login/Signup). I prefer to use the Google Firebase's authentication functionality.

Week 8

Note: I have completed Core 1, submitted Core 2, but not completed the discussion tasks. However, I decided to continue with the custom project as the resubmission for those discussions would not be available until 6 Oct 2023 (end of week 9)

To begin with, I have set up the repository for the Custom Project via the Github Classroom link on Canvas:
https://github.com/SoftDevMobDev-2023-Classrooms/customproject1-trungkiennguyen22082004

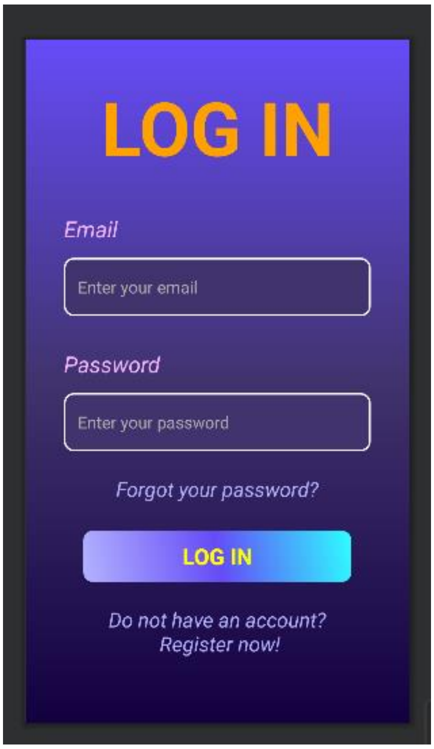


customproject1-trungkiennguyen22082004

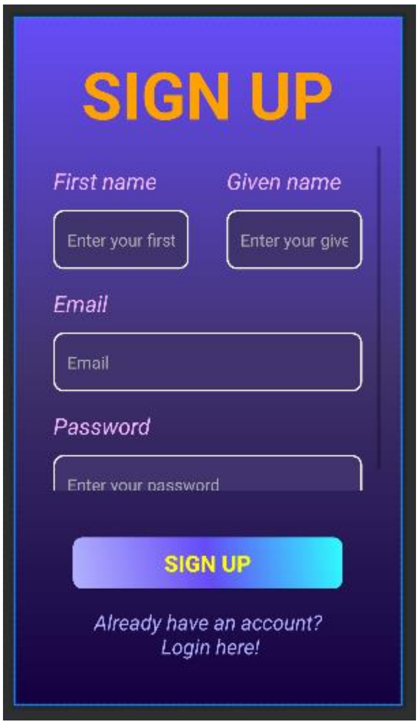
trungkiennguyen22082004 9/28/2023 Commit - Initial Commit 022d7f0 1 hour ago 1 commit

.idea	9/28/2023 Commit - Initial Commit	1 hour ago
app	9/28/2023 Commit - Initial Commit	1 hour ago
gradle/wrapper	9/28/2023 Commit - Initial Commit	1 hour ago
report	9/28/2023 Commit - Initial Commit	1 hour ago
.gitignore	9/28/2023 Commit - Initial Commit	1 hour ago
build.gradle	9/28/2023 Commit - Initial Commit	1 hour ago
gradle.properties	9/28/2023 Commit - Initial Commit	1 hour ago
gradlew	9/28/2023 Commit - Initial Commit	1 hour ago
gradlew.bat	9/28/2023 Commit - Initial Commit	1 hour ago
settings.gradle	9/28/2023 Commit - Initial Commit	1 hour ago

For the Authentication task, I have implemented two activities, including LoginActivity and SignupActivity.



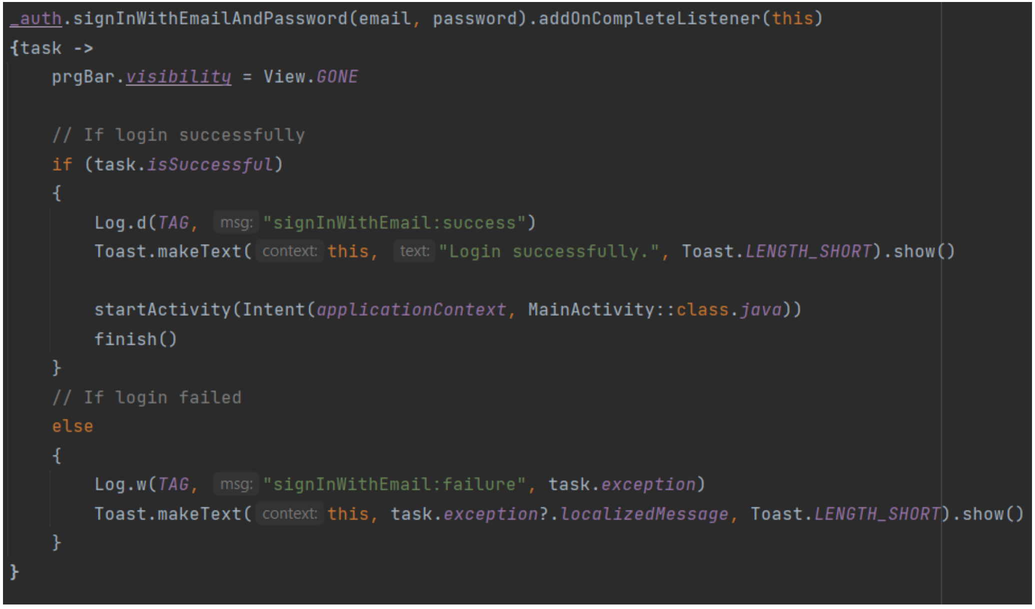
LoginActivity's layout



SignupActivity's layout

I have use the Google Firebase's functionality of Authentication using Email/Password:

- Login: Using the method “signInWithEmailAndPassword()” of FirebaseAuth

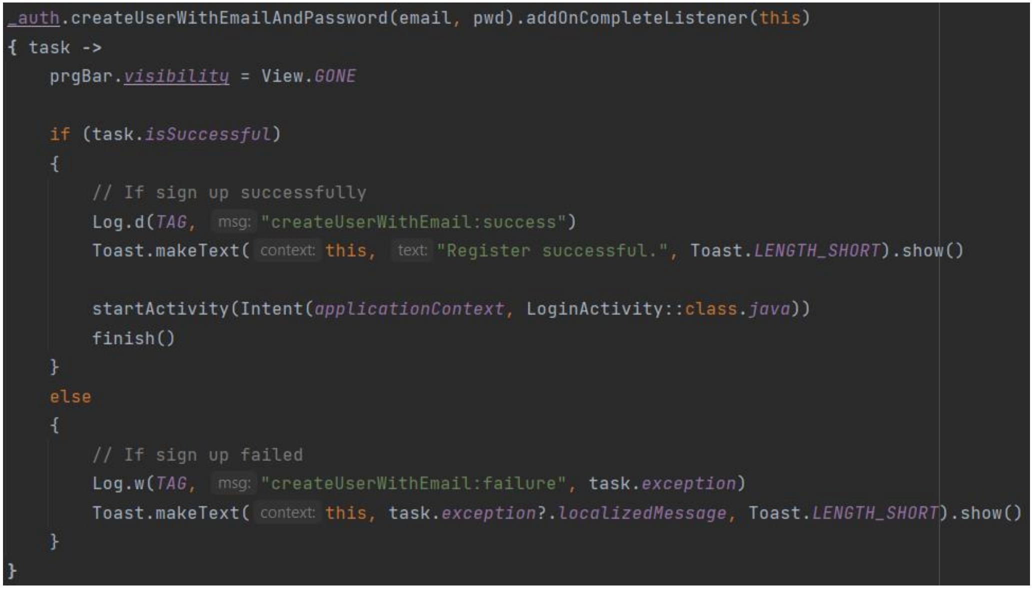


```
_auth.signInWithEmailAndPassword(email, password).addOnCompleteListener(this)
{task ->
    prgBar.visibility = View.GONE

    // If login successfully
    if (task.isSuccessful)
    {
        Log.d(TAG,  msg: "signInWithEmail:success")
        Toast.makeText( context: this,  text: "Login successfully.", Toast.LENGTH_SHORT).show()

        startActivity(Intent(applicationContext, MainActivity::class.java))
        finish()
    }
    // If login failed
    else
    {
        Log.w(TAG,  msg: "signInWithEmail:failure", task.exception)
        Toast.makeText( context: this, task.exception?.localizedMessage, Toast.LENGTH_SHORT).show()
    }
}
```

- Signup: Using the method “createUserWithEmailAndPassword()” of FirebaseAuth

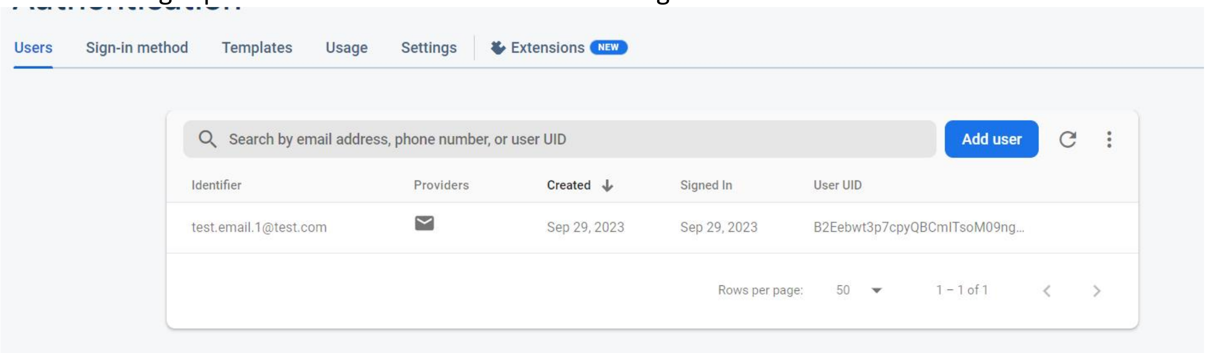


```
_auth.createUserWithEmailAndPassword(email, pwd).addOnCompleteListener(this)
{ task ->
    prgBar.visibility = View.GONE

    if (task.isSuccessful)
    {
        // If sign up successfully
        Log.d(TAG,  msg: "createUserWithEmail:success")
        Toast.makeText( context: this,  text: "Register successful.", Toast.LENGTH_SHORT).show()

        startActivity(Intent(applicationContext, LoginActivity::class.java))
        finish()
    }
    else
    {
        // If sign up failed
        Log.w(TAG,  msg: "createUserWithEmail:failure", task.exception)
        Toast.makeText( context: this, task.exception?.localizedMessage, Toast.LENGTH_SHORT).show()
    }
}
```

I have use “Signup” to create some new users for testing:



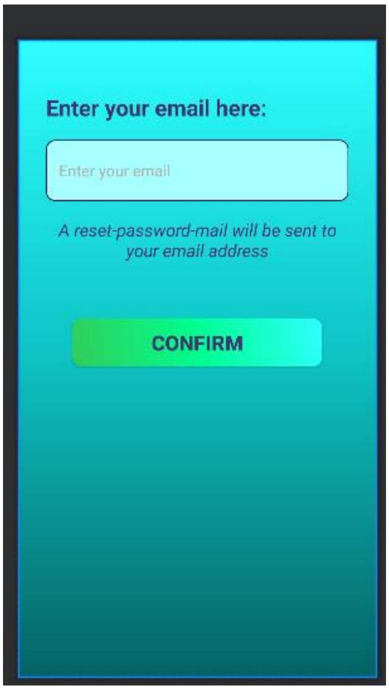
Identifier	Providers	Created ↓	Signed In	User UID
test.email.1@test.com	✉	Sep 29, 2023	Sep 29, 2023	B2Eebwt3p7cpyQBCmlTsoM09ng...

Rows per page: 50 1 – 1 of 1

Week 9

Note: I have completed Core 1, completed Core 2, I have just submitted the Redo for discussion tasks since they had been available in the Friday morning.

Firstly, I continued my work on the Authentication functionality of my custom app, adding the forgot-password and reset-password feature:



Enter your email here:

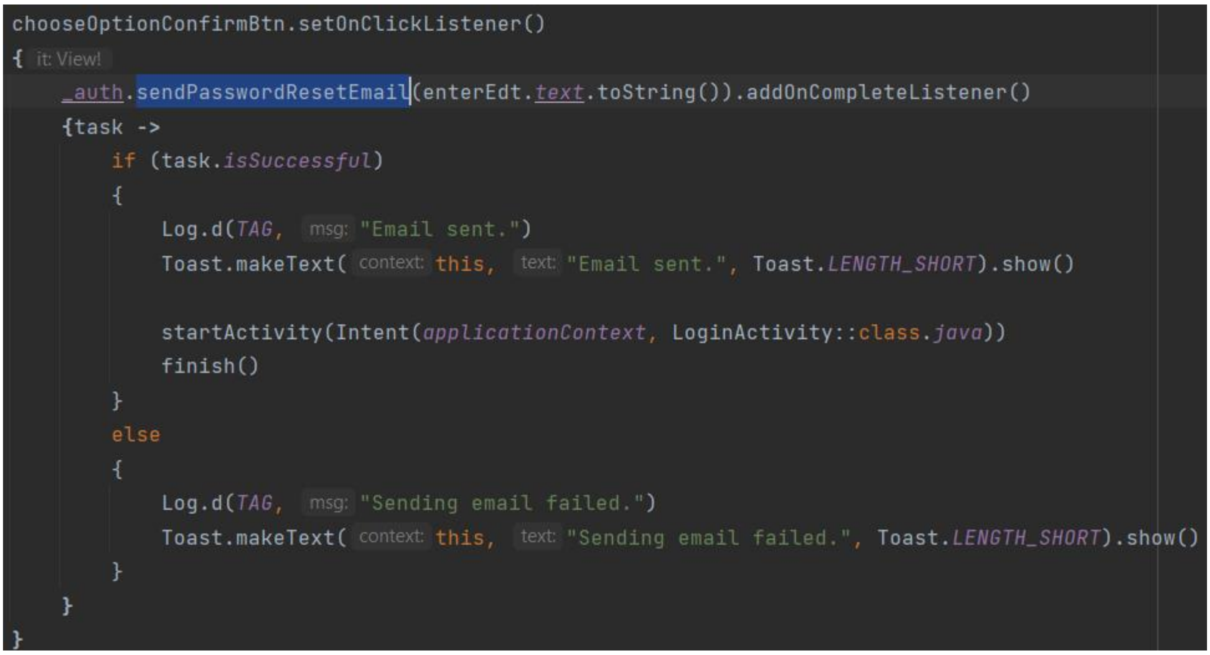
Enter your email

A reset-password-mail will be sent to your email address

CONFIRM

ForgotPasswordActivity's layout

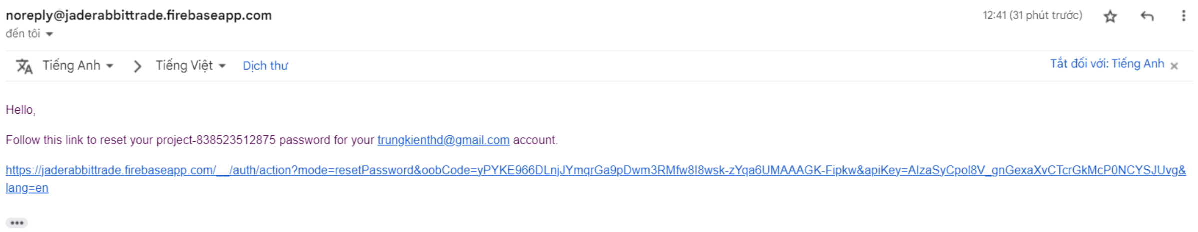
I have used the "sendPasswordResetEmal()" method to sent a resetting-password email to the entered email address:



```
chooseOptionConfirmBtn.setOnClickListener()
{ it: View!
    _auth.sendPasswordResetEmail(enterEdt.text.toString()).addOnCompleteListener()
    {task ->
        if (task.isSuccessful)
        {
            Log.d(TAG, msg: "Email sent.")
            Toast.makeText( context: this, text: "Email sent.", Toast.LENGTH_SHORT).show()

            startActivity(Intent(applicationContext, LoginActivity::class.java))
            finish()
        }
        else
        {
            Log.d(TAG, msg: "Sending email failed.")
            Toast.makeText( context: this, text: "Sending email failed.", Toast.LENGTH_SHORT).show()
        }
    }
}
```

I have tested that functionality, the resetting-password email will look like:



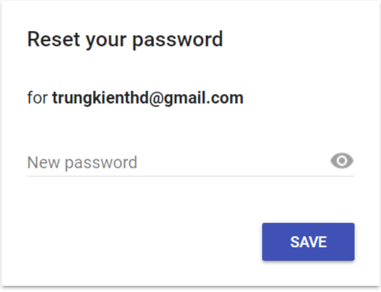
noreply@jaderabbittrade.firebaseapp.com
đến tôi

Hello,

Follow this link to reset your project-838523512875 password for your trungkienthd@gmail.com account.

https://jaderabbittrade.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=yPYKE966DLnjJYmqrGa9pDwm3RMfw8l8wsk-zYqa6UMAAAGK-Fipkw&apiKey=AIzaSyCpol8V_gnGexaXvCTcrGkMcP0NCYSJUvg&lang=en

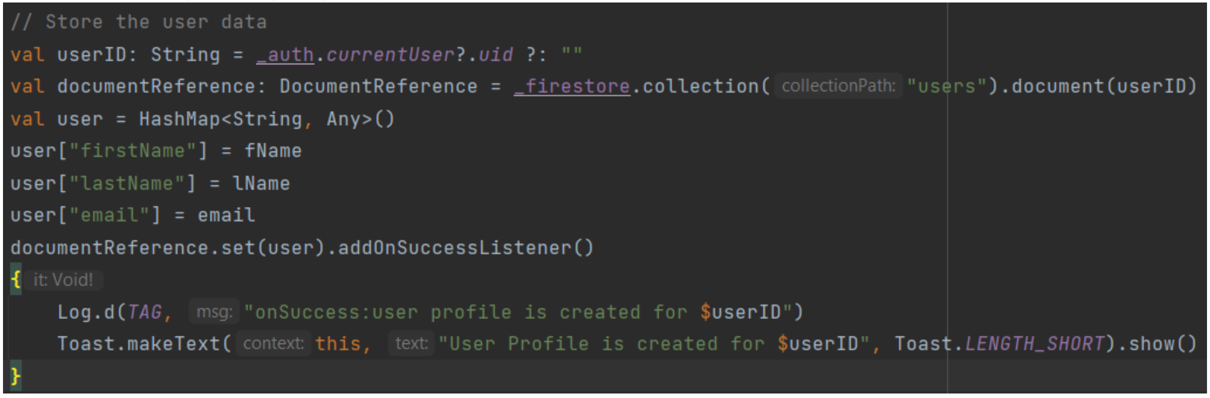
And this is the resetting-password interface:



Reset your password

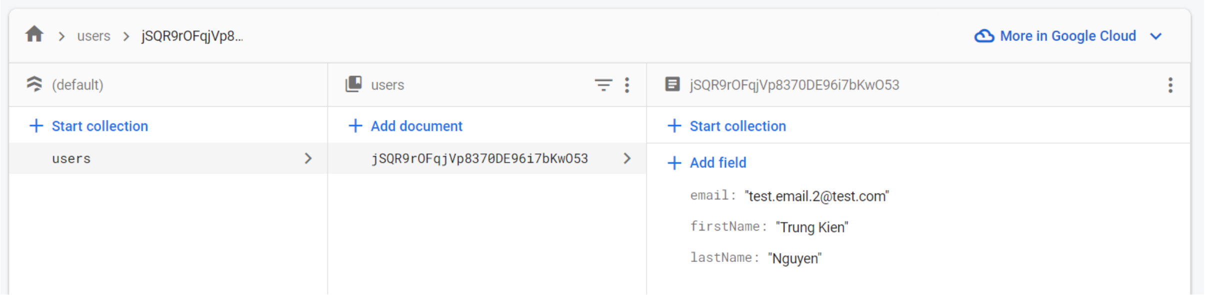
for **trungkienthd@gmail.com**

Next, I stored the User data (First Name, Last Name, …) in Firebase's feature of Cloud Firestore
After create a Cloud Firestore Database, I have added the "Firestore.DocumentReference.set(user)" method to store the First name, last name and email of the user to the Firebase Cloud Firestore's database during the signing up process:



```
// Store the user data
val userID: String = _auth.currentUser?.uid ?: ""
val documentReference: DocumentReference = _firestore.collection( collectionPath: "users").document(userID)
val user = HashMap<String, Any>()
user["firstName"] = fName
user["lastName"] = lName
user["email"] = email
documentReference.set(user).addOnSuccessListener()
{ it: Void!
    Log.d(TAG,  msg: "onSuccess:user profile is created for $userID")
    Toast.makeText( context: this,  text: "User Profile is created for $userID", Toast.LENGTH_SHORT).show()
}
```

I have created a new account for testing this function, and this is the output:



email: "test.email.2@test.com"
firstName: "Trung Kien"
lastName: "Nguyen"

In the following weeks, I will continue to make the Market activity (for displaying digital assets of cryptocurrency), Transaction activity (for displaying history transaction of the user), and probably the User Activity (for viewing and modifying the user's details)

Week 10

[NOTE: as of week 10, if you have not completed Core 1, not completed Core, not submitted Core 3 nor an extension task, you will also need to justify why you should be encouraged to continue with a

custom project. There is no point focusing on this task when the basics are not complete and your progress report will be marked as incomplete.]

Week 11

[NOTE: as of week 11, if you have not submitted/completed all other Core/Extension tasks, you will also need to justify why you should be encouraged to continue with a custom project. There is no point focusing on this task when the basics are not complete and your progress report will be marked as incomplete.]

Level 1: Design evidence

Level 2: App evidence

Level 3: Extended research evidence

References