wonderCE881 Assignment 2: Cryptocurrency Converter App Report

Android Application for Cryptocurrency Conversion

Abstract

Cryptocurrencies are virtual currencies that are based on the internet and do not have any governing body. They are introduced into the internet ecosystem and are generally used outside the confines of the established banking system. They are utilised for online value exchange and transfer. Bitcoin and other cryptocurrencies have evolved from being solely associated with geeks and radicals to being taken into consideration by central banks as a method to adopt digital money. Cryptocurrencies can be completely transferred between digital addresses and are only available in digital form.

INTRODUCTION

Everyone knows Cryptocurrencies have huge potential now as well as in the future with over 320 million crypto users worldwide. And most of the crypto holders trade their currency to make profit. And even more, a lot of people's sources of income is coming only from crypto trading. So, the one who just wants to convert the rate of one crypto currency with another is such a difficult task inside the trading platform app and we can't instantly see the trading fees of those platforms and we can't compare it in one single application.

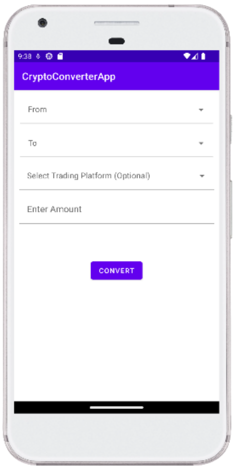
Cryptocurrency Converter addresses the application of conversion of the real time rate of one crypto currency with another. And even more, it shows the trade charges and the sum amount if we trade crypto with some recognized platforms such as Binance, Coinbase etc. Since it is very simple and with only one feature, the size of this app will also minimum, and the main advantage of this is it can seamlessly run in any lower end device, and it consume less computational power. The main problem with this kind of app are people stick with it if there are no ads and distraction. So, I will make this app free to download and free to use for all users. The main business model I will try to implement in this project is to be affiliated marketer for lot of international crypto currency exchange platforms.

An app that converts currencies is primarily designed to provide users with quick access to accurate and current exchange rates. These mobile applications support accurate real-time results while on the go and assist company owners and vacationers in tracking fluctuating currency exchange rates. A currency converter app can offer a variety of useful functions in addition to monitoring exchange rates and enabling real-time multi-currency conversion.

Finding the most recent exchange rates for hundreds of currencies is made easier with the use of a currency converter app. In a foreign exchange market, currency exchange rates are constantly changing. Business owners may evaluate the exchange rates of various currencies and forecast their future indices with the aid of a currency converter Application.

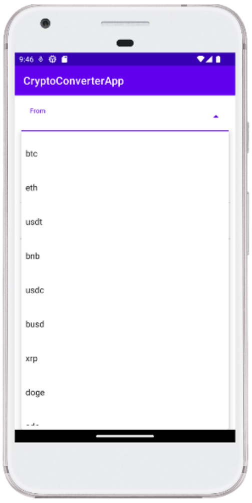
DESIGN AND FEATURES

The design o the application is very simple. All layouts designed in single screen, and all features present in single screen itself. I tried to make this app very user friendly because users like to use it when all features present on single screen and no more navigations required. The main layout of the screen is shown below.



Main Layout

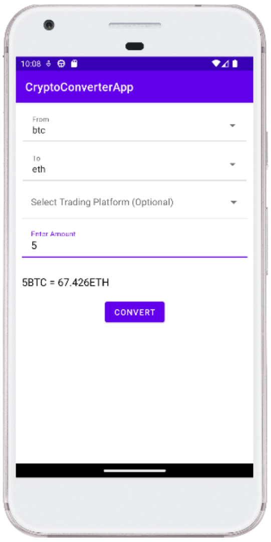
There are five input functions, firstly talking about the selecting the first currency from which you want to convert. There are lot of currencies to select, but the main problem with this application is, we need to know the shortform of those currencies, since all title is retrieved from the API from CoinGecko. For instance, the famous currency i.e., Bitcoin has its short code as BTC, Ethereum called ETH, and same for all those currencies.



Selecting the from currency

Next, the second selection i.e., selecting the currency that you want to convert. And the option are exact for this as for from currencies. We can select the same currency; this app doesn't show any error with that input. It will just show the equal values on both sides with same currency name mentioned at the end.

The third main option is to enter the amount of from currency to get started conversion. And soon after we hit the convert button it calculates and convert that value with the from currency and shows the output just below that function.



From
btc

To
eth

Select Trading Platform (Optional)

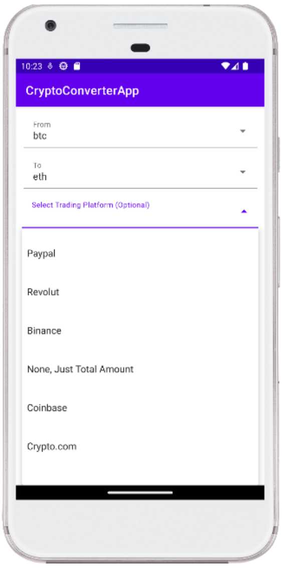
Enter Amount
5

5BTC = 67.426ETH

Conversion from Bitcoin to Ethereum

From above screen, we can see I converted Bitcoin to Ethereum. And in the amount input I put the value 5. That means I am calculating how much Ethereum will be equal to five bitcoins. As it is fetched from the API it shows the updated value. As of now as we can see five Bitcoins are equals to 67.426 Ethereum. This is the main function of the app to convert from one currency to another.

Next one is optional, i.e., my additional feature first talking about the options available to input. As of now I only put some trusted and popular trading platforms to this application, but in future it is easily updatable. Platforms like PayPal, Binance, Coinbase which has different percentage of maker fees.



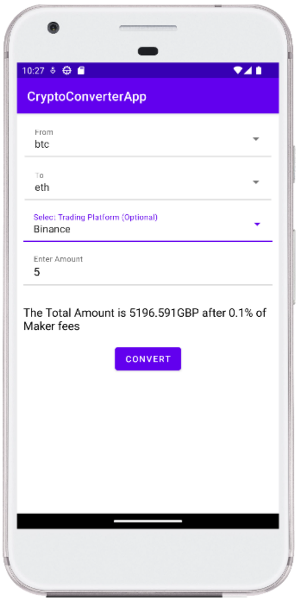
From
btc

To
eth

Select Trading Platform (Optional)

Options to select trading platform

So, the second function of this app to calculate total amount needed to be paid by user if he wanted to trade with selected platform, which include the current price of the currency with some percentage of the maker charges.



From
btc

To
eth

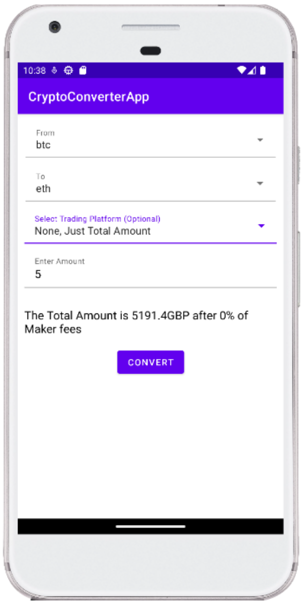
Select Trading Platform (Optional)
Binance

Enter Amount
5

The Total Amount is 5196.591GBP after 0.1% of Maker fees

In this case I selected the same currency as we converted previously. But we can select any of this to calculate the total amount. Here I selected the platform Binance which takes 0.1% of makers fees per trade. So, after we put the value, it simply multiplies with that value with current price of the currency and adds the percentage to calculate the total amount in GBP needed per trade. As we can see above it is showing the output of total amount of around 5196.5 GBP and displays the percentage of makers fees of the selected platform.

There is one more option in that selection and i.e., "None" this is I included because if we want just to know the total amount without any maker fees. This will just multiply the number of currencies with their current price and gives the output.



From
btc

To
eth

Select Trading Platform (Optional)
None, Just Total Amount

Enter Amount
5

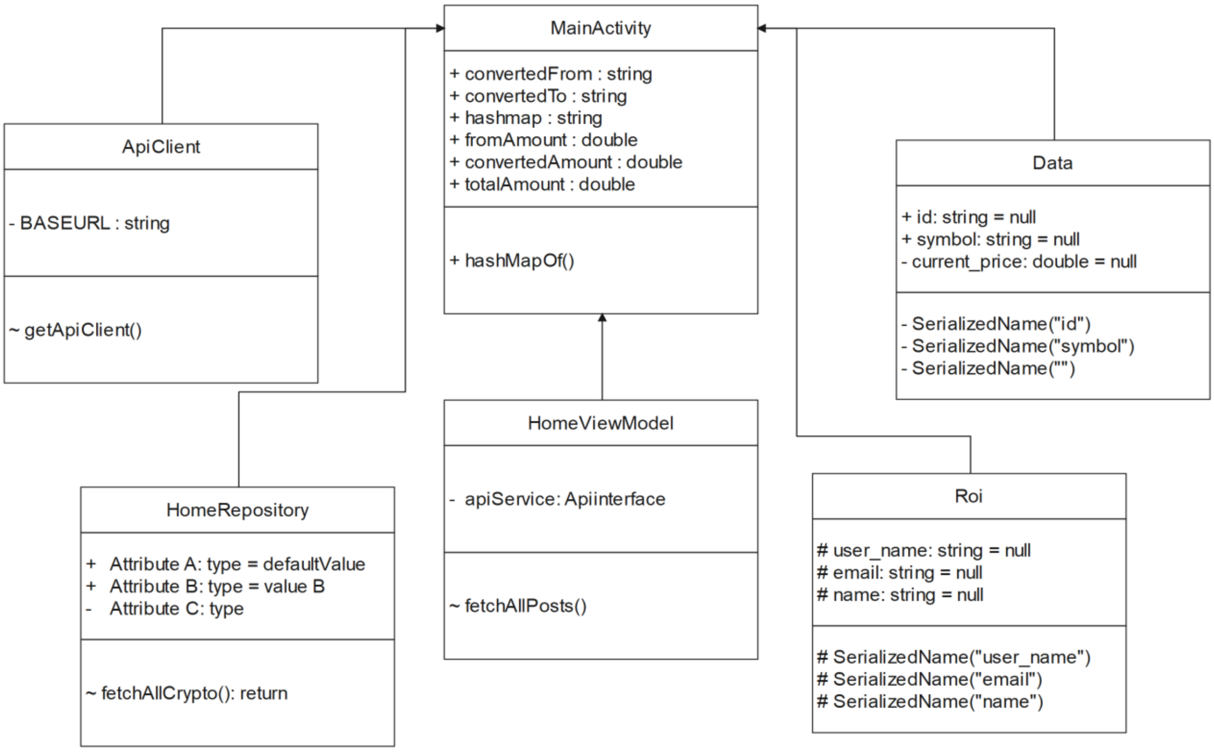
The Total Amount is 5191.4GBP after 0% of Maker fees

Total amount without maker fees

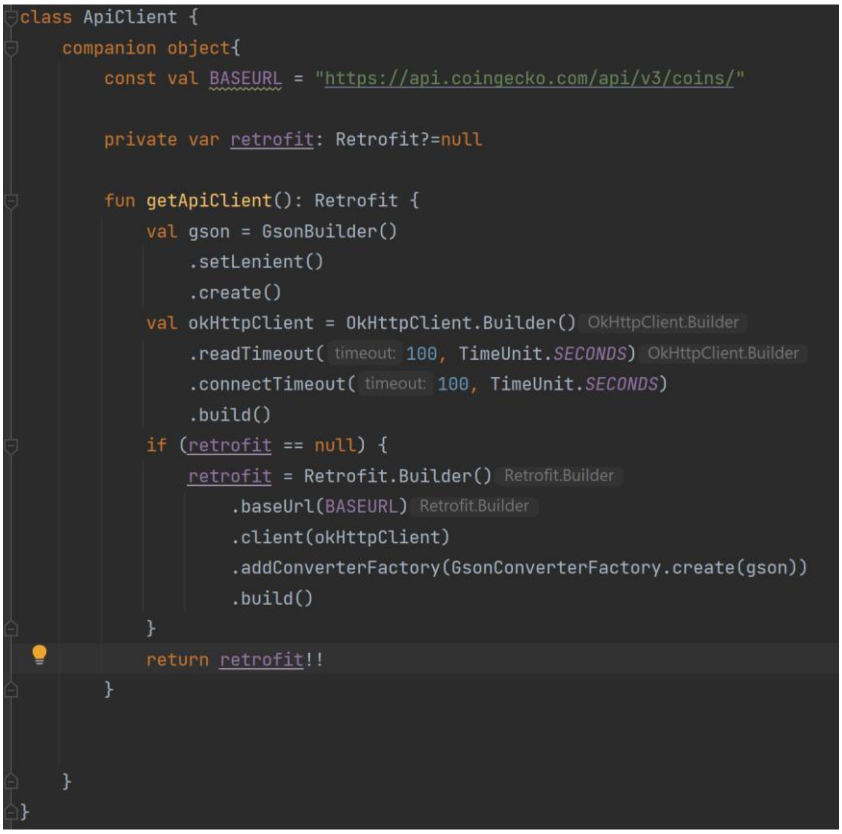
IMPLEMENTATION

We utilised Android Studio and CoinGecko API to retrieve data in order to build this application. User-friendliness of the application was put to the test here by Making its design straightforward, user-friendly, and beneficial for its core purpose. It was challenging to choose the appropriate environment and development kit that would meet every demand of our application because there are many different Android versions, and the operating system is constantly updating older versions with new capabilities. After overcoming the challenges, we finally put our application's concept into practise.

I required six classed for this project including the main activity class in that inherits the remaining other classes.With the help of the UML diagram, we can easily understand how the class with their attributes inherits with other classes in the application.



To get the live data in this project I am using a third-party API service offered by well recognized crypto data service company called CoinGecko. CoinGecko is one of the largest platforms in the world which offers a wide range of crypto currencies data including real time price of every crypto coin in multiple global currencies and offers free API service to get access to the Crypto data. In this project I am using two API offered by CoinGecko The below API link I will be using list the crypto according to market cap the base URL of this API is as below which is provided by the CoinGecko and i.e., https://api.coingecko.com/api/v3/coins/ The below code is used to fetch the data from API server and the code is shown from the screenshot from the class apiClient.



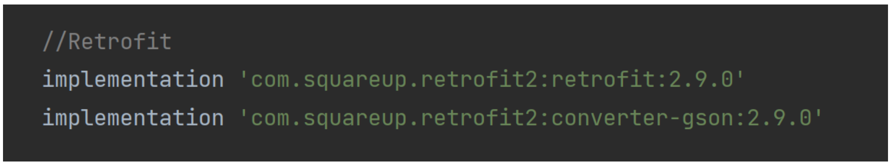
```kotlin
class ApiClient {
    companion object{
        const val BASEURL = "https://api.coingecko.com/api/v3/coins/"

        private var retrofit: Retrofit?=null

        fun getApiClient(): Retrofit {
            val gson = GsonBuilder()
                .setLenient()
                .create()
            val okHttpClient = OkHttpClient.Builder()
                .readTimeout( 100, TimeUnit.SECONDS)
                .connectTimeout( 100, TimeUnit.SECONDS)
                .build()
            if (retrofit == null) {
                retrofit = Retrofit.Builder()
                    .baseUrl(BASEURL)
                    .client(okHttpClient)
                    .addConverterFactory(GsonConverterFactory.create(gson))
                    .build()
            }
            return retrofit!!
        }


    }
}
```

Basically, I used four dependencies, those are retrofit, OkHttp. GSON, Material UI. Here above I injected retrofit because our project is based on sending and receiving the data from server to device and visa versa. So retrofit manages the process of receiving, sending and creating HTTP request and responses.

In that API server there, different attributes present for the particular currencies, but I retrieves the only necessary data required for the conversion between two currencies. And those are id, symbol, and current price. And remaining ones are ignored.

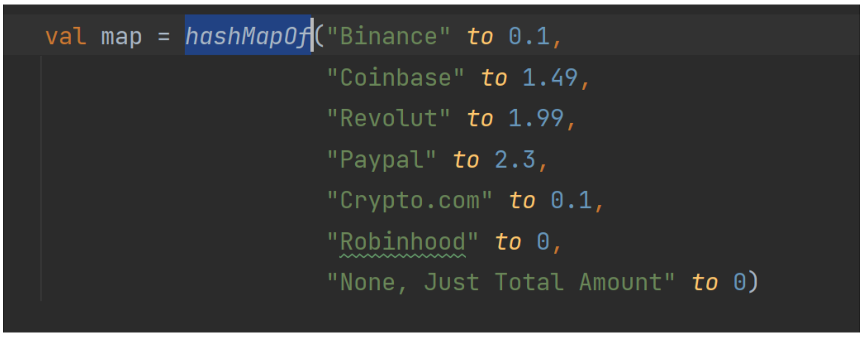
I used OkHttp dependency because it provides the implementation of the HttpURLConnection and Apache Client Interface without using any other dependencies.

The dependencies URL used in the build.gradle module is shown below.



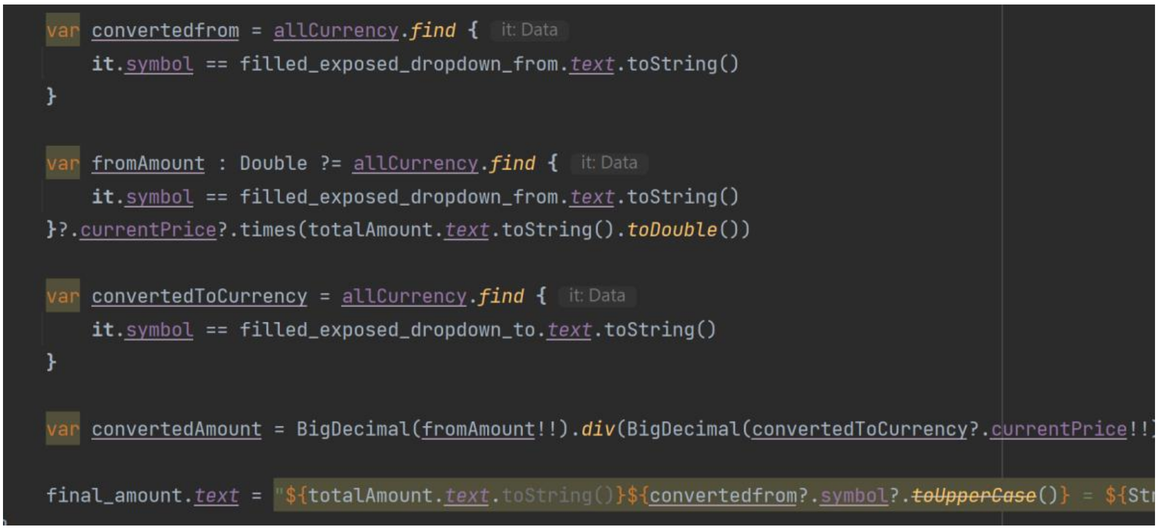
```groovy
//Retrofit
implementation 'com.squareup.retrofit2:retrofit:2.9.0'
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
```

Coming next, while selecting the trading platform, I did included only fer trading platforms with their makeup fees but this can be easily upgradable just by putting the name and percentage value in the function,



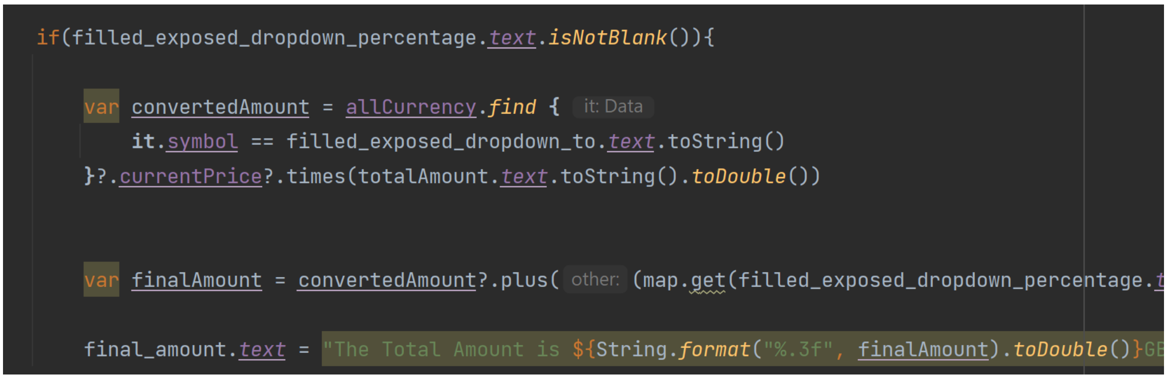
```
val map = hashMapOf("Binance" to 0.1,
                    "Coinbase" to 1.49,
                    "Revolut" to 1.99,
                    "Paypal" to 2.3,
                    "Crypto.com" to 0.1,
                    "Robinhood" to 0,
                    "None, Just Total Amount" to 0)
```

The base code to perform the conversion between the two currency is as shown below.



```
var convertedfrom = allCurrency.find {  it: Data
    it.symbol == filled_exposed_dropdown_from.text.toString()
}

var fromAmount : Double ?= allCurrency.find {  it: Data
    it.symbol == filled_exposed_dropdown_from.text.toString()
}?.currentPrice?.times(totalAmount.text.toString().toDouble())

var convertedToCurrency = allCurrency.find {  it: Data
    it.symbol == filled_exposed_dropdown_to.text.toString()
}

var convertedAmount = BigDecimal(fromAmount!!).div(BigDecimal(convertedToCurrency?.currentPrice!!

final_amount.text = "${totalAmount.text.toString()}${convertedfrom?.symbol?.toUpperCase()} = ${St
```

And to calculate the total amount needed to calculate the makeup fees is mentioned below from the main activity.

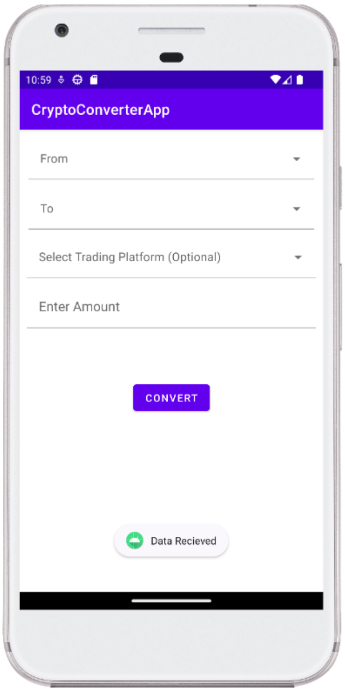


```
if(filled_exposed_dropdown_percentage.text.isNotBlank()){

    var convertedAmount = allCurrency.find {  it: Data
        it.symbol == filled_exposed_dropdown_to.text.toString()
    }?.currentPrice?.times(totalAmount.text.toString().toDouble())


    var finalAmount = convertedAmount?.plus( other: (map.get(filled_exposed_dropdown_percentage.

    final_amount.text = "The Total Amount is ${String.format("%.3f", finalAmount).toDouble()}G
```

And throughout the conversion I used GSON library, to convert the API values to XML format so it can easily extractable to execute the operation.

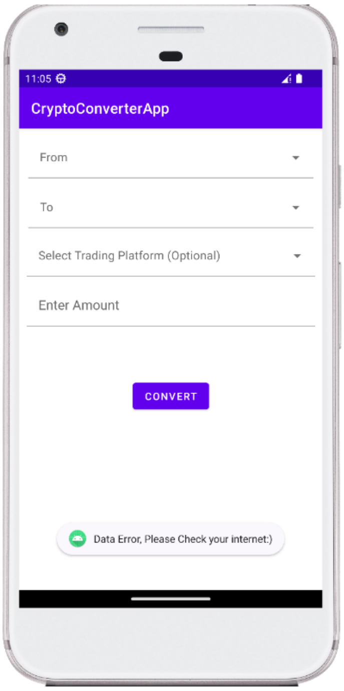
TESTING

The first testing is whether the device is connected to the internet or not. If it connected it shows temporary message that “Data Received”, and if its not connected to the internet it shows error message “Data Error, Please Check your Internet:)”. Below are the screen shots with both conditions.



Data Recieved

Device connected to the Internet



Data Error, Please Check your internet:)

Device not connected to the Internet

CONCLUSION AND FUTURE WORK

The use of a cryptocurrency converter tool will make it easier to gather, examine, and understand all the necessary information about cryptocurrencies under one roof. My proposal is considerably more doable and simpler to implement than many other alternatives. The system is dynamic, making future adjustments simple. This programme offers the ability to compute the maker fees from numerous cryptocurrency trading platforms in addition to just converting values. Everyone can use the cryptocurrency converter application, including forex traders, travellers from other countries, and analysts, with ease for both social and professional purposes. Utilizing the resources and methodologies at hand, development was carried out while keeping user-friendliness as a priority. The application has been tested with live data and has provided a successful result. Hence the software has proved to work efficiently.

Future Direction

The application can be enhanced as follows:

1) Ability to show the live chart of the cryptocurrencies.
2) Ability to perform to add favourite currencies.
3) Ability to show the total amount in other global currencies like INR, USD etc.
4) Ability to list the currencies based on the market cap or other kind of sorting.
5) Ability to trade currencies within the app.

REFERENCES

[1] https://www.youtube.com/watch?v=ct5etYgB5pQ

[2] https://www.geeksforgeeks.org/how-to-build-a-cryptocurrency-tracker-android-app/

[3] https://www.youtube.com/watch?v=hHM5c0hYkQA

[4] https://www.codespeedy.com/create-real-time-currency-converter-app-in-android-studio/

[5] https://stackoverflow.com/questions/70646910/retrieving-data-from-api-android-studio-kotlin

[6] https://stackoverflow.com/questions/56890869/convert-api-response-to-json

[7] https://github.com/evant/kotlin-inject

[8] https://developer.android.com/training/dependency-injection

[9] https://www.geeksforgeeks.org/retrofit-with-kotlin-coroutine-in-android/