Procedure to create the app for chat connect- A real time chat and communication app

Team ID:Team-591094

Chat Connect is a sample project built using the Android Compose UI toolkit. It demonstrates how to create a simple chat app using the Compose libraries. The app allows users to send and receive text messages. The project showcases the use of Compose's declarative UI and state management capabilities. It also includes examples of how to handle input and navigation using composable functions and how to use data from a firebase to populate the UI.

Learning Outcomes:

By end of this project:

- You'll be able to work on Android studio and build an app.
- You'll be able to integrate the database accordingly.

Project Workflow:

- Users register into the application.
- After registration, user logins into the application.
- User enters into the main page

Tasks:

1.Required initial steps

2.Creating a new project

3.Integrating Firebase and Authentication

4.Creating UI files

5.Running the application.

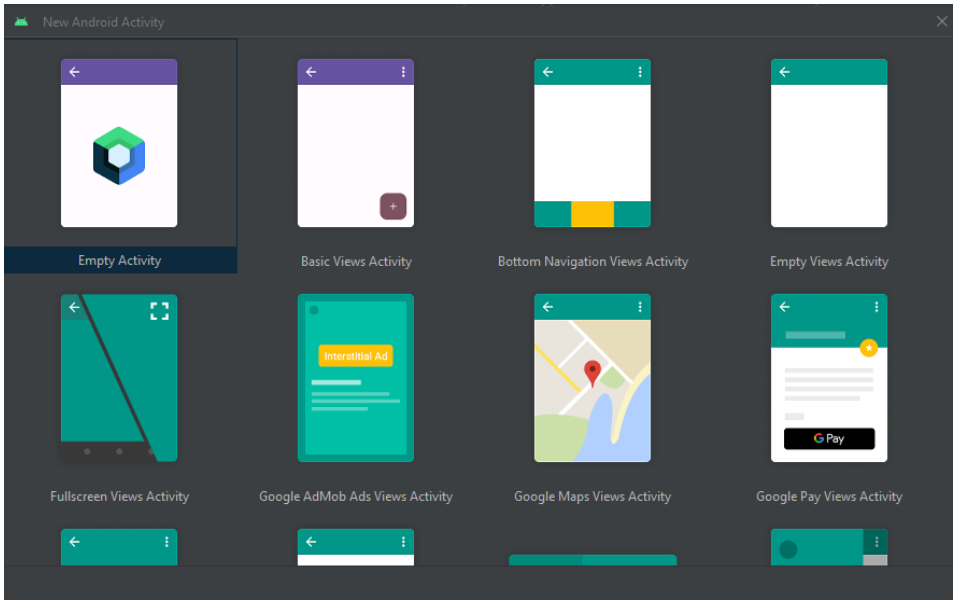
Task 1:

Required initial steps: https://developer.android.com/studio/install

Task 2: Creating a new project.

Step 1: Android studio > File > New > New Project > Empty Compose Activity

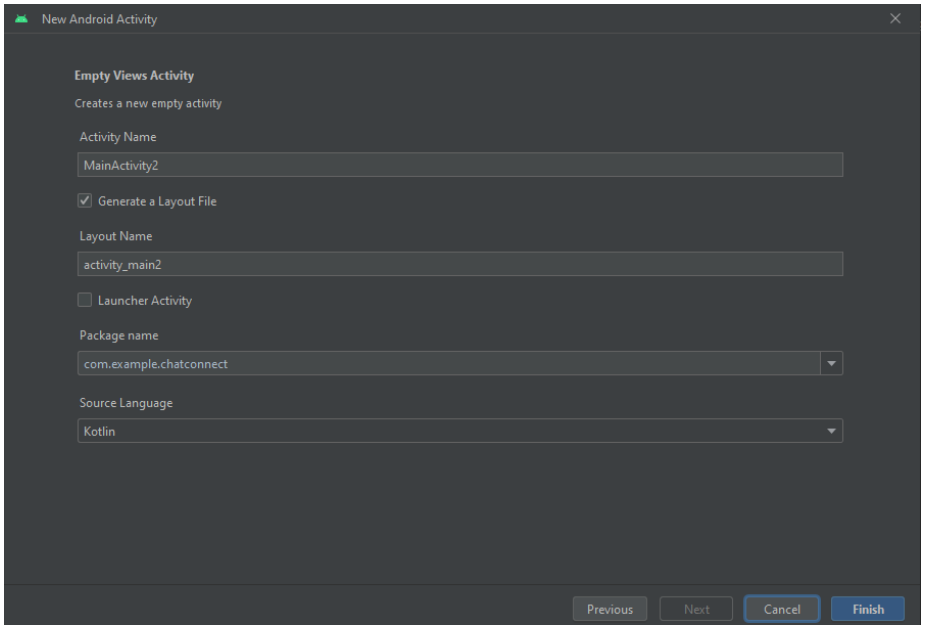
Step 2: Click on Next button.



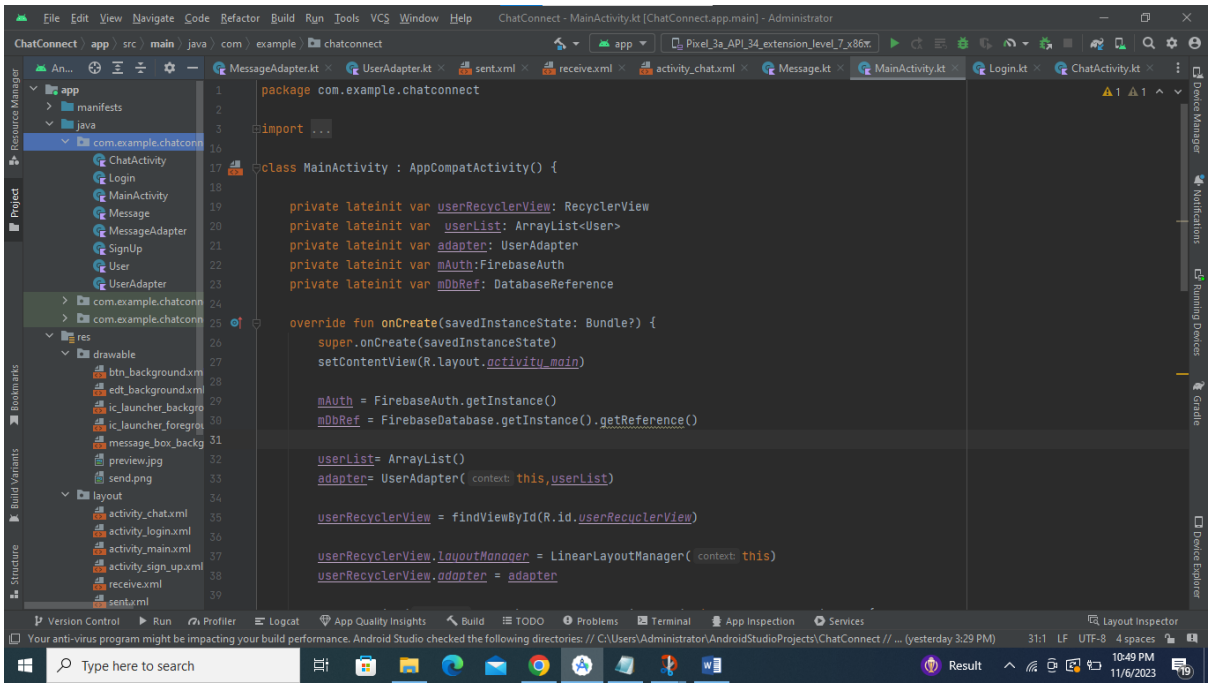
Step 3: Give name to the new project.

Step 4: Give the Minimum SDK value

Step 5: Click Finish



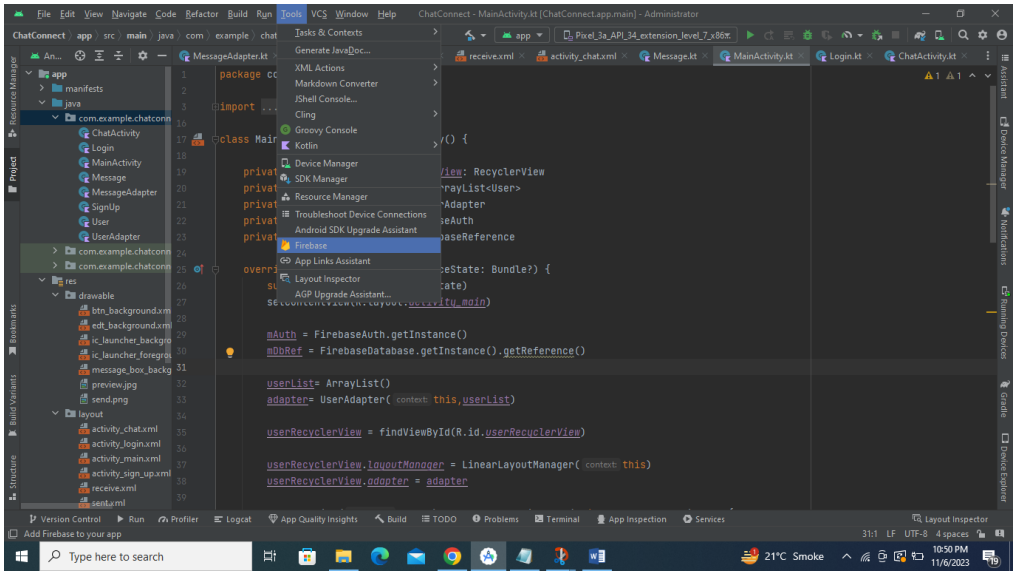
Main Activty file



Task 3:

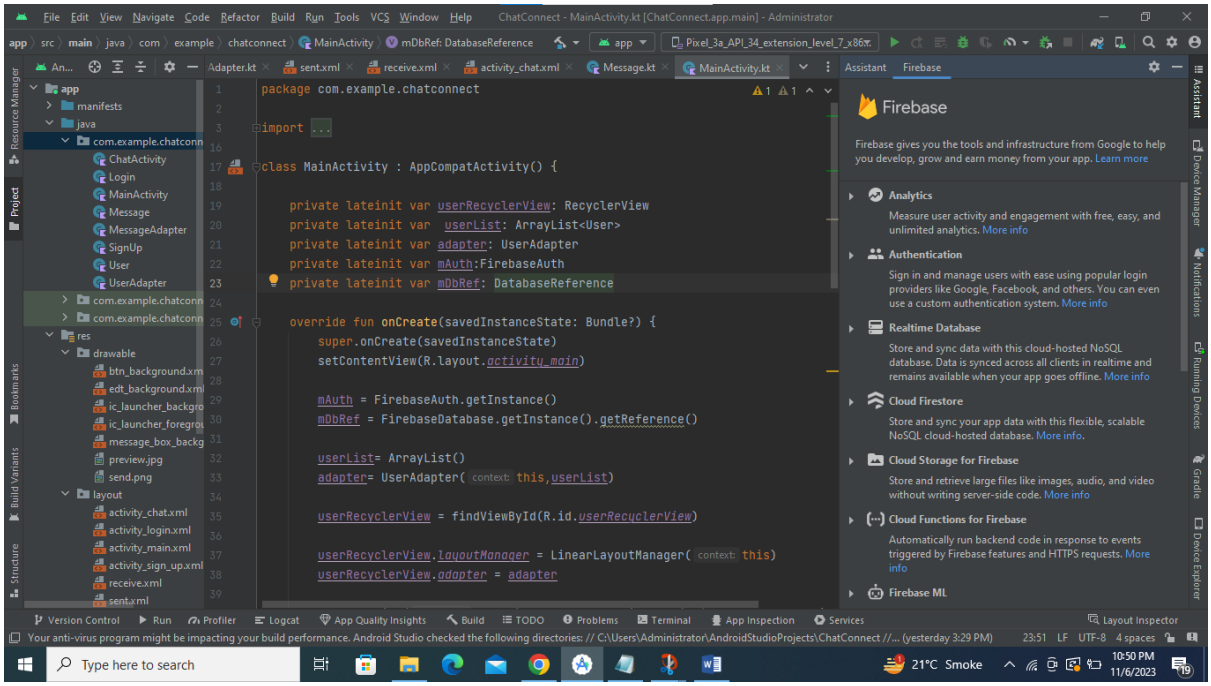
Integrating Firebase

- Menu bar > Tools > Firebase.

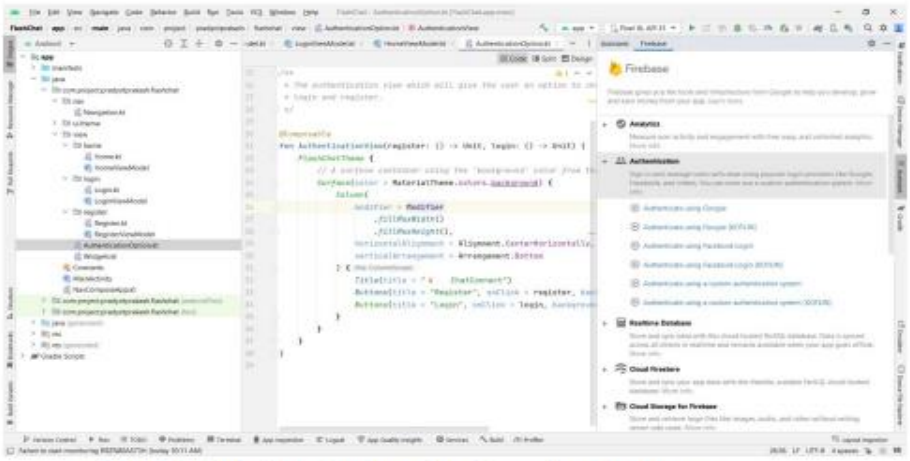


After clicking the fire base tab will open on the side

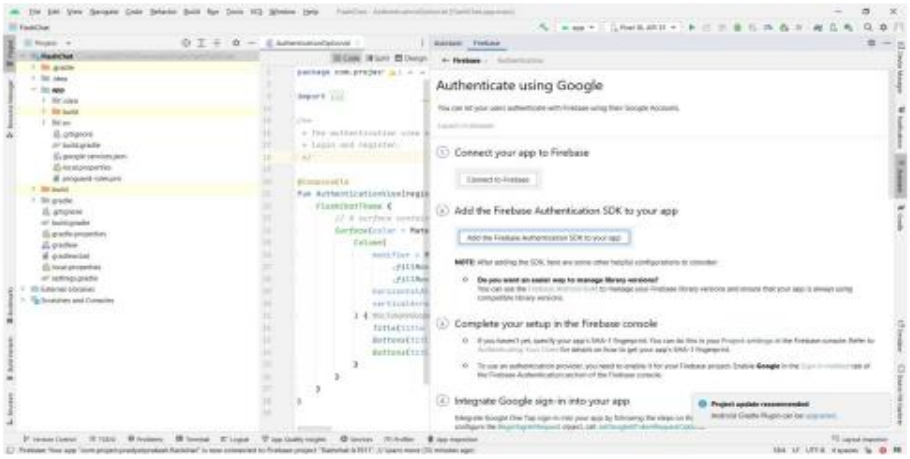
The open authentication



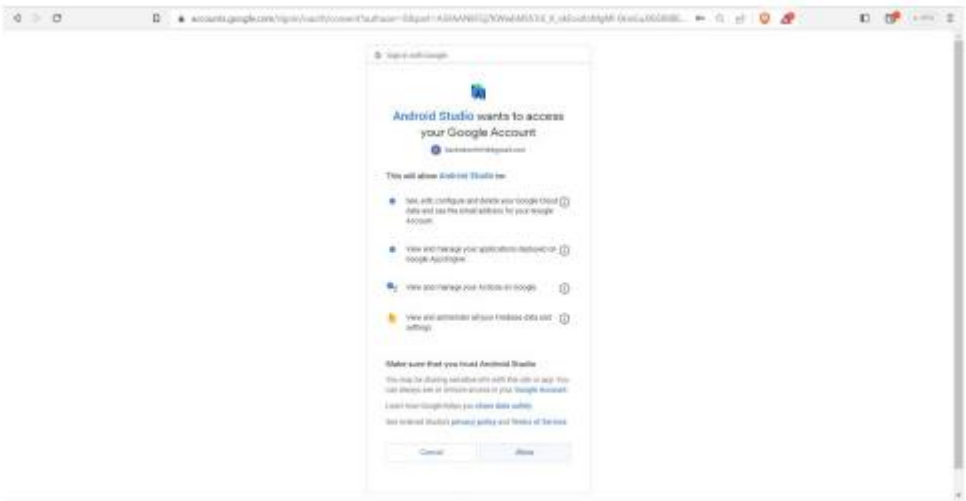
You got to follow some of these procedures



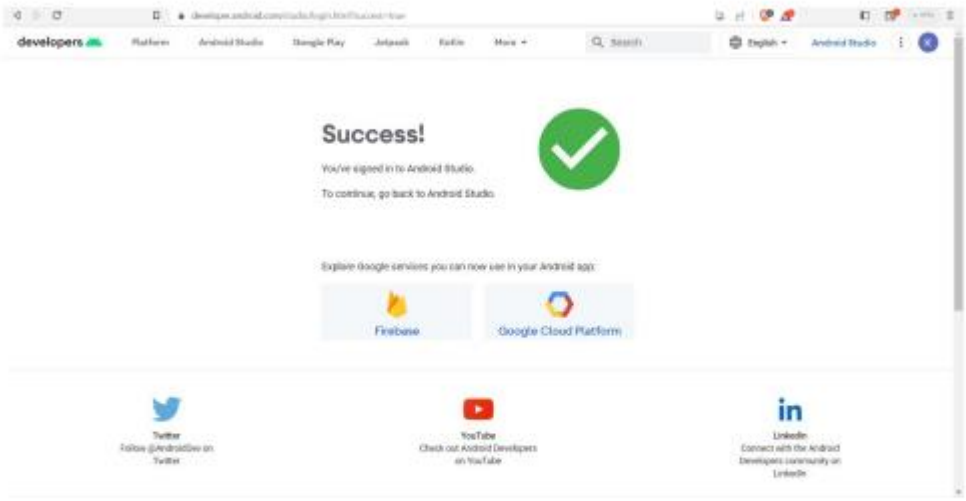
- After that Authenticate using Google will open



- In that Click on point one Connect To Firebase
- Now you will be redirected to Google sign-in page in your respective browser.
- After signing into your google account, below shown interface will appear



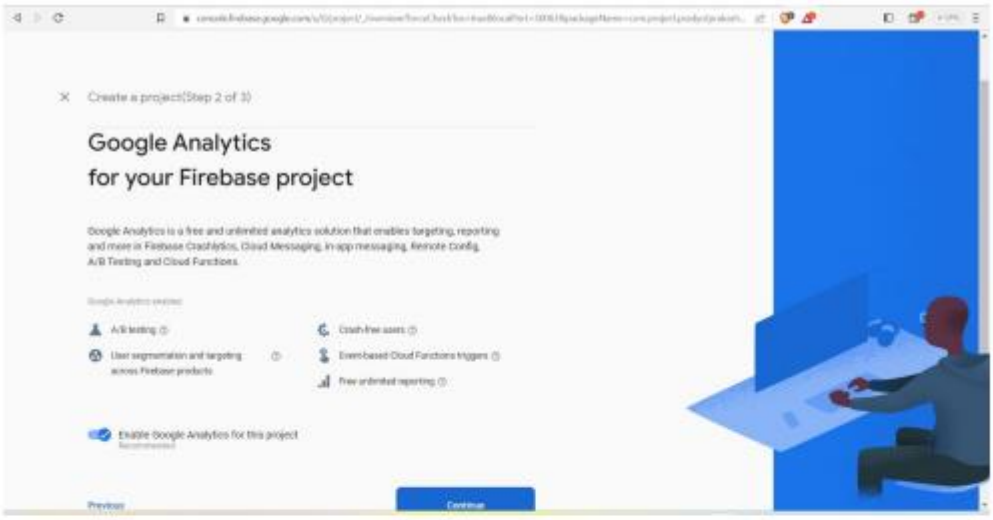
- Click on **Allow**



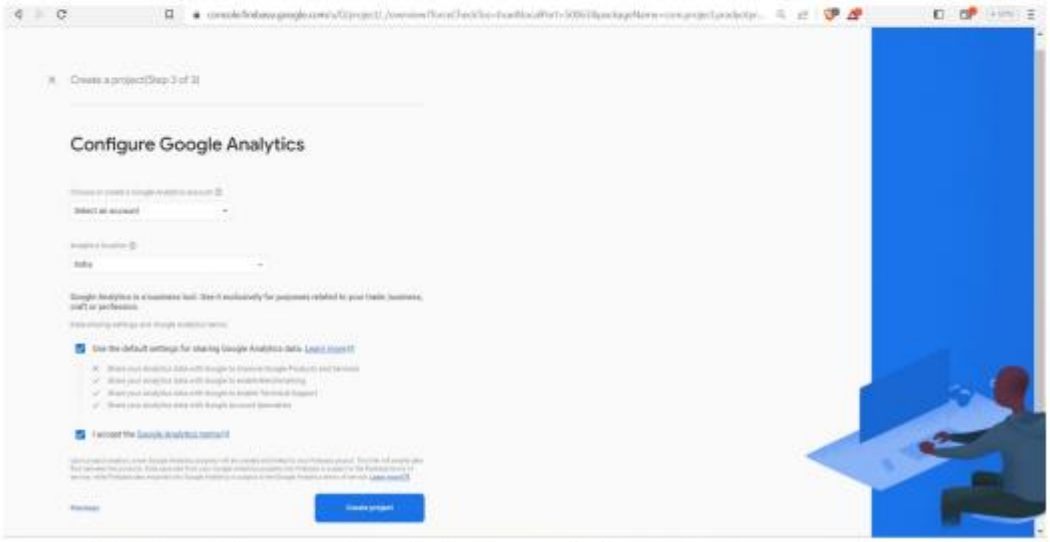
- Now, your Google account is connected to Android
- Now Click on **Firebase**.

- Now the Firebase page will open.

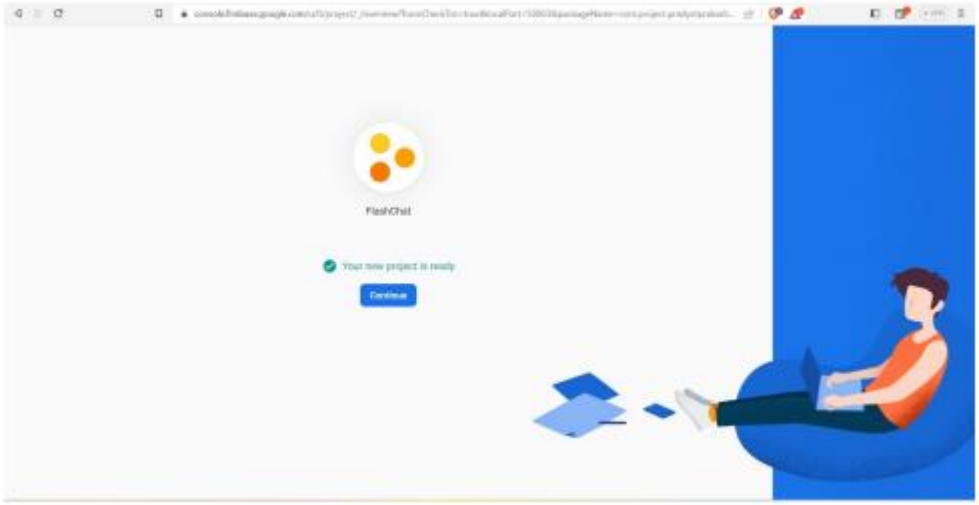
- Click on **continue**



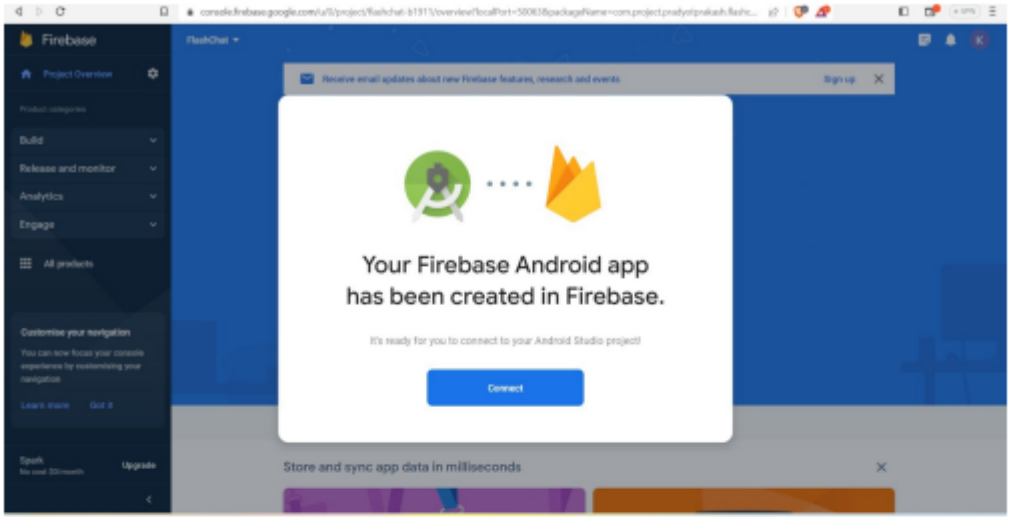
- Now enable the Google Analytics for this project and
- Click on **continue**



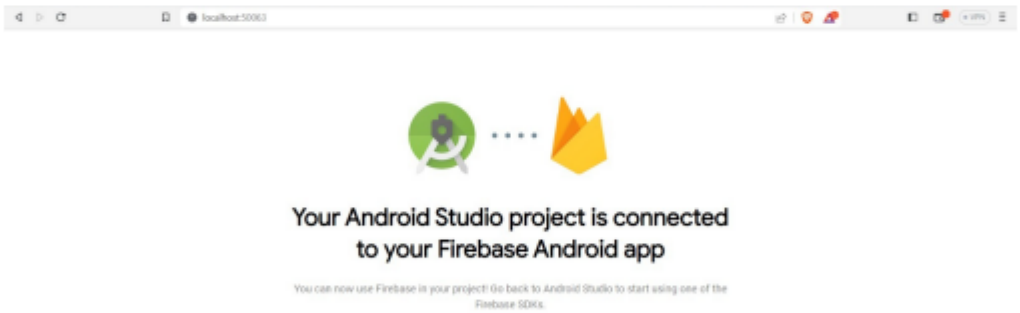
- Set Analytics location to India and check the boxes.
- Now Click on **Create project**



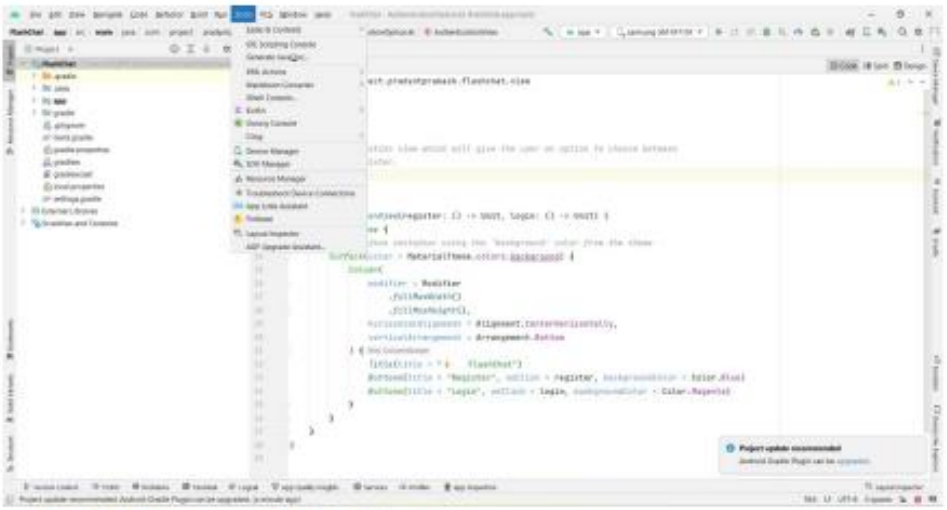
- Click on **Continue**



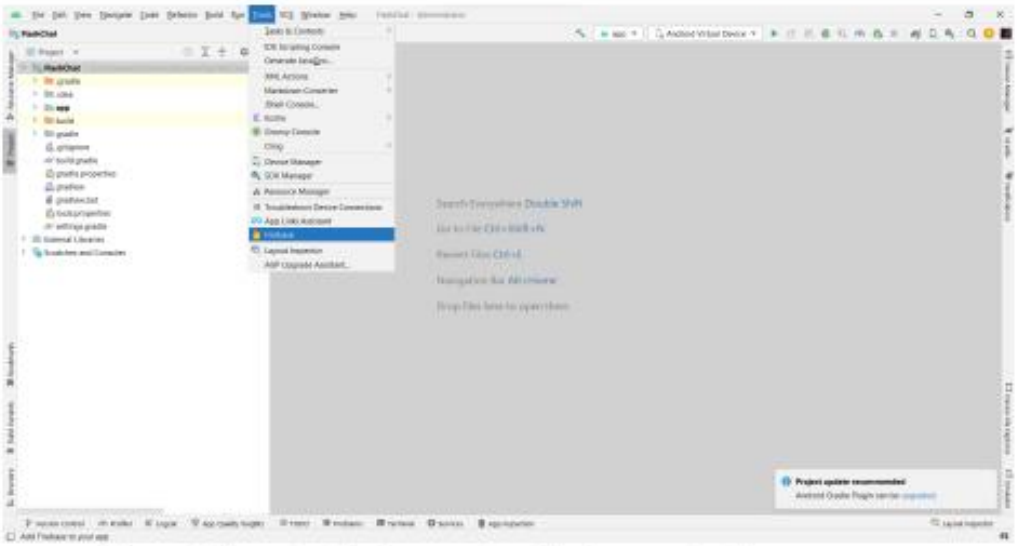
- Click on **Connect**



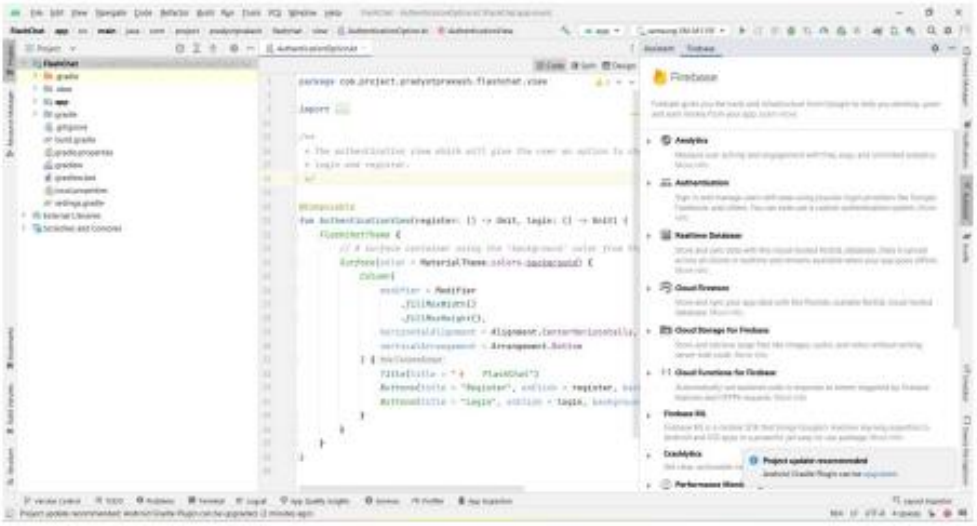
- This shows that your project is connected to Firebase.



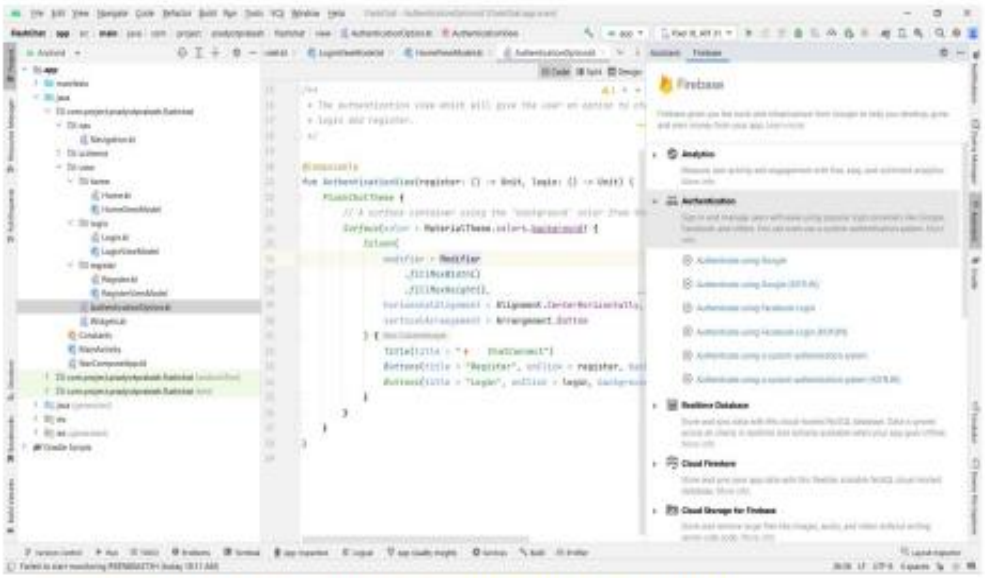
- Menu bar > Tools > Firebase.



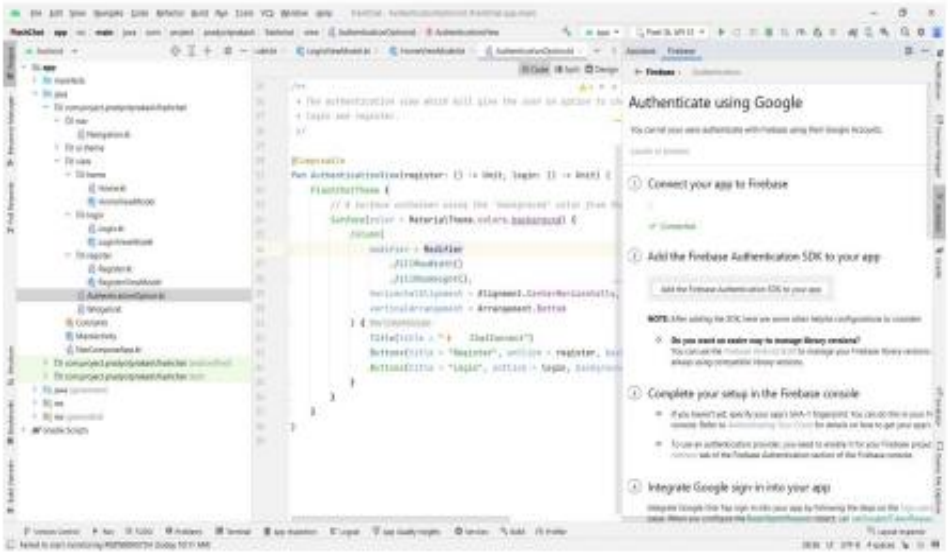
- After clicking the Firebase , the Firebase tab will open.



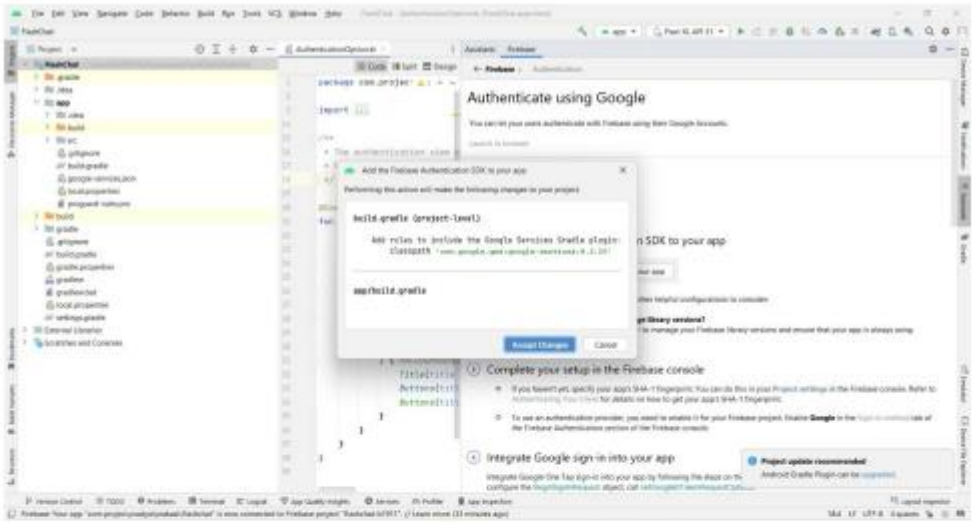
- In Authentication > Go to Authenticate using Google.



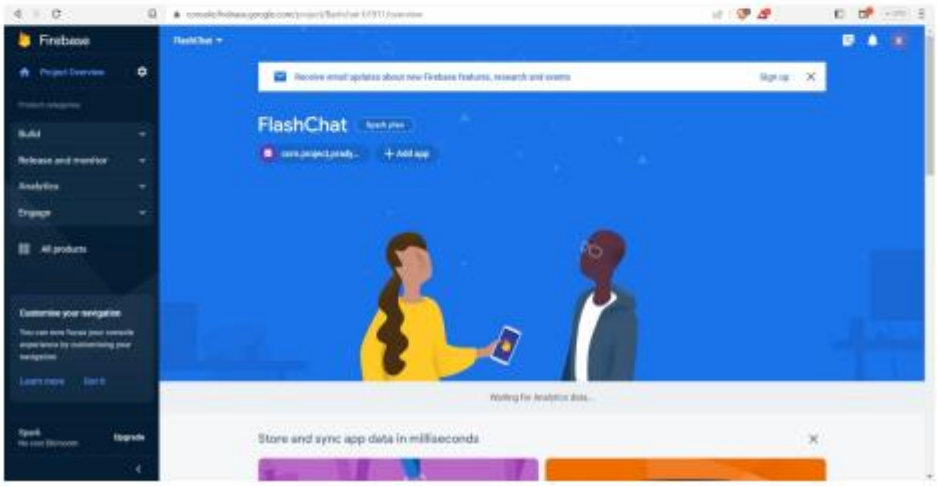
- After that, Authentication using Google will open.



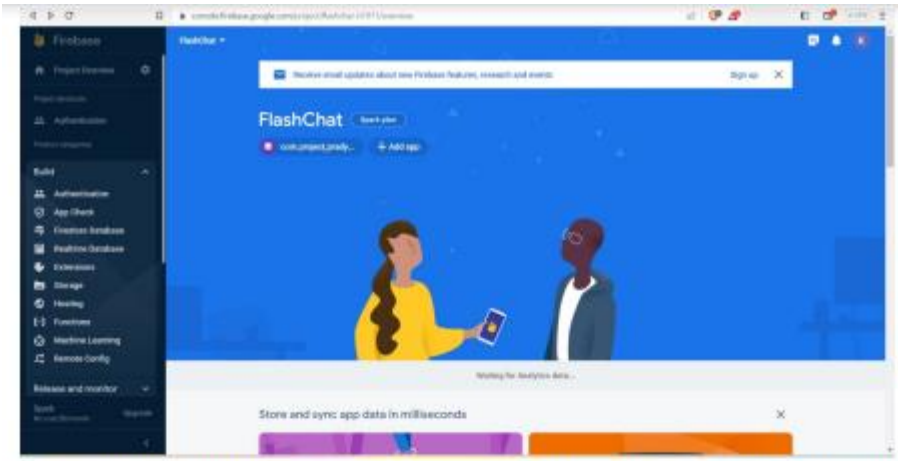
- Now you can see that it shows **Connected.**
- Now click on the second point “Add the Firebase Authentication SDK to your app”.



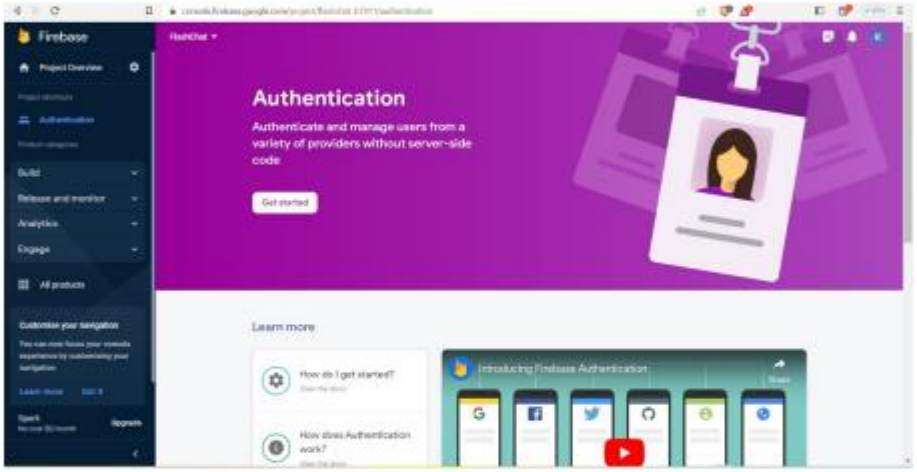
- Now a small tab will open and Click on **Accept Changes**.



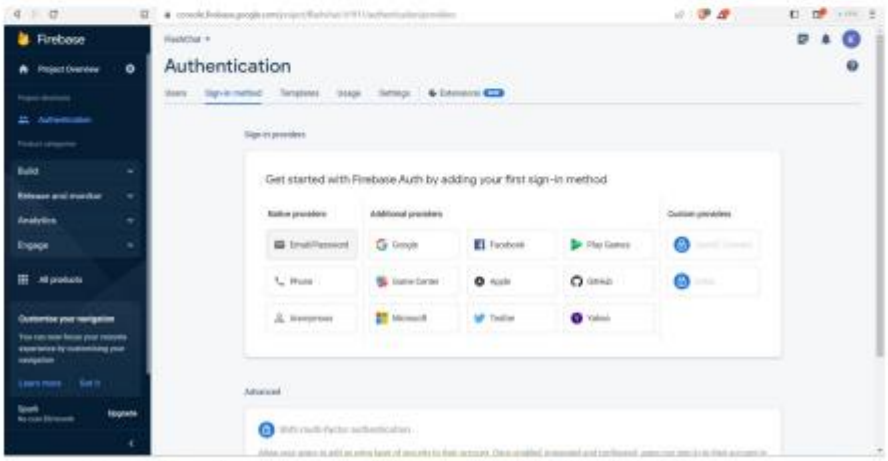
- Now go to your respective browser and search firebase or https://firebase.google.com/
- Open Firebase official website



- In the right side panel Click on **Build** < Click on **Authentication**.

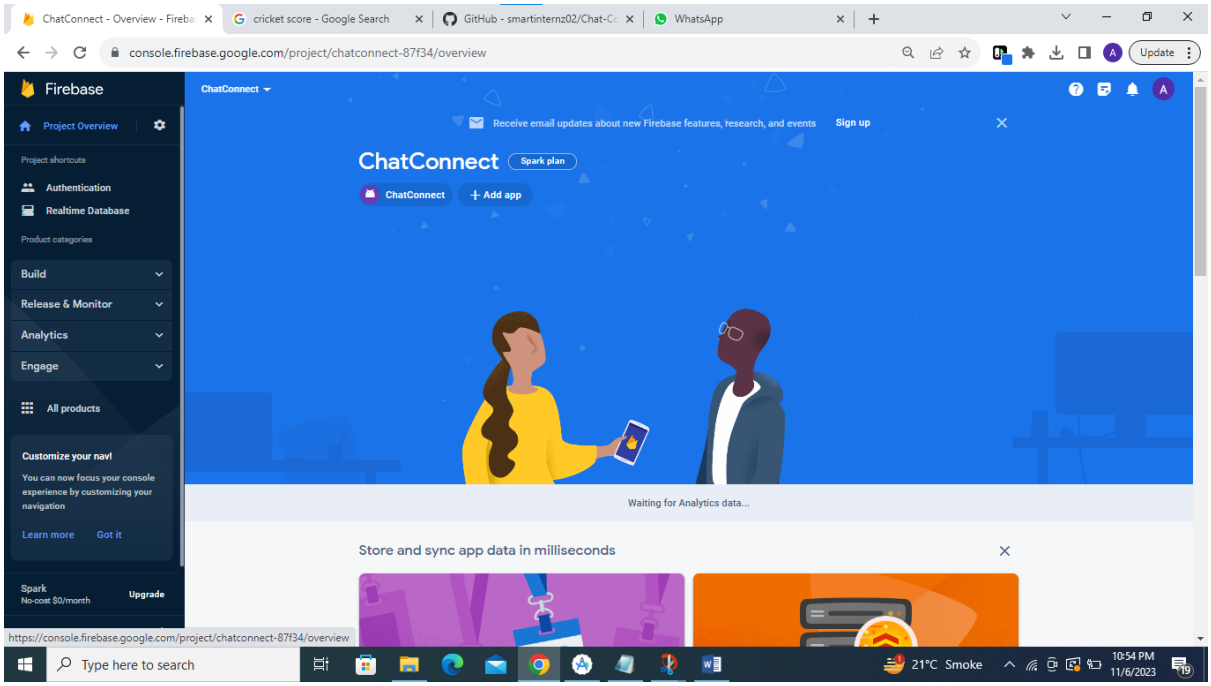


- Click on **Get started**.



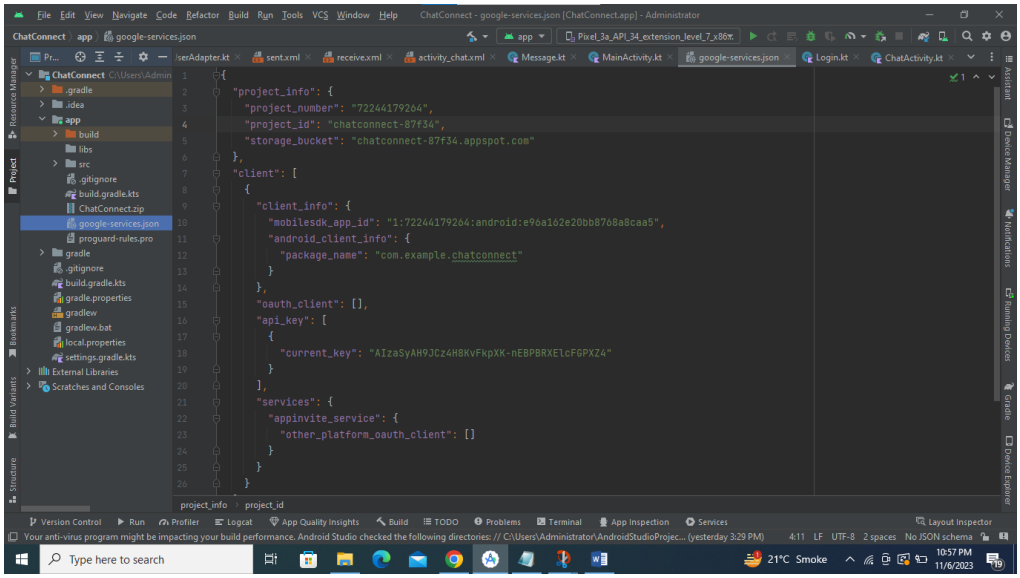
- Now the above page will open for **Authentication**.
- Go to Sign-in-method.
- Click on **Email/Password**.

After following those procedures we have firebase ready for our app



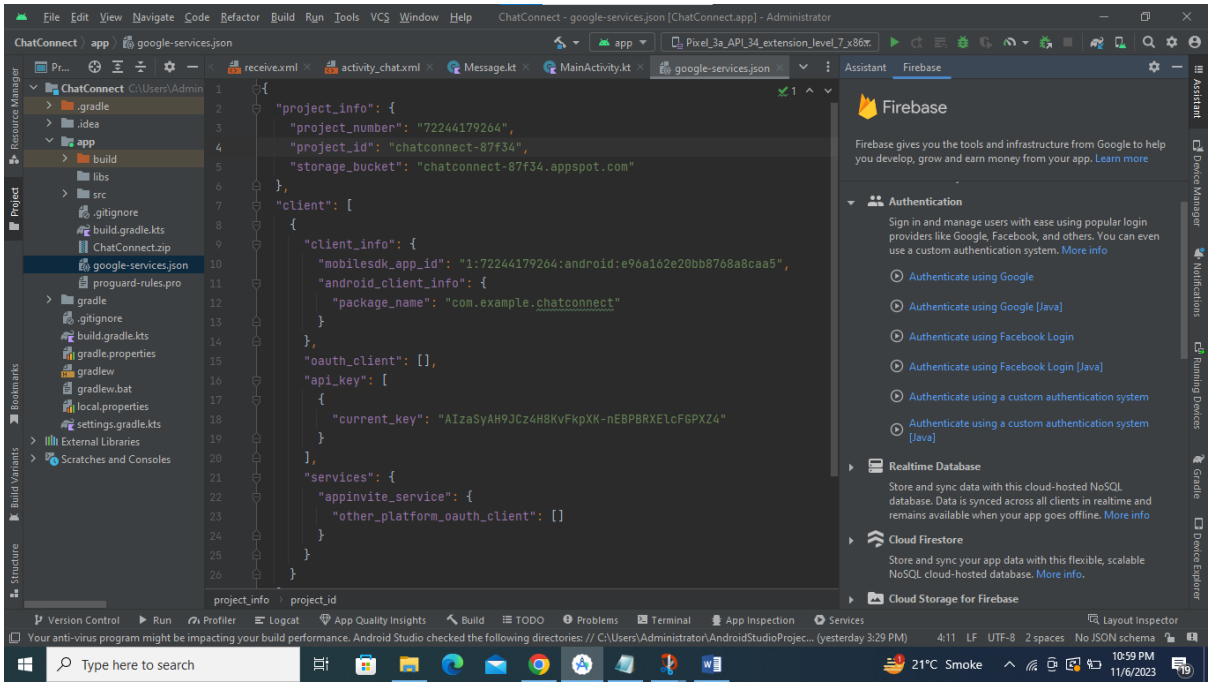
Before this we gotta add some files into our app that is like

One is google.services.json file to our project we show it below

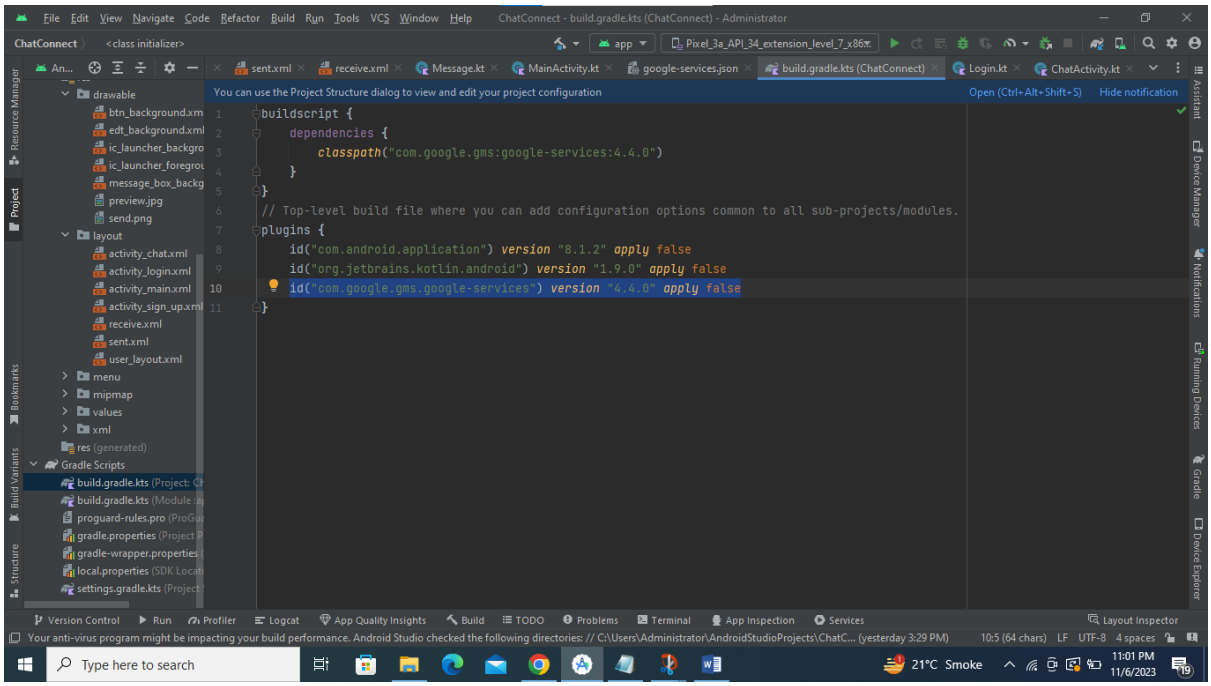


This is like one of the main file we got to copy-paste In our app.

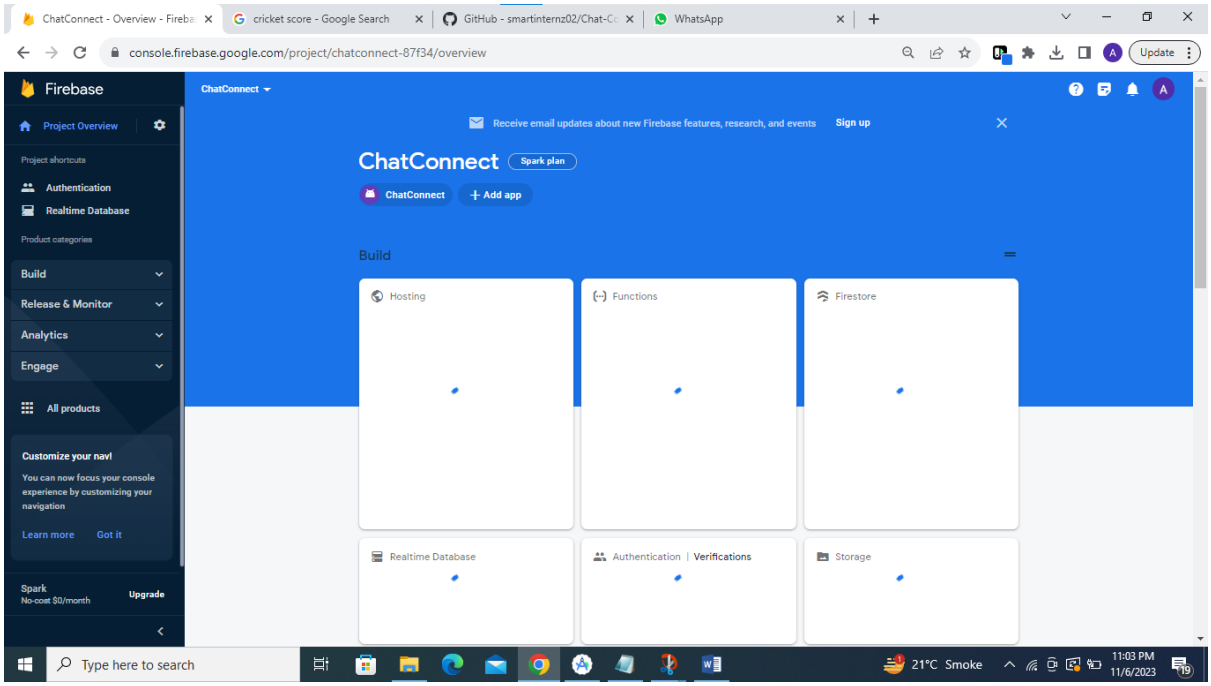
After this we got to paste gradle things to our app



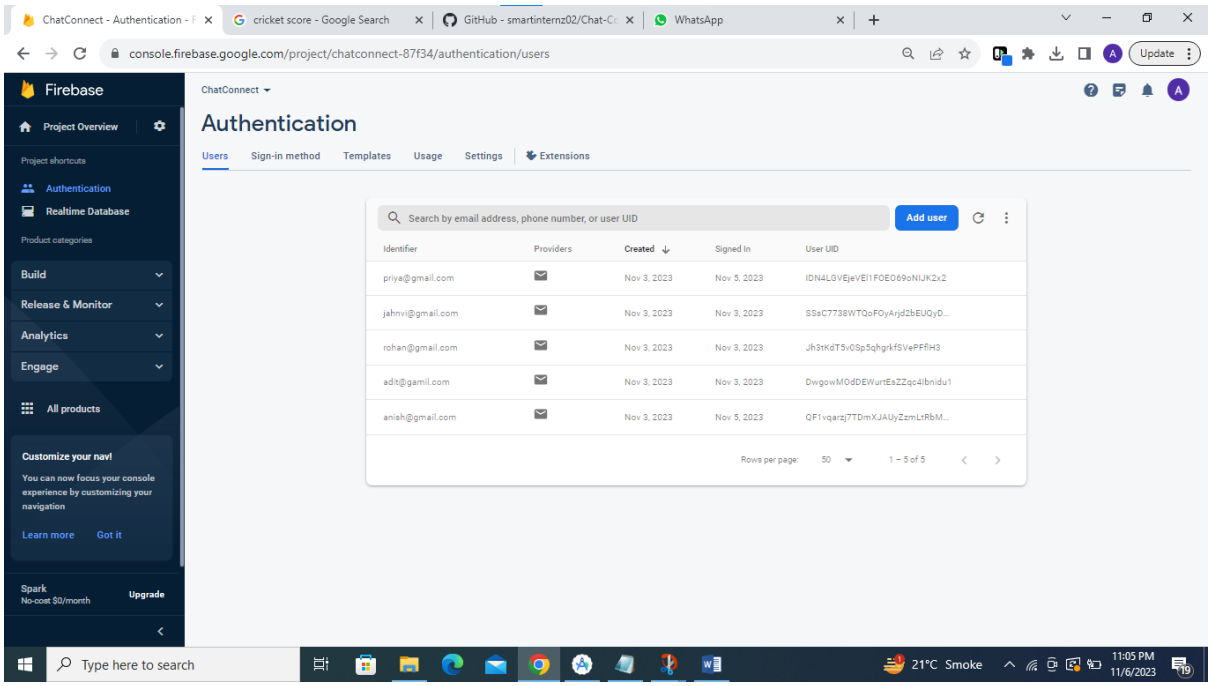
We pasted this in our gradle the on highlighted

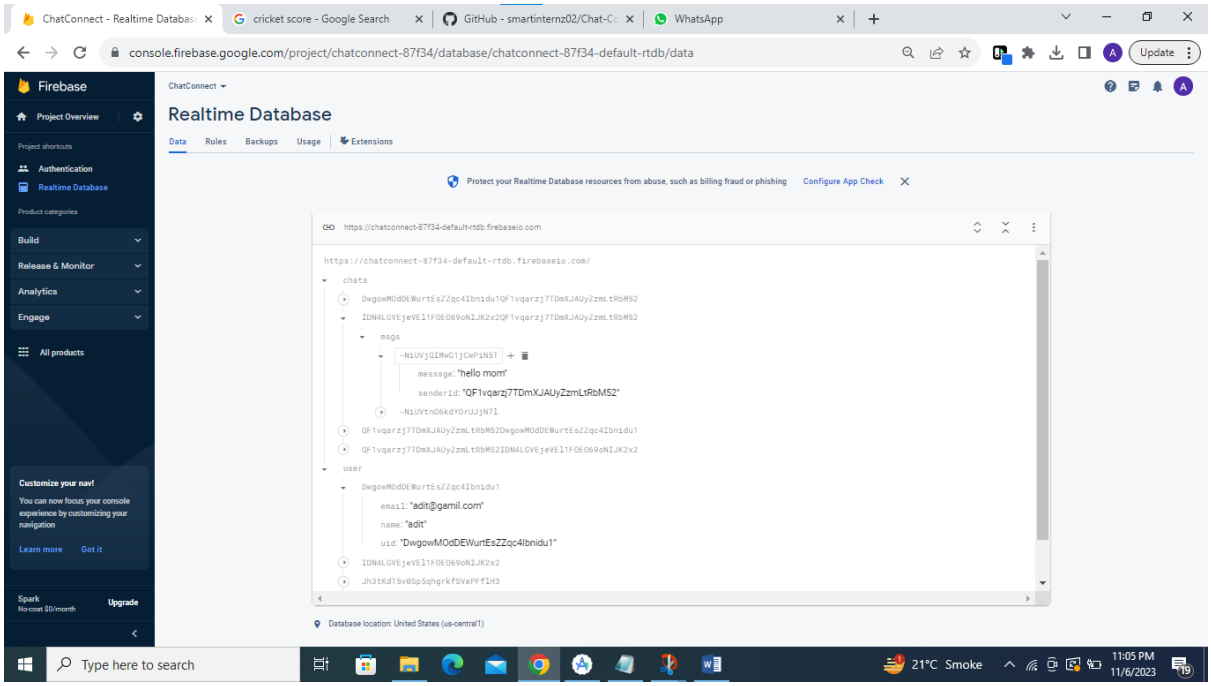


This our app’s firebase overview



We personally have created authentication and Realtime database

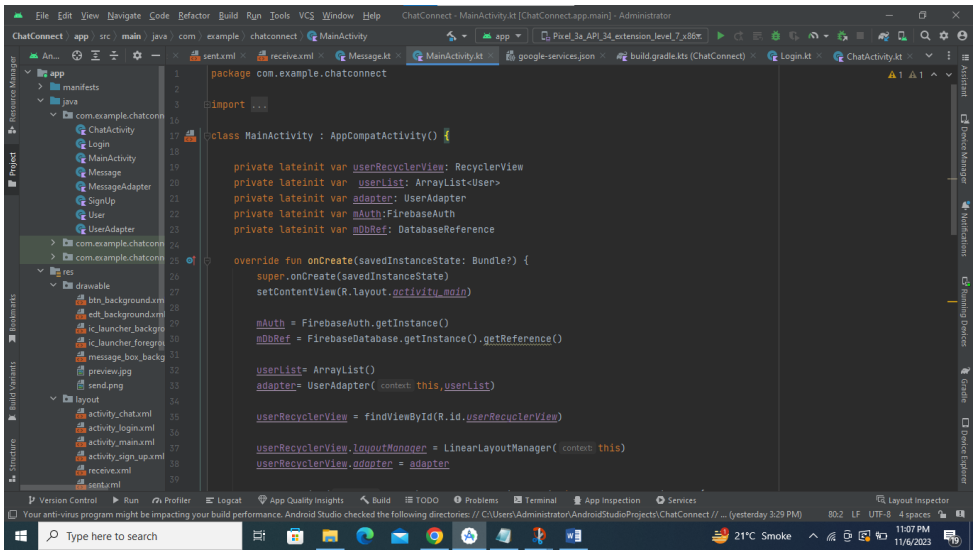




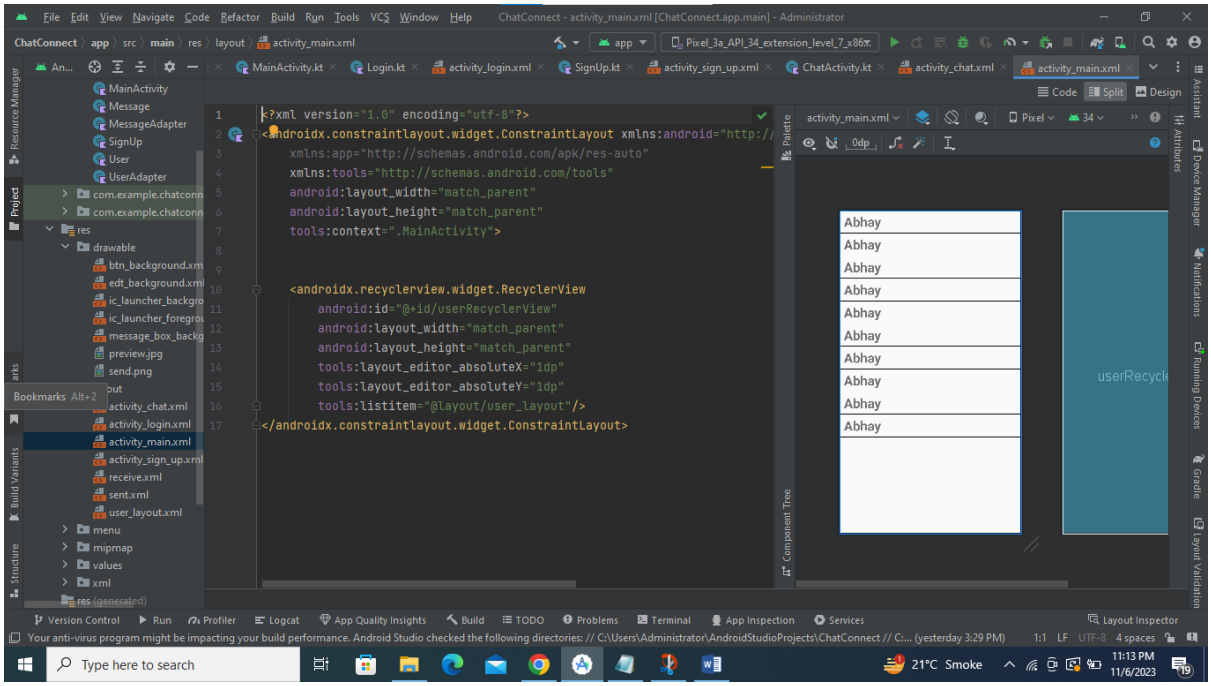
As you can see above we have made uid for each user and each message they send

Task 4:

MainActivity.kt file

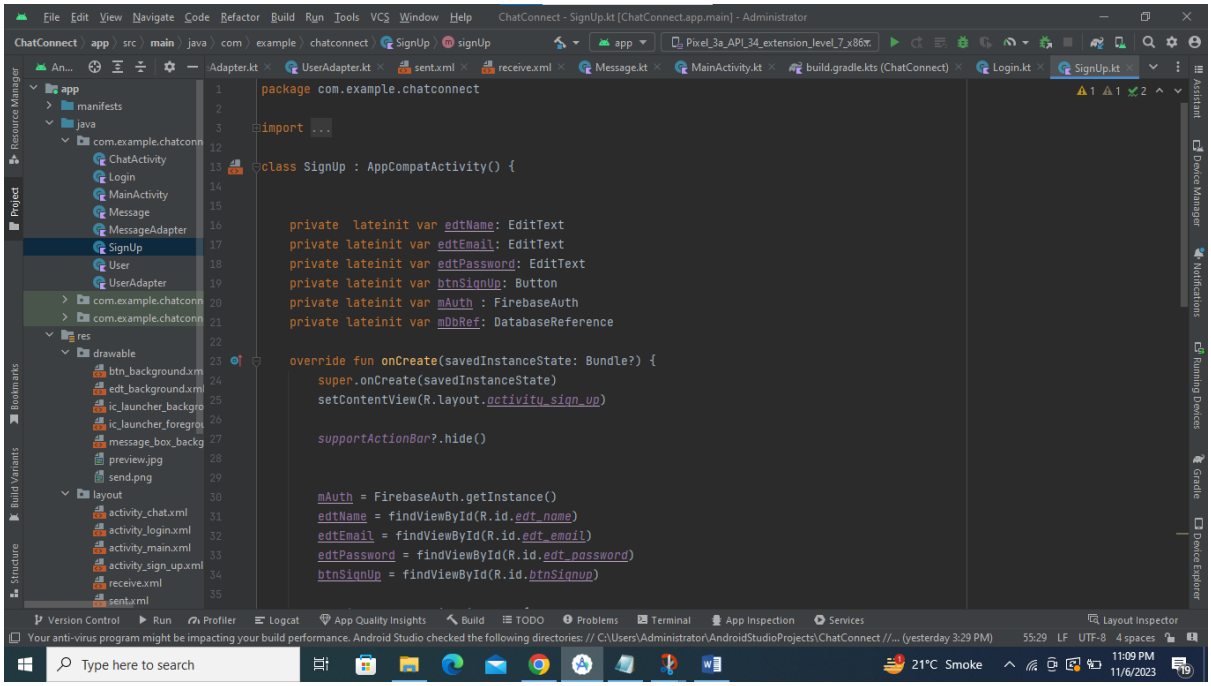


Xml file for main activity

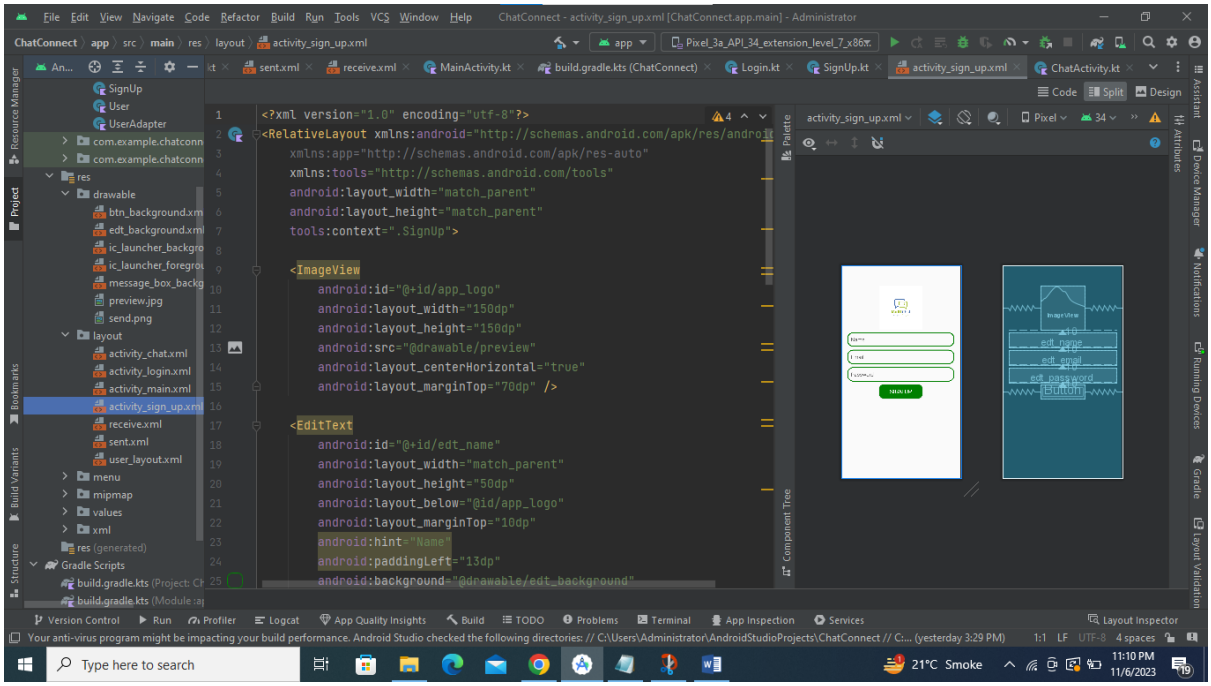


After main activity we made login and signup page for our app

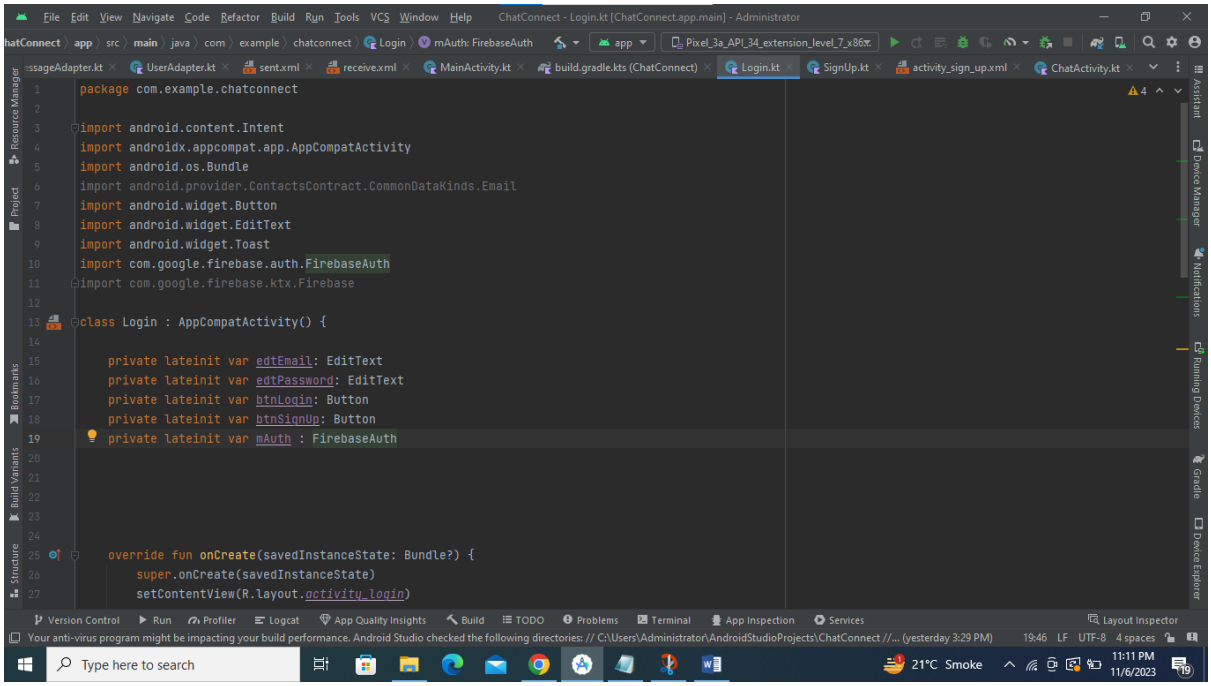
Signup.kt



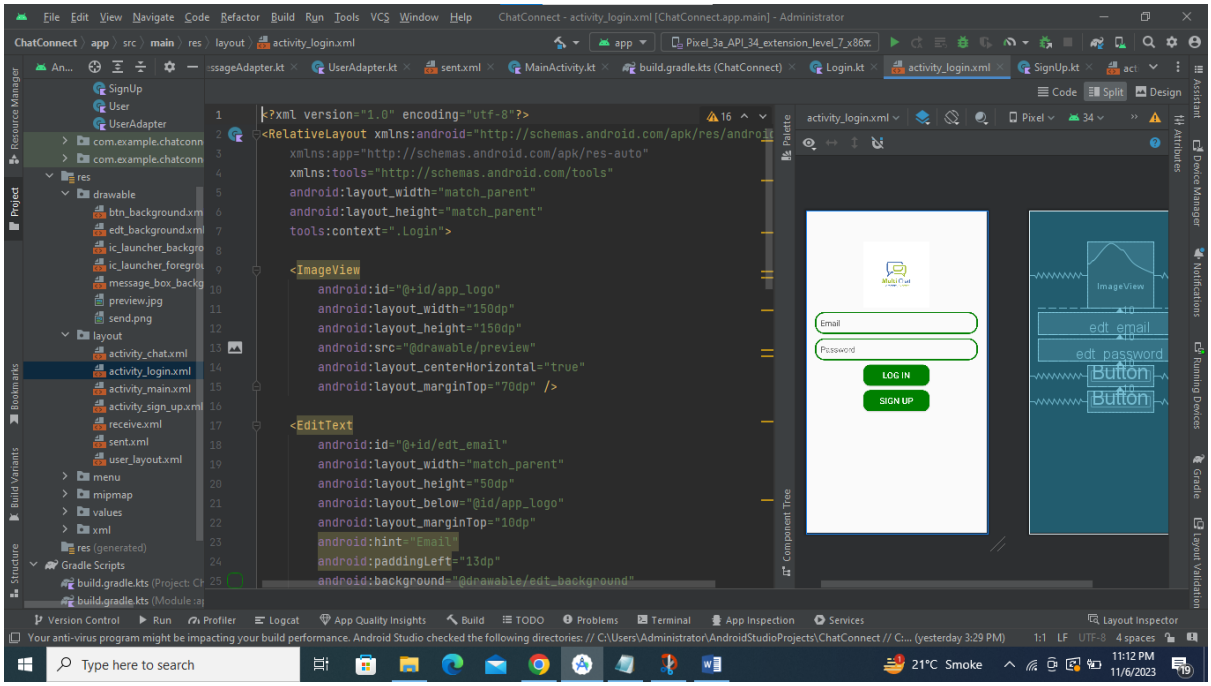
Our xml file for it



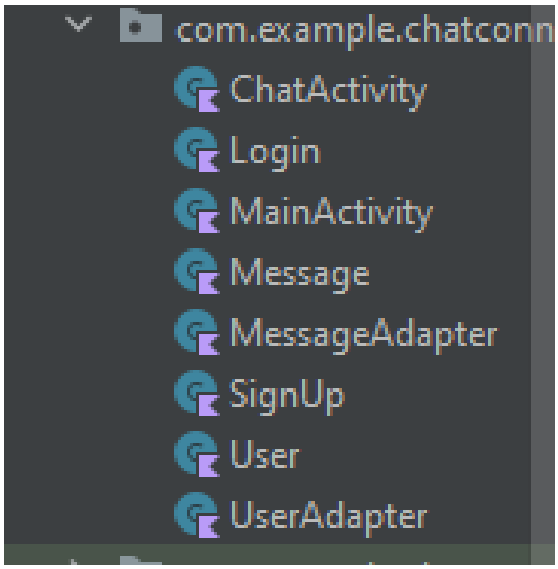
For log in page now



Xml file foe login page

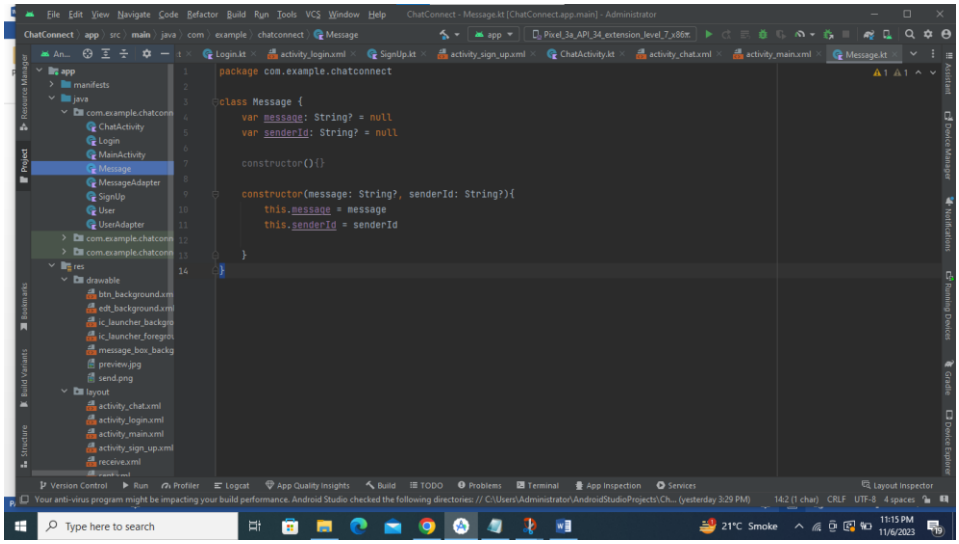


After this we made some more of our files like these:



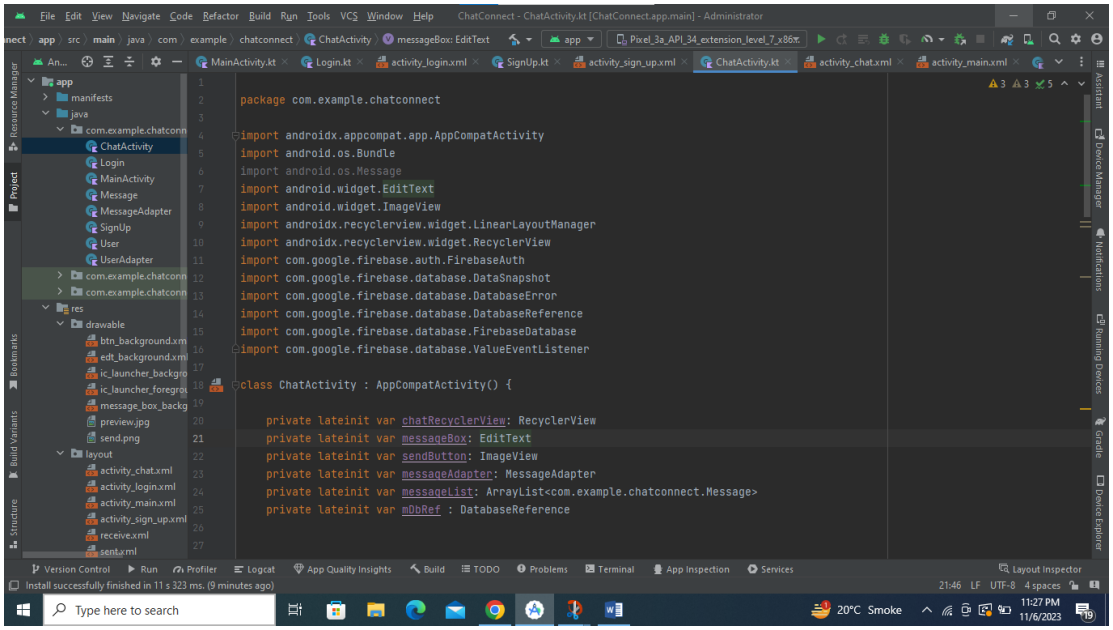
We will put screen shot of all with their kotlin and xml file below

Message.kt

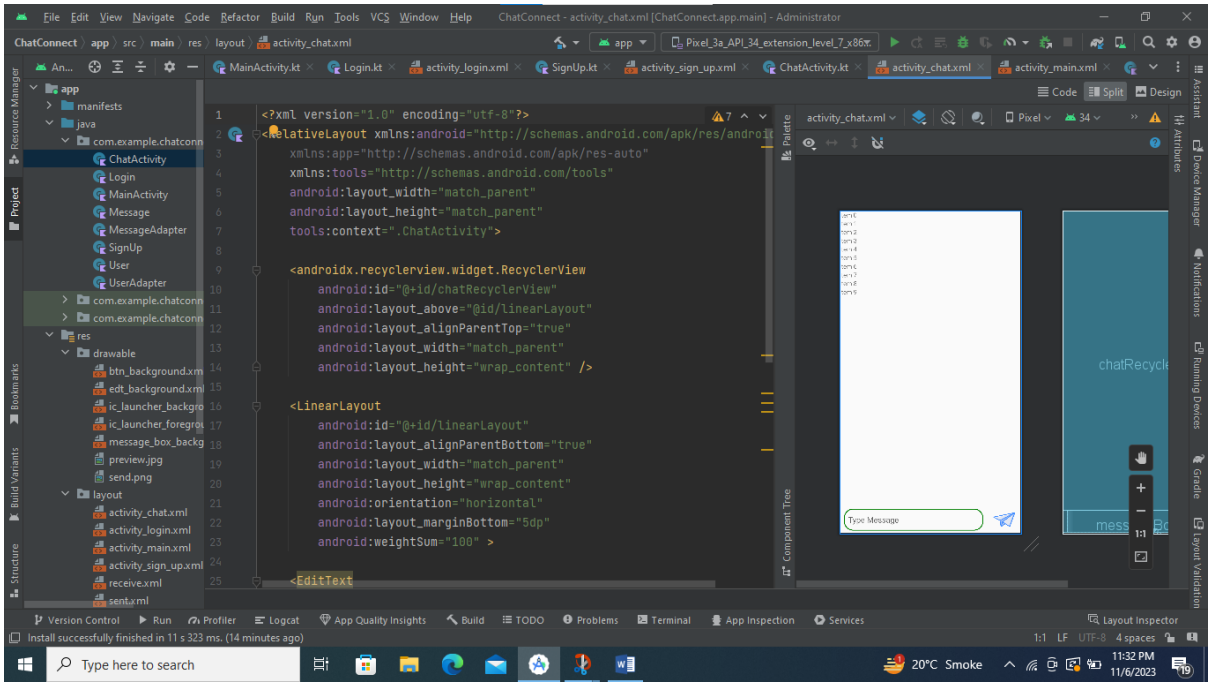


```
package com.example.chatconnect

class Message {
    var message: String? = null
    var senderId: String? = null

    constructor(){}

    constructor(message: String?, senderId: String?){
        this.message = message
        this.senderId = senderId

    }
}
```

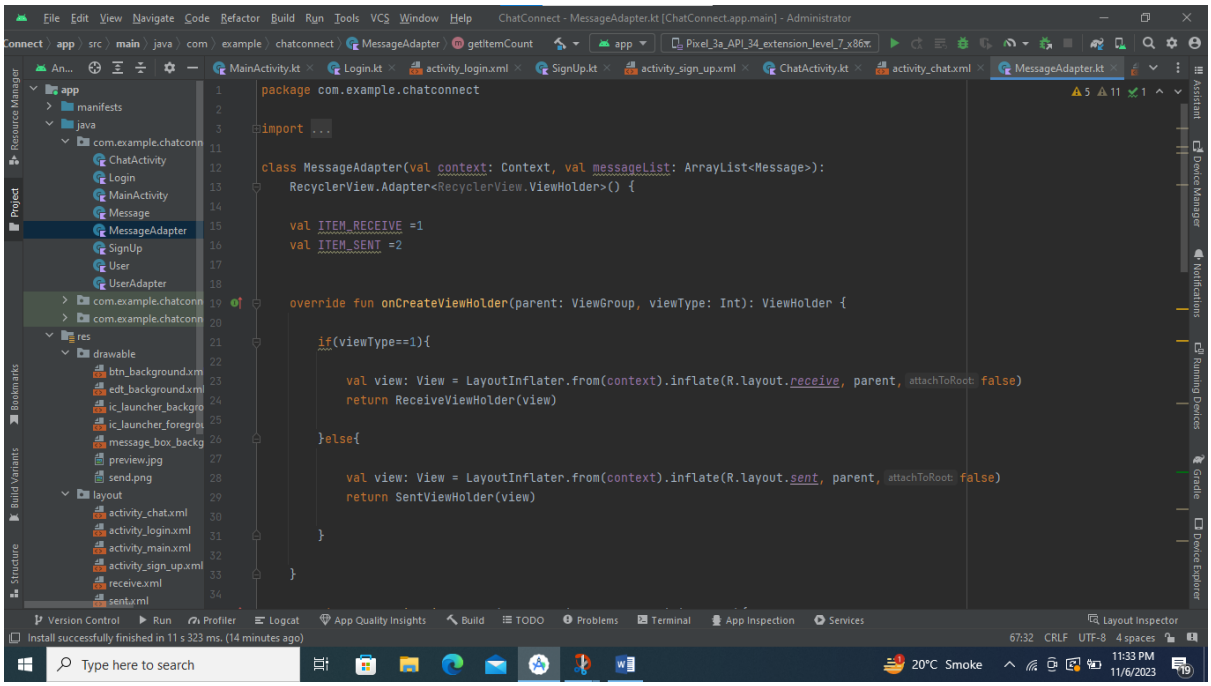
Chat activity



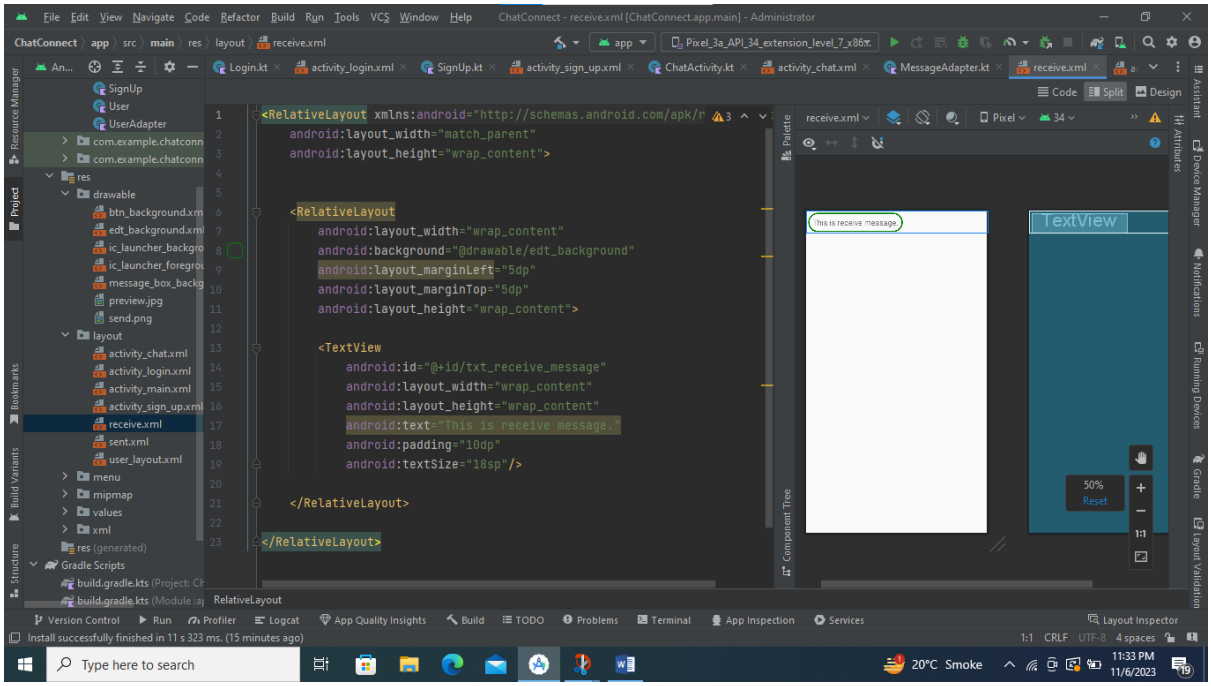
```
package com.example.chatconnect

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Message
import android.widget.EditText
import android.widget.ImageView
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase
import com.google.firebase.database.ValueEventListener

class ChatActivity : AppCompatActivity() {

    private lateinit var chatRecyclerView: RecyclerView
    private lateinit var messageBox: EditText
    private lateinit var sendButton: ImageView
    private lateinit var messageAdapter: MessageAdapter
    private lateinit var messageList: ArrayList<com.example.chatconnect.Message>
    private lateinit var mDbRef : DatabaseReference
```

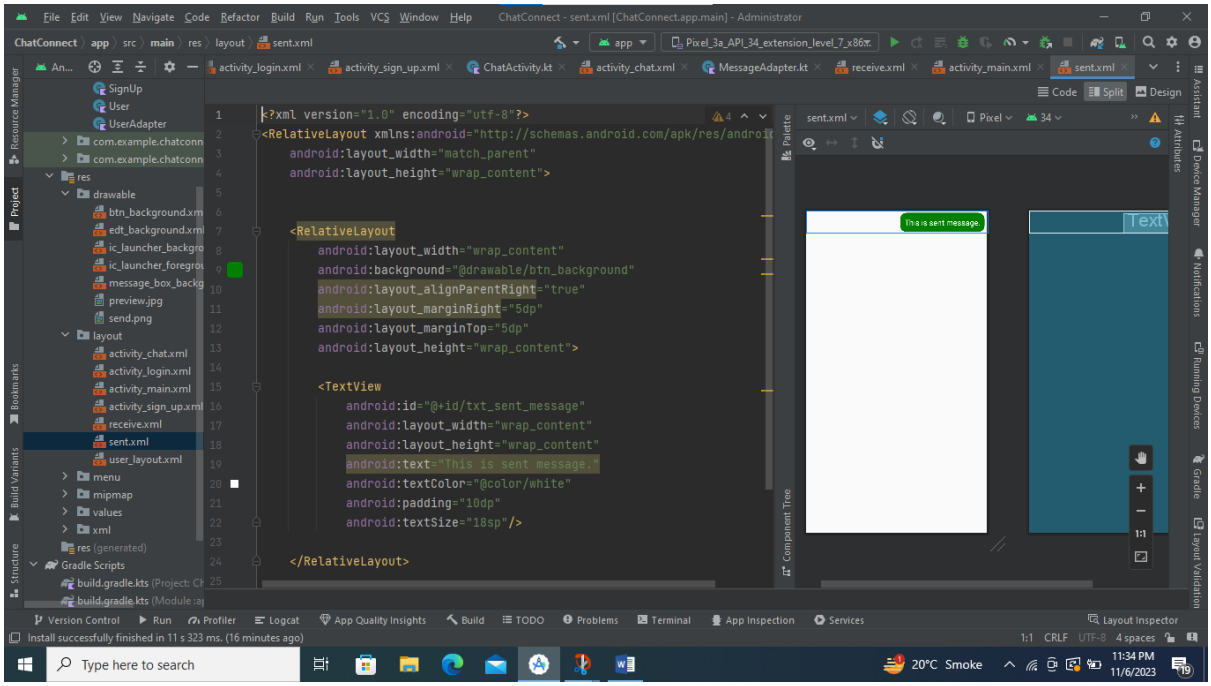


Message adapter

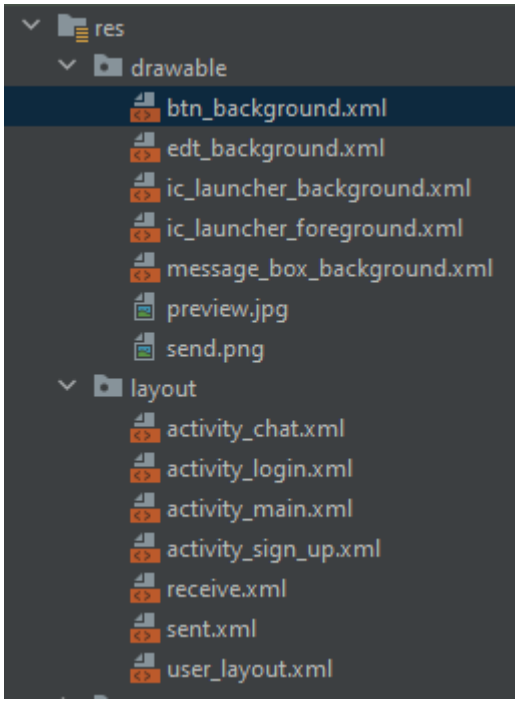


Receive and sent xml file

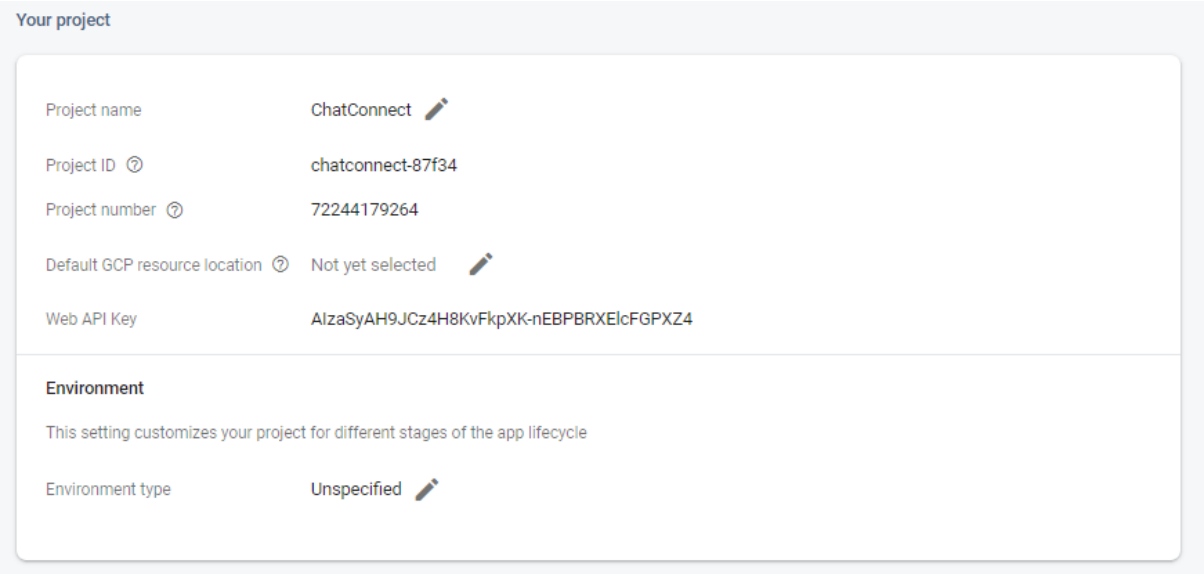




These are some layout and drawable we use



This is our firebase project settings

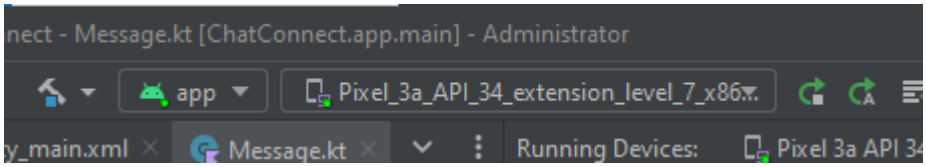


For concluding that we have made this project.

Running the application.

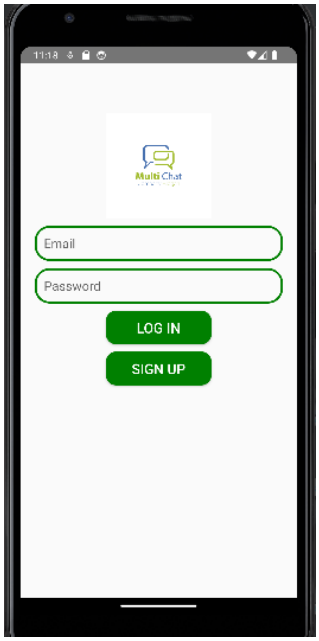
Step 1: Run apps on a hardware device
https://developer.android.com/studio/run/device

Step 2: Run the application in Mobile

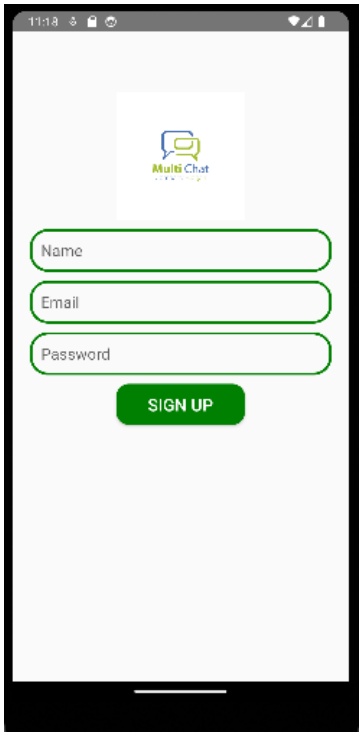


Final Output of the Application:

Login page

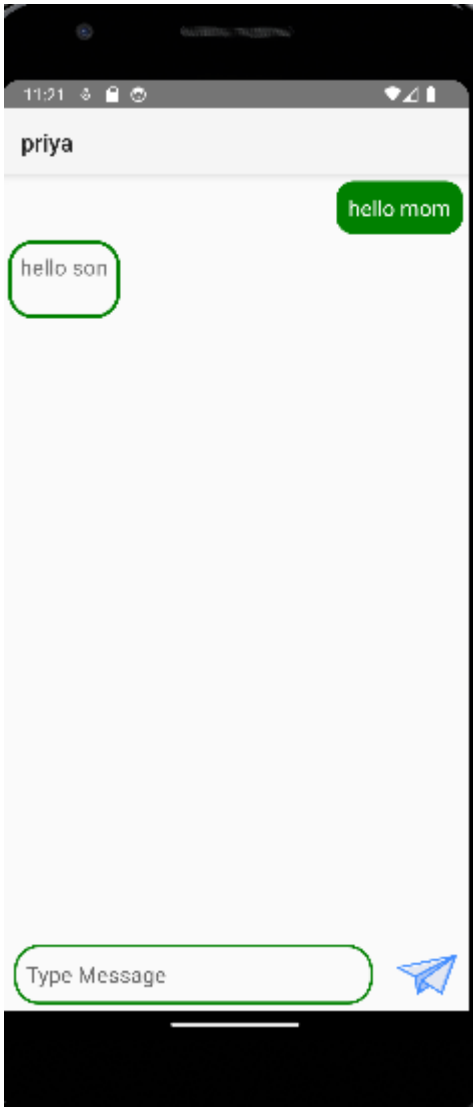


Sign up page



Main page

Sender and reciver chat



Our successful compiler ss