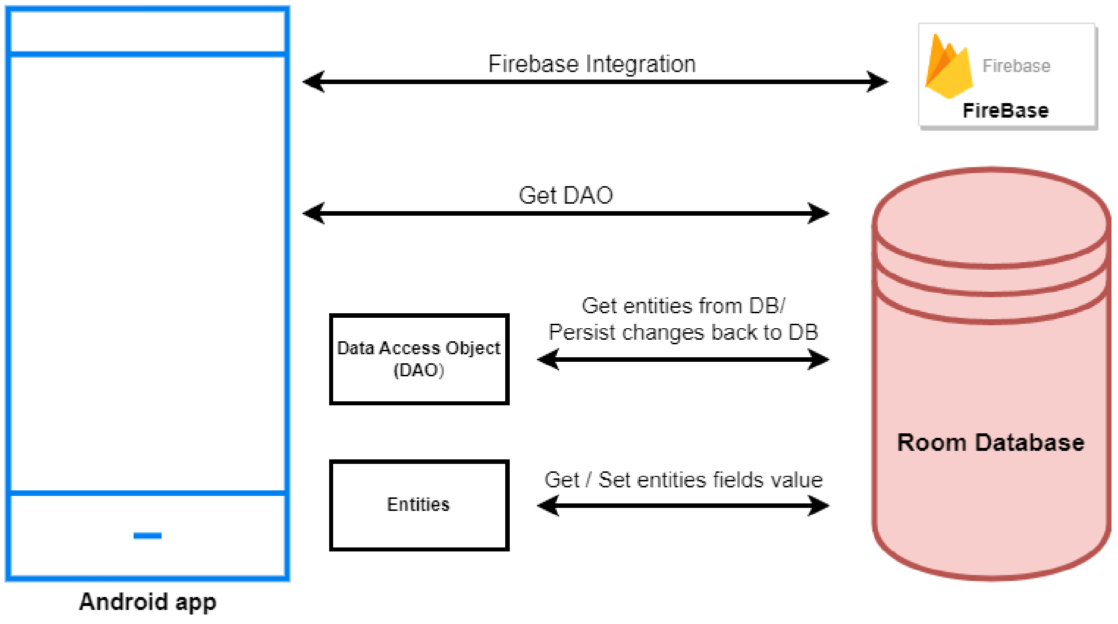
# ChatConnect - A Real-Time Chat and Communication App

Project Based Experiential Learning Program

# ChatConnect - A Real-Time Chat and Communication App

ChatConnect 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.

## Architecture



## 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:

- User Register and login using Phone number and OTP.
- Search for user and chat.
- And can change profile details.

## Tasks:

1.Required initial steps
2.Creating a new project
3.Integrating Firebase and Authentication
4.Creating UI files
5.Running the application.

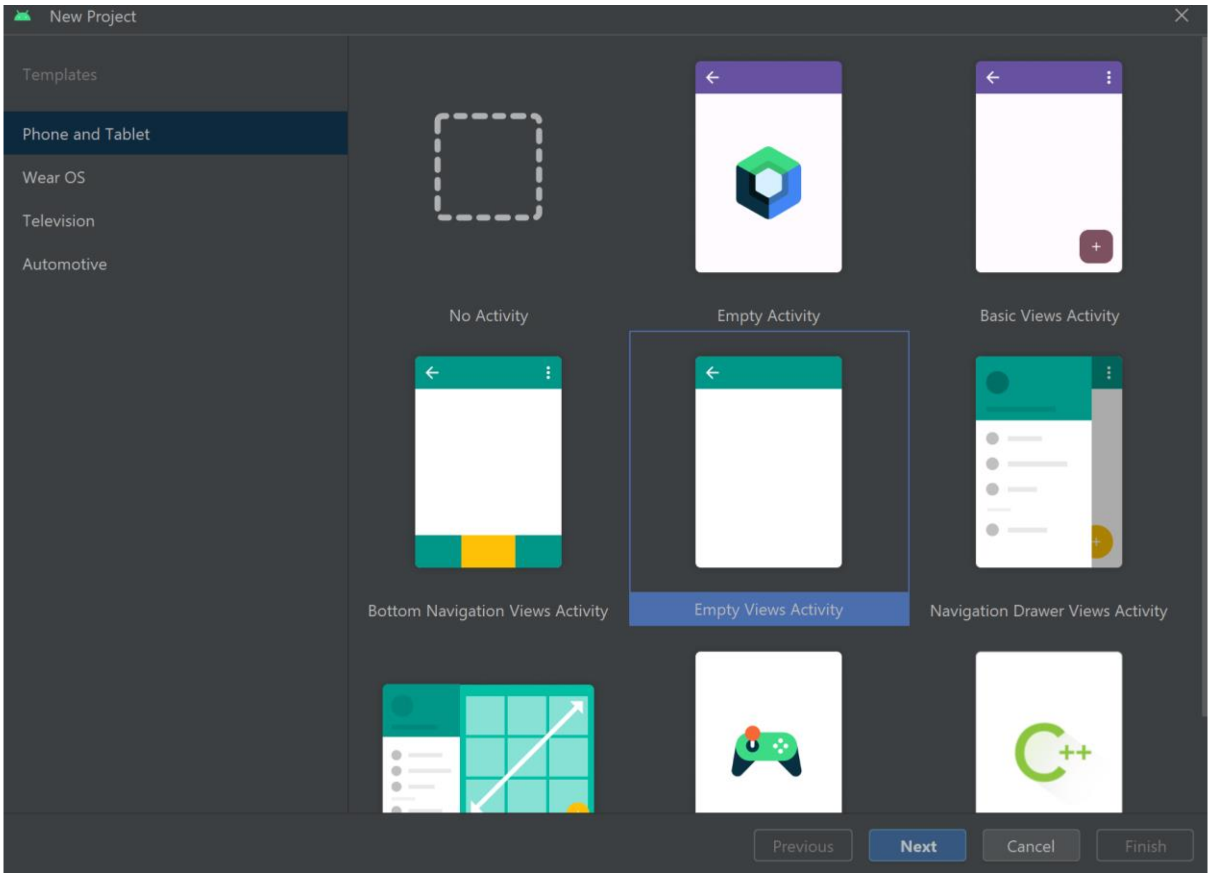
## Task 1:

Required initial steps:

https://developer.android.com/studio/install

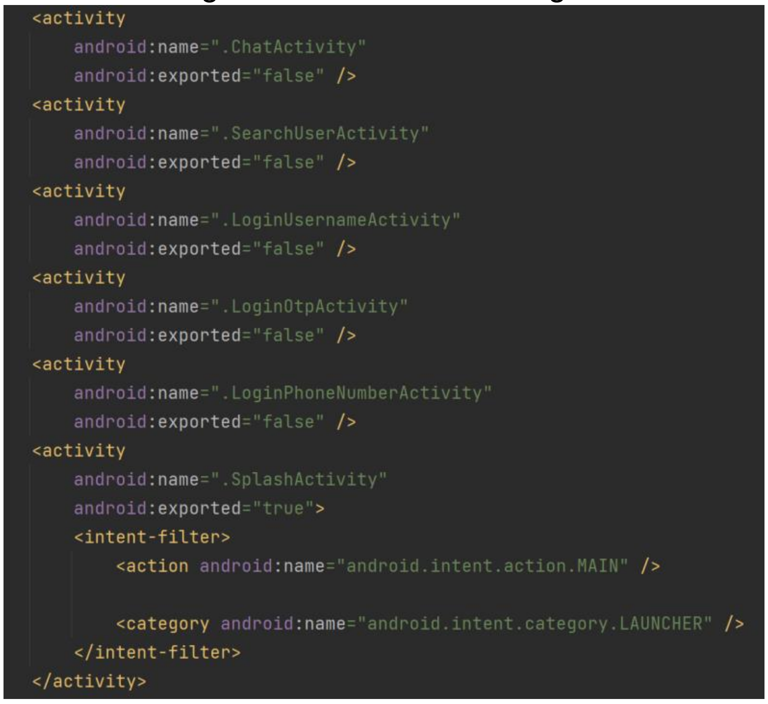
## Task 2:

Creating a new project.
Step 1: Android studio > File > New > New Project > Empty View Activity Step 2 : Click on **Next** button.



Step 3: Give name to the new project.
Step 4: Give the Minimum SDK value(We used API 26:Android 8.0(Oreo) in this project)
Step 5: we did this project in java so select java
Step 6: Click Finish and do following steps

- Main activity file (This contains chatting related code)
- https://github.com/smartinternz02/SI-GuidedProject-587527-1696962169/blob/master/Project%20Code/app/src/main/java/com/example/chatconnect/MainActivity.java
- The first screen we want to see is logo so create a SplashActivity as given below
- https://github.com/smartinternz02/SI-GuidedProject-587527-1696962169/blob/master/Project%20Code/app/src/main/java/com/example/chatconnect/SplashActivity.java
- we must change the manifest file change it like shown below



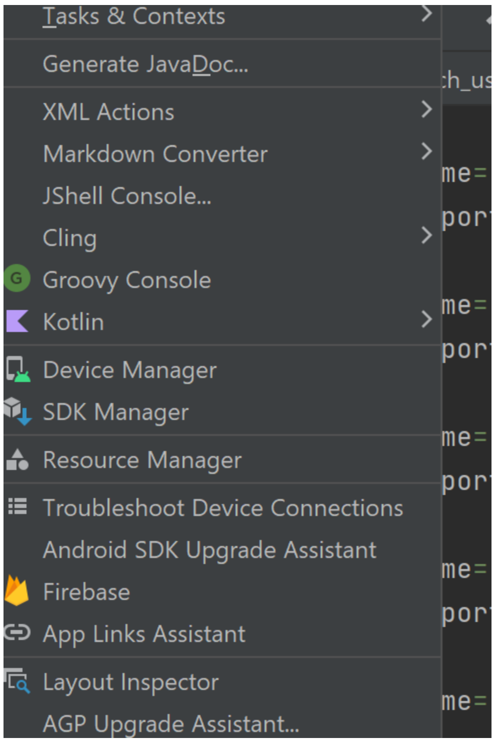
```
<activity
    android:name=".ChatActivity"
    android:exported="false" />
<activity
    android:name=".SearchUserActivity"
    android:exported="false" />
<activity
    android:name=".LoginUsernameActivity"
    android:exported="false" />
<activity
    android:name=".LoginOtpActivity"
    android:exported="false" />
<activity
    android:name=".LoginPhoneNumberActivity"
    android:exported="false" />
<activity
    android:name=".SplashActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

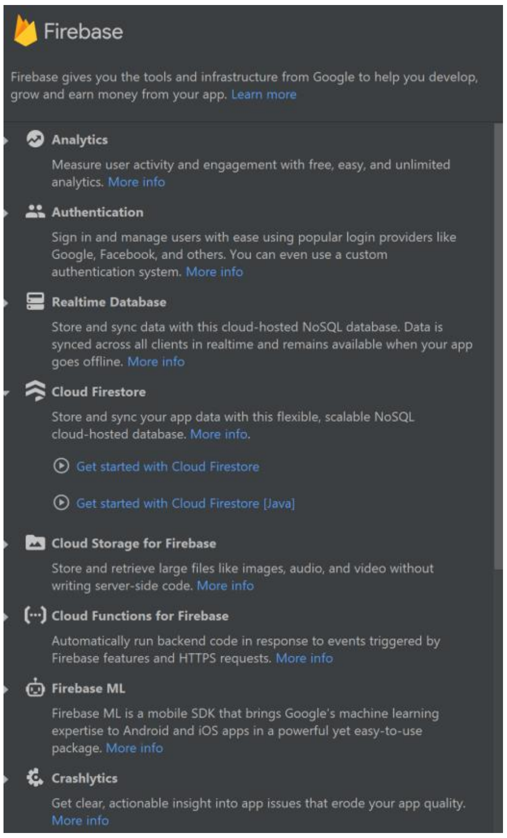
## Task 3 :

Integrating Firebase

- Menu bar > Tools > Firebase.>Cloud Firestore



- After clicking the Firebase, Firebase tab will open.



Firebase

Firebase gives you the tools and infrastructure from Google to help you develop, grow and earn money from your app. Learn more

**Analytics**

Measure user activity and engagement with free, easy, and unlimited analytics. More info

**Authentication**

Sign in and manage users with ease using popular login providers like Google, Facebook, and others. You can even use a custom authentication system. More info

**Realtime Database**

Store and sync data with this cloud-hosted NoSQL database. Data is synced across all clients in realtime and remains available when your app goes offline. More info

**Cloud Firestore**

Store and sync your app data with this flexible, scalable NoSQL cloud-hosted database. More info.

Get started with Cloud Firestore

Get started with Cloud Firestore [Java]

**Cloud Storage for Firebase**

Store and retrieve large files like images, audio, and video without writing server-side code. More info

**Cloud Functions for Firebase**

Automatically run backend code in response to events triggered by Firebase features and HTTPS requests. More info

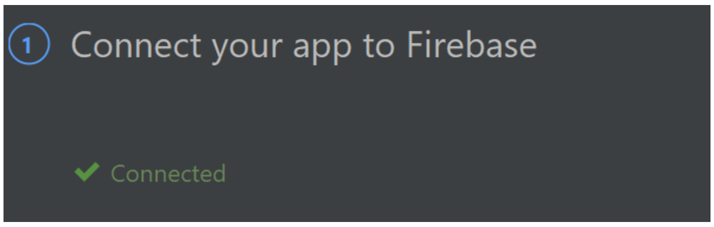
**Firebase ML**

Firebase ML is a mobile SDK that brings Google's machine learning expertise to Android and iOS apps in a powerful yet easy-to-use package. More info

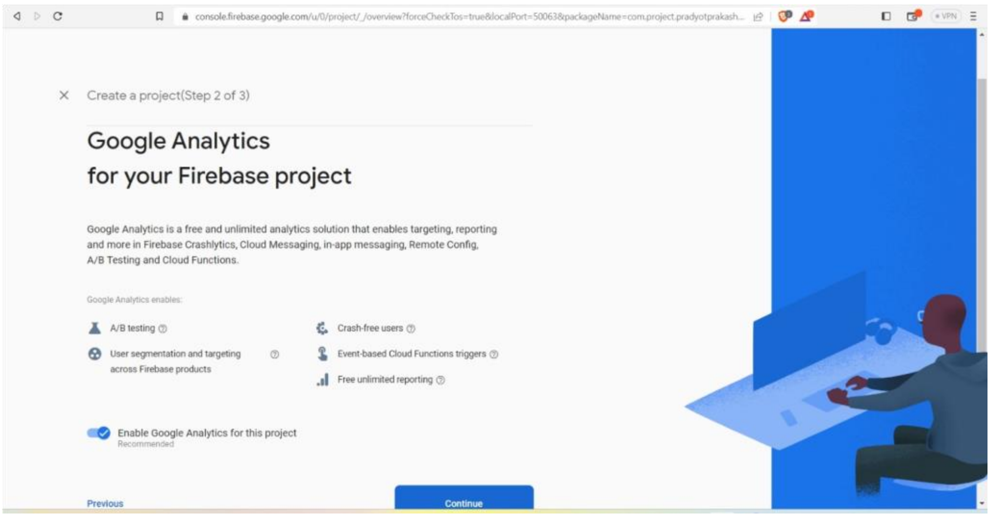
**Crashlytics**

Get clear, actionable insight into app issues that erode your app quality. More info

- In Cloud FireStore > Go to Get started with Cloud FireStore. And click on connect your app with firestore



- 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, Allow android studio to access your google account.
- Now Click on **Firebase**, Now the Firebase page will open and write name for the project and click continue
- Click on **continue**


console.firebase.google.com/u/0/project/_/overview?forceCheckTos=true&localPort=50063&packageName=com.project.pradyotprakash...

Create a project(Step 2 of 3)

Google Analytics
for your Firebase project

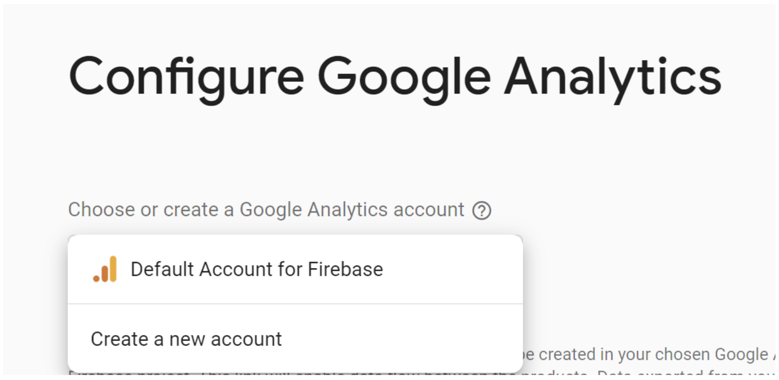
Google Analytics is a free and unlimited analytics solution that enables targeting, reporting and more in Firebase Crashlytics, Cloud Messaging, in-app messaging, Remote Config, A/B Testing and Cloud Functions.

Google Analytics enables:

A/B testing
User segmentation and targeting across Firebase products
Crash-free users
Event-based Cloud Functions triggers
Free unlimited reporting

Enable Google Analytics for this project
Recommended

- Now enable the Google Analytics for this project and
- Click on **continue.**
- Set Default Account for firebase.

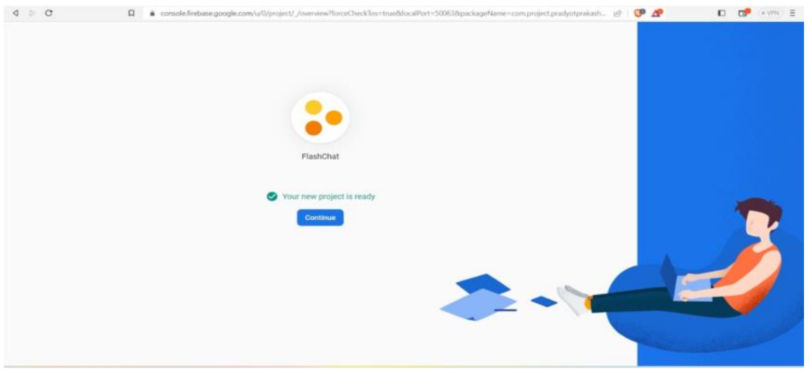

Configure Google Analytics

Choose or create a Google Analytics account

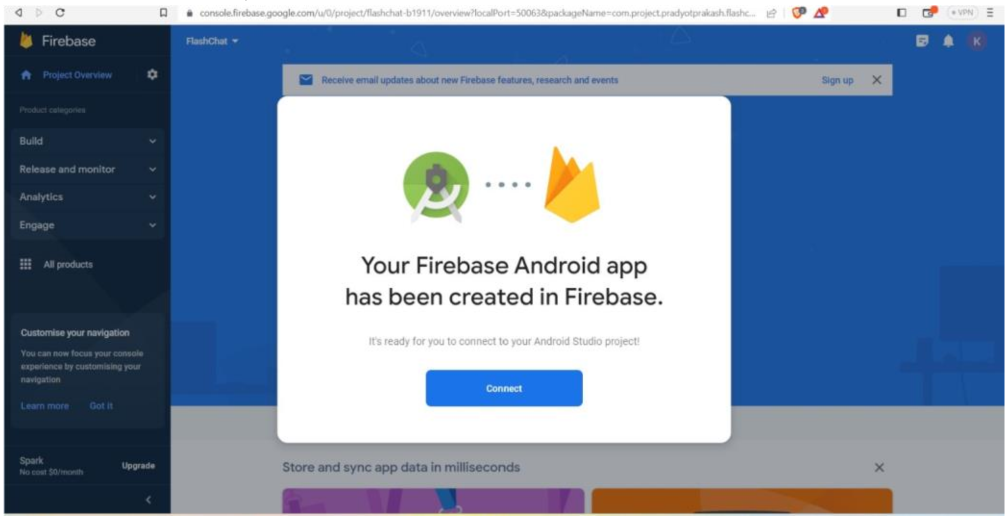
Default Account for Firebase
Create a new account

e created in your chosen Google

- Now Click on **Create project** and after click on **continue.**



- Click connect, firebase will be connected to android studio


Firebase
Project Overview
Build
Release and monitor
Analytics
Engage
All products
Receive email updates about new Firebase features, research and events
Sign up
Connect
Store and sync app data in milliseconds

Your Firebase Android app has been created in Firebase.

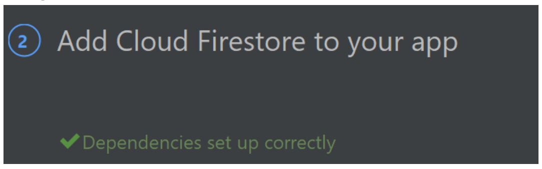
It's ready for you to connect to your Android Studio project!

- This shows that your project is connected to Firebase.

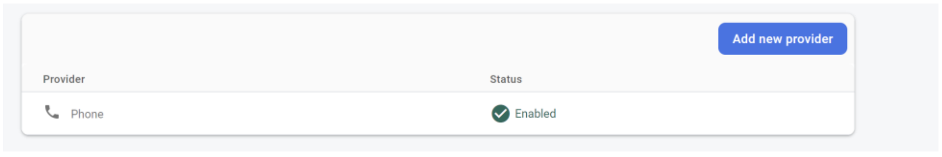
- Open Android Studio Menu bar > Tools > Firebase.

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

- In Cloud Firestore> Click on get started with cloud firestore and click add dependencies

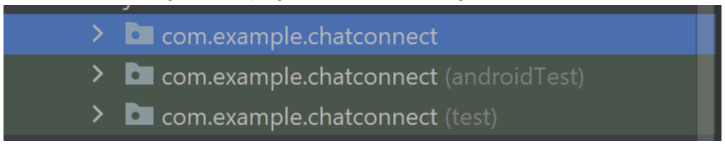

2 Add Cloud Firestore to your app
Dependencies set up correctly
.

- Go to Firebase in google and click on your project and go to Authentication>Sing-in methods>enable



| Provider | Status |
|---|---|
| Phone | ✓ Enabled |

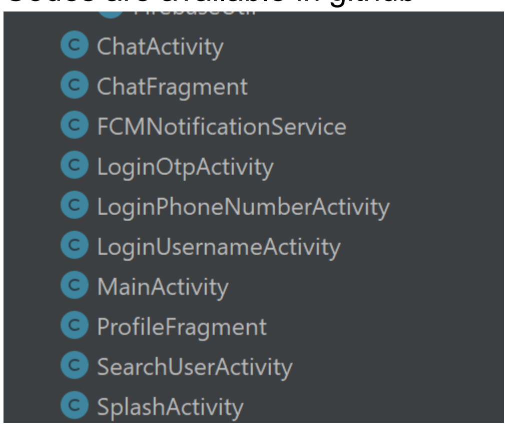
- Go to Firebase (in tools) in android studio as did above and go to Authentication>Authentication using custom authentication>add dependencies
- Do the same for Cloud Messaging
- Right click on com.example.chatconnect the one in blue and go to new>Activity>Empty view Activity



```
> com.example.chatconnect
> com.example.chatconnect (androidTest)
> com.example.chatconnect (test)
```

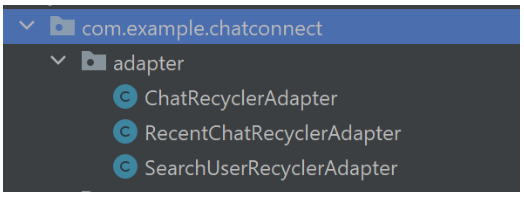
- Make sure every gradle dependencies are called, **<u>compare both the files in github and your file and write dependencies that are not present in your build.gradle.kts (Module:app) file.</u>**

- Follow above steps and create LoginPhoneNumberActivity, LoginOtpActivity, LoginUsernameActivity, SearchUserActivity, ChatActivity
- Codes are available in github



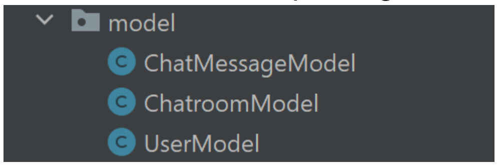
```
ChatActivity
ChatFragment
FCMNotificationService
LoginOtpActivity
LoginPhoneNumberActivity
LoginUsernameActivity
MainActivity
ProfileFragment
SearchUserActivity
SplashActivity
```

- Now right click on com.example.chatconnect new>Fragment>BlankFragment and create ChatFragment, ProfileFragment
- Now again click on com.example.chatconnect new>package>name it adapter and then right click on package and create following java classes.



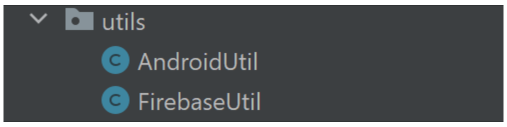
```
v com.example.chatconnect
  v adapter
      ChatRecyclerAdapter
      RecentChatRecyclerAdapter
      SearchUserRecyclerAdapter
```

- Now create another package model with following java classes

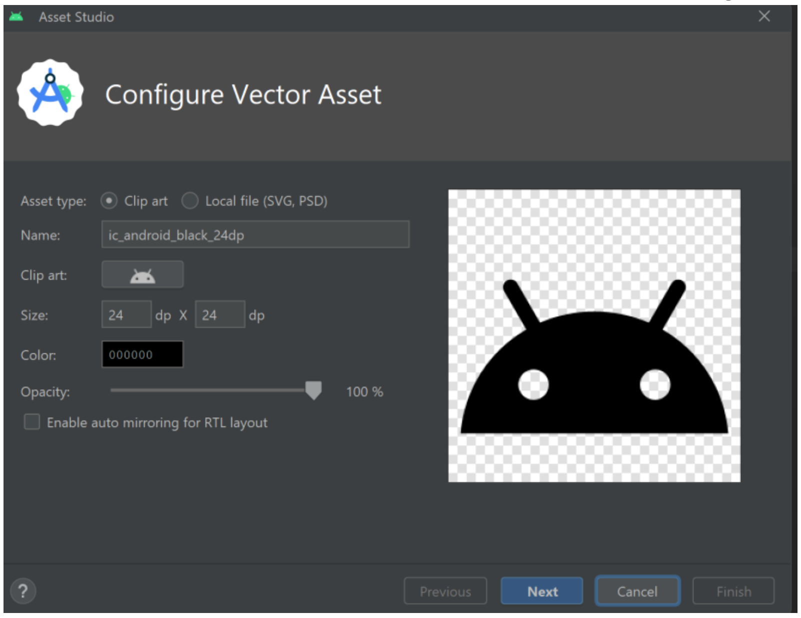


```
v model
    ChatMessageModel
    ChatroomModel
    UserModel
```

- Now create another package named utils with following java classes

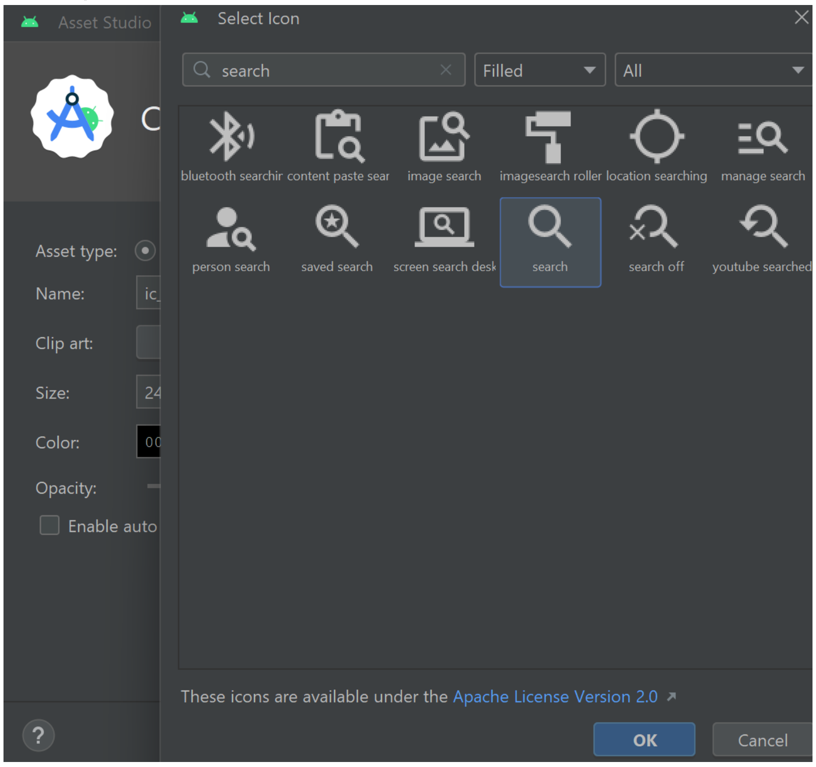

```
utils
    AndroidUtil
    FirebaseUtil
```

- You can find all the codes of adapter, model, utils in the github.
- Now create drawable resource files by res>drawable and right click on drawable>new(if you want custom you can write code). There are some inbuilt icons that you can use
- To create a drawable file from built in icons first right click on drawable>new>vector asset it will look like following image.



Asset type: [✓ Clip art] Local file (SVG, PSD)
Name: ic_android_black_24dp
Clip art:
Size: 24 dp X 24 dp
Color: 000000
Opacity: 100 %
Enable auto mirroring for RTL layout

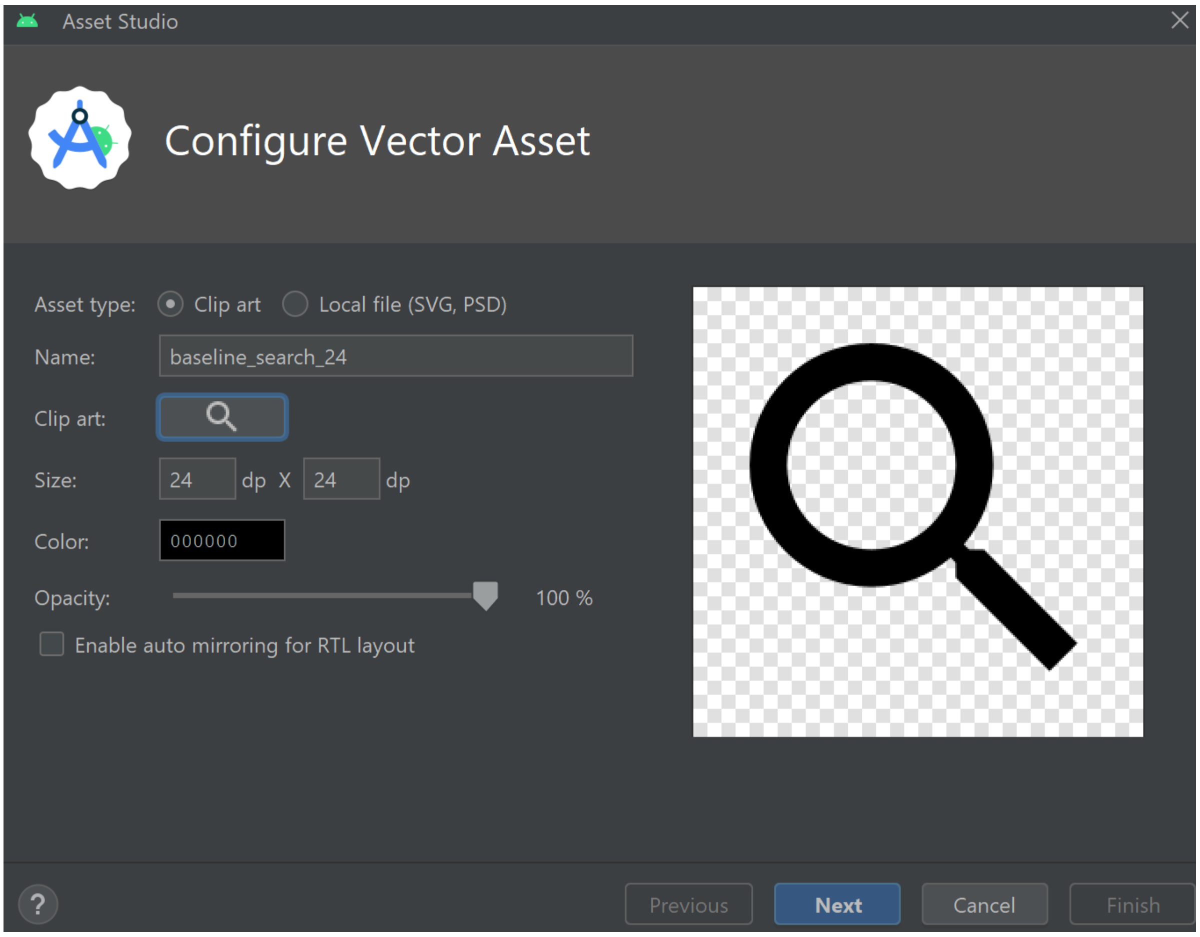
- Now click on clip art image and then search for the art you are looing for an example is shown for search icon.



search | Filled | All

These icons are available under the Apache License Version 2.0

- Now click ok and then it will look like this.



Asset Studio

Configure Vector Asset

Asset type: [✓ Clip art] Local file (SVG, PSD)

Name: baseline_search_24

Clip art:

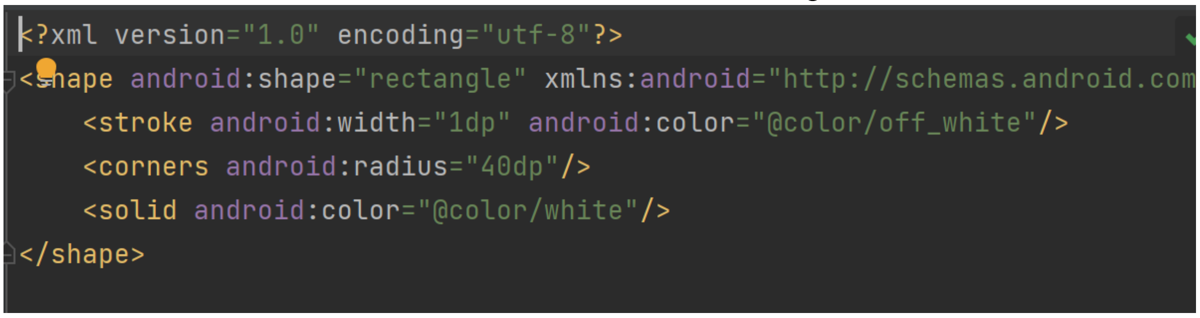
Size: 24 dp X 24 dp

Color: 000000

Opacity: 100 %

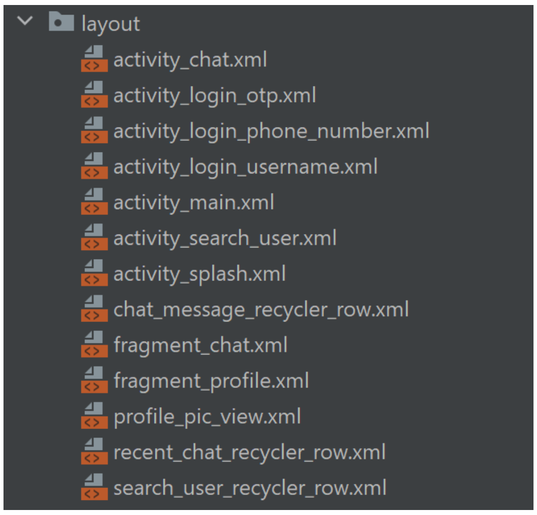
Enable auto mirroring for RTL layout

- You can adjust size as you wish.
- You can design a drawable file by right clicking on drawable and then select drawable resource file and then write a code like following.



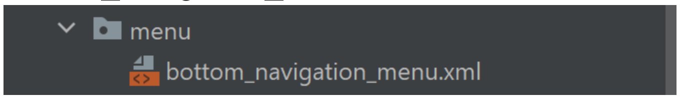
```
<?xml version="1.0" encoding="utf-8"?>
<shape android:shape="rectangle" xmlns:android="http://schemas.android.com
    <stroke android:width="1dp" android:color="@color/off_white"/>
    <corners android:radius="40dp"/>
    <solid android:color="@color/white"/>
</shape>
```

- Or you can copy paste images from your device into the drawable file.
- Now let's see App layout files, For the activities and fragments we created a layout file will be generated. Other than these create chat_message_recycler_row.xml, profile_pic_view.xml, recent_chat_recycler_row.xml, search_user_recycler_row.xml
- Now write the codes for each file as given in the github. You can modify the colours and design according to your need



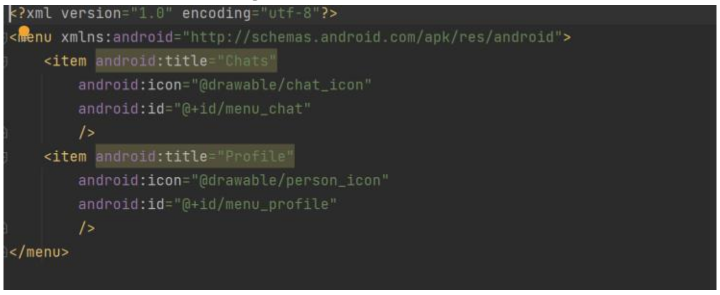
```
layout
    activity_chat.xml
    activity_login_otp.xml
    activity_login_phone_number.xml
    activity_login_username.xml
    activity_main.xml
    activity_search_user.xml
    activity_splash.xml
    chat_message_recycler_row.xml
    fragment_chat.xml
    fragment_profile.xml
    profile_pic_view.xml
    recent_chat_recycler_row.xml
    search_user_recycler_row.xml
```

- Now create a directory named menu and create a xml file name bottom_navigation_menu.xml in it.



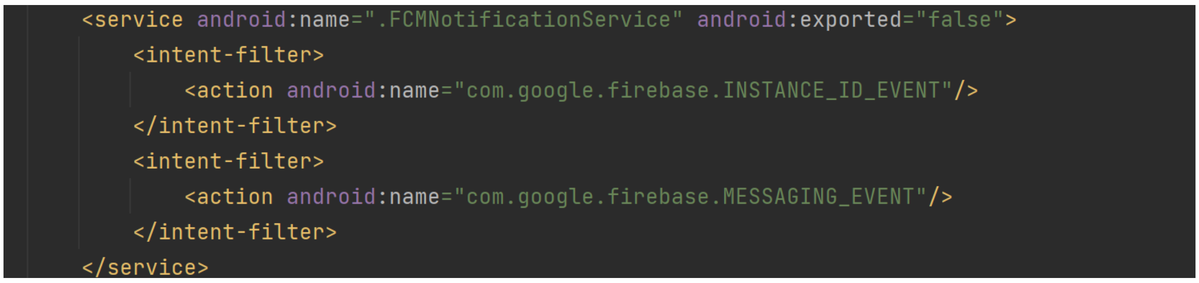
```
menu
    bottom_navigation_menu.xml
```

- Write the following code in it.



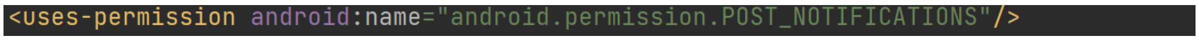
```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:title="Chats"
        android:icon="@drawable/chat_icon"
        android:id="@+id/menu_chat"
        />
    <item android:title="Profile"
        android:icon="@drawable/person_icon"
        android:id="@+id/menu_profile"
        />
</menu>
```

- Now write the following code in AndroidManifest file so that you will receive notifications.



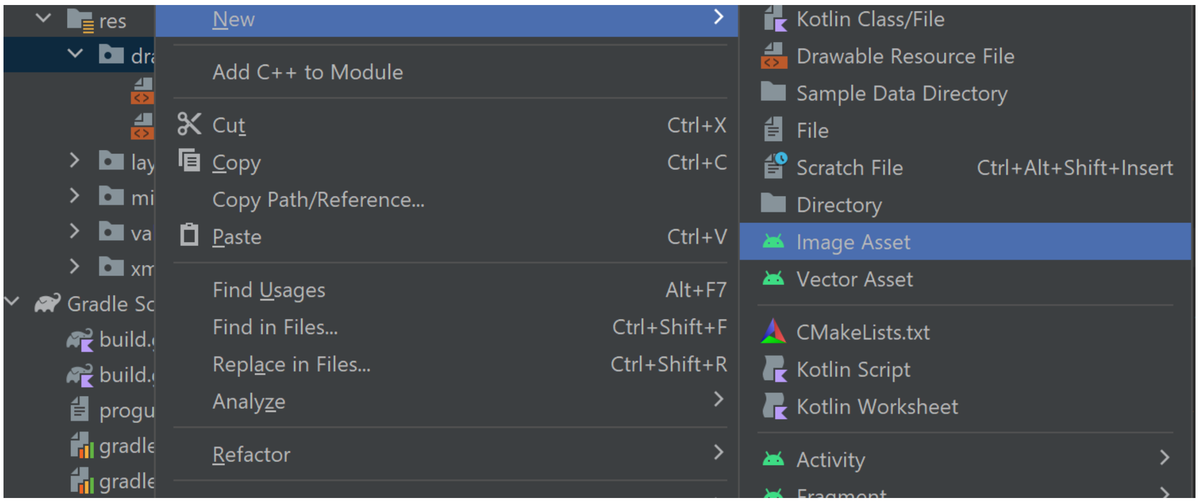
```xml
<service android:name=".FCMNotificationService" android:exported="false">
    <intent-filter>
        <action android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
    </intent-filter>
    <intent-filter>
        <action android:name="com.google.firebase.MESSAGING_EVENT"/>
    </intent-filter>
</service>
```

- Now add this code in AndroidManifest file, so that app will ask permission from user to allow notifications.

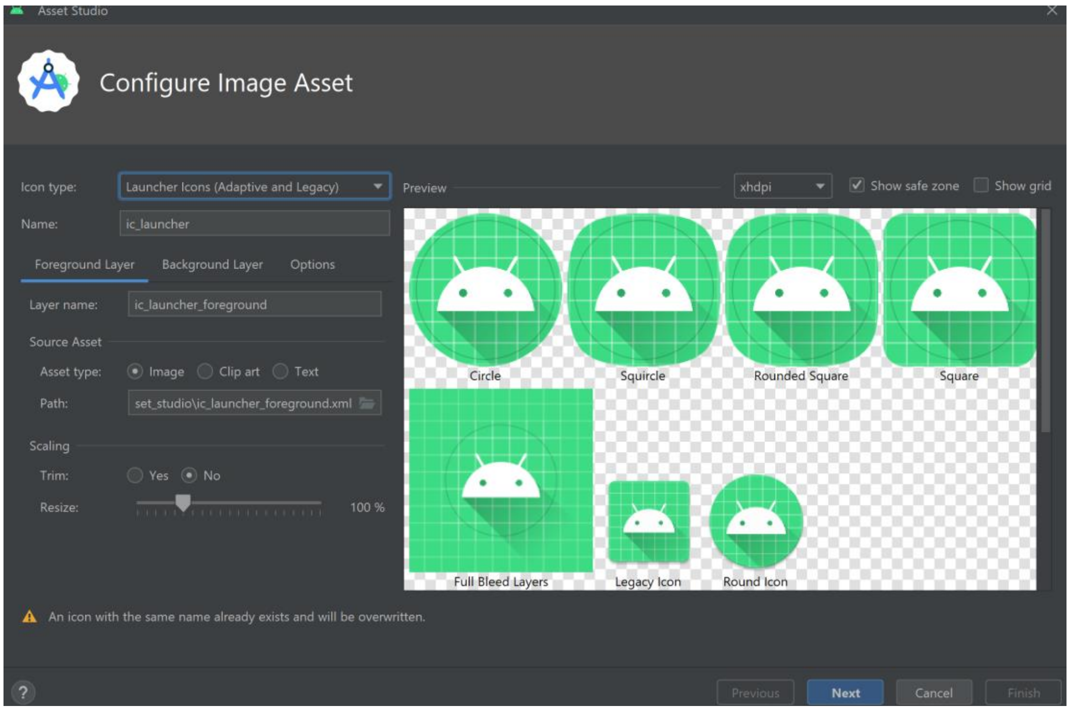


```xml
<uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
```

- Add the above codes like this. https://github.com/smartinternz02/SI-GuidedProject-587527-1696962169/blob/master/Project%20Code/app/src/main/AndroidManifest.xml
- Now let’s create icon for our app.
- Go to res>drawable>right click>Image asset



- Now it will show this.


Asset Studio

Configure Image Asset

Icon type: Launcher Icons (Adaptive and Legacy)

Name: ic_launcher

Foreground Layer | Background Layer | Options

Layer name: ic_launcher_foreground

Source Asset

Asset type: [✓ Image] Clip art Text

Path: set_studio\ic_launcher_foreground.xml

Scaling

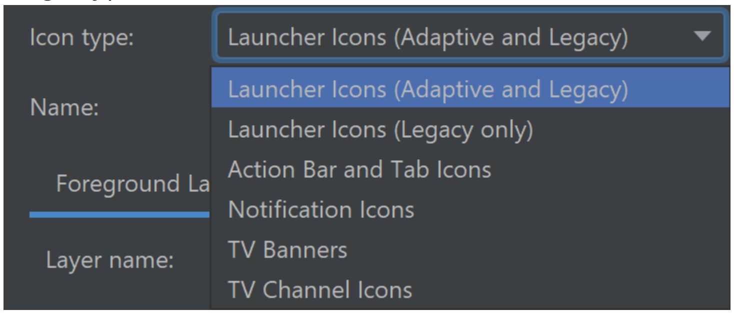
Trim: Yes [✓ No]

Resize: 100 %

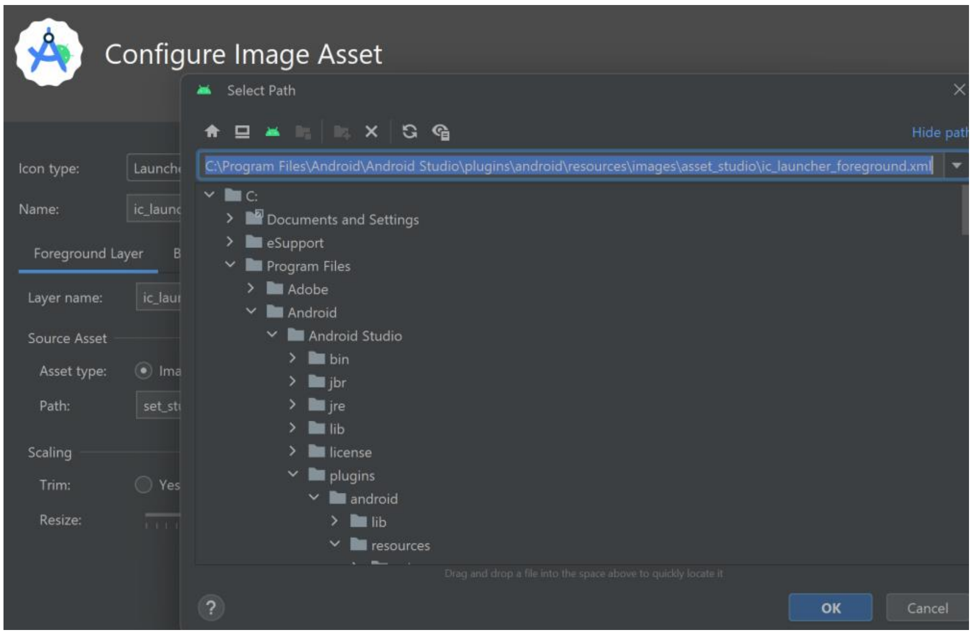
Preview: xhdpi [✓ Show safe zone] Show grid

An icon with the same name already exists and will be overwritten.

- Now click on Icon type selection bar and select Launcher Icons(Adaptive and Legacy)


Icon type: Launcher Icons (Adaptive and Legacy)

- Now in the path you can select the path of whatever image you want to be your icon. You can select clipart or Text as well for icon.


Configure Image Asset

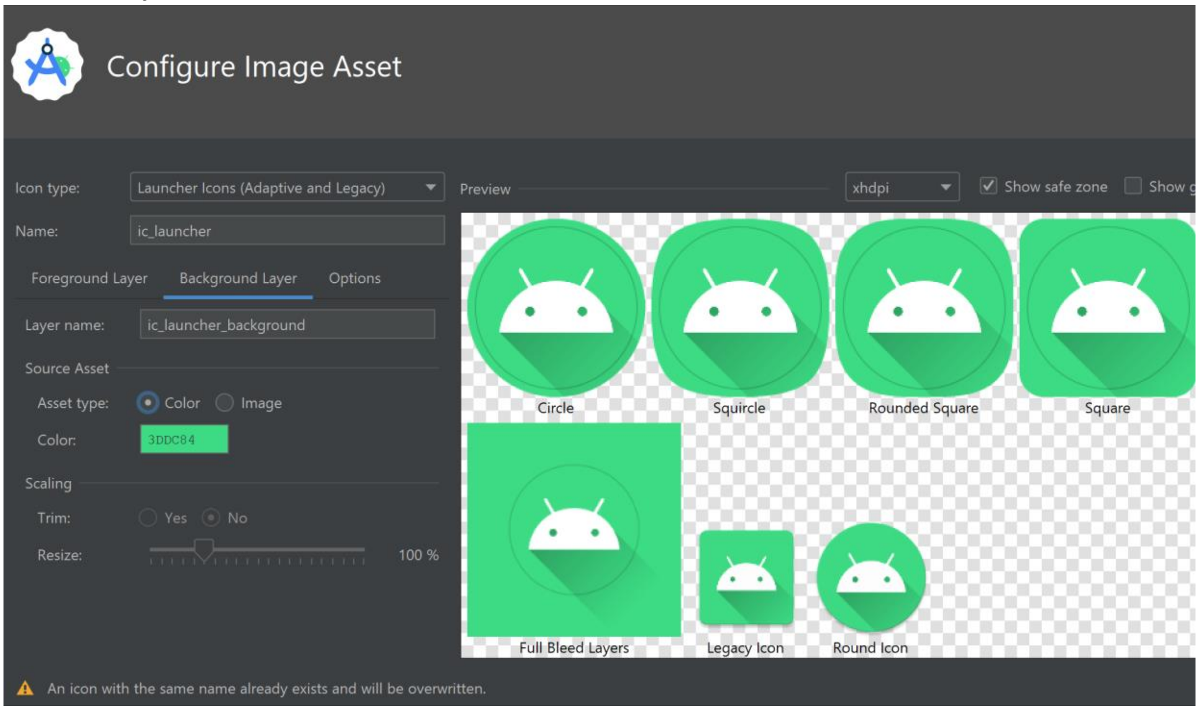
Select Path

C:\Program Files\Android\Android Studio\plugins\android\resources\images\asset_studio\ic_launcher_foreground.xml

```
C:
  Documents and Settings
  eSupport
  Program Files
    Adobe
    Android
      Android Studio
        bin
        jbr
        jre
        lib
        license
        plugins
          android
            lib
            resources
```

Drag and drop a file into the space above to quickly locate it

- Select path and click ok.
- You can resize using Resize bar.
- And now you can edit the background layer as well.
- Similarly in background layer as well you can select file path like you did before or you can select colour.


Configure Image Asset

Icon type: Launcher Icons (Adaptive and Legacy)

Name: ic_launcher

Foreground Layer | Background Layer | Options

Layer name: ic_launcher_background

Source Asset

Asset type: [✓ Color] Image

Color: 3DDC84

Scaling

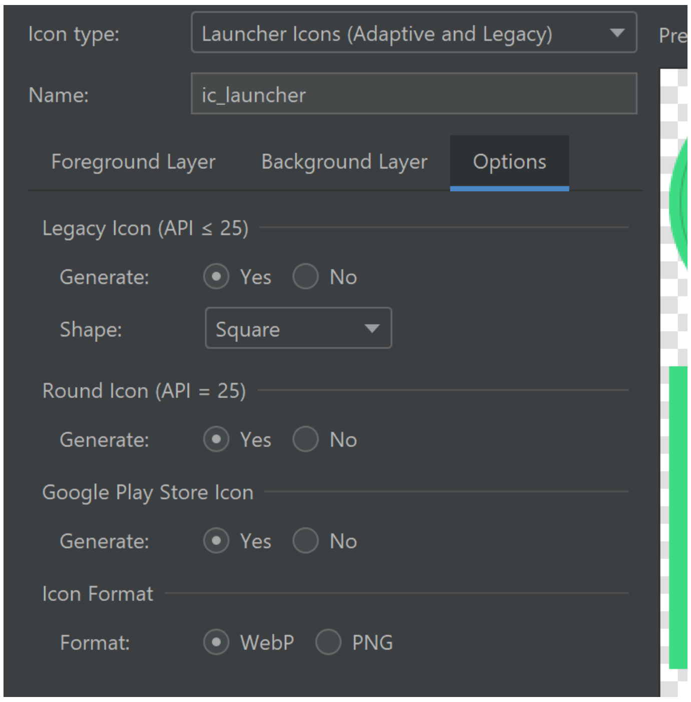
Trim: Yes No

Resize: 100 %

Preview: xhdpi ✓ Show safe zone

An icon with the same name already exists and will be overwritten.

- As you can see there is a warning that icon name already exist in both Foreground layer and background layer make sure to give a unique name.
- Now you can resize using the Resize bar.
- You can explore other options as well.



Icon type: Launcher Icons (Adaptive and Legacy)
Name: ic_launcher

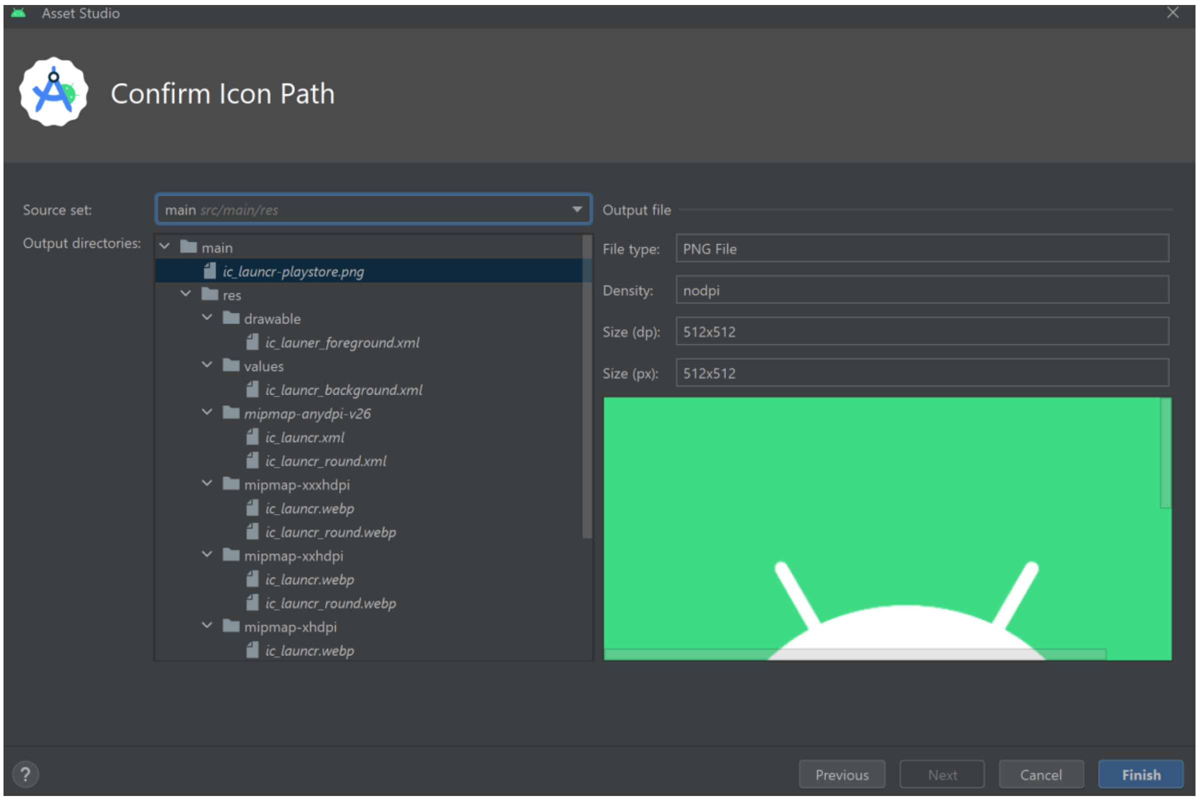
Legacy Icon (API ≤ 25)
Generate: [✓ Yes] No
Shape: Square

Round Icon (API = 25)
Generate: [✓ Yes] No

Google Play Store Icon
Generate: [✓ Yes] No

Icon Format
Format: [✓ WebP] PNG

- Click next and then Finish.

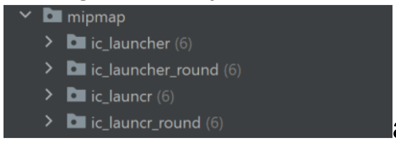


Source set: main src/main/res
Output directories:

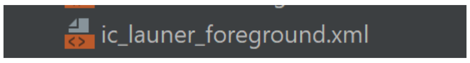
```
main
  ic_launcr-playstore.png
  res
    drawable
      ic_launer_foreground.xml
    values
      ic_launcr_background.xml
    mipmap-anydpi-v26
      ic_launcr.xml
      ic_launcr_round.xml
    mipmap-xxxhdpi
      ic_launcr.webp
      ic_launcr_round.webp
    mipmap-xxhdpi
      ic_launcr.webp
      ic_launcr_round.webp
    mipmap-xhdpi
      ic_launcr.webp
```

Output file
File type: PNG File
Density: nodpi
Size (dp): 512x512
Size (px): 512x512

- The following ic_launcr will be created in mipmap, I changed icon name, Background layer name and foreground layer names so they will be unique.

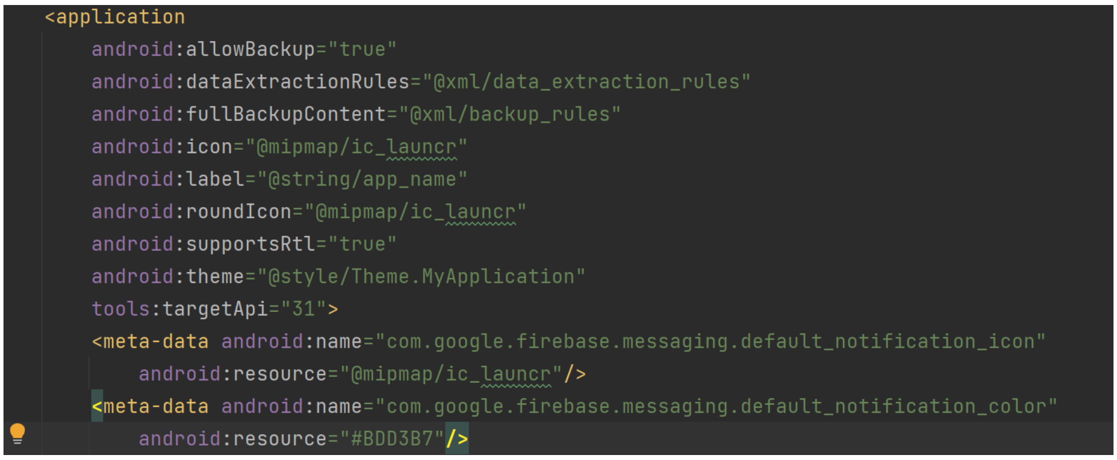
- 

```
mipmap
  ic_launcher (6)
  ic_launcher_round (6)
  ic_launcr (6)
  ic_launcr_round (6)
```

and in drawable 

ic_launer_foreground.xml

- Now go to AndroidManifest file and erite the following code.



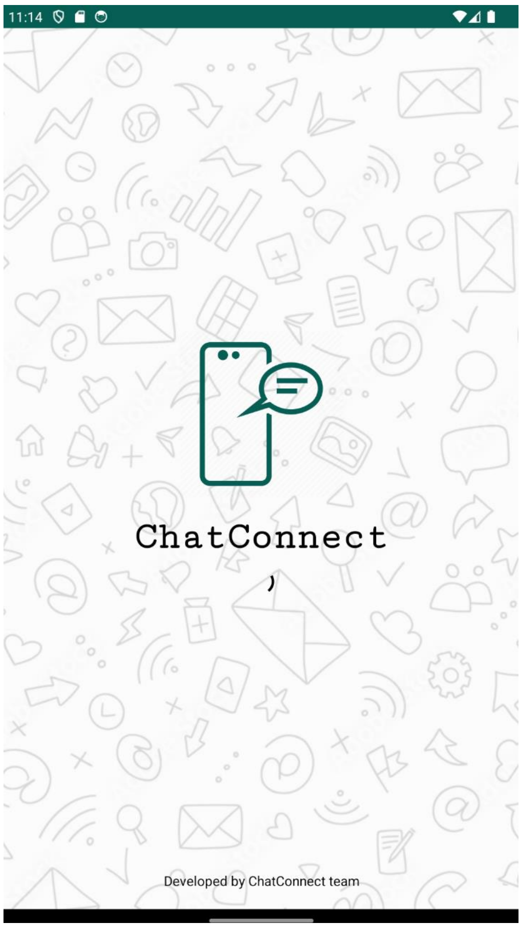
```
<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcr"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcr"
    android:supportsRtl="true"
    android:theme="@style/Theme.MyApplication"
    tools:targetApi="31">
    <meta-data android:name="com.google.firebase.messaging.default_notification_icon"
        android:resource="@mipmap/ic_launcr"/>
    <meta-data android:name="com.google.firebase.messaging.default_notification_color"
        android:resource="#BDD3B7"/>
```

- To demonstrate I used different name, see the above changes carefully and to see the placement of the code in AndroidManifest file refer github.
- Now you are done with every code.

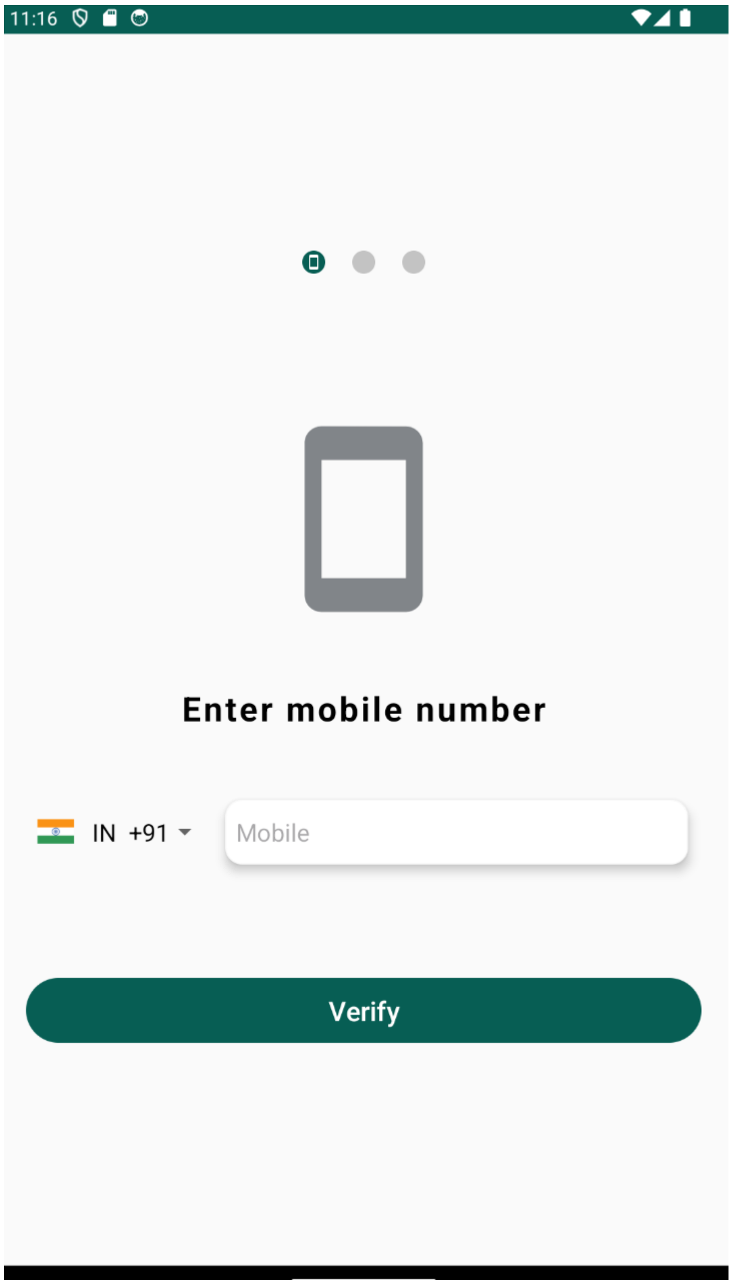
**GITHUB LINK: https://github.com/smartinternz02/SI-GuidedProject-587527-1696962169/tree/master/Project%20Code**

## Final Output of the Application:

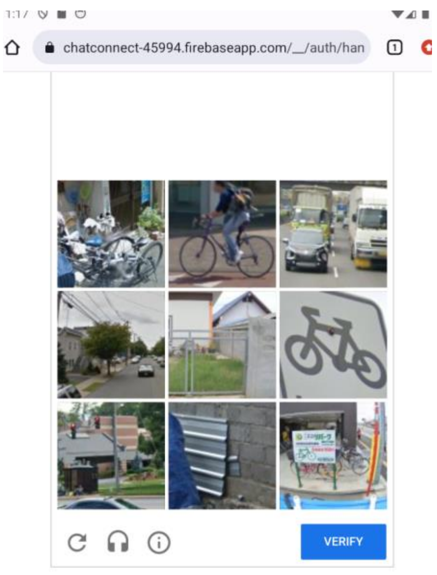
First Screen:



**<u>Sign-in and Sign-up using Phone number and OTP</u>**:

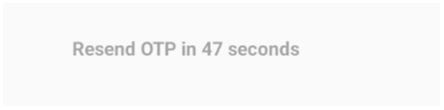


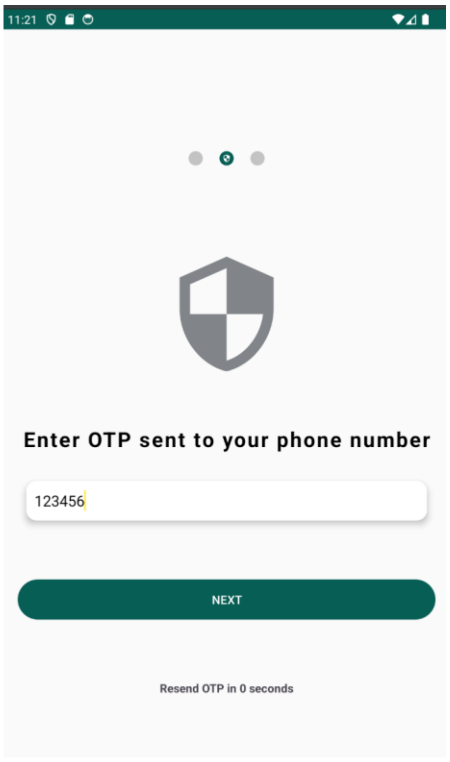
- Next a verification will happen



**OTP Screen**

- "OTP sent successful" this message will be seen on screen and Enter OTP on screen.
- Resend OTP will be available after 60sec.

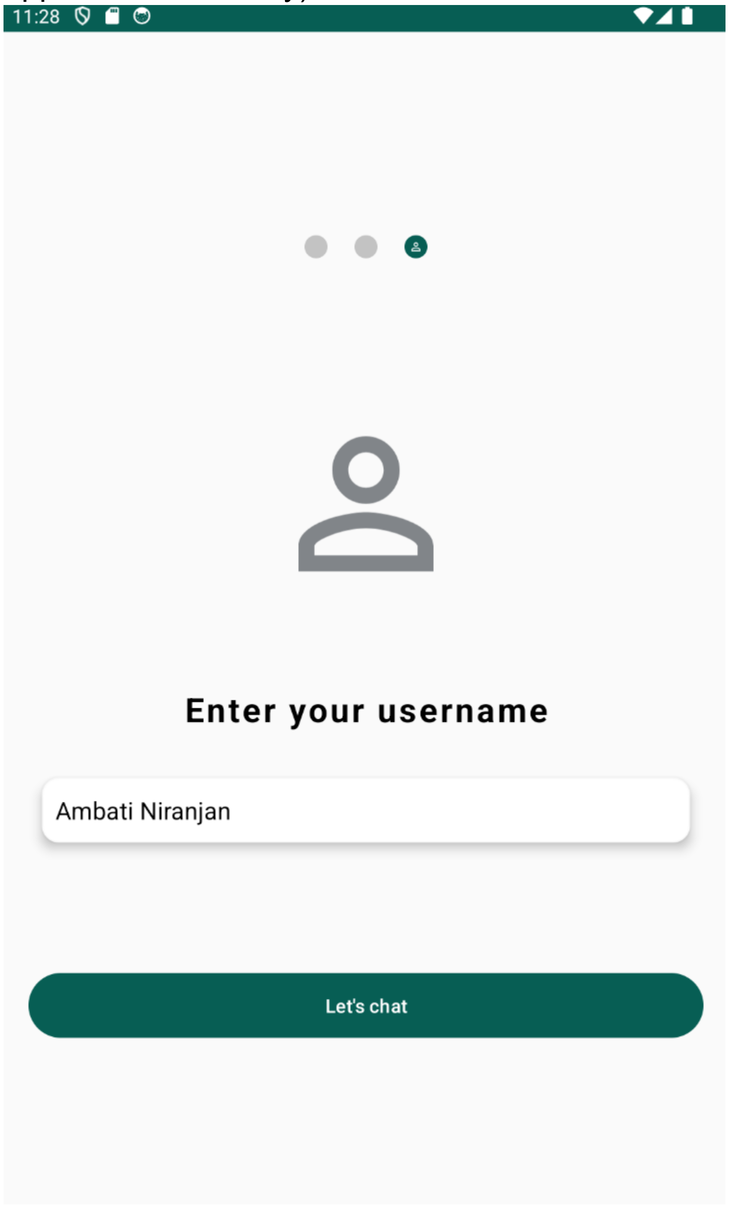


- 

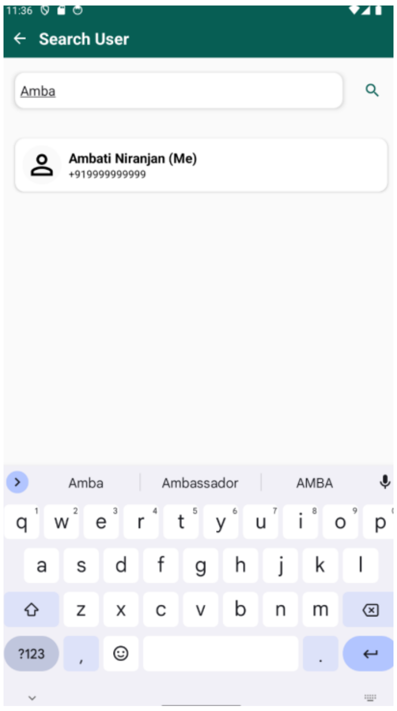
Enter OTP sent to your phone number

123456

- Click Next
- Write Username (if you are already an user the username you kept before will appear automatically) after click Lets Chat.



- Even if you are a old user if it is a new device the Chat Screen will be empty but all your chats are stored you just need to search for that user and then click on that user and you can see Chat, Now if you go back Chat room have a user and your previous chat. For new user it is the same process go to search and select user you want to chat with (**User name are Case sensitive**).

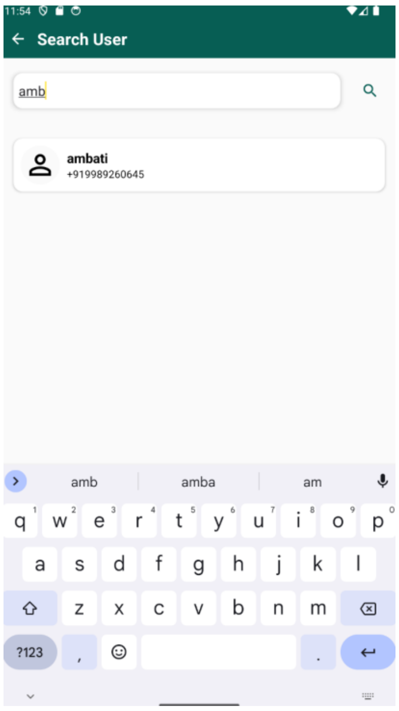


11:36

Search User

Amba

Ambati Niranjan (Me)
+919999999999



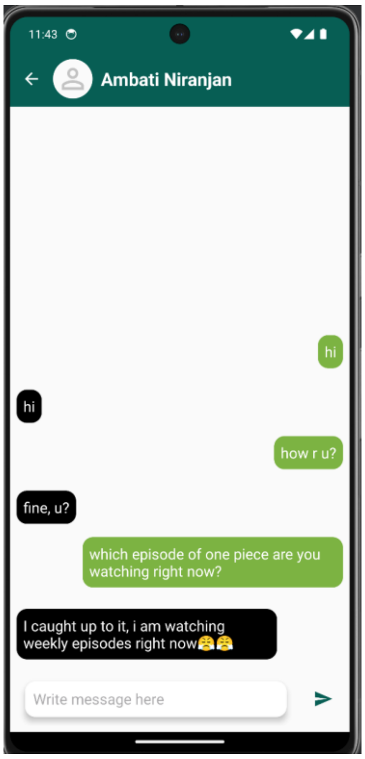
11:54

Search User

amb

ambati
+919989260645

- Now chat with user



11:43

Ambati Niranjan

hi

hi

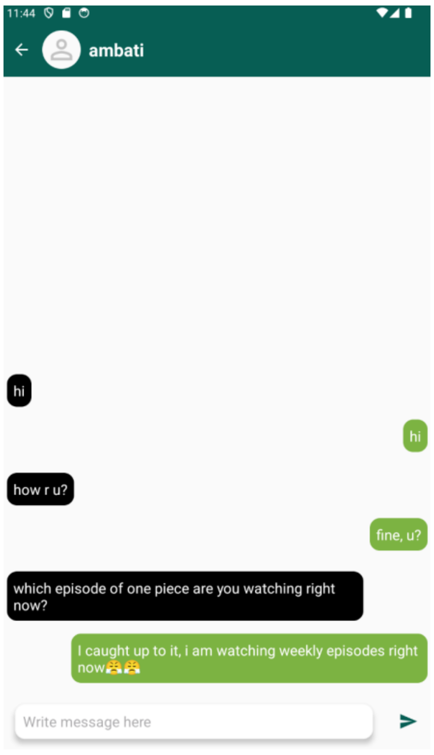
how r u?

fine, u?

which episode of one piece are you watching right now?

I caught up to it, i am watching weekly episodes right now😤😤

Write message here



11:44

ambati

hi

hi

how r u?

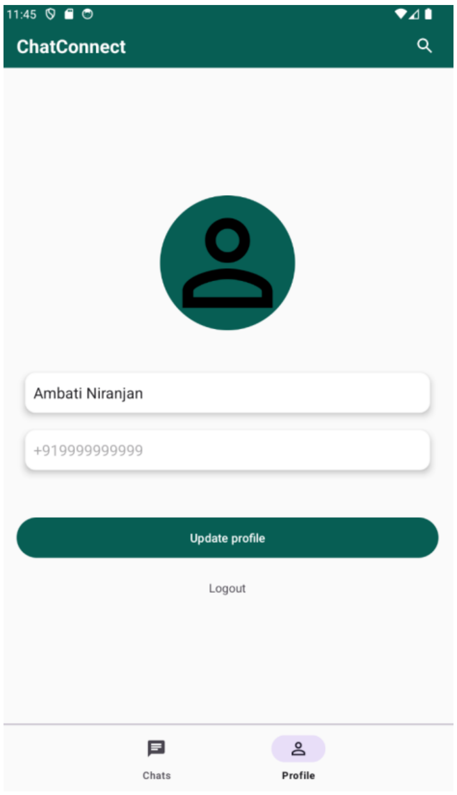
fine, u?

which episode of one piece are you watching right now?

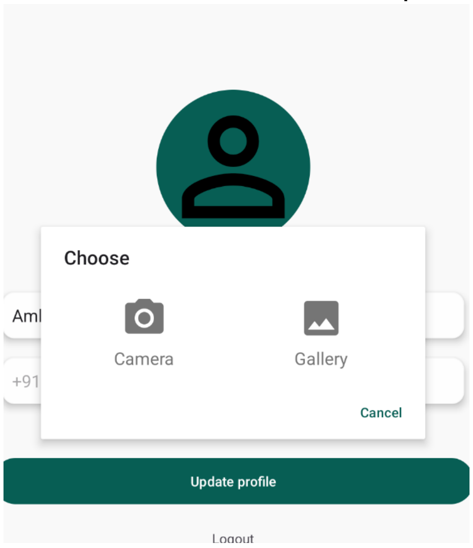
I caught up to it, i am watching weekly episodes right now😤😤

Write message here

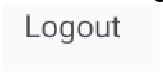
- You can update your profile as well and upload a pic as well.



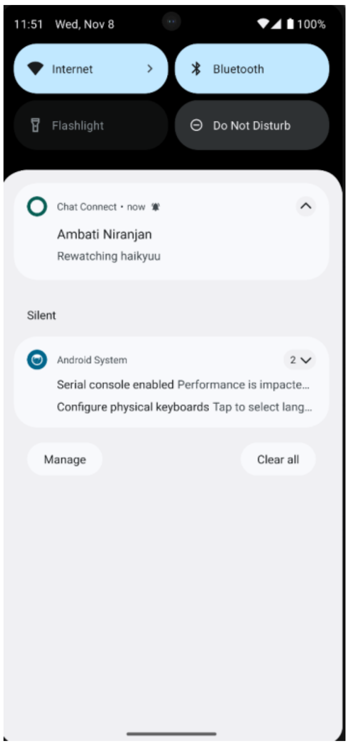
- Click on Person icon to add profile pic



- You can logout by clicking on logout



- You will get notifications as well


11:51 Wed, Nov 8 100%

Chat Connect • now

Ambati Niranjan

Rewatching haikyuu

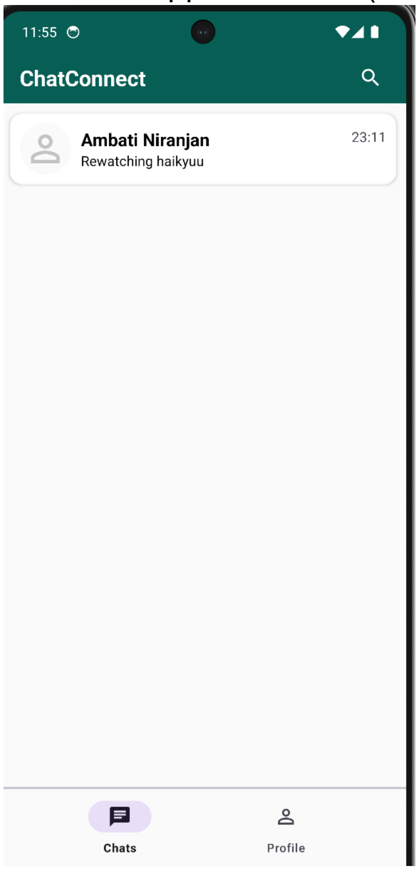
Silent

Android System 2

Serial console enabled Performance is impacte...

Configure physical keyboards Tap to select lang...

- But you need to have at least one interaction in order to receive a notification and click on notification to go to chat.
- So this is about.
- Chats will appear like this(shown for user ambati)


11:55

ChatConnect

Ambati Niranjan 23:11

Rewatching haikyuu