# ANDROID STUDIO- JAVA

In android studio

Help -> Keymap reference – download it and reference

# Development

## Log

1. Import Log

import android.util.Log;

1. Declare a variable for string

private static final String *TAG* = "MY STRING";

1. Display message

Log.*i*(*TAG*, "onCreate");

Complete code

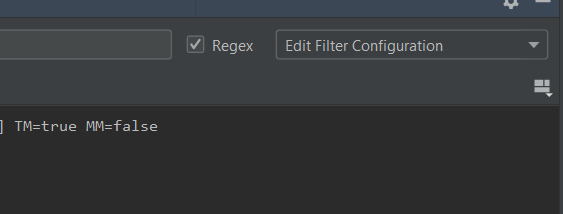
package com.example.a1helloworld;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.util.Log;  
  
import android.os.Bundle;  
  
public class MainActivity extends AppCompatActivity {  
  
 private static final String *TAG* = "MY STRING";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 Log.*i*(*TAG*, "onCreate");  
 }  
}

## Life Cycle

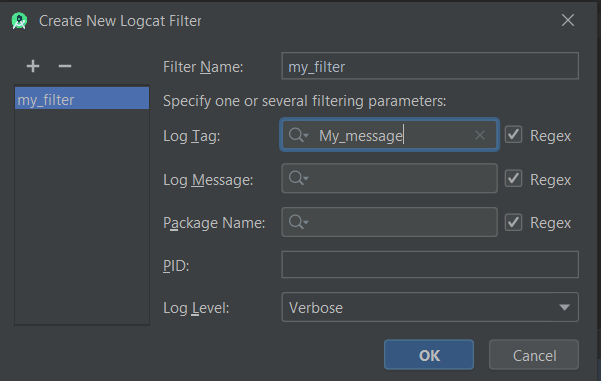
1. In AppCompatActivity -> Cntrl + O
2. Search onStart method

package com.example.a1helloworld;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.util.Log;  
  
import android.os.Bundle;  
  
public class MainActivity extends AppCompatActivity {  
  
 private static final String *TAG* = " My\_message";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 Log.*i*(*TAG*, "onCreate");  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
 Log.*i*(*TAG*, "onStart");  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
  
 Log.*i*(*TAG*, "onResume");  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
 Log.*i*(*TAG*, "onPause");  
 }  
  
 @Override  
 protected void onStop() {  
 super.onStop();  
 Log.*i*(*TAG*, "onStop");  
 }  
  
 @Override  
 protected void onRestart() {  
 super.onRestart();  
 Log.*i*(*TAG*, "onRestart");  
 }  
}

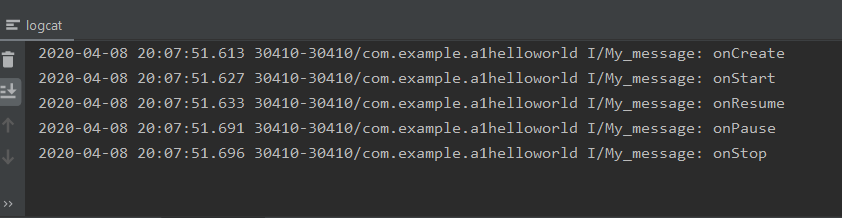
1. click logcat in bottom left corner of android studio
2. choose edit filter in options tag



1. create a filter



Output

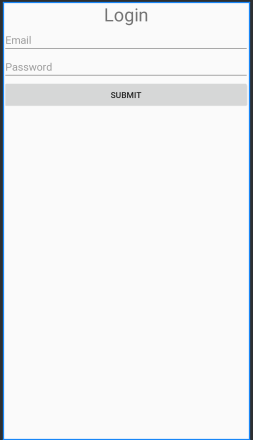


## Simple APP

<https://www.youtube.com/watch?v=Kpyf6s-vPxg&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=12>

### Create Login page UI

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Login"  
 android:textAlignment="center"  
 android:textSize="30sp" />  
  
 <EditText  
 android:id="@+id/editTextEmail"  
 android:layout\_below="@+id/textView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="Email"  
 android:inputType="textEmailAddress" />  
 <EditText  
 android:id="@+id/editTextPass"  
 android:layout\_below="@+id/editTextEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="Password"  
 android:inputType="numberPassword" />  
 <Button  
 android:id="@+id/submitBtn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Submit"  
 android:layout\_below="@+id/editTextPass"  
 />  
</RelativeLayout>



## Create UI using java

<https://www.youtube.com/watch?v=Z8hiM9LEANU&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=13>

1. import

import android.app.Activity;  
  
import android.os.Bundle;  
import android.widget.RelativeLayout;  
import android.widget.Button;

1. initialize object

public class MainActivity extends Activity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
   
 RelativeLayout myRelLayout = new RelativeLayout(this);  
 Button myBtn = new Button(this);  
 }  
}

complete code

package com.example.a2ui;  
  
import android.app.Activity;  
  
import android.os.Bundle;  
import android.widget.RelativeLayout;  
import android.widget.Button;  
  
public class MainActivity extends Activity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
  
 RelativeLayout myRelLayout = new RelativeLayout(this);  
 Button myBtn = new Button(this);  
 myRelLayout.addView(myBtn);  
 setContentView(myRelLayout);  
 }  
}

### Add properties to widget

<https://www.youtube.com/watch?v=9YCXCMzEMpo&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=14>

1. import color

complete code

package com.example.a2ui;  
  
import android.app.Activity;  
  
import android.os.Bundle;  
import android.widget.RelativeLayout;  
import android.widget.Button;  
import android.graphics.Color;  
  
public class MainActivity extends Activity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
  
 RelativeLayout myRelLayout = new RelativeLayout(this);  
 myRelLayout.setBackgroundColor(Color.*BLUE*);  
 Button myBtn = new Button(this);  
 myBtn.setBackgroundColor(Color.*GREEN*);  
 myBtn.setText("Click me");  
 myRelLayout.addView(myBtn);  
 setContentView(myRelLayout);  
 }  
}

1. add params

RelativeLayout.LayoutParams btnDetails = new RelativeLayout.LayoutParams(  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT*,  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT*);  
btnDetails.addRule(RelativeLayout.*CENTER\_HORIZONTAL*);  
btnDetails.addRule(RelativeLayout.*CENTER\_VERTICAL*);  
  
myRelLayout.addView(myBtn, btnDetails);

Complete code

package com.example.a2ui;  
  
import android.app.Activity;  
  
import android.os.Bundle;  
import android.widget.RelativeLayout;  
import android.widget.Button;  
import android.graphics.Color;  
  
public class MainActivity extends Activity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
  
 RelativeLayout myRelLayout = new RelativeLayout(this);  
 myRelLayout.setBackgroundColor(Color.*BLUE*);  
 Button myBtn = new Button(this);  
 myBtn.setBackgroundColor(Color.*GREEN*);  
 myBtn.setText("Click me");  
  
 setContentView(myRelLayout);  
  
 RelativeLayout.LayoutParams btnDetails = new RelativeLayout.LayoutParams(  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT*,  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT* );  
 btnDetails.addRule(RelativeLayout.*CENTER\_HORIZONTAL*);  
 btnDetails.addRule(RelativeLayout.*CENTER\_VERTICAL*);  
  
 myRelLayout.addView(myBtn, btnDetails);  
 }  
}

1. create edit text

package com.example.a2ui;  
  
import android.app.Activity;  
  
import android.os.Bundle;  
import android.widget.EditText;  
import android.widget.RelativeLayout;  
import android.widget.Button;  
import android.graphics.Color;  
import android.widget.EditText;  
  
public class MainActivity extends Activity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
  
 RelativeLayout myRelLayout = new RelativeLayout(this);  
 myRelLayout.setBackgroundColor(Color.*BLUE*);  
 Button myBtn = new Button(this);  
 myBtn.setBackgroundColor(Color.*GREEN*);  
 myBtn.setText("Click me");  
  
 setContentView(myRelLayout);  
  
 RelativeLayout.LayoutParams btnDetails = new RelativeLayout.LayoutParams(  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT*,  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT* );  
 btnDetails.addRule(RelativeLayout.*CENTER\_HORIZONTAL*);  
 btnDetails.addRule(RelativeLayout.*CENTER\_VERTICAL*);  
  
 myRelLayout.addView(myBtn, btnDetails);  
  
 EditText username = new EditText(this);  
  
 myBtn.setId(1);  
 username.setId(2);  
 RelativeLayout.LayoutParams usernameDetails = new RelativeLayout.LayoutParams(  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT*,  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT* );  
  
 usernameDetails.addRule(RelativeLayout.*ABOVE*, myBtn.getId());  
 usernameDetails.addRule(RelativeLayout.*CENTER\_HORIZONTAL*);  
 usernameDetails.setMargins(0,0,0,50);  
 myRelLayout.addView(username, usernameDetails);  
 }  
}

### DIP to pixels

<https://www.youtube.com/watch?v=Fw6pnAyQECE&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=16>

1. import

import android.content.res.Resources;  
import android.util.TypedValue;

1. calculate pixel

Resources r = getResources();  
int px = (int) TypedValue.*applyDimension*(TypedValue.*COMPLEX\_UNIT\_DIP*, 200, r.getDisplayMetrics());  
username.setWidth(px);

Complete code

package com.example.a2ui;  
  
import android.app.Activity;  
  
import android.os.Bundle;  
import android.widget.EditText;  
import android.widget.RelativeLayout;  
import android.widget.Button;  
import android.graphics.Color;  
import android.content.res.Resources;  
import android.util.TypedValue;  
  
public class MainActivity extends Activity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
  
 RelativeLayout myRelLayout = new RelativeLayout(this);  
 myRelLayout.setBackgroundColor(Color.*BLUE*);  
 Button myBtn = new Button(this);  
 myBtn.setBackgroundColor(Color.*GREEN*);  
 myBtn.setText("Click me");  
  
 setContentView(myRelLayout);  
  
 RelativeLayout.LayoutParams btnDetails = new RelativeLayout.LayoutParams(  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT*,  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT* );  
 btnDetails.addRule(RelativeLayout.*CENTER\_HORIZONTAL*);  
 btnDetails.addRule(RelativeLayout.*CENTER\_VERTICAL*);  
  
 myRelLayout.addView(myBtn, btnDetails);  
  
 EditText username = new EditText(this);  
  
 myBtn.setId(1);  
 username.setId(2);  
 RelativeLayout.LayoutParams usernameDetails = new RelativeLayout.LayoutParams(  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT*,  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT* );  
  
 Resources r = getResources();  
 int px = (int) TypedValue.*applyDimension*(TypedValue.*COMPLEX\_UNIT\_DIP*, 200, r.getDisplayMetrics());  
 username.setWidth(px);  
  
 usernameDetails.addRule(RelativeLayout.*ABOVE*, myBtn.getId());  
 usernameDetails.addRule(RelativeLayout.*CENTER\_HORIZONTAL*);  
 usernameDetails.setMargins(0,0,0,50);  
 myRelLayout.addView(username, usernameDetails);  
  
  
 }  
}

### Grid layout

<https://www.youtube.com/watch?v=DAcqmiPcXds&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=17>

### Event Handler

<https://www.youtube.com/watch?v=DAcqmiPcXds&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=17>

#### Event Listner and callback method

MainActivity.java

Button mybtn = (Button) findViewById(R.id.*button*);  
mybtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Toast.*makeText*(MainActivity.this, "Btn clicked...", Toast.*LENGTH\_SHORT*).show();  
 }  
});

Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:text="Button" />  
</RelativeLayout>

### Multiple Event Listners

#### Onclick & Onlong click listner

MainActivity.java

package com.example.a2ui;  
  
import android.app.Activity;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
public class MainActivity extends Activity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 Button mybtn = (Button) findViewById(R.id.*button*);  
 mybtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Toast.*makeText*(MainActivity.this, "Btn clicked...", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 mybtn.setOnLongClickListener(new View.OnLongClickListener() {  
 @Override  
 public boolean onLongClick(View v) {  
 Toast.*makeText*(MainActivity.this, "Btn long click.", Toast.*LENGTH\_SHORT*).show();  
 return true;  
 }  
 });  
 }  
}

## Gestures

<https://www.youtube.com/watch?v=mb7C9cBrjDU&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=21>

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/my\_msg"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:textSize="30dp"  
 android:text="My text" />  
</RelativeLayout>

MainActivity.java

package com.example.a2ui;  
  
import android.os.Bundle;  
import android.view.GestureDetector;  
import android.view.MotionEvent;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity implements GestureDetector.OnGestureListener, GestureDetector.OnDoubleTapListener {  
  
 private TextView my\_msg;  
 private GestureDetector gestureDetector;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 my\_msg = (TextView) findViewById(R.id.*my\_msg*);  
 this.gestureDetector = new GestureDetector(this, this);  
 gestureDetector.setOnDoubleTapListener(this);  
 }  
  
 @Override  
 public boolean onTouchEvent(MotionEvent event) {  
 this.gestureDetector.onTouchEvent(event);  
 return super.onTouchEvent(event);  
 }  
  
 @Override  
 public boolean onSingleTapConfirmed(MotionEvent e) {  
 my\_msg.setText("onSingleTapConfirmed");  
 return true;  
 }  
  
 @Override  
 public boolean onDoubleTap(MotionEvent e) {  
 my\_msg.setText("onDoubleTap");  
 return true;  
 }  
  
 @Override  
 public boolean onDoubleTapEvent(MotionEvent e) {  
 my\_msg.setText("onDoubleTapEvent");  
 return true;  
 }  
  
 @Override  
 public boolean onDown(MotionEvent e) {  
 my\_msg.setText("onDown");  
 return true;  
 }  
  
 @Override  
 public void onShowPress(MotionEvent e) {  
 my\_msg.setText("onShowPress");  
  
 }  
  
 @Override  
 public boolean onSingleTapUp(MotionEvent e) {  
 my\_msg.setText("onSingleTapUp");  
 return true;  
 }  
  
 @Override  
 public boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {  
 my\_msg.setText("onScroll");  
 return true;  
 }  
  
 @Override  
 public void onLongPress(MotionEvent e) {  
 my\_msg.setText("onLongPress");  
 }  
  
 @Override  
 public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {  
 my\_msg.setText("onFling");  
 return true;  
 }  
}

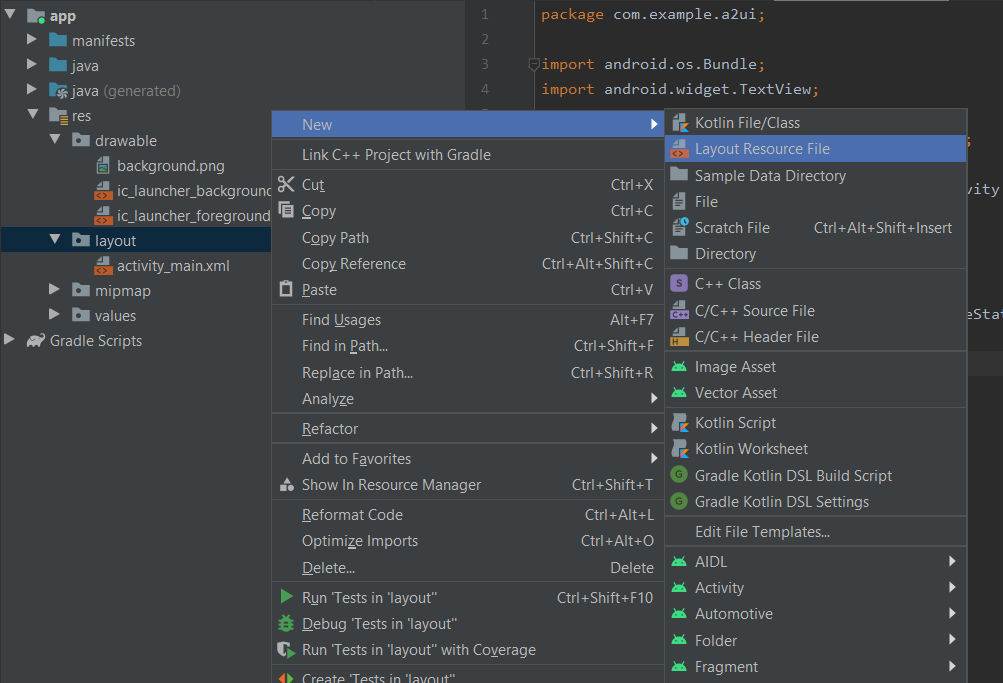
## Fragments

<https://www.youtube.com/watch?v=ztbp59WSk-A&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=23>

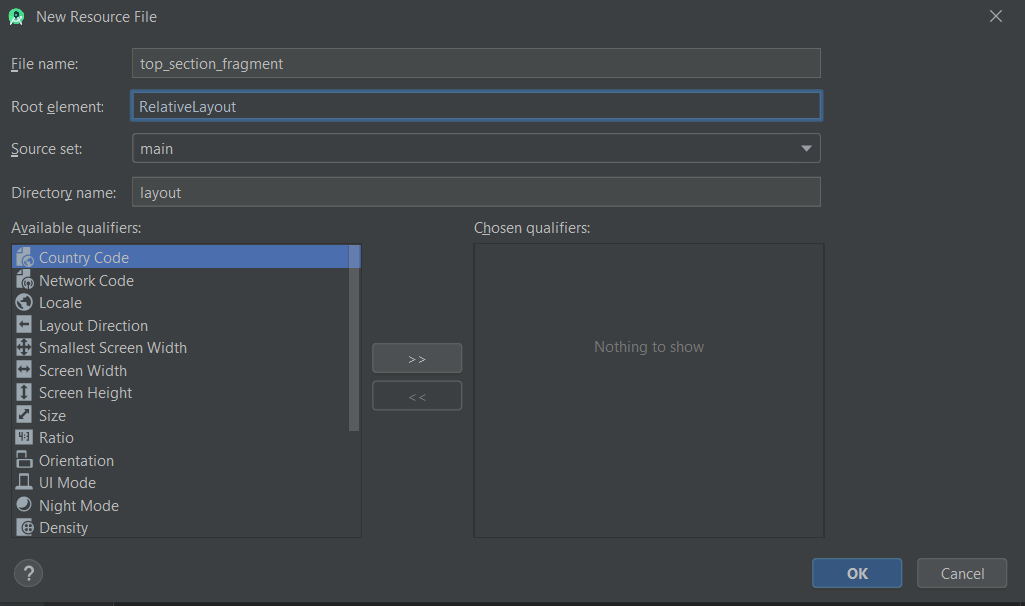
1. paste image in : res > drawable (image name should not include capital letters or numbers)



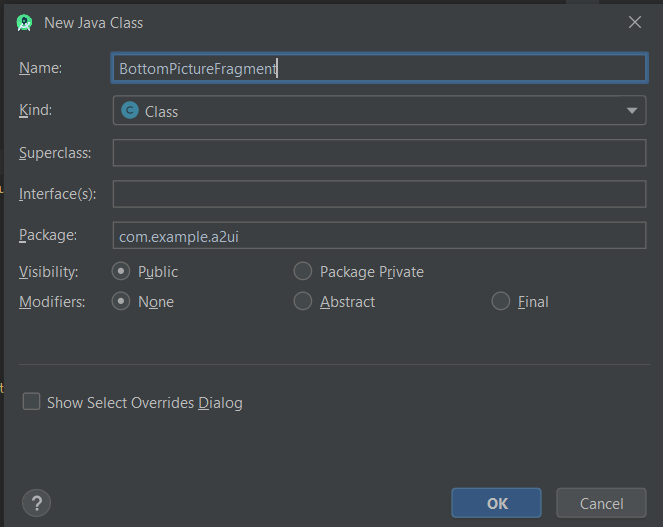
1. create new layout file : right click layout folder -> new -> layout resource file



1. create layout (top\_section\_fragment & bottom\_section\_fragment)

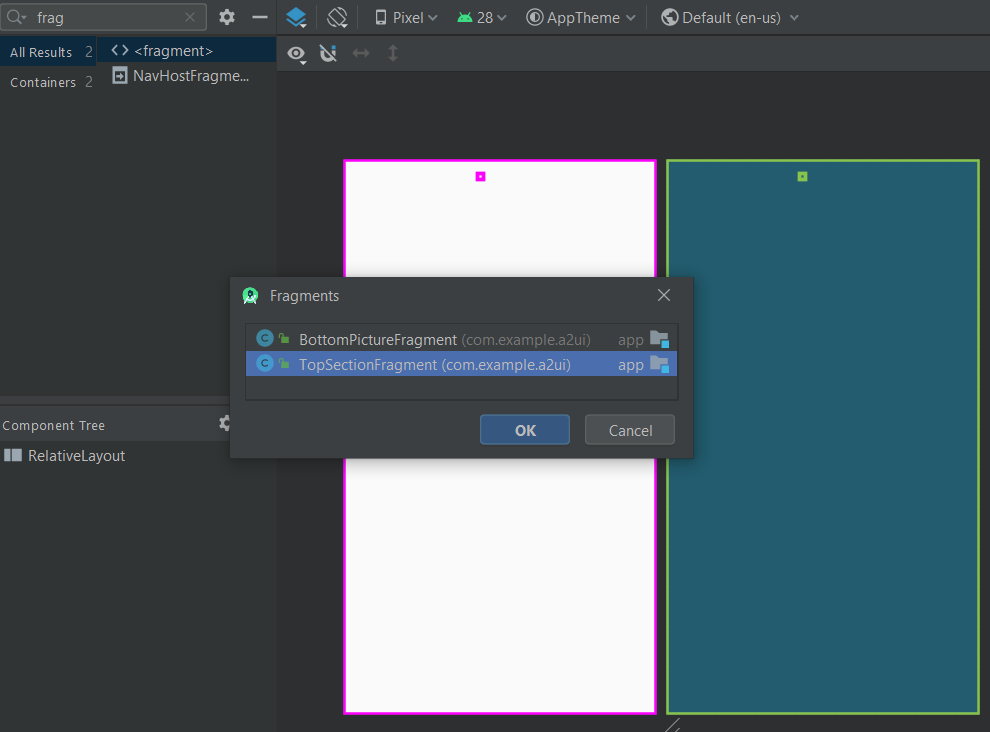


1. create jave class : java>com.example.appname (rightclick)>new > java class =(create two class – TopSectionFragment & BottomSectionFragment)

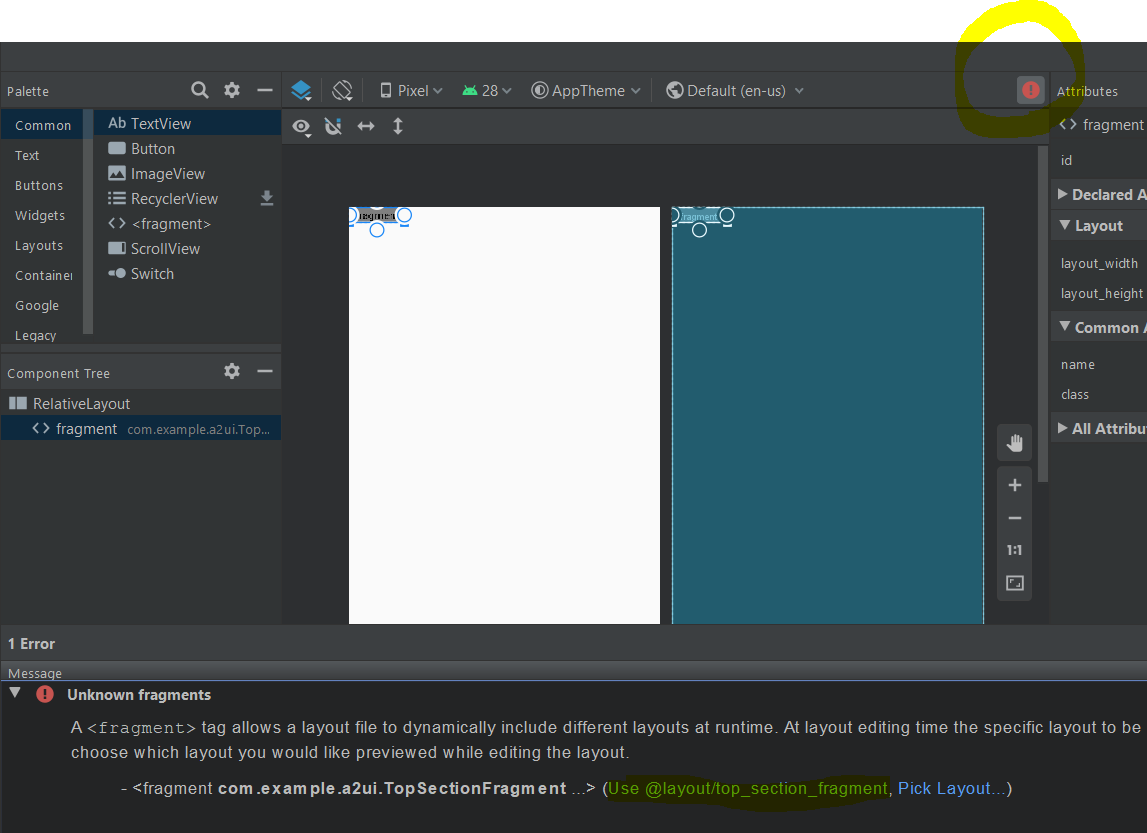


Activity\_main.xml

1. drag and drop fragment



1. solve fragment error – select top\_section\_fragment



Top\_section\_fragment.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent" android:layout\_height="match\_parent">  
 <EditText  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/topTextInput"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="15dp"  
 android:width="300dp"  
 />  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/button"  
 android:layout\_marginTop="55dp"  
 android:layout\_centerHorizontal="true"  
 android:text="Click me"  
 />  
</RelativeLayout>

TopSectionFragment.java

package com.example.a2ui;  
  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
  
public class TopSectionFragment extends Fragment {  
 @Nullable  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*top\_section\_fragment*, container, false);  
 return view;  
 }  
}

BottonPictureFragment.java

package com.example.a2ui;  
  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
  
public class TopSectionFragment extends Fragment {  
 @Nullable  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*top\_section\_fragment*, container, false);  
 return view;  
 }  
}

bottom\_section\_fragment.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent" android:layout\_height="match\_parent"  
 android:background="@drawable/background"  
 >  
  
</RelativeLayout>

Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
  
 <fragment  
 android:id="@+id/fragment"  
 android:name="com.example.a2ui.TopSectionFragment"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 tools:layout="@layout/top\_section\_fragment" />  
 <fragment  
 android:id="@+id/fragment2"  
 android:name="com.example.a2ui.BottomPictureFragment"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/fragment"  
 android:layout\_marginTop="50dp"  
 tools:layout="@layout/bottom\_section\_fragment" />  
</RelativeLayout>

Output



## WorkoutAdvisor – APP

<https://www.youtube.com/watch?v=8Js_Q0AWRLU&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=27>

1. empty activity

### spinner

Strings.xml

<string-array name="workout\_types">  
 <item>Chest</item>  
 <item>Biceps</item>  
 <item>Triceps</item>  
 <item>Shoulders</item>  
</string-array>

Activity\_main.xml

<Spinner  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/woroutType"  
 android:layout\_marginTop="35dp"  
 android:layout\_alignParentTop="true"  
 android:entries="@array/workout\_types"  
 />

MainActivity.java

Spinner workouttype = (Spinner) findViewById(R.id.*woroutType*);  
String workout = String.*valueOf*(workouttype.getSelectedItem());

Complete code

MainActivity.java

package com.example.workoutadvisor2;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Spinner;  
import android.widget.TextView;  
  
public class MainActivity extends Activity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 }  
  
 public void onClickFindWorkout(View view){  
 TextView textView = (TextView) findViewById(R.id.*workout*);  
 Spinner workouttype = (Spinner) findViewById(R.id.*woroutType*);  
 String workout = String.*valueOf*(workouttype.getSelectedItem());  
 textView.setText(workout);  
 }  
}

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
 <Spinner  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/woroutType"  
 android:layout\_marginTop="35dp"  
 android:layout\_alignParentTop="true"  
 android:entries="@array/workout\_types"  
 />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/workouts\_here"  
 android:id="@+id/workout"  
 android:layout\_below="@+id/findWorkout"  
 android:layout\_centerHorizontal="true"  
 />  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/findWorkout"  
 android:layout\_below="@+id/woroutType"  
 android:layout\_centerHorizontal="true"  
 android:text="@string/click\_me"  
 android:onClick="onClickFindWorkout"  
 />  
  
  
</RelativeLayout>

String.xml

<resources>  
 <string name="app\_name">WorkoutAdvisor2</string>  
 <string name="workouts\_here">Workouts Here</string>  
 <string name="click\_me">Click me</string>  
 <string-array name="workout\_types">  
 <item>Chest</item>  
 <item>Biceps</item>  
 <item>Triceps</item>  
 <item>Shoulders</item>  
 </string-array>  
</resources>

1. create WorkoutExpert java class
2. package com.example.workoutadvisor2;  
     
   import java.util.ArrayList;  
   import java.util.List;  
     
   public class WorkoutExpert {  
    List<String> getWorkouts (String workouttypes){  
    List <String> workout = new ArrayList<String>();  
    if (workouttypes.equals("Chest")){  
    workout.add("Bench Press");  
    workout.add("Cable Flys");  
    }  
    else if(workouttypes.equals("Triceps")){  
    workout.add("Triceps Ext");  
    workout.add("Triceps Push down");  
    }  
    else if(workouttypes.equals("Biceps")){  
    workout.add("Biceps Curls");  
    }  
    else if(workouttypes.equals("Sholder")){  
    workout.add("Sholder press");  
    }  
    return workout;  
    }  
   }

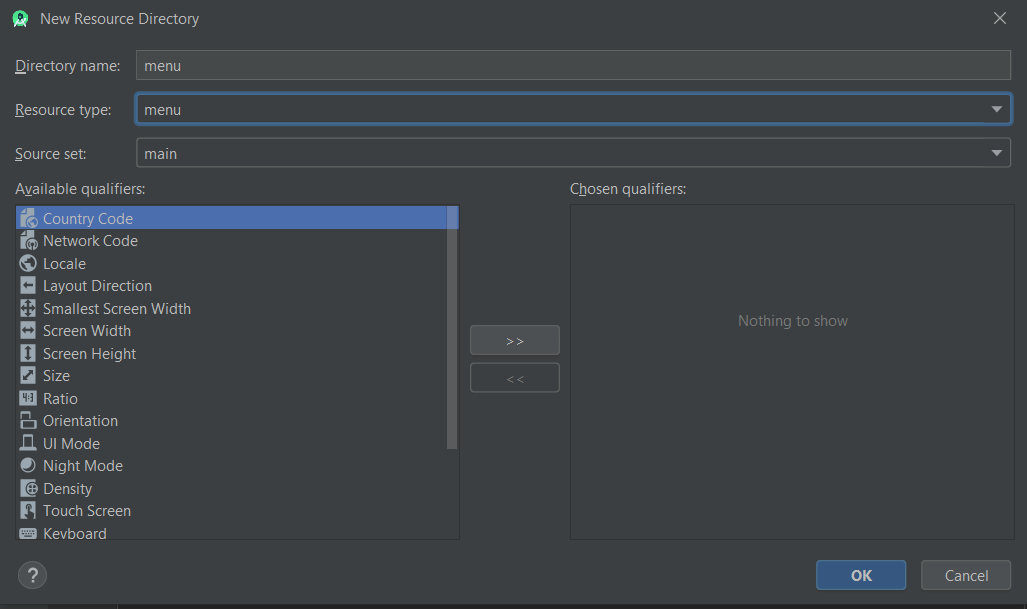
MainActivity.java

### Create an object – external java file

package com.example.workoutadvisor2;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Spinner;  
import android.widget.TextView;  
  
import java.util.List;  
  
public class MainActivity extends Activity {  
  
 private WorkoutExpert workoutExpert = new WorkoutExpert();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 }  
  
 public void onClickFindWorkout(View view){  
 TextView textView = (TextView) findViewById(R.id.*workout*);  
 Spinner workouttype = (Spinner) findViewById(R.id.*woroutType*);  
 String workout = String.*valueOf*(workouttype.getSelectedItem());  
// textView.setText(workout);  
 List<String> workoutList = workoutExpert.getWorkouts(workout);  
  
 StringBuilder workoutFormatter = new StringBuilder();  
 for(String work: workoutList){  
 workoutFormatter.append(work).append("\n");  
 }  
 textView.setText(workoutFormatter);  
 }  
  
}

## Overflow Menu

1. create new folder menu in res folder



1. right click menu folder -> new -> menu resource file -> name it “menu\_main”

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 tools:context="com.example.overflowmenu.MainActivity">  
 <group android:checkableBehavior="single">  
 <item  
 android:id="@+id/menu\_red"  
 android:orderInCategory="1"  
 app:showAsAction="never"  
 android:title="@string/red\_string"  
 />  
 <item  
 android:id="@+id/menu\_blue"  
 android:orderInCategory="2"  
 app:showAsAction="never"  
 android:title="@string/blue\_string"  
 />  
 <item  
 android:id="@+id/menu\_green"  
 android:orderInCategory="3"  
 app:showAsAction="never"  
 android:title="@string/green\_string"  
 />  
 </group>  
  
</menu>

Strings.xml

<resources>  
 <string name="app\_name">OverflowMenu</string>  
 <string name="red\_string">Red</string>  
 <string name="blue\_string">Blue</string>  
 <string name="green\_string">Green</string>  
</resources>

MainActivity.java

package com.example.overflowmenu;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.graphics.Color;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.widget.RelativeLayout;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*menu\_main*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 RelativeLayout main\_view = (RelativeLayout) findViewById(R.id.*main\_view*);  
 switch (item.getItemId()){  
 case R.id.*menu\_red*:  
 if(item.isChecked())  
 item.setChecked(false);  
 else item.setChecked(true);  
 main\_view.setBackgroundColor(Color.*RED*);  
 return true;  
 case R.id.*menu\_blue*:  
 if(item.isChecked())  
 item.setChecked(false);  
 else item.setChecked(true);  
 main\_view.setBackgroundColor(Color.*BLUE*);  
 return true;  
 case R.id.*menu\_green*:  
 if(item.isChecked())  
 item.setChecked(false);  
 else item.setChecked(true);  
 main\_view.setBackgroundColor(Color.*GREEN*);  
 return true;  
 default: return super.onOptionsItemSelected(item);  
 }  
 }  
}

## Animation and transition

<https://www.youtube.com/watch?v=ZAsknWvR09E&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=34>

MainActivity.java

package com.example.overflowmenu;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.graphics.Color;  
import android.os.Bundle;  
import android.transition.TransitionManager;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.RelativeLayout;  
  
public class MainActivity extends AppCompatActivity {  
  
 ViewGroup myLayout;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 myLayout = (ViewGroup) findViewById(R.id.*main\_view*);  
 myLayout.setOnTouchListener(new View.OnTouchListener() {  
 @Override  
 public boolean onTouch(View v, MotionEvent event) {  
 moveButton();  
 return true;  
 }  
 });  
 }  
  
 private void moveButton() {  
 TransitionManager.*beginDelayedTransition*(myLayout);  
 View myButton = findViewById(R.id.*button*);  
 RelativeLayout.LayoutParams positionRules = new RelativeLayout.LayoutParams(  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT*,  
 RelativeLayout.LayoutParams.*WRAP\_CONTENT* );  
 positionRules.addRule(RelativeLayout.*ALIGN\_PARENT\_BOTTOM*, RelativeLayout.*TRUE*);  
 myButton.setLayoutParams(positionRules);  
  
// code for expanding the button  
 ViewGroup.LayoutParams sizeRules = myButton.getLayoutParams();  
 sizeRules.width = 450;  
 sizeRules.height = 300;  
 myButton.setLayoutParams(sizeRules);  
 }  
  
  
}

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/main\_view"  
 tools:context=".MainActivity">  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/button"  
 android:text="Animation"  
 />  
  
</RelativeLayout>

## Intent

<https://www.youtube.com/watch?v=pMMWo96j4JI&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=35>

1. create new activity: right click com.example.appname->new->Activity->empty activity
2. create intent

MainActivity.java

package com.example.overflowmenu;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
  
public class MainActivity extends AppCompatActivity {  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 }  
  
 public void move2second(View view) {  
 Intent i = new Intent(this, Main2Activity.class);  
 startActivity(i);  
 }  
  
  
}

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/main\_view"  
 tools:context=".MainActivity">  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/button"  
 android:text="Animation"  
 android:onClick="move2second"  
 />  
  
</RelativeLayout>

### Send data

<https://www.youtube.com/watch?v=-GxReMxnXkQ&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=36>

MainActivity.java

Intent i = new Intent(this, Main2Activity.class);  
i.putExtra("mymsg", "Hello");  
startActivity(i);

Main2Activity.java

package com.example.overflowmenu;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.Toast;  
  
public class Main2Activity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main2*);  
  
 Bundle firstData = getIntent().getExtras();  
 if (firstData == null){  
 return;  
 }  
 String firstMsg = firstData.getString("mymsg");  
 Toast.*makeText*(this, firstMsg, Toast.*LENGTH\_SHORT*).show();  
 }  
}

### Send Broadcast Intent – Not Working

<https://www.youtube.com/watch?v=uVpleldaTnU&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=37>

MainActivity.java

package com.example.overflowmenu;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
  
public class MainActivity extends AppCompatActivity {  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 }  
  
 public void move2second(View view) {  
 Intent i = new Intent(this, Main2Activity.class);  
 i.setAction("com.example.overflowmenu");  
 i.addFlags(Intent.*FLAG\_INCLUDE\_STOPPED\_PACKAGES*);  
 sendBroadcast(i);  
 }  
}

### BroadCastReceiver – Not Working

<https://www.youtube.com/watch?v=ys-ot3_Nc44&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=38>

1. create new – No activity
2. right click com.example.appname -> new -> other -> Broad cast receiver

AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.broadcastreceiver">  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme">  
 <receiver  
 android:name=".MyReceiver"  
 android:enabled="true"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="com.example.overflowmenu">  
  
 </action>  
 </intent-filter>  
 </receiver>  
 </application>  
  
</manifest>

MyReceiver.java

package com.example.broadcastreceiver;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.widget.Toast;  
  
public class MyReceiver extends BroadcastReceiver {  
  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 // *TODO: This method is called when the BroadcastReceiver is receiving* // an Intent broadcast.  
 Toast.*makeText*(context, "Broadcast received..", Toast.*LENGTH\_SHORT*).show();  
 }  
}

## Thread

<https://www.youtube.com/watch?v=FP71FR2T-Q4&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=39>

MainActivity.java

package com.example.thread;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.os.Handler;  
import android.os.Message;  
import android.view.View;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity {  
  
 Handler handler = new Handler(){  
 @Override  
 public void handleMessage(Message msg) {  
 TextView mytextView = (TextView) findViewById(R.id.*textView*);  
 mytextView.setText("Button Clicked");  
 }  
 };  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 }  
 public void ClickMyButton(View view){  
 Runnable r = new Runnable() {  
 @Override  
 public void run() {  
 long futuretime = System.*currentTimeMillis*()+10000;  
 while (System.*currentTimeMillis*()< futuretime){  
 synchronized (this){  
 try{  
 wait(futuretime - System.*currentTimeMillis*());  
 }catch (Exception e){  
  
 }  
 }  
 }  
 handler.sendEmptyMessage(0);  
 }  
 };  
 Thread mythred = new Thread(r);  
 mythred.start();  
  
  
 }  
}

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 android:layout\_centerHorizontal="true"  
 android:id="@+id/textView"/>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/button"  
 android:text="@string/click\_me"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/textView"  
 android:onClick="ClickMyButton"  
 />  
  
</RelativeLayout>

## Intent Service

<https://www.youtube.com/watch?v=e04Mv9aju0w&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=41>

1. Create new class MyIntentServiece.java

package com.example.thread;  
  
import android.app.IntentService;  
import android.content.Intent;  
import android.widget.Toast;  
  
import androidx.annotation.Nullable;  
  
public class MyIntentServiece extends IntentService {  
 private static final String *TAG* = "com.example.thread";  
 public MyIntentServiece() {  
 super("MyIntentServiece");  
 }  
  
 @Override  
 protected void onHandleIntent(@Nullable Intent intent) {  
 Toast.*makeText*(this, "Serviece Started", Toast.*LENGTH\_SHORT*).show();  
 }  
}

MainActivity.java

package com.example.thread;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
  
public class MainActivity extends AppCompatActivity {  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 Intent intent = new Intent(this, MyIntentServiece.class);  
 startService(intent);  
 }  
 public void ClickMyButton(View view){  
  
 }  
}

AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.thread">  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme">  
 <activity android:name=".MainActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <service android:name=".MyIntentServiece" />  
 </application>  
  
</manifest>

### Create service using service class

1. Right click com.example.thread -> new -> service -> Service (Not Intent service)

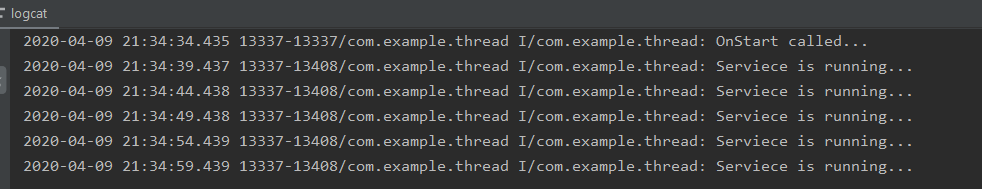
MyServiece.java

package com.example.thread;  
  
import android.app.Service;  
import android.content.Intent;  
import android.os.IBinder;  
import android.util.Log;  
import android.widget.Toast;  
  
public class MyService extends Service {  
 private static final String *TAG* = "com.example.thread";  
  
 public MyService() {  
 }  
  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 Log.*i*(*TAG*, "OnStart called...");  
 Runnable r = new Runnable() {  
 @Override  
 public void run() {  
 for (int i=0; i<5; i++) {  
 long futuretime = System.*currentTimeMillis*() + 5000;  
 while (System.*currentTimeMillis*() < futuretime) {  
 synchronized (this) {  
 try {  
 wait(futuretime - System.*currentTimeMillis*());  
 Log.*i*(*TAG*, "Serviece is running...");  
 } catch (Exception e) {  
  
 }  
 }  
 }  
 }  
  
 }  
 };  
 Thread mythred = new Thread(r);  
 mythred.start();  
 return Service.*START\_STICKY*;  
 }  
  
 @Override  
 public void onDestroy() {  
 Log.*i*(*TAG*, "onDestroy called...");  
 }  
  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
}

MainActivity.java

package com.example.thread;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
  
public class MainActivity extends AppCompatActivity {  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 Intent intent = new Intent(this, MyService.class);  
 startService(intent);  
 }  
 public void ClickMyButton(View view){  
  
 }  
}

output



### Bound Service

<https://www.youtube.com/watch?v=Wy1cN57aCfM&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=43>

1. Right click com.example.appname -> new -> Service -> Service(not Intent Service)

package com.example.thread;  
  
import android.app.Service;  
import android.content.Intent;  
import android.os.Binder;  
import android.os.IBinder;  
  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Locale;  
  
public class MyService extends Service {  
 private static final String *TAG* = "com.example.thread";  
 private final IBinder myBinder = new MyLocalBinder();  
  
 public MyService() {  
 }  
 public String getCurrentTime(){  
 SimpleDateFormat df = new SimpleDateFormat("HH:mm:ss", Locale.*US*);  
 return (df.format(new Date()));  
 }  
  
  
 @Override  
 public IBinder onBind(Intent intent) {  
 return myBinder;  
 }  
  
 public class MyLocalBinder extends Binder {  
 MyService getService(){  
 return MyService.this;  
 }  
 }  
}

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 android:layout\_centerHorizontal="true"  
 android:id="@+id/textView"/>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/button"  
 android:text="@string/click\_me"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/textView"  
 android:onClick="showTime"  
 />  
  
</RelativeLayout>

MainActivity.java

package com.example.thread;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.ComponentName;  
import android.content.Context;  
import android.content.Intent;  
import android.content.ServiceConnection;  
import android.os.Bundle;  
import android.os.IBinder;  
import android.view.View;  
import android.widget.TextView;  
  
import com.example.thread.MyService.MyLocalBinder;  
  
public class MainActivity extends AppCompatActivity {  
  
 MyService myService;  
 boolean isbound = false;  
  
 public void showTime(View view){  
 String currenttime = myService.getCurrentTime();  
 TextView myText = (TextView) findViewById(R.id.*textView*);  
 myText.setText(currenttime);  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 Intent intent = new Intent(this, MyService.class);  
 bindService(intent, myConnection, Context.*BIND\_AUTO\_CREATE*);  
 }  
 private ServiceConnection myConnection = new ServiceConnection() {  
 @Override  
 public void onServiceConnected(ComponentName name, IBinder service) {  
 MyLocalBinder myLocalBinder = (MyLocalBinder) service;  
 myService = myLocalBinder.getService();  
 isbound = true;  
 }  
  
 @Override  
 public void onServiceDisconnected(ComponentName name) {  
 isbound = false;  
 }  
 };  
}

## ListView

### Simple listview

<https://www.youtube.com/watch?v=-PqetZ2yV3s&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=46>

Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <ListView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/mylistview"  
 />  
  
</RelativeLayout>

MainActivity.java

package com.example.listview;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.ListAdapter;  
import android.widget.ListView;  
import android.widget.Toast;  
  
public class MainActivity extends Activity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 String [] fruits = {"Apple", "Orange", "Peach", "Mango"};  
 ListAdapter myAdapter = new ArrayAdapter<String>(this, android.R.layout.*simple\_list\_item\_1*, fruits);  
 ListView listView = (ListView) findViewById(R.id.*mylistview*);  
 listView.setAdapter(myAdapter);  
  
 listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 String fruit = String.*valueOf*(parent.getItemAtPosition(position));  
 Toast.*makeText*(MainActivity.this, fruit, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
}

### Custom ListView

<https://www.youtube.com/watch?v=frBzVEl3efk&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=48&frags=wn>

1. Paste this image in drawable folder



1. Create custom\_list\_row.xml in Layout folder

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="horizontal" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/apple"  
 android:maxWidth="80dp"  
 android:maxHeight="80dp"  
 android:layout\_margin="5dp"  
 android:id="@+id/myImage"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="My message"  
 android:id="@+id/myText"  
 />  
  
</LinearLayout>

1. Create CustomAdapter.java

package com.example.listview;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
  
class CustomAdapter extends ArrayAdapter<String> {  
 public CustomAdapter(@NonNull Context context, String[] fruits) {  
 super(context, R.layout.*custom\_list\_row* ,fruits);  
 }  
  
 @NonNull  
 @Override  
 public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 LayoutInflater myInflater = LayoutInflater.*from*(getContext());  
 View CustomView = myInflater.inflate(R.layout.*custom\_list\_row*, parent, false);  
 String singlefruititem = getItem(position);  
 TextView myText = (TextView) CustomView.findViewById(R.id.*myText*);  
  
 ImageView myImage = (ImageView) CustomView.findViewById(R.id.*myImage*);  
 myText.setText(singlefruititem);  
 myImage.setImageResource(R.drawable.*apple*);  
 return CustomView;  
 }  
}

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <ListView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/mylistview"  
 />  
  
</RelativeLayout>

MainActivity.java

package com.example.listview;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.ListAdapter;  
import android.widget.ListView;  
import android.widget.Toast;  
  
public class MainActivity extends Activity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 String [] fruits = {"Apple", "Orange", "Peach", "Mango"};  
 ListAdapter myAdapter = new CustomAdapter(this, fruits);  
 ListView listView = (ListView) findViewById(R.id.*mylistview*);  
 listView.setAdapter(myAdapter);  
  
 listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 String fruit = String.*valueOf*(parent.getItemAtPosition(position));  
 Toast.*makeText*(MainActivity.this, fruit, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
}



## SQLite

1. Create layout resource file

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/editText"  
 />  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/btn\_add"  
 android:text="Add"  
 android:onClick="addButtonClick"  
 android:layout\_below="@+id/editText"  
 />  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/btn\_delete"  
 android:text="Delete"  
 android:layout\_marginLeft="100dp"  
 android:onClick="deleteButtonClick"  
 android:layout\_below="@+id/editText"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/myText"  
 android:text="My Text"  
 android:layout\_below="@+id/btn\_delete"  
 />  
  
</RelativeLayout>

1. Create new Product.java file in com.example.appname

package com.example.sqlite;  
  
public class Product {  
 private int id;  
 private String product;  
  
 public Product(){  
  
 }  
  
 public Product(String product) {  
 this.product = product;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public String getProduct() {  
 return product;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public void setProduct(String product) {  
 this.product = product;  
 }  
}

1. Create new java class for “MyDBHandler.java”

package com.example.sqlite;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
import androidx.annotation.Nullable;  
  
public class MyDBHandler extends SQLiteOpenHelper {  
 private static final int *DATABASE\_VERSION* = 1;  
 private static final String *DATABASE\_NAME* = "products.db";  
 public static final String *TABLE\_PRODUCTS* = "products";  
 public static final String *COLUMN\_ID* = "id";  
 public static final String *COLUMN\_PRODUCTNAME* = "productname";  
  
 public MyDBHandler(@Nullable Context context, @Nullable String name, @Nullable SQLiteDatabase.CursorFactory factory, int version) {  
 super(context, *DATABASE\_NAME*, factory, *DATABASE\_VERSION*);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 String query = "CREATE TABLE "+ *TABLE\_PRODUCTS* + " ( " + *COLUMN\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT, " + *COLUMN\_PRODUCTNAME* + " TEXT " + ");";  
 db.execSQL(query);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 String query = "DROP TABLE IF EXISTS "+*TABLE\_PRODUCTS*;  
 db.execSQL(query);  
 onCreate(db);  
 }  
 public void addProduct(Product product){  
 ContentValues contentValues = new ContentValues();  
 contentValues.put(*COLUMN\_PRODUCTNAME*, product.getProduct());  
 SQLiteDatabase db = getWritableDatabase();  
 db.insert(*TABLE\_PRODUCTS*, null, contentValues);  
 db.close();  
 }  
  
 public void deleteProduct(String productname){  
 SQLiteDatabase db = getWritableDatabase();  
 String query = "DELETE FROM "+ *TABLE\_PRODUCTS* + " WHERE " + *COLUMN\_PRODUCTNAME* + " = \""+ productname + "\";";  
 db.execSQL(query);  
 }  
 public String database2String(){  
 String dbString = "";  
 SQLiteDatabase db = getWritableDatabase();  
 String query = "SELECT \* FROM "+ *TABLE\_PRODUCTS* + " WHERE 1";  
 Cursor c = db.rawQuery(query, null);  
 c.moveToFirst();  
 while (!c.isAfterLast()){  
 if(c.getString(c.getColumnIndex(*COLUMN\_PRODUCTNAME*)) != null){  
 dbString += c.getString(c.getColumnIndex(*COLUMN\_PRODUCTNAME*));  
 dbString += "\n";  
 }  
 c.moveToNext();  
 }  
 db.close();  
 return dbString;  
 }  
}

1. MainActivity.java

package com.example.sqlite;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.TextView;  
  
public class MainActivity extends Activity {  
  
 EditText editText;  
 TextView textView;  
 MyDBHandler myDBHandler;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 editText = (EditText) findViewById(R.id.*editText*);  
 textView = (TextView) findViewById(R.id.*myText*);  
  
 myDBHandler = new MyDBHandler(this, null, null, 1);  
 printDatabase();  
 }  
  
 public void printDatabase() {  
 String dbstring = myDBHandler.database2String();  
 textView.setText(dbstring);  
 editText.setText("");  
 }  
 public void addButtonClick(View view){  
 Product product = new Product(editText.getText().toString());  
 myDBHandler.addProduct(product);  
 printDatabase();  
 }  
 public void deleteButtonClick(View view){  
 String inputText = editText.getText().toString();  
 myDBHandler.deleteProduct(inputText);  
 printDatabase();  
 }  
}

## Video View

<https://www.youtube.com/watch?v=nk0BIwZP8RM&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=55>

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <VideoView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/myVideoView"  
 />  
  
</RelativeLayout>

MainActivity.java

package com.example.videoplay;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.VideoView;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 final VideoView videoView = (VideoView) findViewById(R.id.*myVideoView*);  
 videoView.setVideoPath("http://techslides.com/demos/sample-videos/small.mp4");  
 videoView.start();  
 }  
}

ANdroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.videoplay">  
 <uses-permission android:name="android.permission.INTERNET" />  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme">  
 <activity android:name=".MainActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

### Capture Image

### Bitmap Image

Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/myImage"  
 android:minHeight="300dp"  
 android:minWidth="300dp"  
 android:layout\_centerHorizontal="true"  
 />  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Capute"  
 android:id="@+id/btn\_capture"  
 android:layout\_below="@+id/myImage"  
 android:layout\_centerHorizontal="true"  
 android:onClick="launchCamera"  
 />  
  
</RelativeLayout>

MainActivity.java

package com.example.captureimage;  
  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.graphics.Bitmap;  
import android.os.Bundle;  
import android.provider.MediaStore;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
  
 static final int *REQUEST\_IMAGE\_CAPTURE* = 1;  
 ImageView imageView;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 Button button = (Button) findViewById(R.id.*btn\_capture*);  
 imageView = (ImageView) findViewById(R.id.*myImage*);  
  
 if (! hasCamera()){  
 button.setEnabled(false);  
 }  
 }  
  
 public boolean hasCamera() {  
 return getPackageManager().hasSystemFeature(PackageManager.*FEATURE\_CAMERA\_ANY*);  
 }  
 public void launchCamera(View view){  
 Intent intent = new Intent(MediaStore.*ACTION\_IMAGE\_CAPTURE*);  
 startActivityForResult(intent, *REQUEST\_IMAGE\_CAPTURE*);  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
  
 if (requestCode == *REQUEST\_IMAGE\_CAPTURE* && resultCode == *RESULT\_OK*) {  
 Toast.*makeText*(this, "Capture Image", Toast.*LENGTH\_SHORT*).show();  
 Bundle extras = data.getExtras();  
 Bitmap photo = (Bitmap) extras.get("data");  
 imageView.setImageBitmap(photo);  
  
 }  
 }  
}

AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.captureimage">  
 <uses-feature android:name="android.hardware.camera2" android:required="true" />  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme">  
 <activity android:name=".MainActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

### Image Effect

<https://www.youtube.com/watch?v=gaQ1izK7v6Y&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=59>



1. Paste image in drawable folder

Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/myImage"  
 android:minHeight="300dp"  
 android:minWidth="300dp"  
 android:layout\_centerHorizontal="true"  
 />  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Capute"  
 android:id="@+id/btn\_capture"  
 android:layout\_below="@+id/myImage"  
 android:layout\_centerHorizontal="true"  
 android:onClick="launchCamera"  
 />  
  
</RelativeLayout>

MainActivity.java

package com.example.captureimage;  
  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.content.res.ResourcesCompat;  
  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.graphics.Bitmap;  
import android.graphics.Color;  
import android.graphics.drawable.BitmapDrawable;  
import android.graphics.drawable.Drawable;  
import android.os.Bundle;  
import android.provider.MediaStore;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
  
 Bitmap myBitmapImg;  
 ImageView imageView;  
 Drawable myFlower;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 imageView = (ImageView)findViewById(R.id.*myImage*);  
 myFlower = ResourcesCompat.*getDrawable*(getResources(), R.drawable.*sunflower*, null);  
 myBitmapImg = ((BitmapDrawable)myFlower).getBitmap();  
  
 Bitmap newphoto = invertImage(myBitmapImg);  
 imageView.setImageBitmap(newphoto);  
 }  
  
 private Bitmap invertImage(Bitmap myBitmapImg) {  
 Bitmap finalImage = Bitmap.*createBitmap*(myBitmapImg.getWidth(),myBitmapImg.getHeight(),myBitmapImg.getConfig());  
  
 int A, R, G, B;  
 int pixelcolor;  
 int height = myBitmapImg.getHeight();  
 int width = myBitmapImg.getWidth();  
 for(int y=0; y<height; y++){  
 for (int x=0; x<width; x++){  
 pixelcolor = myBitmapImg.getPixel(x,y);  
 A = Color.*alpha*(pixelcolor);  
 R = 255 - Color.*red*(pixelcolor);  
 G = 255 - Color.*green*(pixelcolor);  
 B = 255 - Color.*blue*(pixelcolor);  
 finalImage.setPixel(x,y,Color.*argb*(A, R, G, B));  
 }  
 }  
 return finalImage;  
 }  
  
}

## Notification – NOt Working

<https://www.youtube.com/watch?v=ikZemSzCip8&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe&index=61>

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Notify"  
 android:id="@+id/btn\_notify"  
 android:onClick="myButtonClicked"  
 />  
  
</RelativeLayout>

MainActivity.java

package com.example.notification;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.NotificationCompat;  
  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
  
 NotificationCompat.Builder notification;  
 private static final int *uniqueID* = 40111;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 notification = new NotificationCompat.Builder(this);  
 notification.setAutoCancel(true);  
  
 }  
 public void myButtonClicked(View view){  
 notification.setSmallIcon(R.mipmap.*ic\_launcher*);  
 notification.setTicker("This is a ticker");  
 notification.setWhen(System.*currentTimeMillis*());  
 notification.setContentTitle("This is my title");  
 notification.setContentText("Here is the text");  
  
 Intent intent = new Intent(this, MainActivity.class);  
 PendingIntent pendingIntent = PendingIntent.*getActivity*(this, 0, intent, PendingIntent.*FLAG\_UPDATE\_CURRENT*);  
 notification.setContentIntent(pendingIntent);  
 NotificationManager nm = (NotificationManager) getSystemService(*NOTIFICATION\_SERVICE*);  
 nm.notify(*uniqueID*, notification.build());  
 Toast.*makeText*(this, "Notification", Toast.*LENGTH\_SHORT*).show();  
 }  
}

## HTTP Connection

### OkHttp - GET

<https://www.youtube.com/watch?v=oGWJ8xD2W6k>

<https://www.studytutorial.in/android-okhttp-post-and-get-request-tutorial>

1. Add userpermission in manifest file

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.httpconnection">  
 <uses-permission android:name="android.permission.INTERNET" />  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme">  
 <activity android:name=".MainActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

1. Add dependency in gradle file

implementation 'com.squareup.okhttp:okhttp:2.5.0'

MainActivity.java

package com.example.httpconnection;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.TextView;  
  
import com.squareup.okhttp.Callback;  
import com.squareup.okhttp.OkHttpClient;  
import com.squareup.okhttp.Request;  
import com.squareup.okhttp.Response;  
  
import java.io.IOException;  
  
public class MainActivity extends AppCompatActivity {  
  
 private TextView textViewResult;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 textViewResult = (TextView) findViewById(R.id.*textViewResult*);  
 OkHttpClient client = new OkHttpClient();  
 String url = "https://reqres.in/api/users?page=2";  
  
 Request request = new Request.Builder()  
 .url(url)  
 .build();  
 client.newCall(request).enqueue(new Callback() {  
 @Override  
 public void onFailure(Request request, IOException e) {  
 e.printStackTrace();  
 }  
  
 @Override  
 public void onResponse(Response response) throws IOException {  
 if (response.isSuccessful()){  
 final String myResponce = response.body().string();  
 MainActivity.this.runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 textViewResult.setText(myResponce);  
 }  
 });  
 }  
 }  
 });  
 }  
}

### OKHTTP – POST

MainActivity.java

package com.example.httpconnection;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.TextView;  
  
import com.squareup.okhttp.Callback;  
import com.squareup.okhttp.MediaType;  
import com.squareup.okhttp.OkHttpClient;  
import com.squareup.okhttp.Request;  
import com.squareup.okhttp.RequestBody;  
import com.squareup.okhttp.Response;  
  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.io.IOException;  
  
public class MainActivity extends AppCompatActivity {  
  
 private TextView textViewResult;  
 public static final MediaType *MEDIA\_TYPE* = MediaType.*parse*("application/json");  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 textViewResult = (TextView) findViewById(R.id.*textViewResult*);  
  
 final OkHttpClient client = new OkHttpClient();  
  
 JSONObject postdata = new JSONObject();  
 try {  
 postdata.put("name", "Abhay Anand");  
 postdata.put("job", "leader");  
 } catch(JSONException e){  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 }  
  
 String url = "https://reqres.in/api/users";  
  
 RequestBody body = RequestBody.*create*(*MEDIA\_TYPE*, postdata.toString());  
  
 final Request request = new Request.Builder()  
 .url(url)  
 .post(body)  
 .addHeader("Content-Type", "application/json")  
 .addHeader("Authorization", "Your Token")  
 .addHeader("cache-control", "no-cache")  
 .build();  
  
 client.newCall(request).enqueue(new Callback() {  
 @Override  
 public void onFailure(Request request, IOException e) {  
 e.printStackTrace();  
 }  
  
 @Override  
 public void onResponse(Response response) throws IOException {  
 if (response.isSuccessful()){  
 final String myResponce = response.body().string();  
 MainActivity.this.runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 textViewResult.setText(myResponce);  
 }  
 });  
 }  
 }  
 });  
 }  
}

### Retrofit

<https://square.github.io/retrofit/>

#### GET

Video tutorial: <https://www.youtube.com/watch?v=4JGvDUlfk7Y&list=PLrnPJCHvNZuCbuD3xpfKzQWOj3AXybSaM>

Json placeholder: <https://jsonplaceholder.typicode.com/>

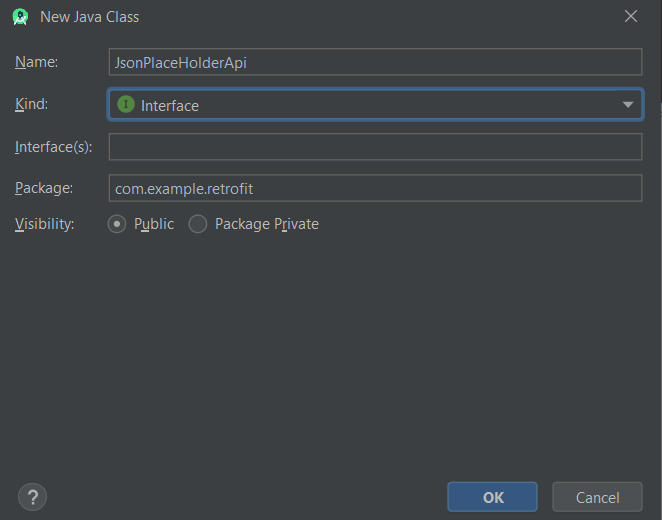
1. Implement retrofit package – Gradle mobile app

implementation 'com.squareup.retrofit2:retrofit:2.4.0'  
implementation 'com.squareup.retrofit2:converter-gson:2.4.0'

1. Create new java class Post.java

package com.example.retrofit;  
  
import com.google.gson.annotations.SerializedName;  
  
public class Post {  
 private int userId;  
 private int id;  
 private String title;  
 @SerializedName("body")  
 private String text;  
  
 public int getUserId() {  
 return userId;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public String getText() {  
 return text;  
 }  
}

1. Create new Java class – type interface



package com.example.retrofit;  
  
import java.util.List;  
  
import retrofit2.Call;  
import retrofit2.http.GET;  
  
public interface JsonPlaceHolderApi {  
 @GET("posts")  
 Call<List<Post>> getPosts();  
}

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="8dp"  
 tools:context=".MainActivity">  
  
 <androidx.core.widget.NestedScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 >  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/tv\_result"  
 android:textColor="#000"  
 android:text="Hello World!"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
 </androidx.core.widget.NestedScrollView>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

MainActivity.java

package com.example.retrofit;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.TextView;  
  
import java.util.List;  
  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
import retrofit2.Retrofit;  
import retrofit2.converter.gson.GsonConverterFactory;  
  
public class MainActivity extends AppCompatActivity {  
  
 private TextView tv\_result;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 tv\_result = (TextView) findViewById(R.id.*tv\_result*);  
  
 Retrofit retrofit = new Retrofit.Builder()  
 .baseUrl("https://jsonplaceholder.typicode.com/")  
 .addConverterFactory(GsonConverterFactory.*create*())  
 .build();  
  
 JsonPlaceHolderApi jsonPlaceHolderApi = retrofit.create(JsonPlaceHolderApi.class);  
 Call<List<Post>> call = jsonPlaceHolderApi.getPosts();  
 call.enqueue(new Callback<List<Post>>() {  
 @Override  
 public void onResponse(Call<List<Post>> call, Response<List<Post>> response) {  
 if(!response.isSuccessful()){  
 tv\_result.setText("Code: "+ response.code());  
 return;  
 }  
  
 List<Post> posts = response.body();  
 for(Post post: posts){  
 String content = "";  
 content += "ID: " + post.getId() + "\n";  
 content += "User ID: " + post.getUserId() + "\n";  
 content += "Title: " + post.getTitle() + "\n";  
 content += "Text: " + post.getText() + "\n\n";  
 tv\_result.append(content);  
 }  
 }  
  
 @Override  
 public void onFailure(Call<List<Post>> call, Throwable t) {  
 tv\_result.setText(t.getMessage());  
 }  
 });  
 }  
}

AndroidManifest.xml

<uses-permission android:name="android.permission.INTERNET"/>

#### GET URL MANIPULATION

<https://www.youtube.com/watch?v=TyJEDhauUeQ&list=PLrnPJCHvNZuCbuD3xpfKzQWOj3AXybSaM&index=2>

1. JsonPlaceholderApi.java

package com.example.retrofit;  
  
import java.util.List;  
import java.util.Map;  
  
import retrofit2.Call;  
import retrofit2.http.GET;  
import retrofit2.http.Path;  
import retrofit2.http.Query;  
import retrofit2.http.QueryMap;  
import retrofit2.http.Url;  
  
public interface JsonPlaceHolderApi {  
 @GET("posts")  
 Call<List<Post>> getPosts(  
 @Query("userId") Integer[] userId,  
 @Query("\_sort") String sort,  
 @Query("\_order") String order  
 );  
  
 @GET("posts")  
 Call<List<Post>> getPosts(@QueryMap Map<String, String> parameters);  
  
 @GET("posts/{id}/comments")  
 Call<List<Comments>> getComments(@Path("id") int postId);  
  
 @GET  
 Call<List<Comments>> getComments(@Url String url);  
}

1. Create java class Comments.java

package com.example.retrofit;  
  
import com.google.gson.annotations.SerializedName;  
  
public class Comments {  
 private int postId;  
 private int id;  
 private String name;  
 private String email;  
 @SerializedName("body")  
 private String text;  
  
 public int getPostId() {  
 return postId;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public String getText() {  
 return text;  
 }  
}

1. MainActivity.java

package com.example.retrofit;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.TextView;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
import retrofit2.Retrofit;  
import retrofit2.converter.gson.GsonConverterFactory;  
  
public class MainActivity extends AppCompatActivity {  
  
 private TextView tv\_result;  
 private JsonPlaceHolderApi jsonPlaceHolderApi;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 tv\_result = (TextView) findViewById(R.id.*tv\_result*);  
  
 Retrofit retrofit = new Retrofit.Builder()  
 .baseUrl("https://jsonplaceholder.typicode.com/")  
 .addConverterFactory(GsonConverterFactory.*create*())  
 .build();  
  
 jsonPlaceHolderApi = retrofit.create(JsonPlaceHolderApi.class);  
// getPosts();  
 getComments();  
 }  
  
 private void getComments() {  
// Call<List<Comments>> call = jsonPlaceHolderApi.getComments(3);  
 Call<List<Comments>> call = jsonPlaceHolderApi.getComments("posts/3/comments");  
 call.enqueue(new Callback<List<Comments>>() {  
 @Override  
 public void onResponse(Call<List<Comments>> call, Response<List<Comments>> response) {  
 if(!response.isSuccessful()){  
 tv\_result.setText("Code: "+ response.code());  
 return;  
 }  
  
 List<Comments> posts = response.body();  
 for(Comments post: posts){  
 String content = "";  
 content += "ID: " + post.getId() + "\n";  
 content += "Post ID: " + post.getPostId() + "\n";  
 content += "Name: " + post.getName() + "\n";  
 content += "Email: " + post.getEmail() + "\n";  
 content += "Text: " + post.getText() + "\n\n";  
 tv\_result.append(content);  
 }  
 }  
  
 @Override  
 public void onFailure(Call<List<Comments>> call, Throwable t) {  
 tv\_result.setText(t.getMessage());  
 }  
 });  
 }  
  
  
 private void getPosts() {  
// Call<List<Post>> call = jsonPlaceHolderApi.getPosts(1,4, "id", "desc");  
// Call<List<Post>> call = jsonPlaceHolderApi.getPosts(new Integer[]{2,3,4}, null, null);  
 Map<String,String> parameters = new HashMap<>();  
 parameters.put("userId", "1");  
 parameters.put("\_sort", "id");  
 parameters.put("\_order", "desc");  
  
 Call<List<Post>> call = jsonPlaceHolderApi.getPosts(parameters);  
 call.enqueue(new Callback<List<Post>>() {  
 @Override  
 public void onResponse(Call<List<Post>> call, Response<List<Post>> response) {  
 if(!response.isSuccessful()){  
 tv\_result.setText("Code: "+ response.code());  
 return;  
 }  
  
 List<Post> posts = response.body();  
 for(Post post: posts){  
 String content = "";  
 content += "ID: " + post.getId() + "\n";  
 content += "User ID: " + post.getUserId() + "\n";  
 content += "Title: " + post.getTitle() + "\n";  
 content += "Text: " + post.getText() + "\n\n";  
 tv\_result.append(content);  
 }  
 }  
  
 @Override  
 public void onFailure(Call<List<Post>> call, Throwable t) {  
 tv\_result.setText(t.getMessage());  
 }  
 });  
 }  
}

#### POST

<https://www.youtube.com/watch?v=GP5OyYDu_mU&list=PLrnPJCHvNZuCbuD3xpfKzQWOj3AXybSaM&index=3>

##### Json

1. Post.java

package com.example.retrofit;  
  
import com.google.gson.annotations.SerializedName;  
  
public class Post {  
 private int userId;  
 private int id;  
 private String title;  
 @SerializedName("body")  
 private String text;  
  
 public Post(int userId, String title, String text) {  
 this.userId = userId;  
 this.title = title;  
 this.text = text;  
 }  
  
 public int getUserId() {  
 return userId;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public String getText() {  
 return text;  
 }  
}

1. JsonPlaceholderApi.java

package com.example.retrofit;  
  
import java.util.List;  
import java.util.Map;  
  
import retrofit2.Call;  
import retrofit2.http.Body;  
import retrofit2.http.Field;  
import retrofit2.http.FormUrlEncoded;  
import retrofit2.http.GET;  
import retrofit2.http.POST;  
import retrofit2.http.Path;  
import retrofit2.http.Query;  
import retrofit2.http.QueryMap;  
import retrofit2.http.Url;  
  
public interface JsonPlaceHolderApi {  
 @GET("posts")  
 Call<List<Post>> getPosts(  
 @Query("userId") Integer[] userId,  
 @Query("\_sort") String sort,  
 @Query("\_order") String order  
 );  
  
 @GET("posts")  
 Call<List<Post>> getPosts(@QueryMap Map<String, String> parameters);  
  
 @GET("posts/{id}/comments")  
 Call<List<Comments>> getComments(@Path("id") int postId);  
  
 @GET  
 Call<List<Comments>> getComments(@Url String url);  
  
 @POST("posts")  
 Call<Post> createPost(@Body Post post);  
  
 @FormUrlEncoded  
 @POST("posts")  
 Call<Post> createPost(  
 @Field("userId") int userId,  
 @Field("title") String title,  
 @Field("body") String text  
 );  
}

1. MainActivity.java

package com.example.retrofit;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.TextView;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
import retrofit2.Retrofit;  
import retrofit2.converter.gson.GsonConverterFactory;  
  
public class MainActivity extends AppCompatActivity {  
  
 private TextView tv\_result;  
 private JsonPlaceHolderApi jsonPlaceHolderApi;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 tv\_result = (TextView) findViewById(R.id.*tv\_result*);  
  
 Retrofit retrofit = new Retrofit.Builder()  
 .baseUrl("https://jsonplaceholder.typicode.com/")  
 .addConverterFactory(GsonConverterFactory.*create*())  
 .build();  
  
 jsonPlaceHolderApi = retrofit.create(JsonPlaceHolderApi.class);  
// getPosts();  
// getComments();  
 createPost();  
 }  
  
 private void createPost() {  
 Post post = new Post(23,"New Title", "New Text");  
 Call<Post> call = jsonPlaceHolderApi.createPost(post);  
 call.enqueue(new Callback<Post>() {  
 @Override  
 public void onResponse(Call<Post> call, Response<Post> response) {  
 if(!response.isSuccessful()){  
 tv\_result.setText("Code: "+ response.code());  
 return;  
 }  
  
 Post postResponse = response.body();  
 String content = "";  
 content += "Code: " + response.code() + "\n";  
 content += "ID: " + postResponse.getId() + "\n";  
 content += "User ID: " + postResponse.getUserId() + "\n";  
 content += "Title: " + postResponse.getTitle() + "\n";  
 content += "Text: " + postResponse.getText() + "\n\n";  
 tv\_result.setText(content);  
 }  
  
 @Override  
 public void onFailure(Call<Post> call, Throwable t) {  
 tv\_result.setText(t.getMessage());  
 }  
 });  
 }  
  
 private void getComments() {  
// Call<List<Comments>> call = jsonPlaceHolderApi.getComments(3);  
 Call<List<Comments>> call = jsonPlaceHolderApi.getComments("posts/3/comments");  
 call.enqueue(new Callback<List<Comments>>() {  
 @Override  
 public void onResponse(Call<List<Comments>> call, Response<List<Comments>> response) {  
 if(!response.isSuccessful()){  
 tv\_result.setText("Code: "+ response.code());  
 return;  
 }  
  
 List<Comments> posts = response.body();  
 for(Comments post: posts){  
 String content = "";  
 content += "ID: " + post.getId() + "\n";  
 content += "Post ID: " + post.getPostId() + "\n";  
 content += "Name: " + post.getName() + "\n";  
 content += "Email: " + post.getEmail() + "\n";  
 content += "Text: " + post.getText() + "\n\n";  
 tv\_result.append(content);  
 }  
 }  
  
 @Override  
 public void onFailure(Call<List<Comments>> call, Throwable t) {  
 tv\_result.setText(t.getMessage());  
 }  
 });  
 }  
  
  
 private void getPosts() {  
// Call<List<Post>> call = jsonPlaceHolderApi.getPosts(1,4, "id", "desc");  
// Call<List<Post>> call = jsonPlaceHolderApi.getPosts(new Integer[]{2,3,4}, null, null);  
 Map<String,String> parameters = new HashMap<>();  
 parameters.put("userId", "1");  
 parameters.put("\_sort", "id");  
 parameters.put("\_order", "desc");  
  
 Call<List<Post>> call = jsonPlaceHolderApi.getPosts(parameters);  
 call.enqueue(new Callback<List<Post>>() {  
 @Override  
 public void onResponse(Call<List<Post>> call, Response<List<Post>> response) {  
 if(!response.isSuccessful()){  
 tv\_result.setText("Code: "+ response.code());  
 return;  
 }  
  
 List<Post> posts = response.body();  
 for(Post post: posts){  
 String content = "";  
 content += "ID: " + post.getId() + "\n";  
 content += "User ID: " + post.getUserId() + "\n";  
 content += "Title: " + post.getTitle() + "\n";  
 content += "Text: " + post.getText() + "\n\n";  
 tv\_result.append(content);  
 }  
 }  
  
 @Override  
 public void onFailure(Call<List<Post>> call, Throwable t) {  
 tv\_result.setText(t.getMessage());  
 }  
 });  
 }  
}

##### FormEncoded



1. JsonPlaceholderApi.java

package com.example.retrofit;  
  
import java.util.List;  
import java.util.Map;  
  
import retrofit2.Call;  
import retrofit2.http.Body;  
import retrofit2.http.Field;  
import retrofit2.http.FieldMap;  
import retrofit2.http.FormUrlEncoded;  
import retrofit2.http.GET;  
import retrofit2.http.POST;  
import retrofit2.http.Path;  
import retrofit2.http.Query;  
import retrofit2.http.QueryMap;  
import retrofit2.http.Url;  
  
public interface JsonPlaceHolderApi {  
 @GET("posts")  
 Call<List<Post>> getPosts(  
 @Query("userId") Integer[] userId,  
 @Query("\_sort") String sort,  
 @Query("\_order") String order  
 );  
  
 @GET("posts")  
 Call<List<Post>> getPosts(@QueryMap Map<String, String> parameters);  
  
 @GET("posts/{id}/comments")  
 Call<List<Comments>> getComments(@Path("id") int postId);  
  
 @GET  
 Call<List<Comments>> getComments(@Url String url);  
  
 @POST("posts")  
 Call<Post> createPost(@Body Post post);  
  
 @FormUrlEncoded  
 @POST("posts")  
 Call<Post> createPost(  
 @Field("userId") int userId,  
 @Field("title") String title,  
 @Field("body") String text  
 );  
  
 @FormUrlEncoded  
 @POST("posts")  
 Call<Post> createPost(@FieldMap Map<String, String> fields);  
}

1. MainActivity.java

package com.example.retrofit;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.TextView;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
import retrofit2.Retrofit;  
import retrofit2.converter.gson.GsonConverterFactory;  
  
public class MainActivity extends AppCompatActivity {  
  
 private TextView tv\_result;  
 private JsonPlaceHolderApi jsonPlaceHolderApi;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 tv\_result = (TextView) findViewById(R.id.*tv\_result*);  
  
 Retrofit retrofit = new Retrofit.Builder()  
 .baseUrl("https://jsonplaceholder.typicode.com/")  
 .addConverterFactory(GsonConverterFactory.*create*())  
 .build();  
  
 jsonPlaceHolderApi = retrofit.create(JsonPlaceHolderApi.class);  
// getPosts();  
// getComments();  
 createPost();  
 }  
  
 private void createPost() {  
// Post post = new Post(23,"New Title", "New Text");  
// Call<Post> call = jsonPlaceHolderApi.createPost(23, "New Title", "New Text");  
 Map<String, String> fields = new HashMap<>();  
 fields.put("userId", "25");  
 fields.put("title", "New Title");  
 fields.put("body", "New Body");  
 Call<Post> call = jsonPlaceHolderApi.createPost(fields);  
 call.enqueue(new Callback<Post>() {  
 @Override  
 public void onResponse(Call<Post> call, Response<Post> response) {  
 if(!response.isSuccessful()){  
 tv\_result.setText("Code: "+ response.code());  
 return;  
 }  
  
 Post postResponse = response.body();  
 String content = "";  
 content += "Code: " + response.code() + "\n";  
 content += "ID: " + postResponse.getId() + "\n";  
 content += "User ID: " + postResponse.getUserId() + "\n";  
 content += "Title: " + postResponse.getTitle() + "\n";  
 content += "Text: " + postResponse.getText() + "\n\n";  
 tv\_result.setText(content);  
 }  
  
 @Override  
 public void onFailure(Call<Post> call, Throwable t) {  
 tv\_result.setText(t.getMessage());  
 }  
 });  
 }  
  
 private void getComments() {  
// Call<List<Comments>> call = jsonPlaceHolderApi.getComments(3);  
 Call<List<Comments>> call = jsonPlaceHolderApi.getComments("posts/3/comments");  
 call.enqueue(new Callback<List<Comments>>() {  
 @Override  
 public void onResponse(Call<List<Comments>> call, Response<List<Comments>> response) {  
 if(!response.isSuccessful()){  
 tv\_result.setText("Code: "+ response.code());  
 return;  
 }  
  
 List<Comments> posts = response.body();  
 for(Comments post: posts){  
 String content = "";  
 content += "ID: " + post.getId() + "\n";  
 content += "Post ID: " + post.getPostId() + "\n";  
 content += "Name: " + post.getName() + "\n";  
 content += "Email: " + post.getEmail() + "\n";  
 content += "Text: " + post.getText() + "\n\n";  
 tv\_result.append(content);  
 }  
 }  
  
 @Override  
 public void onFailure(Call<List<Comments>> call, Throwable t) {  
 tv\_result.setText(t.getMessage());  
 }  
 });  
 }  
  
  
 private void getPosts() {  
// Call<List<Post>> call = jsonPlaceHolderApi.getPosts(1,4, "id", "desc");  
// Call<List<Post>> call = jsonPlaceHolderApi.getPosts(new Integer[]{2,3,4}, null, null);  
 Map<String,String> parameters = new HashMap<>();  
 parameters.put("userId", "1");  
 parameters.put("\_sort", "id");  
 parameters.put("\_order", "desc");  
  
 Call<List<Post>> call = jsonPlaceHolderApi.getPosts(parameters);  
 call.enqueue(new Callback<List<Post>>() {  
 @Override  
 public void onResponse(Call<List<Post>> call, Response<List<Post>> response) {  
 if(!response.isSuccessful()){  
 tv\_result.setText("Code: "+ response.code());  
 return;  
 }  
  
 List<Post> posts = response.body();  
 for(Post post: posts){  
 String content = "";  
 content += "ID: " + post.getId() + "\n";  
 content += "User ID: " + post.getUserId() + "\n";  
 content += "Title: " + post.getTitle() + "\n";  
 content += "Text: " + post.getText() + "\n\n";  
 tv\_result.append(content);  
 }  
 }  
  
 @Override  
 public void onFailure(Call<List<Post>> call, Throwable t) {  
 tv\_result.setText(t.getMessage());  
 }  
 });  
 }  
}

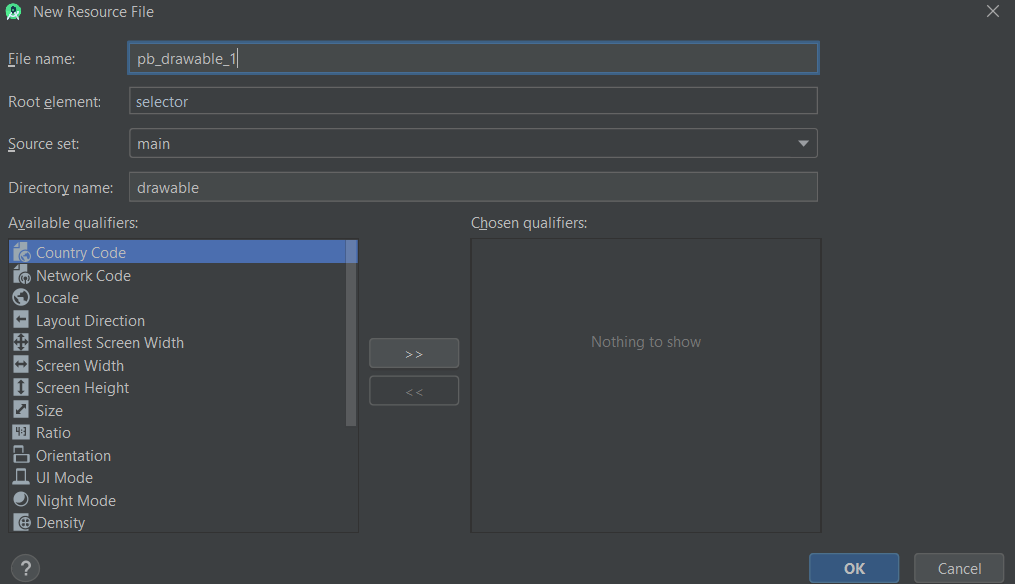
### Connect Non secured site – http

<https://better-coding.com/solved-android-cannot-send-data-to-the-server-cleartext-communication-to-not-permitted-by-network-security-policy/>

## Progress Bar

<https://www.youtube.com/watch?v=2qkgqgeC5r4>

1. Create new resource file in drawable folder – pb\_drawable\_1.xml



<?xml version="1.0" encoding="utf-8"?>  
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">  
 <item>  
 <shape>  
 <solid android:color="@android:color/white"/>  
 </shape>  
  
 </item>  
 <item>  
 <clip>  
 <shape>  
 <gradient android:angle="45"  
 android:endColor="#3eff1c"  
 android:startColor="#04678e"  
 />  
 </shape>  
 </clip>  
 </item>  
</layer-list>

1. Create another drawable file- pb\_drawable\_2.xml

<?xml version="1.0" encoding="utf-8"?>  
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">  
 <item android:id="@android:id/background">  
 <shape>  
 <gradient android:startColor="#424242"  
 android:endColor="#757575"  
 android:centerColor="#616161"  
 android:angle="-90" />  
 <size android:width="15dp"  
 android:height="15dp" />  
 <corners android:radius="6dp" />  
 </shape>  
 </item>  
 <item android:id="@android:id/progress">  
 <clip>  
 <layer-list>  
 <item android:bottom="2dp"  
 android:left="2dp"  
 android:top="2dp"  
 android:right="2dp"  
 >  
 <shape>  
 <gradient android:angle="-90"  
 android:centerColor="#43a047"  
 android:endColor="#388e3c"  
 android:startColor="#4caf50"/>  
 <corners android:radius="4dp" />  
 </shape>  
 </item>  
 </layer-list>  
 </clip>  
 </item>  
</layer-list>

1. Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:padding="16dp"  
 tools:context=".MainActivity">  
  
  
 <Button  
 android:id="@+id/btnStart"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="12dp"  
 android:text="Submit"  
 />  
 <ProgressBar  
 android:id="@+id/progressBarAnim"  
 style="?android:attr/progressBarStyleHorizontal"  
 android:layout\_width="match\_parent"  
 android:layout\_marginBottom="12dp"  
  
 android:secondaryProgressTint="@color/colorPrimaryDark"  
  
 android:progressTint="@color/colorAccent"  
 android:layout\_height="wrap\_content" />  
 <ProgressBar  
 android:id="@+id/progressBar2"  
 style="?android:attr/progressBarStyleHorizontal"  
 android:layout\_width="match\_parent"  
 android:indeterminate="true"  
 android:visibility="invisible"  
 android:progressTint="@color/colorPrimary"  
 android:secondaryProgressTint="@color/colorPrimaryDark"  
 android:layout\_marginBottom="12dp"  
 android:layout\_height="wrap\_content" />  
 <ProgressBar  
 android:id="@+id/progressBar3"  
 style="?android:attr/progressBarStyleHorizontal"  
 android:layout\_width="match\_parent"  
 android:layout\_marginBottom="12dp"  
 android:indeterminate="true"  
 android:indeterminateDuration="5000"  
 android:indeterminateDrawable="@drawable/pb\_drawable\_1"  
 android:layout\_height="wrap\_content" />  
 <ProgressBar  
 android:id="@+id/progressBar4"  
 style="?android:attr/progressBarStyleHorizontal"  
 android:layout\_width="match\_parent"  
 android:layout\_marginBottom="12dp"  
 android:indeterminate="true"  
 android:indeterminateDrawable="@drawable/pb\_drawable\_2"  
 android:layout\_height="wrap\_content" />  
</LinearLayout>

1. MainActivity.java

package com.example.progressbar;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.animation.Animator;  
import android.animation.AnimatorListenerAdapter;  
import android.animation.ObjectAnimator;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.ProgressBar;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
  
 private ProgressBar progressBarAnimator;  
 private ObjectAnimator prograssAnimator;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
   
 init();  
 prograssAnimator.setDuration(7000);  
 findViewById(R.id.*btnStart*).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 prograssAnimator.start();  
 findViewById(R.id.*progressBar2*).setVisibility(View.*VISIBLE*);  
 }  
 });  
 prograssAnimator.addListener(new AnimatorListenerAdapter() {  
 @Override  
 public void onAnimationEnd(Animator animation) {  
 super.onAnimationEnd(animation);  
 Toast.*makeText*(MainActivity.this, "OP Completed", Toast.*LENGTH\_SHORT*).show();  
 progressBarAnimator.setVisibility(View.*GONE*);  
 }  
 });  
 }  
  
 private void init() {  
 progressBarAnimator = findViewById(R.id.*progressBarAnim*);  
 prograssAnimator = ObjectAnimator.*ofInt*(progressBarAnimator, "progress", 0, 100);  
  
 }  
}

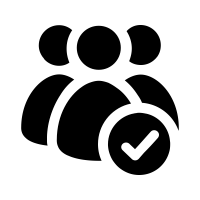
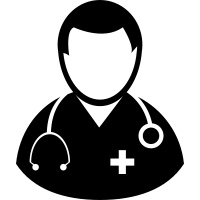
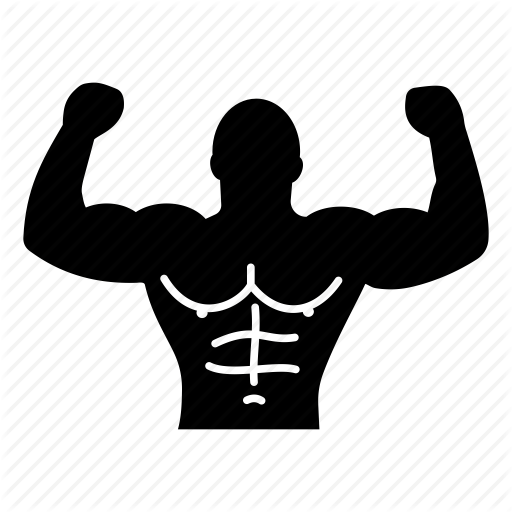
## Grid Layout and CardView

<https://www.youtube.com/watch?v=VUPM387qyrw>

1. Implement cardView

implementation "androidx.cardview:cardview:1.0.0"

1. Images



1. Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/admin\_home\_bg"  
 android:weightSum="10"  
 tools:context=".MainActivity">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="2">  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/textGrid"  
 android:textSize="34dp"  
 android:text="GRID LAYOUT"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:textColor="@android:color/white"  
 />  
 </RelativeLayout>  
  
 <GridLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="6"  
 android:columnCount="2"  
 android:rowCount="2"  
 android:alignmentMode="alignMargins"  
 android:columnOrderPreserved="false"  
 android:padding="14dp"  
 >  
 <!-- Row 1-->  
 <!-- Column 1-->  
 <androidx.cardview.widget.CardView  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_columnWeight="1"  
 android:layout\_rowWeight="1"  
 android:layout\_marginBottom="16dp"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 app:cardElevation="8dp"  
 app:cardCornerRadius="8dp"  
 >  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="16dp"  
 android:orientation="vertical"  
 android:layout\_gravity="center\_horizontal|center\_vertical"  
 >  
 <ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/gym\_members"  
 android:layout\_gravity="center\_horizontal"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Gym Members"  
 android:textColor="@android:color/black"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:textAlignment="center" />  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
 <!-- Column 2-->  
 <androidx.cardview.widget.CardView  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_columnWeight="1"  
 android:layout\_rowWeight="1"  
 android:layout\_marginBottom="16dp"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 app:cardElevation="8dp"  
 app:cardCornerRadius="8dp"  
 >  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="16dp"  
 android:orientation="vertical"  
 android:layout\_gravity="center\_horizontal|center\_vertical"  
 >  
 <ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/gym\_expert"  
 android:layout\_gravity="center\_horizontal"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Gym Experts"  
 android:textColor="@android:color/black"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:textAlignment="center" />  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
 <!-- Row 2-->  
 <!-- Column 1-->  
 <androidx.cardview.widget.CardView  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_columnWeight="1"  
 android:layout\_rowWeight="1"  
 android:layout\_marginBottom="16dp"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 app:cardElevation="8dp"  
 app:cardCornerRadius="8dp"  
 >  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="16dp"  
 android:orientation="vertical"  
 android:layout\_gravity="center\_horizontal|center\_vertical"  
 >  
 <ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/doctors"  
 android:layout\_gravity="center\_horizontal"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Doctors"  
 android:textColor="@android:color/black"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:textAlignment="center" />  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
 <!-- Column 2-->  
 <androidx.cardview.widget.CardView  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_columnWeight="1"  
 android:layout\_rowWeight="1"  
 android:layout\_marginBottom="16dp"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 app:cardElevation="8dp"  
 app:cardCornerRadius="8dp"  
 >  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="16dp"  
 android:orientation="vertical"  
 android:layout\_gravity="center\_horizontal|center\_vertical"  
 >  
 <ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/attendance"  
 android:layout\_gravity="center\_horizontal"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Attendance"  
 android:textColor="@android:color/black"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:textAlignment="center" />  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
 </GridLayout>  
  
</LinearLayout>

## Date picker Dialogue box

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context=".MainActivity">

<EditText

android:id="@+id/date"

android:layout\_width="200dp"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:background="#d4d4d4"

android:hint="Select Date..."

android:padding="15dp"

android:textColor="#897"

android:textColorHint="#090"

android:textSize="20sp"

android:textStyle="bold" />

</RelativeLayout>

MainActivity.java

package example.abhiandroid.datepickerexample;

import android.app.DatePickerDialog;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.DatePicker;

import android.widget.EditText;

import java.util.Calendar;

public class MainActivity extends AppCompatActivity {

EditText date;

DatePickerDialog datePickerDialog;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

// initiate the date picker and a button

date = (EditText) findViewById(R.id.date);

// perform click event on edit text

date.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// calender class's instance and get current date , month and year from calender

final Calendar c = Calendar.getInstance();

int mYear = c.get(Calendar.YEAR); // current year

int mMonth = c.get(Calendar.MONTH); // current month

int mDay = c.get(Calendar.DAY\_OF\_MONTH); // current day

// date picker dialog

datePickerDialog = new DatePickerDialog(MainActivity.this,

new DatePickerDialog.OnDateSetListener() {

@Override

public void onDateSet(DatePicker view, int year,

int monthOfYear, int dayOfMonth) {

// set day of month , month and year value in the edit text

date.setText(dayOfMonth + "/"

+ (monthOfYear + 1) + "/" + year);

}

}, mYear, mMonth, mDay);

datePickerDialog.show();

}

});

}

}

## Open Gallery

Intent intent = new Intent();

intent.setType("image/\*");

intent.setAction(Intent.ACTION\_GET\_CONTENT);

startActivityForResult(Intent.createChooser(intent, "Select Picture"),SELECT\_IMAGE);

OnActivityResult for get image.

public void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == SELECT\_IMAGE) {

if (resultCode == Activity.RESULT\_OK) {

if (data != null) {

try {

Bitmap bitmap = MediaStore.Images.Media.getBitmap(getActivity().getContentResolver(), data.getData());

} catch (IOException e) {

e.printStackTrace();

}

}

} else if (resultCode == Activity.RESULT\_CANCELED) {

Toast.makeText(getActivity(), "Canceled", Toast.LENGTH\_SHORT).show();

}

}

}

AndroidManiferst.xml

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" android:maxSdkVersion="18" />

## SQL

### Retreive data using foreign key

This is the result of separating a single table in two:

Table users:

user\_id (pk, ai)

email

password

last\_login

Table data:

user\_id (fk to users.user\_id)

data\_1

data\_2

To select a single record when there was only one table:

SELECT users.email, users.password, data.data\_1, data.data\_2

FROM users,data

WHERE users.email='$user\_email' AND users.user\_id=data.user\_id";

How do I get all records from both tables having the rows connected by users.user\_id=data.user\_id?

Row1: email, password, data\_1, data2

Row2: email, password, data\_1, data2

Row3: email, password, data\_1, data2

Row4: email, password, data\_1, data2

## Firebase Chat App –NOT WORKING

<https://www.youtube.com/watch?v=fJWFeW09qeE&list=PLzLFqCABnRQftQQETzoVMuteXzNiXmnj8>

1. Create project “firebase chat app”
2. Gradle file

implementation 'com.google.firebase:firebase-database:16.0.2'  
implementation 'com.google.firebase:firebase-auth:16.0.1'  
implementation 'com.google.firebase:firebase-core:16.0.1'

1. MainActivity
2. Tools->firebase
3. Authentication-> email & password authentication->connect to firebase
4. It will open firebase in browser create new project
5. Add firebase authentication to app – xlick button in android studio

# Firebase - WHATSAPP

<https://www.youtube.com/watch?v=XkBYzLQ60to&list=PLxefhmF0pcPmtdoud8f64EpgapkclCllj&index=2>

## Connect to firebase

1. Create new project – whatsapp –kitcat
2. Colors.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="colorPrimary">#006257</color>  
 <color name="colorPrimaryDark">#005048</color>  
 <color name="colorAccent">#E6E6E6</color>  
</resources>

1. Tools-> firebase ->authentication -> browser (chatapp)
2. Add firebase auth to app
3. Firebase -> real time database -> save and retrieve data
4. Add real time database to your app
5. Fireabase -> storage -> upload and download data from cloud storage
6. Add storage to app

## Toolbar

1. Styles.xml

<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">

1. Res -> rc -> new layout resource file-> app\_bar\_layout.xml = root element – toolbar

<?xml version="1.0" encoding="utf-8"?>  
<androidx.appcompat.widget.Toolbar xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/main\_app\_bar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@color/colorPrimary"  
 android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">  
  
</androidx.appcompat.widget.Toolbar>

1. Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/colorAccent"  
 tools:context=".MainActivity">  
  
 <include  
 android:id="@+id/main\_page\_toolbar"  
 layout="@layout/app\_bar\_layout"  
 ></include>  
  
</RelativeLayout>

1. MainActivity.java

package com.example.whatsapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.os.Bundle;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Toolbar mToolbar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mToolbar = (Toolbar) findViewById(R.id.*main\_page\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("WhatsApp");  
 }  
}

## TAB Layout – fragment view

1. Dependency

implementation 'com.android.support:appcompat-v7:28.0.0'  
implementation 'com.android.support:design:28.0.0'

1. Create 3 new fragments
2. Com.example.appname -> rc -> new -> fragment -> fragment(blank) = (ChatsFragment, GroupsFragment, ContactFragment), uncheck – include fragment factory methords.
3. Create new class – TabsAccessorAdapter.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentManager;  
import androidx.fragment.app.FragmentPagerAdapter;  
  
public class TabsAccessorAdapter extends FragmentPagerAdapter {  
 public TabsAccessorAdapter(@NonNull FragmentManager fm) {  
 super(fm);  
 }  
  
 @NonNull  
 @Override  
 public Fragment getItem(int position) {  
  
 switch (position){  
 case 0:  
 ChatsFragment chatsFragment = new ChatsFragment();  
 return chatsFragment;  
 case 1:  
 GroupsFragment groupsFragment = new GroupsFragment();  
 return groupsFragment;  
 case 2:  
 ContactsFragment contactsFragment = new ContactsFragment();  
 return contactsFragment;  
 default:  
 return null;  
 }  
  
  
 }  
  
 @Override  
 public int getCount() {  
 return 3;  
 }  
  
 @Nullable  
 @Override  
 public CharSequence getPageTitle(int position) {  
 switch (position){  
 case 0:  
 return "Chats";  
 case 1:  
 return "Groups";  
 case 2:  
 return "Contacts";  
 default:  
 return null;  
 }  
 }  
}

1. Fragment\_chats.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ChatsFragment">  
  
 <!-- *TODO: Update blank fragment layout* -->  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Chat Fragment"  
 android:layout\_centerInParent="true"  
 android:textSize="26dp"  
 android:textStyle="bold" />  
  
</RelativeLayout>

1. Fragment\_groups.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".GroupsFragment">  
  
 <!-- *TODO: Update blank fragment layout* -->  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Groups Fragment"  
 android:layout\_centerInParent="true"  
 android:textSize="26dp"  
 android:textStyle="bold" />  
  
</RelativeLayout>

1. Fragment\_contacs.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ContactsFragment">  
  
 <!-- *TODO: Update blank fragment layout* -->  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Contact Fragment"  
 android:layout\_centerInParent="true"  
 android:textSize="26dp"  
 android:textStyle="bold"/>  
  
</RelativeLayout>

1. Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/colorAccent"  
 tools:context=".MainActivity">  
  
  
  
 <com.google.android.material.appbar.AppBarLayout  
 android:id="@+id/appBarLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
 <include  
 android:id="@+id/main\_page\_toolbar"  
 layout="@layout/app\_bar\_layout"  
 ></include>  
 <com.google.android.material.tabs.TabLayout  
 android:id="@+id/main\_tabs"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">  
  
 </com.google.android.material.tabs.TabLayout>  
 </com.google.android.material.appbar.AppBarLayout>  
   
   
 <androidx.viewpager.widget.ViewPager  
 android:id="@+id/main\_tabs\_pager"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/appBarLayout">  
  
 </androidx.viewpager.widget.ViewPager>  
  
</RelativeLayout>

1. MainActivity.java

package com.example.whatsapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.viewpager.widget.ViewPager;  
  
import android.os.Bundle;  
  
import com.google.android.material.tabs.TabLayout;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Toolbar mToolbar;  
 private ViewPager myViewPager;  
 private TabLayout myTabLayout;  
  
 private TabsAccessorAdapter myTabsAccessorAdapter;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mToolbar = (Toolbar) findViewById(R.id.*main\_page\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("WhatsApp");  
  
 myViewPager = (ViewPager) findViewById(R.id.*main\_tabs\_pager*);  
 myTabsAccessorAdapter = new TabsAccessorAdapter(getSupportFragmentManager());  
 myViewPager.setAdapter(myTabsAccessorAdapter);  
  
 myTabLayout = (TabLayout) findViewById(R.id.*main\_tabs*);  
 myTabLayout.setupWithViewPager(myViewPager);  
 }  
}

## UserAuth – firebase

<https://www.youtube.com/watch?v=yjBCXa01dZc&list=PLxefhmF0pcPmtdoud8f64EpgapkclCllj&index=7>

1. Create new empty activity – LoginActivity
2. Rc drawable folder -> new -> drawable resource file – file name = buttons

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android">  
 <solid  
 android:color="@color/colorPrimary">  
  
 </solid>  
</shape>

1. Rc drawable folder -> new -> drawable resource file – file name = inputs

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android">  
<stroke  
 android:color="@color/colorPrimary"  
 android:width="5dp"  
 ></stroke>  
</shape>

1. Create new activity RegisterActivity
2. Download icons:- <https://www.youtube.com/redirect?redir_token=shfrYpdGdh7iZjGrBM1RP6m5T5d8MTU4OTA0MDQwNEAxNTg4OTU0MDA0&v=yjBCXa01dZc&q=https%3A%2F%2Fdrive.google.com%2Fopen%3Fid%3D1SS8plszKPkyiR5CfeKufhUrCSVl_z5pY&event=video_description>
3. Paste this in drawable folder

### Designing

1. Activity\_login.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".LoginActivity">  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <ImageView  
 android:id="@+id/login\_image"  
 android:layout\_width="match\_parent"  
 android:layout\_height="250dp"  
 android:src="@drawable/login\_photo"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:scaleType="fitCenter"  
 />  
 <EditText  
 android:id="@+id/login\_email"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/login\_image"  
 android:hint="Email"  
 android:inputType="textEmailAddress"  
 android:background="@drawable/inputs"  
 android:padding="4dp"  
 android:drawableStart="@drawable/email"  
 android:layout\_marginTop="25dp"  
 android:layout\_marginLeft="17dp"  
 android:layout\_marginRight="17dp"  
 />  
  
 <EditText  
 android:id="@+id/login\_password"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/login\_email"  
 android:hint="Password"  
 android:inputType="textPassword"  
 android:background="@drawable/inputs"  
 android:padding="4dp"  
 android:drawableStart="@drawable/password"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="17dp"  
 android:layout\_marginRight="17dp"  
 />  
  
 <TextView  
 android:id="@+id/forget\_password\_link"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/login\_password"  
 android:text="@string/forget\_password"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="23dp"  
 android:textAlignment="textEnd"  
 android:textSize="17dp"  
 android:textStyle="bold"  
 android:textColor="@color/colorPrimaryDark"  
 />  
  
 <Button  
 android:id="@+id/login\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/forget\_password\_link"  
 android:padding="4dp"  
 android:layout\_marginTop="4dp"  
 android:layout\_marginLeft="19dp"  
 android:layout\_marginRight="19dp"  
 android:text="@string/login"  
 android:textAllCaps="false"  
 android:textSize="24dp"  
 android:textColor="@android:color/background\_light"  
 android:background="@drawable/buttons"  
 />  
 <TextView  
 android:id="@+id/need\_new\_account\_link"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/login\_button"  
 android:text="Need new Account?"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="23dp"  
 android:textAlignment="center"  
 android:textSize="17dp"  
 android:textStyle="bold"  
 android:textColor="@color/colorPrimaryDark"  
 />  
  
 <TextView  
 android:id="@+id/login\_using"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/need\_new\_account\_link"  
 android:text="Or Login using your"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="23dp"  
 android:textAlignment="center"  
 android:textSize="17dp"  
 />  
 <Button  
 android:id="@+id/phone\_login\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/login\_using"  
 android:padding="4dp"  
 android:layout\_marginTop="4dp"  
 android:layout\_marginLeft="19dp"  
 android:layout\_marginRight="19dp"  
 android:text="@string/phone"  
 android:textAllCaps="false"  
 android:textSize="14sp"  
 android:textColor="@android:color/background\_light"  
 android:background="@drawable/buttons"  
 android:drawableStart="@drawable/phone"  
 />  
 </RelativeLayout>  
 </ScrollView>  
</RelativeLayout>

1. MainActivity.java

package com.example.whatsapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.viewpager.widget.ViewPager;  
  
import android.content.Intent;  
import android.os.Bundle;  
  
import com.google.android.material.tabs.TabLayout;  
import com.google.firebase.auth.FirebaseUser;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Toolbar mToolbar;  
 private ViewPager myViewPager;  
 private TabLayout myTabLayout;  
  
 private TabsAccessorAdapter myTabsAccessorAdapter;  
  
 private FirebaseUser currentUser;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mToolbar = (Toolbar) findViewById(R.id.*main\_page\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("WhatsApp");  
  
 myViewPager = (ViewPager) findViewById(R.id.*main\_tabs\_pager*);  
 myTabsAccessorAdapter = new TabsAccessorAdapter(getSupportFragmentManager());  
 myViewPager.setAdapter(myTabsAccessorAdapter);  
  
 myTabLayout = (TabLayout) findViewById(R.id.*main\_tabs*);  
 myTabLayout.setupWithViewPager(myViewPager);  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 if(currentUser == null){  
 SendUserToLoginActivity();  
 }  
 }  
  
 private void SendUserToLoginActivity() {  
  
 Intent loginIntent = new Intent(MainActivity.this, LoginActivity.class);  
 startActivity(loginIntent);  
 }  
}

1. LoginActivity.java

package com.example.whatsapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
  
import com.google.firebase.auth.FirebaseUser;  
  
public class LoginActivity extends AppCompatActivity {  
  
 private FirebaseUser currentUser;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 if(currentUser != null){  
 SendUserToMainActivity();  
 }  
 }  
  
 private void SendUserToMainActivity() {  
 Intent mainIntent = new Intent(LoginActivity.this, MainActivity.class);  
 startActivity(mainIntent);  
 }  
  
}

1. ActivityRegister.java

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".RegisterActivity">  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <ImageView  
 android:id="@+id/register\_image"  
 android:layout\_width="match\_parent"  
 android:layout\_height="250dp"  
 android:src="@drawable/signup\_photo"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:scaleType="fitCenter"  
 />  
 <EditText  
 android:id="@+id/register\_email"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/register\_image"  
 android:hint="Email"  
 android:inputType="textEmailAddress"  
 android:background="@drawable/inputs"  
 android:padding="4dp"  
 android:drawableStart="@drawable/email"  
 android:layout\_marginTop="25dp"  
 android:layout\_marginLeft="17dp"  
 android:layout\_marginRight="17dp"  
 />  
  
 <EditText  
 android:id="@+id/register\_password"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/register\_email"  
 android:hint="Password"  
 android:inputType="textPassword"  
 android:background="@drawable/inputs"  
 android:padding="4dp"  
 android:drawableStart="@drawable/password"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="17dp"  
 android:layout\_marginRight="17dp"  
 />  
  
  
  
 <Button  
 android:id="@+id/register\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/register\_password"  
 android:padding="4dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="19dp"  
 android:layout\_marginRight="19dp"  
 android:text="@string/create\_account"  
 android:textAllCaps="false"  
 android:textSize="24dp"  
 android:textColor="@android:color/background\_light"  
 android:background="@drawable/buttons"  
 />  
 <TextView  
 android:id="@+id/already\_have\_account\_link"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/register\_button"  
 android:text="Already have an Account?"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="23dp"  
 android:textAlignment="center"  
 android:textSize="17dp"  
 android:textStyle="bold"  
 android:textColor="@color/colorPrimaryDark"  
 />  
 </RelativeLayout>  
 </ScrollView>  
  
 </RelativeLayout>

### Create new account – firebase

1. Browser firebase console-> authentication -> sign in methords -> enable email login
2. LoginActivity.java

package com.example.whatsapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
  
import com.google.firebase.auth.FirebaseUser;  
  
public class LoginActivity extends AppCompatActivity {  
  
 private FirebaseUser currentUser;  
 private Button LoginButton, PhoneLoginButton;  
 private EditText UserEmail, UserPassword;  
 private TextView NeedNewAccountLink, ForgetPasswordLink;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
   
 InitializeFields();  
 }  
  
 private void InitializeFields() {  
 LoginButton = (Button) findViewById(R.id.*login\_button*);  
 PhoneLoginButton = (Button) findViewById(R.id.*phone\_login\_button*);  
 UserEmail = (EditText) findViewById(R.id.*login\_email*);  
 UserPassword = (EditText) findViewById(R.id.*login\_password*);  
 NeedNewAccountLink = (TextView) findViewById(R.id.*need\_new\_account\_link*);  
 ForgetPasswordLink = (TextView) findViewById(R.id.*forget\_password\_link*);  
 }  
  
 public void setNeedNewAccountLink(View view){  
 SendUserToRegisterActivity();  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 if(currentUser != null){  
 SendUserToMainActivity();  
 }  
 }  
  
 private void SendUserToMainActivity() {  
 Intent mainIntent = new Intent(LoginActivity.this, MainActivity.class);  
 startActivity(mainIntent);  
 finish();  
 }  
 private void SendUserToRegisterActivity() {  
 Intent mainIntent = new Intent(LoginActivity.this, RegisterActivity.class);  
 startActivity(mainIntent);  
 finish();  
 }  
}

1. RegisterActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
  
public class RegisterActivity extends AppCompatActivity {  
  
 private Button CreateAccountButton, PhoneLoginButton;  
 private EditText UserEmail, UserPassword;  
 private TextView AlreadyHavaAccountLink;  
  
 private FirebaseAuth mAuth;  
 private ProgressDialog loadingBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_register*);  
  
 InitializeFields();  
  
 mAuth = FirebaseAuth.*getInstance*();  
 }  
  
 private void InitializeFields() {  
 CreateAccountButton = (Button) findViewById(R.id.*register\_button*);  
 UserEmail = (EditText) findViewById(R.id.*register\_email*);  
 UserPassword = (EditText) findViewById(R.id.*register\_password*);  
 AlreadyHavaAccountLink = (TextView) findViewById(R.id.*already\_have\_account\_link*);  
 loadingBar = new ProgressDialog(this);  
 }  
  
 public void setAlreadyHavaAccountLink(View view){  
 SendUserToLoginActivity();  
 }  
  
 private void SendUserToLoginActivity() {  
 Intent mainIntent = new Intent(RegisterActivity.this, LoginActivity.class);  
 startActivity(mainIntent);  
 finish();  
 }  
  
 public void setCreateAccountButton(View view){  
 CreateNewAccount();  
 }  
  
 private void CreateNewAccount() {  
 String email = UserEmail.getText().toString();  
 String password = UserPassword.getText().toString();  
  
 if(TextUtils.*isEmpty*(email)){  
 Toast.*makeText*(this, "Please enter email", Toast.*LENGTH\_SHORT*).show();  
 }  
 if(TextUtils.*isEmpty*(password)){  
 Toast.*makeText*(this, "Please enter password", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 loadingBar.setTitle("Creating new account");  
 loadingBar.setMessage("Please wait while we are creating new account for you");  
 loadingBar.setCanceledOnTouchOutside(true);  
 loadingBar.show();  
 mAuth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
 SendUserToLoginActivity();  
 Toast.*makeText*(RegisterActivity.this, "Account Created Successfully", Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
 }else {  
 String message = task.getException().toString();  
 Toast.*makeText*(RegisterActivity.this, "Error: "+message, Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
 }  
 }  
 });  
 }  
 }  
}

1. Activity\_regisster.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".RegisterActivity">  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <ImageView  
 android:id="@+id/register\_image"  
 android:layout\_width="match\_parent"  
 android:layout\_height="250dp"  
 android:src="@drawable/signup\_photo"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:scaleType="fitCenter"  
 />  
 <EditText  
 android:id="@+id/register\_email"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/register\_image"  
 android:hint="Email"  
 android:inputType="textEmailAddress"  
 android:background="@drawable/inputs"  
 android:padding="4dp"  
 android:drawableStart="@drawable/email"  
 android:layout\_marginTop="25dp"  
 android:layout\_marginLeft="17dp"  
 android:layout\_marginRight="17dp"  
 />  
  
 <EditText  
 android:id="@+id/register\_password"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/register\_email"  
 android:hint="Password"  
 android:inputType="textPassword"  
 android:background="@drawable/inputs"  
 android:padding="4dp"  
 android:drawableStart="@drawable/password"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="17dp"  
 android:layout\_marginRight="17dp"  
 />  
  
  
  
 <Button  
 android:id="@+id/register\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/register\_password"  
 android:padding="4dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="19dp"  
 android:layout\_marginRight="19dp"  
 android:text="@string/create\_account"  
 android:textAllCaps="false"  
 android:textSize="24dp"  
 android:textColor="@android:color/background\_light"  
 android:background="@drawable/buttons"  
 android:onClick="setCreateAccountButton"  
 />  
 <TextView  
 android:id="@+id/already\_have\_account\_link"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/register\_button"  
 android:text="Already have an Account?"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="23dp"  
 android:textAlignment="center"  
 android:textSize="17dp"  
 android:textStyle="bold"  
 android:textColor="@color/colorPrimaryDark"  
 android:onClick="setAlreadyHavaAccountLink"  
 />  
 </RelativeLayout>  
 </ScrollView>  
  
 </RelativeLayout>

1. Activity\_login.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".LoginActivity">  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <ImageView  
 android:id="@+id/login\_image"  
 android:layout\_width="match\_parent"  
 android:layout\_height="250dp"  
 android:src="@drawable/login\_photo"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:scaleType="fitCenter"  
 />  
 <EditText  
 android:id="@+id/login\_email"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/login\_image"  
 android:hint="Email"  
 android:inputType="textEmailAddress"  
 android:background="@drawable/inputs"  
 android:padding="4dp"  
 android:drawableStart="@drawable/email"  
 android:layout\_marginTop="25dp"  
 android:layout\_marginLeft="17dp"  
 android:layout\_marginRight="17dp"  
 />  
  
 <EditText  
 android:id="@+id/login\_password"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/login\_email"  
 android:hint="Password"  
 android:inputType="textPassword"  
 android:background="@drawable/inputs"  
 android:padding="4dp"  
 android:drawableStart="@drawable/password"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="17dp"  
 android:layout\_marginRight="17dp"  
 />  
  
 <TextView  
 android:id="@+id/forget\_password\_link"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/login\_password"  
 android:text="@string/forget\_password"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="23dp"  
 android:textAlignment="textEnd"  
 android:textSize="17dp"  
 android:textStyle="bold"  
 android:textColor="@color/colorPrimaryDark"  
 />  
  
 <Button  
 android:id="@+id/login\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/forget\_password\_link"  
 android:padding="4dp"  
 android:layout\_marginTop="4dp"  
 android:layout\_marginLeft="19dp"  
 android:layout\_marginRight="19dp"  
 android:text="@string/login"  
 android:textAllCaps="false"  
 android:textSize="24dp"  
 android:textColor="@android:color/background\_light"  
 android:background="@drawable/buttons"  
 />  
 <TextView  
 android:id="@+id/need\_new\_account\_link"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/login\_button"  
 android:text="Need new Account?"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="23dp"  
 android:textAlignment="center"  
 android:textSize="17dp"  
 android:textStyle="bold"  
 android:textColor="@color/colorPrimaryDark"  
 android:onClick="setNeedNewAccountLink"  
 />  
  
 <TextView  
 android:id="@+id/login\_using"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/need\_new\_account\_link"  
 android:text="Or Login using your"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="23dp"  
 android:textAlignment="center"  
 android:textSize="17dp"  
 />  
 <Button  
 android:id="@+id/phone\_login\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/login\_using"  
 android:padding="4dp"  
 android:layout\_marginTop="4dp"  
 android:layout\_marginLeft="19dp"  
 android:layout\_marginRight="19dp"  
 android:text="@string/phone"  
 android:textAllCaps="false"  
 android:textSize="14sp"  
 android:textColor="@android:color/background\_light"  
 android:background="@drawable/buttons"  
 android:drawableStart="@drawable/phone"  
 />  
 </RelativeLayout>  
 </ScrollView>  
</RelativeLayout>

### SignIn – firebase

1. LoginActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
public class LoginActivity extends AppCompatActivity {  
  
 private FirebaseUser currentUser;  
 private FirebaseAuth mAuth;  
 private Button LoginButton, PhoneLoginButton;  
 private EditText UserEmail, UserPassword;  
 private TextView NeedNewAccountLink, ForgetPasswordLink;  
 private ProgressDialog loadingBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUser = mAuth.getCurrentUser();  
 InitializeFields();  
 }  
  
 private void InitializeFields() {  
 LoginButton = (Button) findViewById(R.id.*login\_button*);  
 PhoneLoginButton = (Button) findViewById(R.id.*phone\_login\_button*);  
 UserEmail = (EditText) findViewById(R.id.*login\_email*);  
 UserPassword = (EditText) findViewById(R.id.*login\_password*);  
 NeedNewAccountLink = (TextView) findViewById(R.id.*need\_new\_account\_link*);  
 ForgetPasswordLink = (TextView) findViewById(R.id.*forget\_password\_link*);  
 loadingBar = new ProgressDialog(this);  
 }  
  
 public void setNeedNewAccountLink(View view){  
 SendUserToRegisterActivity();  
 }  
 public void setLoginButton(View view){  
 AllowUserToLogin();  
 }  
  
 private void AllowUserToLogin() {  
 String email = UserEmail.getText().toString();  
 String password = UserPassword.getText().toString();  
  
 if(TextUtils.*isEmpty*(email)){  
 Toast.*makeText*(this, "Please enter email", Toast.*LENGTH\_SHORT*).show();  
 }  
 if(TextUtils.*isEmpty*(password)){  
 Toast.*makeText*(this, "Please enter password", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 loadingBar.setTitle("Sign In");  
 loadingBar.setMessage("Please wait while we are authenticating your account");  
 loadingBar.setCanceledOnTouchOutside(true);  
 loadingBar.show();  
 mAuth.signInWithEmailAndPassword(email, password)  
 .addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
 SendUserToMainActivity();  
 Toast.*makeText*(LoginActivity.this, "Logged in successful", Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
 }else {  
 String message = task.getException().toString();  
 Toast.*makeText*(LoginActivity.this, "Error: "+message, Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
 }  
 }  
 });  
 }  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 if(currentUser != null){  
 SendUserToMainActivity();  
 }  
 }  
  
 private void SendUserToMainActivity() {  
 Intent mainIntent = new Intent(LoginActivity.this, MainActivity.class);  
 startActivity(mainIntent);  
 finish();  
 }  
 private void SendUserToRegisterActivity() {  
 Intent mainIntent = new Intent(LoginActivity.this, RegisterActivity.class);  
 startActivity(mainIntent);  
 finish();  
 }  
}

1. MainActivity.java

package com.example.whatsapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.viewpager.widget.ViewPager;  
  
import android.content.Intent;  
import android.os.Bundle;  
  
import com.google.android.material.tabs.TabLayout;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Toolbar mToolbar;  
 private ViewPager myViewPager;  
 private TabLayout myTabLayout;  
  
 private TabsAccessorAdapter myTabsAccessorAdapter;  
  
 private FirebaseUser currentUser;  
 private FirebaseAuth mAuth;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUser = mAuth.getCurrentUser();  
  
 mToolbar = (Toolbar) findViewById(R.id.*main\_page\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("WhatsApp");  
  
 myViewPager = (ViewPager) findViewById(R.id.*main\_tabs\_pager*);  
 myTabsAccessorAdapter = new TabsAccessorAdapter(getSupportFragmentManager());  
 myViewPager.setAdapter(myTabsAccessorAdapter);  
  
 myTabLayout = (TabLayout) findViewById(R.id.*main\_tabs*);  
 myTabLayout.setupWithViewPager(myViewPager);  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 if(currentUser == null){  
 SendUserToLoginActivity();  
 }  
 }  
  
 private void SendUserToLoginActivity() {  
  
 Intent loginIntent = new Intent(MainActivity.this, LoginActivity.class);  
 startActivity(loginIntent);  
 finish();  
 }  
}

## Drop down menu

1. Res -> rc -> resource directory -> name= menu – type = menu
2. Rc – menu -> menu resource file -> file name = options\_menu

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
 <item  
 android:title="@string/find\_friends"  
 android:id="@+id/main\_find\_friends\_option"  
 app:showAsAction="never"></item>  
  
 <item  
 android:title="@string/settings"  
 android:id="@+id/main\_settings\_option"  
 app:showAsAction="never"></item>  
  
 <item  
 android:title="@string/logout"  
 android:id="@+id/main\_logout\_option"  
 app:showAsAction="never"></item>  
</menu>

1. MainActivity.java

package com.example.whatsapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.viewpager.widget.ViewPager;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
  
import com.google.android.material.tabs.TabLayout;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Toolbar mToolbar;  
 private ViewPager myViewPager;  
 private TabLayout myTabLayout;  
  
 private TabsAccessorAdapter myTabsAccessorAdapter;  
  
 private FirebaseUser currentUser;  
 private FirebaseAuth mAuth;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUser = mAuth.getCurrentUser();  
  
 mToolbar = (Toolbar) findViewById(R.id.*main\_page\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("WhatsApp");  
  
 myViewPager = (ViewPager) findViewById(R.id.*main\_tabs\_pager*);  
 myTabsAccessorAdapter = new TabsAccessorAdapter(getSupportFragmentManager());  
 myViewPager.setAdapter(myTabsAccessorAdapter);  
  
 myTabLayout = (TabLayout) findViewById(R.id.*main\_tabs*);  
 myTabLayout.setupWithViewPager(myViewPager);  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 if(currentUser == null){  
 SendUserToLoginActivity();  
 }  
 }  
  
 private void SendUserToLoginActivity() {  
  
 Intent loginIntent = new Intent(MainActivity.this, LoginActivity.class);  
 startActivity(loginIntent);  
 finish();  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 super.onCreateOptionsMenu(menu);  
 getMenuInflater().inflate(R.menu.*options\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 super.onOptionsItemSelected(item);  
  
 if(item.getItemId() == R.id.*main\_logout\_option*){  
 mAuth.signOut();  
 SendUserToLoginActivity();  
 }  
 if(item.getItemId() == R.id.*main\_settings\_option*){  
  
 }  
 if(item.getItemId() == R.id.*main\_find\_friends\_option*){  
  
 }  
  
 return true;  
 }  
}

## Create FirebaseDatabase

1. In browser create new realtime database – test mode
2. Android manifest.xml

<uses-permission android:name="android.permission.INTERNET"></uses-permission>

1. RegisterActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
public class RegisterActivity extends AppCompatActivity {  
  
 private Button CreateAccountButton, PhoneLoginButton;  
 private EditText UserEmail, UserPassword;  
 private TextView AlreadyHavaAccountLink;  
  
 private FirebaseAuth mAuth;  
 private DatabaseReference RootRef;  
 private ProgressDialog loadingBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_register*);  
  
 InitializeFields();  
  
 mAuth = FirebaseAuth.*getInstance*();  
 RootRef = FirebaseDatabase.*getInstance*().getReference();  
 }  
  
 private void InitializeFields() {  
 CreateAccountButton = (Button) findViewById(R.id.*register\_button*);  
 UserEmail = (EditText) findViewById(R.id.*register\_email*);  
 UserPassword = (EditText) findViewById(R.id.*register\_password*);  
 AlreadyHavaAccountLink = (TextView) findViewById(R.id.*already\_have\_account\_link*);  
 loadingBar = new ProgressDialog(this);  
 }  
  
 public void setAlreadyHavaAccountLink(View view){  
 SendUserToLoginActivity();  
 }  
  
 private void SendUserToLoginActivity() {  
 Intent mainIntent = new Intent(RegisterActivity.this, LoginActivity.class);  
 startActivity(mainIntent);  
 finish();  
 }  
  
 public void setCreateAccountButton(View view){  
 CreateNewAccount();  
 }  
  
 private void CreateNewAccount() {  
 String email = UserEmail.getText().toString();  
 String password = UserPassword.getText().toString();  
  
 if(TextUtils.*isEmpty*(email)){  
 Toast.*makeText*(this, "Please enter email", Toast.*LENGTH\_SHORT*).show();  
 }  
 if(TextUtils.*isEmpty*(password)){  
 Toast.*makeText*(this, "Please enter password", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 loadingBar.setTitle("Creating new account");  
 loadingBar.setMessage("Please wait while we are creating new account for you");  
 loadingBar.setCanceledOnTouchOutside(true);  
 loadingBar.show();  
 mAuth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
 String currentUserId = mAuth.getCurrentUser().getUid();  
 RootRef.child("Users").child(currentUserId).setValue("");  
  
 SendUserToMainActivity();  
 Toast.*makeText*(RegisterActivity.this, "Account Created Successfully", Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
 }else {  
 String message = task.getException().toString();  
 Toast.*makeText*(RegisterActivity.this, "Error: "+message, Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
 }  
 }  
 });  
 }  
 }  
  
 private void SendUserToMainActivity() {  
 Intent mainIntent = new Intent(RegisterActivity.this, MainActivity.class);  
 mainIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(mainIntent);  
 finish();  
 }  
}

1. In browser delete current user and – create new user using app

## SettingsActivity

1. Create new activity – settingsActivity
2. Circle Image View:- <https://github.com/hdodenhof/CircleImageView>
3. Add dependency to gradle

implementation 'de.hdodenhof:circleimageview:3.1.0'

1. Settings\_activity.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".SettingsActivity">  
  
 <de.hdodenhof.circleimageview.CircleImageView  
  
 android:id="@+id/set\_profile\_image"  
 android:layout\_width="250dp"  
 android:layout\_height="250dp"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="45dp"  
 android:src="@drawable/profile\_image"  
 app:civ\_border\_width="5dp"  
 app:civ\_border\_color="@color/colorPrimary"/>  
 <EditText  
 android:id="@+id/set\_user\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/set\_profile\_image"  
 android:inputType="textMultiLine"  
 android:hint="@string/username"  
 android:textAlignment="center"  
 android:drawableStart="@drawable/edit\_profile"  
 android:textColor="@color/colorPrimaryDark"  
 android:textStyle="bold"  
 android:background="@drawable/inputs"  
 android:layout\_marginStart="15dp"  
 android:layout\_marginEnd="15dp"  
 android:padding="6dp"  
 android:layout\_marginTop="10dp"  
 />  
  
 <EditText  
 android:id="@+id/set\_profile\_status"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/set\_user\_name"  
 android:inputType="textMultiLine"  
 android:hint="@string/hey\_i\_am\_available\_now"  
 android:textAlignment="center"  
 android:drawableStart="@drawable/edit\_status"  
 android:textColor="@color/colorPrimaryDark"  
 android:textStyle="bold"  
 android:background="@drawable/inputs"  
 android:layout\_marginStart="15dp"  
 android:layout\_marginEnd="15dp"  
 android:padding="6dp"  
 android:layout\_marginTop="10dp"  
 />  
  
 <Button  
 android:id="@+id/update\_settings\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/set\_profile\_status"  
 android:text="Update"  
 android:textAlignment="center"  
 android:textColor="@android:color/background\_light"  
 android:textStyle="bold"  
 android:background="@drawable/buttons"  
 android:layout\_marginStart="15dp"  
 android:layout\_marginEnd="15dp"  
 android:padding="6dp"  
 android:layout\_marginTop="10dp"  
 android:textAllCaps="false"  
 android:textSize="20dp"  
 android:onClick="setUpdateAccountSettings"  
 />  
  
</RelativeLayout>

1. MainActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.viewpager.widget.ViewPager;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.widget.Toast;  
  
import com.google.android.material.tabs.TabLayout;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
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;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Toolbar mToolbar;  
 private ViewPager myViewPager;  
 private TabLayout myTabLayout;  
  
 private TabsAccessorAdapter myTabsAccessorAdapter;  
  
 private FirebaseUser currentUser;  
 private FirebaseAuth mAuth;  
 private DatabaseReference RootRef;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUser = mAuth.getCurrentUser();  
 RootRef = FirebaseDatabase.*getInstance*().getReference();  
  
 mToolbar = (Toolbar) findViewById(R.id.*main\_page\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("WhatsApp");  
  
 myViewPager = (ViewPager) findViewById(R.id.*main\_tabs\_pager*);  
 myTabsAccessorAdapter = new TabsAccessorAdapter(getSupportFragmentManager());  
 myViewPager.setAdapter(myTabsAccessorAdapter);  
  
 myTabLayout = (TabLayout) findViewById(R.id.*main\_tabs*);  
 myTabLayout.setupWithViewPager(myViewPager);  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 if(currentUser == null){  
 SendUserToLoginActivity();  
 }else {  
 VarifyUserExistance();  
 }  
 }  
  
 private void VarifyUserExistance() {  
 String currentUserID = mAuth.getCurrentUser().getUid();  
 RootRef.child("Users").child(currentUserID).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if((dataSnapshot.child("name").exists())){  
 Toast.*makeText*(MainActivity.this, "Welcome", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 SendUserToSettingsActivity();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 super.onCreateOptionsMenu(menu);  
 getMenuInflater().inflate(R.menu.*options\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 super.onOptionsItemSelected(item);  
  
 if(item.getItemId() == R.id.*main\_logout\_option*){  
 mAuth.signOut();  
 SendUserToLoginActivity();  
 }  
 if(item.getItemId() == R.id.*main\_settings\_option*){  
 SendUserToSettingsActivity();  
 }  
 if(item.getItemId() == R.id.*main\_find\_friends\_option*){  
  
 }  
  
 return true;  
 }  
  
 private void SendUserToLoginActivity() {  
  
 Intent loginIntent = new Intent(MainActivity.this, LoginActivity.class);  
 loginIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(loginIntent);  
 finish();  
 }  
  
 private void SendUserToSettingsActivity() {  
  
 Intent loginIntent = new Intent(MainActivity.this, SettingsActivity.class);  
 loginIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(loginIntent);  
 finish();  
 }  
}

1. SettingsActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
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;  
  
import java.util.HashMap;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class SettingsActivity extends AppCompatActivity {  
  
 private Button UpdateAccountSettings;  
 private EditText userName, userStatus;  
 private CircleImageView userProfileImage;  
 private String currentUserId;  
 private FirebaseAuth mAuth;  
 private DatabaseReference RootRef;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_settings*);  
  
 InitializeFields();  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUserId = mAuth.getCurrentUser().getUid();  
 RootRef = FirebaseDatabase.*getInstance*().getReference();  
  
 userName.setVisibility(View.*INVISIBLE*);  
 RetrieveUserInfo();  
 }  
  
 private void InitializeFields() {  
 UpdateAccountSettings = (Button) findViewById(R.id.*update\_settings\_button*);  
 userName = (EditText) findViewById(R.id.*set\_user\_name*);  
 userStatus = (EditText) findViewById(R.id.*set\_profile\_status*);  
 userProfileImage = (CircleImageView) findViewById(R.id.*set\_profile\_image*);  
 }  
  
 public void setUpdateAccountSettings(View view){  
 String setUserName = userName.getText().toString();  
 String setUserStatus = userStatus.getText().toString();  
 if(TextUtils.*isEmpty*(setUserName)){  
 Toast.*makeText*(this, "Please enter username", Toast.*LENGTH\_SHORT*).show();  
 }  
 if(TextUtils.*isEmpty*(setUserStatus)){  
 Toast.*makeText*(this, "Please enter status", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 HashMap<String, String> profileMap = new HashMap<>();  
 profileMap.put("uid", currentUserId);  
 profileMap.put("name", setUserName);  
 profileMap.put("status", setUserStatus);  
 RootRef.child("Users").child(currentUserId).setValue(profileMap)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(SettingsActivity.this, "Profile updated successfully", Toast.*LENGTH\_SHORT*).show();  
 SendUserToMainActivity();  
 }  
 else {  
 String message = task.getException().toString();  
 Toast.*makeText*(SettingsActivity.this, "Error: "+ message, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 }  
  
 private void SendUserToMainActivity() {  
 Intent mainIntent = new Intent(SettingsActivity.this, MainActivity.class);  
 mainIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(mainIntent);  
 finish();  
 }  
  
 private void RetrieveUserInfo() {  
 RootRef.child("Users").child(currentUserId)  
 .addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if((dataSnapshot.exists()) && (dataSnapshot.hasChild("name") && (dataSnapshot.hasChild("image")))){  
  
 String retrieveUserName = dataSnapshot.child("name").getValue().toString();  
 String retrieveStatus = dataSnapshot.child("status").getValue().toString();  
 String retrieveProfileImage = dataSnapshot.child("image").getValue().toString();  
  
 userName.setText(retrieveUserName);  
 userStatus.setText(retrieveStatus);  
 }  
 else if((dataSnapshot.exists()) && (dataSnapshot.hasChild("name"))){  
 String retrieveUserName = dataSnapshot.child("name").getValue().toString();  
 String retrieveStatus = dataSnapshot.child("status").getValue().toString();  
  
 userName.setText(retrieveUserName);  
 userStatus.setText(retrieveStatus);  
 }else{  
 userName.setVisibility(View.*VISIBLE*);  
 Toast.*makeText*(SettingsActivity.this, "Please set and update your profile information.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
}

## Create Group

1. Menu/Options\_menu.xml

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
 <item  
 android:title="@string/find\_friends"  
 android:id="@+id/main\_find\_friends\_option"  
 app:showAsAction="never"></item>  
  
 <item  
 android:title="Create Group"  
 android:id="@+id/main\_create\_group\_option"  
 app:showAsAction="never"></item>  
  
 <item  
 android:title="@string/settings"  
 android:id="@+id/main\_settings\_option"  
 app:showAsAction="never"></item>  
  
 <item  
 android:title="@string/logout"  
 android:id="@+id/main\_logout\_option"  
 app:showAsAction="never"></item>  
</menu>

1. Styles.xml

<resources>  
  
 <!-- Base application theme. -->  
 <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">  
 <!-- Customize your theme here. -->  
 <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="colorAccent">@color/colorAccent</item>  
 </style>  
  
 <style name="AlertDialog" parent="Theme.AppCompat.Light.Dialog"></style>  
  
</resources>

1. MainActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.viewpager.widget.ViewPager;  
  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.tabs.TabLayout;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
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;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Toolbar mToolbar;  
 private ViewPager myViewPager;  
 private TabLayout myTabLayout;  
  
 private TabsAccessorAdapter myTabsAccessorAdapter;  
  
 private FirebaseUser currentUser;  
 private FirebaseAuth mAuth;  
 private DatabaseReference RootRef;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUser = mAuth.getCurrentUser();  
 RootRef = FirebaseDatabase.*getInstance*().getReference();  
  
 mToolbar = (Toolbar) findViewById(R.id.*main\_page\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("WhatsApp");  
  
 myViewPager = (ViewPager) findViewById(R.id.*main\_tabs\_pager*);  
 myTabsAccessorAdapter = new TabsAccessorAdapter(getSupportFragmentManager());  
 myViewPager.setAdapter(myTabsAccessorAdapter);  
  
 myTabLayout = (TabLayout) findViewById(R.id.*main\_tabs*);  
 myTabLayout.setupWithViewPager(myViewPager);  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 if(currentUser == null){  
 SendUserToLoginActivity();  
 }else {  
 VarifyUserExistance();  
 }  
 }  
  
 private void VarifyUserExistance() {  
 String currentUserID = mAuth.getCurrentUser().getUid();  
 RootRef.child("Users").child(currentUserID).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if((dataSnapshot.child("name").exists())){  
 Toast.*makeText*(MainActivity.this, "Welcome", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 SendUserToSettingsActivity();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 super.onCreateOptionsMenu(menu);  
 getMenuInflater().inflate(R.menu.*options\_menu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 super.onOptionsItemSelected(item);  
  
 if(item.getItemId() == R.id.*main\_logout\_option*){  
 mAuth.signOut();  
 SendUserToLoginActivity();  
 }  
 if(item.getItemId() == R.id.*main\_create\_group\_option*){  
 RequestNewGroup();  
 }  
 if(item.getItemId() == R.id.*main\_settings\_option*){  
 SendUserToSettingsActivity();  
 }  
 if(item.getItemId() == R.id.*main\_find\_friends\_option*){  
  
 }  
  
 return true;  
 }  
  
 private void RequestNewGroup() {  
 AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this, R.style.*AlertDialog*);  
 builder.setTitle("Enter Group Name: ");  
  
 final EditText groupNameField = new EditText(MainActivity.this);  
 groupNameField.setHint("e.g Coding Cafe");  
 builder.setView(groupNameField);  
  
 builder.setPositiveButton("Create", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 String groupName = groupNameField.getText().toString();  
 if(TextUtils.*isEmpty*(groupName)){  
 Toast.*makeText*(MainActivity.this, "Please write group name", Toast.*LENGTH\_SHORT*).show();  
 }  
 else {  
 CreateNewGroup(groupName);  
 }  
 }  
 });  
  
 builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 dialog.cancel();  
 }  
 });  
  
 builder.show();  
 }  
  
 private void CreateNewGroup(final String groupName) {  
 RootRef.child("Groups").child(groupName).setValue("")  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(MainActivity.this, groupName + " group is created successfully", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
  
 private void SendUserToLoginActivity() {  
  
 Intent loginIntent = new Intent(MainActivity.this, LoginActivity.class);  
 loginIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(loginIntent);  
 finish();  
 }  
  
 private void SendUserToSettingsActivity() {  
  
 Intent loginIntent = new Intent(MainActivity.this, SettingsActivity.class);  
 loginIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(loginIntent);  
 finish();  
 }  
}

## Retrieve groups list to listview

1. Fragment\_groups.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".GroupsFragment">  
  
 <!-- *TODO: Update blank fragment layout* -->  
 <ListView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/listView">  
  
 </ListView>  
  
</RelativeLayout>

1. GroupsFragment.java

package com.example.whatsapp;  
  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
  
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;  
  
import java.util.ArrayList;  
import java.util.HashSet;  
import java.util.Iterator;  
import java.util.Set;  
  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \*/*public class GroupsFragment extends Fragment {  
  
 private View groupFragmentView;  
 private ListView list\_view;  
 private ArrayAdapter<String> arrayAdapter;  
 private ArrayList<String> list\_of\_groups = new ArrayList<>();  
 private DatabaseReference GroupRef;  
  
 public GroupsFragment() {  
 // Required empty public constructor  
 }  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 groupFragmentView = inflater.inflate(R.layout.*fragment\_groups*, container, false);  
 GroupRef = FirebaseDatabase.*getInstance*().getReference().child("Groups");  
 InitializeFields();  
   
 RetrieveAndDisplayGroups();  
 return groupFragmentView;  
 }  
  
  
 private void InitializeFields() {  
 list\_view = (ListView) groupFragmentView.findViewById(R.id.*listView*);  
 arrayAdapter = new ArrayAdapter<String>(getContext(), android.R.layout.*simple\_list\_item\_1*, list\_of\_groups);  
 list\_view.setAdapter(arrayAdapter);  
 }  
  
 private void RetrieveAndDisplayGroups() {  
 GroupRef.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 Set<String> set = new HashSet<>();  
 Iterator iterator = dataSnapshot.getChildren().iterator();  
 while (iterator.hasNext()){  
 set.add(((DataSnapshot)iterator.next()).getKey());  
 }  
 list\_of\_groups.clear();  
 list\_of\_groups.addAll(set);  
 arrayAdapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
}

## GroupChatActivity

1. Create new emoty activity – GroupChatActivity
2. Activity\_group\_chat.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".GroupChatActivity">  
  
 <include  
 android:id="@+id/group\_chat\_bar\_layout"  
 layout="@layout/app\_bar\_layout"  
 >  
  
 </include>  
  
 <ScrollView  
 android:id="@+id/my\_scroll\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/group\_chat\_bar\_layout"  
 android:layout\_above="@+id/myLinearLayout">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
 <TextView  
 android:id="@+id/group\_chat\_text\_display"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:textAllCaps="false"  
 android:textSize="20dp"  
 android:textColor="@android:color/background\_dark"  
 android:layout\_marginStart="2dp"  
 android:layout\_marginEnd="2dp"  
 android:layout\_marginBottom="50dp"  
 />  
 </LinearLayout>  
 </ScrollView>  
 <LinearLayout  
 android:id="@+id/myLinearLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentStart="true"  
 >  
 <EditText  
 android:id="@+id/input\_group\_message"  
 android:layout\_width="220dp"  
 android:layout\_height="wrap\_content"  
 android:hint="Write a message here..."  
 android:padding="17dp"  
 />  
 <ImageButton  
 android:id="@+id/send\_message\_button"  
 android:layout\_width="50dp"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/send\_message"  
 android:onClick="setSendImageButton"  
 />  
 </LinearLayout>  
</RelativeLayout>

1. GroupChatActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.ScrollView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.ChildEventListener;  
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;  
  
import java.text.SimpleDateFormat;  
import java.util.Calendar;  
import java.util.HashMap;  
import java.util.Iterator;  
  
public class GroupChatActivity extends AppCompatActivity {  
  
 private Toolbar mToolbar;  
 private ImageButton SendImageButton;  
 private EditText userMessageInput;  
 private ScrollView mScrollView;  
 private TextView displayTextMessage;  
  
 private FirebaseAuth mAuth;  
 private DatabaseReference UsersRef, GroupNameRef, GroupMessageKeyRef;  
  
 private String currentGroupName, currentUserId, currentUserName, currentDate, currentTime;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_group\_chat*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUserId = mAuth.getCurrentUser().getUid();  
 UsersRef = FirebaseDatabase.*getInstance*().getReference().child("Users");  
  
 currentGroupName = getIntent().getExtras().get("groupName").toString();  
 GroupNameRef = FirebaseDatabase.*getInstance*().getReference().child("Groups").child(currentGroupName);  
  
 InitializeFields();  
   
 GetUserInfo();  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
 GroupNameRef.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
 if(dataSnapshot.exists()){  
 DisplayMessages(dataSnapshot);  
 }  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
 if(dataSnapshot.exists()){  
 DisplayMessages(dataSnapshot);  
 }  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 private void DisplayMessages(DataSnapshot dataSnapshot) {  
 Iterator iterator = dataSnapshot.getChildren().iterator();  
 while (iterator.hasNext()){  
 String chatDate = (String) ((DataSnapshot)iterator.next()).getValue();  
 String chatMessage = (String) ((DataSnapshot)iterator.next()).getValue();  
 String chatName = (String) ((DataSnapshot)iterator.next()).getValue();  
 String chatTime = (String) ((DataSnapshot)iterator.next()).getValue();  
  
 displayTextMessage.append(chatName + " :\n" + chatMessage + "\n" + chatTime + " " + chatDate + "\n\n\n");  
 }  
 }  
  
 private void InitializeFields() {  
 mToolbar = (Toolbar)findViewById(R.id.*group\_chat\_bar\_layout*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle(currentGroupName);  
  
 SendImageButton = (ImageButton) findViewById(R.id.*send\_message\_button*);  
 userMessageInput = (EditText) findViewById(R.id.*input\_group\_message*);  
 displayTextMessage = (TextView) findViewById(R.id.*group\_chat\_text\_display*);  
 mScrollView = (ScrollView) findViewById(R.id.*my\_scroll\_view*);  
 }  
  
 private void GetUserInfo() {  
 UsersRef.child(currentUserId).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 currentUserName = dataSnapshot.child("name").getValue().toString();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void setSendImageButton(View view){  
 // save message info to db  
 String message = userMessageInput.getText().toString();  
 String messageKey = GroupNameRef.push().getKey();  
 if(TextUtils.*isEmpty*(message)){  
 Toast.*makeText*(this, "Please write message", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 Calendar calForDate = Calendar.*getInstance*();  
 SimpleDateFormat currentDateFormat = new SimpleDateFormat("MMM dd, yyyy");  
 currentDate = currentDateFormat.format(calForDate.getTime());  
  
 Calendar calForTime = Calendar.*getInstance*();  
 SimpleDateFormat currentTimeFormat = new SimpleDateFormat("hh:mm a");  
 currentTime = currentTimeFormat.format(calForTime.getTime());  
  
 HashMap<String, Object> groupMessageKey = new HashMap<>();  
  
 GroupNameRef.updateChildren(groupMessageKey);  
 GroupMessageKeyRef = GroupNameRef.child(messageKey);  
  
 HashMap<String, Object> messageInfoMap = new HashMap<>();  
 messageInfoMap.put("name", currentUserName);  
 messageInfoMap.put("message", message);  
 messageInfoMap.put("date", currentDate);  
 messageInfoMap.put("time", currentTime);  
  
 GroupMessageKeyRef.updateChildren(messageInfoMap);  
 }  
  
 userMessageInput.setText("");  
 }  
}

1. GroupsFragment.java

package com.example.whatsapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
  
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;  
  
import java.util.ArrayList;  
import java.util.HashSet;  
import java.util.Iterator;  
import java.util.Set;  
  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \*/*public class GroupsFragment extends Fragment {  
  
 private View groupFragmentView;  
 private ListView list\_view;  
 private ArrayAdapter<String> arrayAdapter;  
 private ArrayList<String> list\_of\_groups = new ArrayList<>();  
 private DatabaseReference GroupRef;  
  
 public GroupsFragment() {  
 // Required empty public constructor  
 }  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 groupFragmentView = inflater.inflate(R.layout.*fragment\_groups*, container, false);  
 GroupRef = FirebaseDatabase.*getInstance*().getReference().child("Groups");  
 InitializeFields();  
   
 RetrieveAndDisplayGroups();  
  
 list\_view.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 String currentGroupName = parent.getItemAtPosition(position).toString();  
  
 Intent intent = new Intent(getContext(), GroupChatActivity.class);  
 intent.putExtra("groupName", currentGroupName);  
 startActivity(intent);  
 }  
 });  
  
 return groupFragmentView;  
 }  
  
  
 private void InitializeFields() {  
 list\_view = (ListView) groupFragmentView.findViewById(R.id.*listView*);  
 arrayAdapter = new ArrayAdapter<String>(getContext(), android.R.layout.*simple\_list\_item\_1*, list\_of\_groups);  
 list\_view.setAdapter(arrayAdapter);  
 }  
  
 private void RetrieveAndDisplayGroups() {  
 GroupRef.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 Set<String> set = new HashSet<>();  
 Iterator iterator = dataSnapshot.getChildren().iterator();  
 while (iterator.hasNext()){  
 set.add(((DataSnapshot)iterator.next()).getKey());  
 }  
 list\_of\_groups.clear();  
 list\_of\_groups.addAll(set);  
 arrayAdapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
}

## Scrolling messages

1. GroupChatActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.ScrollView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.ChildEventListener;  
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;  
  
import java.text.SimpleDateFormat;  
import java.util.Calendar;  
import java.util.HashMap;  
import java.util.Iterator;  
  
public class GroupChatActivity extends AppCompatActivity {  
  
 private Toolbar mToolbar;  
 private ImageButton SendImageButton;  
 private EditText userMessageInput;  
 private ScrollView mScrollView;  
 private TextView displayTextMessage;  
  
 private FirebaseAuth mAuth;  
 private DatabaseReference UsersRef, GroupNameRef, GroupMessageKeyRef;  
  
 private String currentGroupName, currentUserId, currentUserName, currentDate, currentTime;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_group\_chat*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUserId = mAuth.getCurrentUser().getUid();  
 UsersRef = FirebaseDatabase.*getInstance*().getReference().child("Users");  
  
 currentGroupName = getIntent().getExtras().get("groupName").toString();  
 GroupNameRef = FirebaseDatabase.*getInstance*().getReference().child("Groups").child(currentGroupName);  
  
 InitializeFields();  
   
 GetUserInfo();  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
 GroupNameRef.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
 if(dataSnapshot.exists()){  
 DisplayMessages(dataSnapshot);  
 }  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
 if(dataSnapshot.exists()){  
 DisplayMessages(dataSnapshot);  
 }  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 private void DisplayMessages(DataSnapshot dataSnapshot) {  
 Iterator iterator = dataSnapshot.getChildren().iterator();  
 while (iterator.hasNext()){  
 String chatDate = (String) ((DataSnapshot)iterator.next()).getValue();  
 String chatMessage = (String) ((DataSnapshot)iterator.next()).getValue();  
 String chatName = (String) ((DataSnapshot)iterator.next()).getValue();  
 String chatTime = (String) ((DataSnapshot)iterator.next()).getValue();  
  
 displayTextMessage.append(chatName + " :\n" + chatMessage + "\n" + chatTime + " " + chatDate + "\n\n\n");  
  
 mScrollView.fullScroll(ScrollView.*FOCUS\_DOWN*);  
 }  
 }  
  
 private void InitializeFields() {  
 mToolbar = (Toolbar)findViewById(R.id.*group\_chat\_bar\_layout*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle(currentGroupName);  
  
 SendImageButton = (ImageButton) findViewById(R.id.*send\_message\_button*);  
 userMessageInput = (EditText) findViewById(R.id.*input\_group\_message*);  
 displayTextMessage = (TextView) findViewById(R.id.*group\_chat\_text\_display*);  
 mScrollView = (ScrollView) findViewById(R.id.*my\_scroll\_view*);  
 }  
  
 private void GetUserInfo() {  
 UsersRef.child(currentUserId).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 currentUserName = dataSnapshot.child("name").getValue().toString();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void setSendImageButton(View view){  
 // save message info to db  
 String message = userMessageInput.getText().toString();  
 String messageKey = GroupNameRef.push().getKey();  
 if(TextUtils.*isEmpty*(message)){  
 Toast.*makeText*(this, "Please write message", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 Calendar calForDate = Calendar.*getInstance*();  
 SimpleDateFormat currentDateFormat = new SimpleDateFormat("MMM dd, yyyy");  
 currentDate = currentDateFormat.format(calForDate.getTime());  
  
 Calendar calForTime = Calendar.*getInstance*();  
 SimpleDateFormat currentTimeFormat = new SimpleDateFormat("hh:mm a");  
 currentTime = currentTimeFormat.format(calForTime.getTime());  
  
 HashMap<String, Object> groupMessageKey = new HashMap<>();  
  
 GroupNameRef.updateChildren(groupMessageKey);  
 GroupMessageKeyRef = GroupNameRef.child(messageKey);  
  
 HashMap<String, Object> messageInfoMap = new HashMap<>();  
 messageInfoMap.put("name", currentUserName);  
 messageInfoMap.put("message", message);  
 messageInfoMap.put("date", currentDate);  
 messageInfoMap.put("time", currentTime);  
  
 GroupMessageKeyRef.updateChildren(messageInfoMap);  
 }  
  
 userMessageInput.setText("");  
 mScrollView.fullScroll(ScrollView.*FOCUS\_DOWN*);  
 }  
}

## Phone Authentication

1. Create new emoty activity – PhoneLoginActivity
2. In browser – firebase console -> authentication -> sign in methords -> enable phone number
3. Phone number verification documentation - <https://firebase.google.com/docs/auth/android/phone-auth>
4. Activity\_phone\_login.xml
5. <?xml version="1.0" encoding="utf-8"?>  
   <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"  
    tools:context=".PhoneLoginActivity">  
     
    <EditText  
    android:id="@+id/phone\_number\_input"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:layout\_alignParentStart="true"  
    android:layout\_marginStart="17dp"  
    android:layout\_marginEnd="17dp"  
    android:layout\_marginTop="180dp"  
    android:background="@drawable/inputs"  
    android:padding="8dp"  
    android:hint="@string/write\_phone\_number"  
    android:textAllCaps="false"  
    android:textSize="15dp"  
    android:drawableStart="@drawable/phone"  
    android:textAlignment="center"  
    android:textStyle="bold"  
    />  
    <EditText  
    android:id="@+id/verification\_code\_input"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:layout\_alignParentStart="true"  
    android:layout\_marginStart="17dp"  
    android:layout\_marginEnd="17dp"  
    android:layout\_marginTop="4dp"  
    android:background="@drawable/inputs"  
    android:padding="5dp"  
    android:layout\_below="@+id/phone\_number\_input"  
    android:hint="@string/write\_verification\_code"  
    android:textAllCaps="false"  
    android:textSize="15dp"  
    android:drawableStart="@drawable/password"  
    android:textAlignment="center"  
    android:textStyle="bold"  
    android:visibility="invisible"  
    />  
     
    <Button  
    android:id="@+id/send\_ver\_code\_button"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@+id/verification\_code\_input"  
    android:text="@string/send\_verification\_code"  
    android:textAllCaps="false"  
    android:layout\_marginStart="17dp"  
    android:layout\_marginEnd="17dp"  
    android:layout\_marginTop="1dp"  
    android:background="@drawable/buttons"  
    android:textColor="@android:color/background\_light"  
    android:textSize="19dp"  
    android:textAlignment="center"  
    android:textStyle="bold"  
    android:onClick="setSendVerificationButton"  
    />  
    <Button  
    android:id="@+id/verify\_button"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@+id/send\_ver\_code\_button"  
    android:text="Verify"  
    android:textAllCaps="false"  
    android:layout\_marginStart="17dp"  
    android:layout\_marginEnd="17dp"  
    android:layout\_marginTop="1dp"  
    android:background="@drawable/buttons"  
    android:textColor="@android:color/background\_light"  
    android:textSize="19dp"  
    android:textAlignment="center"  
    android:textStyle="bold"  
    android:visibility="invisible"  
    android:onClick="setVerifyButton"  
    />  
     
   </RelativeLayout>
6. Activity\_login.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".LoginActivity">  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <ImageView  
 android:id="@+id/login\_image"  
 android:layout\_width="match\_parent"  
 android:layout\_height="250dp"  
 android:src="@drawable/login\_photo"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:scaleType="fitCenter"  
 />  
 <EditText  
 android:id="@+id/login\_email"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/login\_image"  
 android:hint="Email"  
 android:inputType="textEmailAddress"  
 android:background="@drawable/inputs"  
 android:padding="4dp"  
 android:drawableStart="@drawable/email"  
 android:layout\_marginTop="25dp"  
 android:layout\_marginLeft="17dp"  
 android:layout\_marginRight="17dp"  
 />  
  
 <EditText  
 android:id="@+id/login\_password"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/login\_email"  
 android:hint="Password"  
 android:inputType="textPassword"  
 android:background="@drawable/inputs"  
 android:padding="4dp"  
 android:drawableStart="@drawable/password"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="17dp"  
 android:layout\_marginRight="17dp"  
 />  
  
 <TextView  
 android:id="@+id/forget\_password\_link"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/login\_password"  
 android:text="@string/forget\_password"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="23dp"  
 android:textAlignment="textEnd"  
 android:textSize="17dp"  
 android:textStyle="bold"  
 android:textColor="@color/colorPrimaryDark"  
 />  
  
 <Button  
 android:id="@+id/login\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/forget\_password\_link"  
 android:padding="4dp"  
 android:layout\_marginTop="4dp"  
 android:layout\_marginLeft="19dp"  
 android:layout\_marginRight="19dp"  
 android:text="@string/login"  
 android:textAllCaps="false"  
 android:textSize="24dp"  
 android:textColor="@android:color/background\_light"  
 android:background="@drawable/buttons"  
 android:onClick="setLoginButton"  
 />  
 <TextView  
 android:id="@+id/need\_new\_account\_link"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/login\_button"  
 android:text="Need new Account?"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="23dp"  
 android:textAlignment="center"  
 android:textSize="17dp"  
 android:textStyle="bold"  
 android:textColor="@color/colorPrimaryDark"  
 android:onClick="setNeedNewAccountLink"  
 />  
  
 <TextView  
 android:id="@+id/login\_using"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@+id/need\_new\_account\_link"  
 android:text="Or Login using your"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="23dp"  
 android:textAlignment="center"  
 android:textSize="17dp"  
 />  
 <Button  
 android:id="@+id/phone\_login\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/login\_using"  
 android:padding="4dp"  
 android:layout\_marginTop="4dp"  
 android:layout\_marginLeft="19dp"  
 android:layout\_marginRight="19dp"  
 android:text="@string/phone"  
 android:textAllCaps="false"  
 android:textSize="14sp"  
 android:textColor="@android:color/background\_light"  
 android:background="@drawable/buttons"  
 android:drawableStart="@drawable/phone"  
 android:onClick="setPhoneLoginButton"  
 />  
 </RelativeLayout>  
 </ScrollView>  
</RelativeLayout>

1. LoginActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
public class LoginActivity extends AppCompatActivity {  
  
 private FirebaseAuth mAuth;  
 private Button LoginButton, PhoneLoginButton;  
 private EditText UserEmail, UserPassword;  
 private TextView NeedNewAccountLink, ForgetPasswordLink;  
 private ProgressDialog loadingBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 InitializeFields();  
 }  
  
 private void InitializeFields() {  
 LoginButton = (Button) findViewById(R.id.*login\_button*);  
 PhoneLoginButton = (Button) findViewById(R.id.*phone\_login\_button*);  
 UserEmail = (EditText) findViewById(R.id.*login\_email*);  
 UserPassword = (EditText) findViewById(R.id.*login\_password*);  
 NeedNewAccountLink = (TextView) findViewById(R.id.*need\_new\_account\_link*);  
 ForgetPasswordLink = (TextView) findViewById(R.id.*forget\_password\_link*);  
 loadingBar = new ProgressDialog(this);  
 }  
  
 public void setNeedNewAccountLink(View view){  
 SendUserToRegisterActivity();  
 }  
 public void setLoginButton(View view){  
 AllowUserToLogin();  
 }  
  
 private void AllowUserToLogin() {  
 String email = UserEmail.getText().toString();  
 String password = UserPassword.getText().toString();  
  
 if(TextUtils.*isEmpty*(email)){  
 Toast.*makeText*(this, "Please enter email", Toast.*LENGTH\_SHORT*).show();  
 }  
 if(TextUtils.*isEmpty*(password)){  
 Toast.*makeText*(this, "Please enter password", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 loadingBar.setTitle("Sign In");  
 loadingBar.setMessage("Please wait while we are authenticating your account");  
 loadingBar.setCanceledOnTouchOutside(true);  
 loadingBar.show();  
 mAuth.signInWithEmailAndPassword(email, password)  
 .addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
 SendUserToMainActivity();  
 Toast.*makeText*(LoginActivity.this, "Logged in successful", Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
 }else {  
 String message = task.getException().toString();  
 Toast.*makeText*(LoginActivity.this, "Error: "+message, Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
 }  
 }  
 });  
 }  
 }  
  
 private void SendUserToMainActivity() {  
 Intent mainIntent = new Intent(LoginActivity.this, MainActivity.class);  
 mainIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(mainIntent);  
 finish();  
 }  
 private void SendUserToRegisterActivity() {  
 Intent mainIntent = new Intent(LoginActivity.this, RegisterActivity.class);  
 startActivity(mainIntent);  
 finish();  
 }  
  
 private void SendUserToPhoneLoginActivity() {  
 Intent mainIntent = new Intent(LoginActivity.this, PhoneLoginActivity.class);  
 startActivity(mainIntent);  
 finish();  
 }  
  
  
 public void setPhoneLoginButton(View view){  
 SendUserToPhoneLoginActivity();  
 }  
}

1. PhoneActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.FirebaseException;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.PhoneAuthCredential;  
import com.google.firebase.auth.PhoneAuthProvider;  
  
import java.util.concurrent.TimeUnit;  
  
public class PhoneLoginActivity extends AppCompatActivity {  
  
 private Button SendVerificationButton, VerifyButton;  
 private EditText InputPhoneNumber, InputVerificationCode;  
 private PhoneAuthProvider.OnVerificationStateChangedCallbacks callbacks;  
 private String mVerificationId;  
 private PhoneAuthProvider.ForceResendingToken mResendToken;  
 private FirebaseAuth mAuth;  
 private ProgressDialog loadingBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_phone\_login*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 InputPhoneNumber = (EditText) findViewById(R.id.*phone\_number\_input*);  
 InputVerificationCode = (EditText) findViewById(R.id.*verification\_code\_input*);  
 SendVerificationButton = (Button) findViewById(R.id.*send\_ver\_code\_button*);  
 VerifyButton = (Button) findViewById(R.id.*verify\_button*);  
 loadingBar = new ProgressDialog(this);  
  
 callbacks = new PhoneAuthProvider.OnVerificationStateChangedCallbacks() {  
 @Override  
 public void onVerificationCompleted(@NonNull PhoneAuthCredential phoneAuthCredential) {  
 signInWithPhoneAuthCredential(phoneAuthCredential);  
 }  
  
 @Override  
 public void onVerificationFailed(@NonNull FirebaseException e) {  
  
 loadingBar.dismiss();  
 Toast.*makeText*(PhoneLoginActivity.this, "Invalid phone number please enter your valid phone number with your country code.", Toast.*LENGTH\_SHORT*).show();  
  
 SendVerificationButton.setVisibility(View.*VISIBLE*);  
 InputPhoneNumber.setVisibility(View.*VISIBLE*);  
  
 VerifyButton.setVisibility(View.*INVISIBLE*);  
 InputVerificationCode.setVisibility(View.*INVISIBLE*);  
 }  
  
 public void onCodeSent(@NonNull String verificationId,  
 @NonNull PhoneAuthProvider.ForceResendingToken token) {  
  
 mVerificationId = verificationId;  
 mResendToken = token;  
  
 loadingBar.dismiss();  
 Toast.*makeText*(PhoneLoginActivity.this, "Code has been send please check", Toast.*LENGTH\_SHORT*).show();  
  
 SendVerificationButton.setVisibility(View.*INVISIBLE*);  
 InputPhoneNumber.setVisibility(View.*INVISIBLE*);  
  
 VerifyButton.setVisibility(View.*VISIBLE*);  
 InputVerificationCode.setVisibility(View.*VISIBLE*);  
 }  
 };  
 }  
  
 private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {  
 mAuth.signInWithCredential(credential)  
 .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 loadingBar.dismiss();  
 Toast.*makeText*(PhoneLoginActivity.this, "Congratulations we logged in successfully", Toast.*LENGTH\_SHORT*).show();  
 SendUserToMainActivity();  
 } else {  
 String message = task.getException().toString();  
 Toast.*makeText*(PhoneLoginActivity.this, "Error: " + message, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
  
 public void setSendVerificationButton(View view){  
  
  
  
 String phoneNumber = InputPhoneNumber.getText().toString();  
 if(TextUtils.*isEmpty*(phoneNumber)){  
 Toast.*makeText*(this, "Please enter your phone number", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 loadingBar.setTitle("Phone Verification");  
 loadingBar.setMessage("Please wait, while we authenticating your phone");  
 loadingBar.setCanceledOnTouchOutside(false);  
 loadingBar.show();  
  
 PhoneAuthProvider.*getInstance*().verifyPhoneNumber(  
 phoneNumber, // Phone number to verify  
 60, // Timeout duration  
 TimeUnit.*SECONDS*, // Unit of timeout  
 PhoneLoginActivity.this, // Activity (for callback binding)  
 callbacks); // OnVerificationStateChangedCallbacks  
 }  
 }  
  
 public void setVerifyButton(View view){  
 SendVerificationButton.setVisibility(View.*INVISIBLE*);  
 InputPhoneNumber.setVisibility(View.*INVISIBLE*);  
  
 String verificationCode = InputVerificationCode.getText().toString();  
 if(TextUtils.*isEmpty*(verificationCode)){  
 Toast.*makeText*(this, "Please write verification code first", Toast.*LENGTH\_SHORT*).show();  
 }else {  
  
 loadingBar.setTitle("Verification Code");  
 loadingBar.setMessage("Please wait, while we verifying your Verification Code");  
 loadingBar.setCanceledOnTouchOutside(false);  
 loadingBar.show();  
  
 PhoneAuthCredential credential = PhoneAuthProvider.*getCredential*(mVerificationId, verificationCode);  
 signInWithPhoneAuthCredential(credential);  
 }  
 }  
  
 private void SendUserToMainActivity() {  
 Intent mainIntent = new Intent(PhoneLoginActivity.this, MainActivity.class);  
 mainIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(mainIntent);  
 finish();  
 }  
}

### This app is not authorized to use Firebase Authentication. Please verifythat the correct package name and SHA-1 are configured in the Firebase Console. [ App validation failed ]

Tutorial for generating sha1 key:- <https://www.youtube.com/watch?v=A59Oy_2yFIQ>

1. Open file explorer and go to this location:- C:\Program Files\Java\jdk-14\bin
2. Shift +rc -> open cmd here
3. Paste this code

keytool -list -v -keystore C:\Users\YOURUSERPROFILENAME\.android\debug.keystore -alias androiddebugkey -storepass android -keypass android

1. This will show the sha1 key and copy this key
2. Go to settings page:- <https://console.firebase.google.com/project/_/settings/general>
3. In browser -> settings-> your apps-> add sha certificate finger print

## IMAGE pICK AND CROPPING

<https://github.com/ArthurHub/Android-Image-Cropper>

1. Gradle add dependency

api 'com.theartofdev.edmodo:android-image-cropper:2.8.+'

1. AndroidManifest.xml

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/>  
<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

1. androidManifest.xml – include this in <application> tag

<activity android:name="com.theartofdev.edmodo.cropper.CropImageActivity"  
 android:theme="@style/Base.Theme.AppCompat"/>

1. activity\_settings.xml

<de.hdodenhof.circleimageview.CircleImageView  
  
 android:id="@+id/set\_profile\_image"  
 android:layout\_width="250dp"  
 android:layout\_height="250dp"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="45dp"  
 android:src="@drawable/profile\_image"  
 app:civ\_border\_width="5dp"  
 app:civ\_border\_color="@color/colorPrimary"  
 android:onClick="setUserProfileImage"  
 />

1. SettingsActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.icu.text.Collator;  
import android.net.Uri;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
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;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
import com.google.firebase.storage.UploadTask;  
import com.theartofdev.edmodo.cropper.CropImage;  
import com.theartofdev.edmodo.cropper.CropImageView;  
  
import java.util.HashMap;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class SettingsActivity extends AppCompatActivity {  
  
 private Button UpdateAccountSettings;  
 private EditText userName, userStatus;  
 private CircleImageView userProfileImage;  
 private String currentUserId;  
 private FirebaseAuth mAuth;  
 private DatabaseReference RootRef;  
 private static final int *GalleryPick* = 1;  
  
 private StorageReference UserProfileImageRef;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_settings*);  
  
 InitializeFields();  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUserId = mAuth.getCurrentUser().getUid();  
 RootRef = FirebaseDatabase.*getInstance*().getReference();  
 UserProfileImageRef = FirebaseStorage.*getInstance*().getReference().child("Profile\_Images");  
  
 userName.setVisibility(View.*INVISIBLE*);  
 RetrieveUserInfo();  
 }  
  
 private void InitializeFields() {  
 UpdateAccountSettings = (Button) findViewById(R.id.*update\_settings\_button*);  
 userName = (EditText) findViewById(R.id.*set\_user\_name*);  
 userStatus = (EditText) findViewById(R.id.*set\_profile\_status*);  
 userProfileImage = (CircleImageView) findViewById(R.id.*set\_profile\_image*);  
 }  
  
 public void setUpdateAccountSettings(View view){  
 String setUserName = userName.getText().toString();  
 String setUserStatus = userStatus.getText().toString();  
 if(TextUtils.*isEmpty*(setUserName)){  
 Toast.*makeText*(this, "Please enter username", Toast.*LENGTH\_SHORT*).show();  
 }  
 if(TextUtils.*isEmpty*(setUserStatus)){  
 Toast.*makeText*(this, "Please enter status", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 HashMap<String, String> profileMap = new HashMap<>();  
 profileMap.put("uid", currentUserId);  
 profileMap.put("name", setUserName);  
 profileMap.put("status", setUserStatus);  
 RootRef.child("Users").child(currentUserId).setValue(profileMap)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(SettingsActivity.this, "Profile updated successfully", Toast.*LENGTH\_SHORT*).show();  
 SendUserToMainActivity();  
 }  
 else {  
 String message = task.getException().toString();  
 Toast.*makeText*(SettingsActivity.this, "Error: "+ message, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 }  
  
 private void SendUserToMainActivity() {  
 Intent mainIntent = new Intent(SettingsActivity.this, MainActivity.class);  
 mainIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(mainIntent);  
 finish();  
 }  
  
 private void RetrieveUserInfo() {  
 RootRef.child("Users").child(currentUserId)  
 .addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if((dataSnapshot.exists()) && (dataSnapshot.hasChild("name") && (dataSnapshot.hasChild("image")))){  
  
 String retrieveUserName = dataSnapshot.child("name").getValue().toString();  
 String retrieveStatus = dataSnapshot.child("status").getValue().toString();  
 String retrieveProfileImage = dataSnapshot.child("image").getValue().toString();  
  
 userName.setText(retrieveUserName);  
 userStatus.setText(retrieveStatus);  
 }  
 else if((dataSnapshot.exists()) && (dataSnapshot.hasChild("name"))){  
 String retrieveUserName = dataSnapshot.child("name").getValue().toString();  
 String retrieveStatus = dataSnapshot.child("status").getValue().toString();  
  
 userName.setText(retrieveUserName);  
 userStatus.setText(retrieveStatus);  
 }else{  
 userName.setVisibility(View.*VISIBLE*);  
 Toast.*makeText*(SettingsActivity.this, "Please set and update your profile information.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void setUserProfileImage(View view){  
// Intent galleryIntent = new Intent();  
// galleryIntent.setAction(Intent.ACTION\_GET\_CONTENT);  
// galleryIntent.setType("image/\*");  
// startActivityForResult(galleryIntent, GalleryPick);  
  
 CropImage.*activity*()  
 .setGuidelines(CropImageView.Guidelines.*ON*)  
 .setAspectRatio(1,1)  
 .start(this);  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
// if(requestCode==GalleryPick && resultCode == RESULT\_OK && data != null){  
// Uri ImageUri = data.getData();  
//  
// }  
  
 if (requestCode == CropImage.*CROP\_IMAGE\_ACTIVITY\_REQUEST\_CODE*) {  
 CropImage.ActivityResult result = CropImage.*getActivityResult*(data);  
  
 if(resultCode == *RESULT\_OK*){  
 Uri resultUri = result.getUri();  
 StorageReference filepath = UserProfileImageRef.child(currentUserId+".jpg");  
 filepath.putFile(resultUri).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(SettingsActivity.this, "Profile image uploaded successfully", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 String message = task.getException().toString();  
 Toast.*makeText*(SettingsActivity.this, "Error: " + message, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 }  
 }  
}

## Retrieve Image

1. Download image – picaso

<https://square.github.io/picasso/>

implementation 'com.squareup.picasso:picasso:2.71828'

1. SettingsActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.icu.text.Collator;  
import android.net.Uri;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
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;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
import com.google.firebase.storage.UploadTask;  
import com.squareup.picasso.Picasso;  
import com.theartofdev.edmodo.cropper.CropImage;  
import com.theartofdev.edmodo.cropper.CropImageView;  
  
import java.security.ProtectionDomain;  
import java.util.HashMap;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class SettingsActivity extends AppCompatActivity {  
  
 private Button UpdateAccountSettings;  
 private EditText userName, userStatus;  
 private CircleImageView userProfileImage;  
 private String currentUserId;  
 private FirebaseAuth mAuth;  
 private DatabaseReference RootRef;  
 private static final int *GalleryPick* = 1;  
  
 private StorageReference UserProfileImageRef;  
  
 private ProgressDialog loadingBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_settings*);  
  
 InitializeFields();  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUserId = mAuth.getCurrentUser().getUid();  
 RootRef = FirebaseDatabase.*getInstance*().getReference();  
 UserProfileImageRef = FirebaseStorage.*getInstance*().getReference().child("Profile\_Images");  
  
 userName.setVisibility(View.*INVISIBLE*);  
 RetrieveUserInfo();  
 }  
  
 private void InitializeFields() {  
 UpdateAccountSettings = (Button) findViewById(R.id.*update\_settings\_button*);  
 userName = (EditText) findViewById(R.id.*set\_user\_name*);  
 userStatus = (EditText) findViewById(R.id.*set\_profile\_status*);  
 userProfileImage = (CircleImageView) findViewById(R.id.*set\_profile\_image*);  
 loadingBar = new ProgressDialog(this);  
 }  
  
 public void setUpdateAccountSettings(View view){  
 String setUserName = userName.getText().toString();  
 String setUserStatus = userStatus.getText().toString();  
 if(TextUtils.*isEmpty*(setUserName)){  
 Toast.*makeText*(this, "Please enter username", Toast.*LENGTH\_SHORT*).show();  
 }  
 if(TextUtils.*isEmpty*(setUserStatus)){  
 Toast.*makeText*(this, "Please enter status", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 HashMap<String, String> profileMap = new HashMap<>();  
 profileMap.put("uid", currentUserId);  
 profileMap.put("name", setUserName);  
 profileMap.put("status", setUserStatus);  
 RootRef.child("Users").child(currentUserId).setValue(profileMap)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(SettingsActivity.this, "Profile updated successfully", Toast.*LENGTH\_SHORT*).show();  
 SendUserToMainActivity();  
 }  
 else {  
 String message = task.getException().toString();  
 Toast.*makeText*(SettingsActivity.this, "Error: "+ message, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 }  
  
 private void SendUserToMainActivity() {  
 Intent mainIntent = new Intent(SettingsActivity.this, MainActivity.class);  
 mainIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(mainIntent);  
 finish();  
 }  
  
 private void RetrieveUserInfo() {  
 RootRef.child("Users").child(currentUserId)  
 .addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if((dataSnapshot.exists()) && (dataSnapshot.hasChild("name") && (dataSnapshot.hasChild("image")))){  
  
 String retrieveUserName = dataSnapshot.child("name").getValue().toString();  
 String retrieveStatus = dataSnapshot.child("status").getValue().toString();  
 String retrieveProfileImage = dataSnapshot.child("image").getValue().toString();  
  
 userName.setText(retrieveUserName);  
 userStatus.setText(retrieveStatus);  
 Picasso.*get*().load(retrieveProfileImage).placeholder(R.drawable.*profile\_image*).into(userProfileImage);  
 }  
 else if((dataSnapshot.exists()) && (dataSnapshot.hasChild("name"))){  
 String retrieveUserName = dataSnapshot.child("name").getValue().toString();  
 String retrieveStatus = dataSnapshot.child("status").getValue().toString();  
  
 userName.setText(retrieveUserName);  
 userStatus.setText(retrieveStatus);  
 }else{  
 userName.setVisibility(View.*VISIBLE*);  
 Toast.*makeText*(SettingsActivity.this, "Please set and update your profile information.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void setUserProfileImage(View view){  
// Intent galleryIntent = new Intent();  
// galleryIntent.setAction(Intent.ACTION\_GET\_CONTENT);  
// galleryIntent.setType("image/\*");  
// startActivityForResult(galleryIntent, GalleryPick);  
  
 CropImage.*activity*()  
 .setGuidelines(CropImageView.Guidelines.*ON*)  
 .setAspectRatio(1,1)  
 .start(this);  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
// if(requestCode==GalleryPick && resultCode == RESULT\_OK && data != null){  
// Uri ImageUri = data.getData();  
//  
// }  
  
 if (requestCode == CropImage.*CROP\_IMAGE\_ACTIVITY\_REQUEST\_CODE*) {  
 CropImage.ActivityResult result = CropImage.*getActivityResult*(data);  
  
 if(resultCode == *RESULT\_OK*){  
  
 loadingBar.setTitle("Set Profile Image");  
 loadingBar.setMessage("Please wait, your profile image is updating");  
 loadingBar.setCanceledOnTouchOutside(true);  
 loadingBar.show();  
  
 Uri resultUri = result.getUri();  
 final StorageReference filepath = UserProfileImageRef.child(currentUserId+".jpg");  
 filepath.putFile(resultUri).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(SettingsActivity.this, "Profile image uploaded successfully", Toast.*LENGTH\_SHORT*).show();  
// final String downloadUrl = task.getResult().getStorage().getDownloadUrl().toString();  
 filepath.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {  
 @Override  
 public void onSuccess(Uri uri) {  
 String downloadUrl = uri.toString();  
 RootRef.child("Users").child(currentUserId).child("image")  
 .setValue(downloadUrl)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 loadingBar.dismiss();  
 Toast.*makeText*(SettingsActivity.this, "Image Saved in database, successfully", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 loadingBar.dismiss();  
 String message = task.getException().toString();  
 Toast.*makeText*(SettingsActivity.this, "Error: " + message, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 });  
  
 }else {  
 loadingBar.dismiss();  
 String message = task.getException().toString();  
 Toast.*makeText*(SettingsActivity.this, "Error: " + message, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 }  
 }  
}

## Find Friends Activity and go to parent activity

1. Create new empty activity- FindFriendsActivity
2. AndroidManifest.xml

<activity android:name=".FindFriendsActivity"  
 android:parentActivityName=".MainActivity"  
 ></activity>

1. MainActivity.java – call this function in options menu find friends click

private void SendUserToFindFriendsActivity() {  
  
 Intent loginIntent = new Intent(MainActivity.this, FindFriendsActivity.class);  
// loginIntent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);  
 startActivity(loginIntent);  
// finish();  
 }

1. Activity\_find\_friends.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".FindFriendsActivity">  
  
 <include  
 layout="@layout/app\_bar\_layout"  
 android:id="@+id/find\_friends\_toolbar"  
 ></include>  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/find\_friends\_recycler\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/find\_friends\_toolbar"  
 android:layout\_alignParentStart="true"  
 android:layout\_centerVertical="true"  
 />  
  
</RelativeLayout>

1. FindFriendsActivity.java

package com.example.whatsapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.os.Bundle;  
  
public class FindFriendsActivity extends AppCompatActivity {  
  
 private Toolbar mToolBar;  
 private RecyclerView FindFriendsRecyclerList;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_find\_friends*);  
  
 FindFriendsRecyclerList = (RecyclerView) findViewById(R.id.*find\_friends\_recycler\_list*);  
 FindFriendsRecyclerList.setLayoutManager(new LinearLayoutManager(this));  
  
 mToolBar = (Toolbar) findViewById(R.id.*find\_friends\_toolbar*);  
 setSupportActionBar(mToolBar);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 getSupportActionBar().setDisplayShowHomeEnabled(true);  
 getSupportActionBar().setTitle("Find Friends");  
 }  
}

## UsersCustomLayout

1. Create new layout resource file -> users\_display\_layout – RelativeLayout
2. GO to this site:- <https://github.com/firebase/FirebaseUI-Android>
3. Implement –

implementation 'com.firebaseui:firebase-ui-database:6.2.1'

1. Create new module class – Contacts.java

package com.example.whatsapp;  
  
public class Contacts {  
 public String name, status, image;  
  
 public Contacts() {  
 }  
  
 public Contacts(String name, String status, String image) {  
 this.name = name;  
 this.status = status;  
 this.image = image;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
 public void setStatus(String status) {  
 this.status = status;  
 }  
  
 public String getImage() {  
 return image;  
 }  
  
 public void setImage(String image) {  
 this.image = image;  
 }  
}

1. Users\_display\_layout.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="15dp"  
 android:orientation="horizontal"  
 >  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/users\_profile\_image"  
 android:layout\_width="85dp"  
 android:layout\_height="85dp"  
 android:src="@drawable/profile\_image"  
 />  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:weightSum="5"  
 >  
 <TextView  
 android:id="@+id/user\_profile\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/username"  
 android:textAlignment="textStart"  
 android:textSize="18dp"  
 android:textStyle="bold"  
 android:layout\_weight="4"  
 />  
 <ImageView  
 android:id="@+id/user\_online\_status"  
 android:layout\_width="18dp"  
 android:layout\_height="18dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_gravity="right"  
 android:layout\_weight="1"  
 android:src="@drawable/online"  
 android:visibility="invisible"  
 />  
 </LinearLayout>  
 <TextView  
 android:id="@+id/user\_status"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/status"  
 android:textAlignment="textStart"  
 android:textSize="16dp"  
 />  
 </LinearLayout>  
 </LinearLayout>  
  
</RelativeLayout>

1. FindFriendsActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.squareup.picasso.Picasso;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class FindFriendsActivity extends AppCompatActivity {  
  
 private Toolbar mToolBar;  
 private RecyclerView FindFriendsRecyclerList;  
 private DatabaseReference UsersRef;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_find\_friends*);  
  
 UsersRef = FirebaseDatabase.*getInstance*().getReference().child("Users");  
  
 FindFriendsRecyclerList = (RecyclerView) findViewById(R.id.*find\_friends\_recycler\_list*);  
 FindFriendsRecyclerList.setLayoutManager(new LinearLayoutManager(this));  
  
 mToolBar = (Toolbar) findViewById(R.id.*find\_friends\_toolbar*);  
 setSupportActionBar(mToolBar);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 getSupportActionBar().setDisplayShowHomeEnabled(true);  
 getSupportActionBar().setTitle("Find Friends");  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 FirebaseRecyclerOptions<Contacts> options = new FirebaseRecyclerOptions.Builder<Contacts>()  
 .setQuery(UsersRef, Contacts.class)  
 .build();  
  
 FirebaseRecyclerAdapter<Contacts, FindFriendViewHolder> adapter = new FirebaseRecyclerAdapter<Contacts, FindFriendViewHolder>(options) {  
 @Override  
 protected void onBindViewHolder(@NonNull FindFriendViewHolder holder, int position, @NonNull Contacts model) {  
 holder.userName.setText(model.getName());  
 holder.userStatus.setText(model.getStatus());  
 Picasso.*get*().load(model.getImage()).placeholder(R.drawable.*profile\_image*).into(holder.profileImage);  
 }  
  
 @NonNull  
 @Override  
 public FindFriendViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*users\_display\_layout*, parent, false);  
 FindFriendViewHolder viewHolder = new FindFriendViewHolder(view);  
 return viewHolder;  
 }  
 };  
  
 FindFriendsRecyclerList.setAdapter(adapter);  
 adapter.startListening();  
 }  
  
 public static class FindFriendViewHolder extends RecyclerView.ViewHolder{  
  
 TextView userName, userStatus;  
 CircleImageView profileImage;  
 public FindFriendViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 userName = itemView.findViewById(R.id.*user\_profile\_name*);  
 userStatus = itemView.findViewById(R.id.*user\_status*);  
 profileImage = itemView.findViewById(R.id.*users\_profile\_image*);  
 }  
 }  
}

## Hnadle Item Click

1. Create new empty activity – ProfileActivity
2. FindFriendsActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.squareup.picasso.Picasso;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class FindFriendsActivity extends AppCompatActivity {  
  
 private Toolbar mToolBar;  
 private RecyclerView FindFriendsRecyclerList;  
 private DatabaseReference UsersRef;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_find\_friends*);  
  
 UsersRef = FirebaseDatabase.*getInstance*().getReference().child("Users");  
  
 FindFriendsRecyclerList = (RecyclerView) findViewById(R.id.*find\_friends\_recycler\_list*);  
 FindFriendsRecyclerList.setLayoutManager(new LinearLayoutManager(this));  
  
 mToolBar = (Toolbar) findViewById(R.id.*find\_friends\_toolbar*);  
 setSupportActionBar(mToolBar);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 getSupportActionBar().setDisplayShowHomeEnabled(true);  
 getSupportActionBar().setTitle("Find Friends");  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 FirebaseRecyclerOptions<Contacts> options = new FirebaseRecyclerOptions.Builder<Contacts>()  
 .setQuery(UsersRef, Contacts.class)  
 .build();  
  
 FirebaseRecyclerAdapter<Contacts, FindFriendViewHolder> adapter = new FirebaseRecyclerAdapter<Contacts, FindFriendViewHolder>(options) {  
 @Override  
 protected void onBindViewHolder(@NonNull FindFriendViewHolder holder, final int position, @NonNull Contacts model) {  
 holder.userName.setText(model.getName());  
 holder.userStatus.setText(model.getStatus());  
 Picasso.*get*().load(model.getImage()).placeholder(R.drawable.*profile\_image*).into(holder.profileImage);  
 holder.itemView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String visit\_user\_id = getRef(position).getKey();  
 Intent profileIntent = new Intent(FindFriendsActivity.this, ProfileActivity.class);  
 profileIntent.putExtra("visit\_user\_id", visit\_user\_id);  
 startActivity(profileIntent);  
 }  
 });  
 }  
  
 @NonNull  
 @Override  
 public FindFriendViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*users\_display\_layout*, parent, false);  
 FindFriendViewHolder viewHolder = new FindFriendViewHolder(view);  
 return viewHolder;  
 }  
 };  
  
 FindFriendsRecyclerList.setAdapter(adapter);  
 adapter.startListening();  
 }  
  
 public static class FindFriendViewHolder extends RecyclerView.ViewHolder{  
  
 TextView userName, userStatus;  
 CircleImageView profileImage;  
 public FindFriendViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 userName = itemView.findViewById(R.id.*user\_profile\_name*);  
 userStatus = itemView.findViewById(R.id.*user\_status*);  
 profileImage = itemView.findViewById(R.id.*users\_profile\_image*);  
 }  
 }  
}

1. Activity\_profile.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/profile\_activity"  
 tools:context=".ProfileActivity">  
   
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/visit\_profile\_image"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="110dp"  
 android:src="@drawable/profile\_image"  
 />  
 <TextView  
 android:id="@+id/visit\_user\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/username"  
 android:layout\_below="@+id/visit\_profile\_image"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="15dp"  
 android:layout\_marginTop="5dp"  
 android:textAlignment="center"  
 android:textAllCaps="false"  
 android:textSize="18dp"  
 android:textStyle="bold"  
 />  
 <TextView  
 android:id="@+id/visit\_profile\_status"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/hey\_i\_am\_available\_now"  
 android:layout\_below="@+id/visit\_user\_name"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="15dp"  
 android:layout\_marginTop="25dp"  
 android:textAlignment="center"  
 android:textAllCaps="false"  
 android:textSize="25dp"  
 android:textStyle="bold"  
 />  
  
 <Button  
 android:id="@+id/send\_message\_request\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/visit\_profile\_status"  
 android:layout\_marginTop="15dp"  
 android:layout\_marginLeft="65dp"  
 android:layout\_marginRight="65dp"  
 android:textColor="@android:color/background\_light"  
 android:background="@drawable/buttons"  
 android:textStyle="bold"  
 android:textSize="20dp"  
 android:text="@string/send\_message"  
 android:textAllCaps="false"  
 />  
</RelativeLayout>

1. ProfilActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
  
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;  
import com.squareup.picasso.Picasso;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ProfileActivity extends AppCompatActivity {  
  
 private String receiverUserID;  
  
 private CircleImageView userProfileImage;  
 private TextView userProfileName, userProfileStatus;  
// private Button SendMessageRequestButton;  
 private DatabaseReference UserRef;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_profile*);  
  
 UserRef = FirebaseDatabase.*getInstance*().getReference().child("Users");  
 receiverUserID = getIntent().getExtras().get("visit\_user\_id").toString();  
  
 Toast.*makeText*(this, "User id: "+ receiverUserID, Toast.*LENGTH\_SHORT*).show();  
  
 InitializeFields();  
   
 RetrieveUserInfo();  
 }  
  
 private void InitializeFields() {  
 userProfileImage = (CircleImageView) findViewById(R.id.*visit\_profile\_image*);  
 userProfileName = (TextView) findViewById(R.id.*visit\_user\_name*);  
 userProfileStatus = (TextView) findViewById(R.id.*visit\_profile\_status*);  
 }  
 private void RetrieveUserInfo() {  
 UserRef.child(receiverUserID).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if((dataSnapshot.exists()) && (dataSnapshot.hasChild("image"))){  
 String userImage = dataSnapshot.child("image").getValue().toString();  
 String userName = dataSnapshot.child("name").getValue().toString();  
 String userStatus = dataSnapshot.child("status").getValue().toString();  
  
 Picasso.*get*().load(userImage).placeholder(R.drawable.*profile\_image*).into(userProfileImage);  
 userProfileName.setText(userName);  
 userProfileStatus.setText(userStatus);  
 }else {  
 String userName = dataSnapshot.child("name").getValue().toString();  
 String userStatus = dataSnapshot.child("status").getValue().toString();  
  
 userProfileName.setText(userName);  
 userProfileStatus.setText(userStatus);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
}

## Send Chat Request

1. Activity\_profile.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/profile\_activity"  
 tools:context=".ProfileActivity">  
   
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/visit\_profile\_image"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="110dp"  
 android:src="@drawable/profile\_image"  
 />  
 <TextView  
 android:id="@+id/visit\_user\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/username"  
 android:layout\_below="@+id/visit\_profile\_image"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="15dp"  
 android:layout\_marginTop="5dp"  
 android:textAlignment="center"  
 android:textAllCaps="false"  
 android:textSize="18dp"  
 android:textStyle="bold"  
 />  
 <TextView  
 android:id="@+id/visit\_profile\_status"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/hey\_i\_am\_available\_now"  
 android:layout\_below="@+id/visit\_user\_name"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="15dp"  
 android:layout\_marginTop="25dp"  
 android:textAlignment="center"  
 android:textAllCaps="false"  
 android:textSize="25dp"  
 android:textStyle="bold"  
 />  
  
 <Button  
 android:id="@+id/send\_message\_request\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/visit\_profile\_status"  
 android:layout\_marginTop="15dp"  
 android:layout\_marginLeft="65dp"  
 android:layout\_marginRight="65dp"  
 android:textColor="@android:color/background\_light"  
 android:background="@drawable/buttons"  
 android:textStyle="bold"  
 android:textSize="20dp"  
 android:text="@string/send\_message"  
 android:textAllCaps="false"  
 />  
  
 <Button  
 android:id="@+id/decline\_message\_request\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/send\_message\_request\_button"  
 android:layout\_marginTop="15dp"  
 android:layout\_marginLeft="65dp"  
 android:layout\_marginRight="65dp"  
 android:textColor="@android:color/background\_light"  
 android:background="@drawable/buttons"  
 android:textStyle="bold"  
 android:textSize="20dp"  
 android:text="@string/cancel\_chat\_request"  
 android:textAllCaps="false"  
 android:visibility="invisible"  
 />  
</RelativeLayout>

1. ProfileActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
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;  
import com.squareup.picasso.Picasso;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ProfileActivity extends AppCompatActivity {  
  
 private String receiverUserID, senderUserId, Current\_State;  
  
 private CircleImageView userProfileImage;  
 private TextView userProfileName, userProfileStatus;  
 private Button SendMessageRequestButton, DeclineMessageRequestButton;  
 private DatabaseReference UserRef, ChatRequestRef, ContactsRef;  
 private FirebaseAuth mAuth;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_profile*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 UserRef = FirebaseDatabase.*getInstance*().getReference().child("Users");  
 ChatRequestRef = FirebaseDatabase.*getInstance*().getReference().child("Chat Requests");  
 ContactsRef = FirebaseDatabase.*getInstance*().getReference().child("Contacts");  
  
 receiverUserID = getIntent().getExtras().get("visit\_user\_id").toString();  
 senderUserId = mAuth.getCurrentUser().getUid();  
  
// Toast.makeText(this, "User id: "+ receiverUserID, Toast.LENGTH\_SHORT).show();  
  
 InitializeFields();  
  
 Current\_State = "new";  
 RetrieveUserInfo();  
 }  
  
 private void InitializeFields() {  
 userProfileImage = (CircleImageView) findViewById(R.id.*visit\_profile\_image*);  
 userProfileName = (TextView) findViewById(R.id.*visit\_user\_name*);  
 userProfileStatus = (TextView) findViewById(R.id.*visit\_profile\_status*);  
 SendMessageRequestButton = (Button) findViewById(R.id.*send\_message\_request\_button*);  
 DeclineMessageRequestButton = (Button) findViewById(R.id.*decline\_message\_request\_button*);  
 }  
 private void RetrieveUserInfo() {  
 UserRef.child(receiverUserID).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if((dataSnapshot.exists()) && (dataSnapshot.hasChild("image"))){  
 String userImage = dataSnapshot.child("image").getValue().toString();  
 String userName = dataSnapshot.child("name").getValue().toString();  
 String userStatus = dataSnapshot.child("status").getValue().toString();  
  
 Picasso.*get*().load(userImage).placeholder(R.drawable.*profile\_image*).into(userProfileImage);  
 userProfileName.setText(userName);  
 userProfileStatus.setText(userStatus);  
  
 ManageChatRequests();  
 }else {  
 String userName = dataSnapshot.child("name").getValue().toString();  
 String userStatus = dataSnapshot.child("status").getValue().toString();  
  
 userProfileName.setText(userName);  
 userProfileStatus.setText(userStatus);  
  
 ManageChatRequests();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 private void ManageChatRequests() {  
 ChatRequestRef.child(senderUserId)  
 .addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild(receiverUserID)){  
 String request\_type = dataSnapshot.child(receiverUserID).child("request\_type").getValue().toString();  
 if(request\_type.equals("sent")){  
 Current\_State = "request\_sent";  
 SendMessageRequestButton.setText("Cancel Chat Request");  
 }  
 else if(request\_type.equals("received")){  
 Current\_State = "request\_received";  
 SendMessageRequestButton.setText("Accept Chat Request");  
 DeclineMessageRequestButton.setVisibility(View.*VISIBLE*);  
 DeclineMessageRequestButton.setEnabled(true);  
 DeclineMessageRequestButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 CancelChatRequest();  
 }  
 });  
 }  
 }  
 else {  
 ContactsRef.child(senderUserId)  
 .addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild(receiverUserID)){  
 Current\_State = "friends";  
 SendMessageRequestButton.setText("Remove this contact");  
  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 if(!senderUserId.equals(receiverUserID)){  
 SendMessageRequestButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 SendMessageRequestButton.setEnabled(false);  
 if(Current\_State.equals("new")){  
 SendChatRequest();  
 }  
 else if(Current\_State.equals("request\_sent")){  
 CancelChatRequest();  
 }  
 else if(Current\_State.equals("request\_received")){  
 AcceptChatRequest();  
 }  
 else if(Current\_State.equals("friends")){  
 RemoveSpecificContact();  
 }  
 }  
 });  
 }else {  
 SendMessageRequestButton.setVisibility(View.*INVISIBLE*);  
 }  
 }  
  
 private void RemoveSpecificContact() {  
 ContactsRef.child(senderUserId).child(receiverUserID)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 ContactsRef.child(receiverUserID).child(senderUserId)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 SendMessageRequestButton.setEnabled(true);  
 Current\_State = "new";  
 SendMessageRequestButton.setText("Sent Chat Request");  
 DeclineMessageRequestButton.setVisibility(View.*INVISIBLE*);  
 DeclineMessageRequestButton.setEnabled(false);  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
  
 private void AcceptChatRequest() {  
 ContactsRef.child(senderUserId).child(receiverUserID).child("Contacts").setValue("Saved")  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 ContactsRef.child(receiverUserID).child(senderUserId).child("Contacts").setValue("Saved")  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 removeRequest(senderUserId, receiverUserID, "friends");  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
  
 private void removeRequest(final String senderUserId, final String receiverUserID, final String curState) {  
 ChatRequestRef.child(senderUserId).child(receiverUserID)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 ChatRequestRef.child(receiverUserID).child(senderUserId)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 SendMessageRequestButton.setEnabled(true);  
 Current\_State = curState;  
 SendMessageRequestButton.setText("Remove this contact");  
 DeclineMessageRequestButton.setVisibility(View.*INVISIBLE*);  
 DeclineMessageRequestButton.setEnabled(false);  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
  
 private void CancelChatRequest() {  
 ChatRequestRef.child(senderUserId).child(receiverUserID)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 ChatRequestRef.child(receiverUserID).child(senderUserId)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 SendMessageRequestButton.setEnabled(true);  
 Current\_State = "new";  
 SendMessageRequestButton.setText("Sent Chat Request");  
 DeclineMessageRequestButton.setVisibility(View.*INVISIBLE*);  
 DeclineMessageRequestButton.setEnabled(false);  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
  
 private void SendChatRequest() {  
 ChatRequestRef.child(senderUserId).child(receiverUserID)  
 .child("request\_type").setValue("sent")  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 ChatRequestRef.child(receiverUserID).child(senderUserId)  
 .child("request\_type").setValue("received")  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 SendMessageRequestButton.setEnabled(true);  
 Current\_State = "request\_sent";  
 SendMessageRequestButton.setText("Cancel Chat Request");  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
}

## Show Contacts

1. Fragement\_contacts.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ContactsFragment">  
  
 <!-- *TODO: Update blank fragment layout* -->  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/contacts\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:text="Contact Fragment"  
 />  
  
</RelativeLayout>

1. ContactsFragment.java

package com.example.whatsapp;  
  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
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;  
import com.squareup.picasso.Picasso;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \*/*public class ContactsFragment extends Fragment {  
  
 private View ContactsView;  
 private RecyclerView myContactList;  
 private DatabaseReference ContactsRef, UsersRef;  
 private FirebaseAuth mAuth;  
 private String currentUserID;  
  
 public ContactsFragment() {  
 // Required empty public constructor  
 }  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 ContactsView = inflater.inflate(R.layout.*fragment\_contacts*, container, false);  
 myContactList = (RecyclerView) ContactsView.findViewById(R.id.*contacts\_list*);  
 myContactList.setLayoutManager(new LinearLayoutManager(getContext()));  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUserID = mAuth.getCurrentUser().getUid();  
 ContactsRef = FirebaseDatabase.*getInstance*().getReference().child("Contacts").child(currentUserID);  
 UsersRef = FirebaseDatabase.*getInstance*().getReference().child("Users");  
  
 return ContactsView;  
 }  
  
 @Override  
 public void onStart() {  
 super.onStart();  
  
 FirebaseRecyclerOptions options = new FirebaseRecyclerOptions.Builder<Contacts>()  
 .setQuery(ContactsRef, Contacts.class)  
 .build();  
 FirebaseRecyclerAdapter<Contacts, ContactsViewHolder> adapter = new FirebaseRecyclerAdapter<Contacts, ContactsViewHolder>(options) {  
 @Override  
 protected void onBindViewHolder(@NonNull final ContactsViewHolder holder, int position, @NonNull Contacts model) {  
 String userIds = getRef(position).getKey();  
 UsersRef.child(userIds).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild("image")){  
 String profileImage = dataSnapshot.child("image").getValue().toString();  
 String profileName = dataSnapshot.child("name").getValue().toString();  
 String profileStatus = dataSnapshot.child("status").getValue().toString();  
  
 holder.userName.setText(profileName);  
 holder.userStatus.setText(profileStatus);  
 Picasso.*get*().load(profileImage).placeholder(R.drawable.*profile\_image*).into(holder.profileImage);  
 }  
 else {  
 String profileName = dataSnapshot.child("name").getValue().toString();  
 String profileStatus = dataSnapshot.child("status").getValue().toString();  
  
 holder.userName.setText(profileName);  
 holder.userStatus.setText(profileStatus);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 @NonNull  
 @Override  
 public ContactsViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*users\_display\_layout*, parent, false);  
 ContactsViewHolder viewHolder = new ContactsViewHolder(view);  
 return viewHolder;  
 }  
 };  
  
 myContactList.setAdapter(adapter);  
 adapter.startListening();  
 }  
  
 public static class ContactsViewHolder extends RecyclerView.ViewHolder{  
  
 TextView userName, userStatus;  
 CircleImageView profileImage;  
  
 public ContactsViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 userName = (TextView) itemView.findViewById(R.id.*user\_profile\_name*);  
 userStatus = (TextView) itemView.findViewById(R.id.*user\_status*);  
 profileImage = (CircleImageView) itemView.findViewById(R.id.*users\_profile\_image*);  
 }  
 }  
}

## RequestFragment

1. Com.exmple.appname -> rc-> new-> fragement-> fragment(blank) = RequestFragment- uncheck – include interface callback
2. Fragment\_request.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".RequestFragment">  
  
 <!-- *TODO: Update blank fragment layout* -->  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/chat\_requests\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 />  
  
</RelativeLayout>

1. RequestFragment.java

package com.example.whatsapp;  
  
import android.app.AlertDialog;  
import android.content.DialogInterface;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
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;  
import com.squareup.picasso.Picasso;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \*/*public class RequestFragment extends Fragment {  
  
 private View RequestFragmentView;  
 private RecyclerView myRequestsList;  
 private DatabaseReference ChatRequestRef, UsersRef, ContactsRef;  
 private FirebaseAuth mAuth;  
 private String currentUserId;  
  
 public RequestFragment() {  
 // Required empty public constructor  
 }  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 RequestFragmentView = inflater.inflate(R.layout.*fragment\_request*, container, false);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUserId = mAuth.getCurrentUser().getUid();  
 UsersRef = FirebaseDatabase.*getInstance*().getReference().child("Users");  
 ChatRequestRef = FirebaseDatabase.*getInstance*().getReference().child("Chat Requests");  
 ContactsRef = FirebaseDatabase.*getInstance*().getReference().child("Contacts");  
  
 myRequestsList = (RecyclerView) RequestFragmentView.findViewById(R.id.*chat\_requests\_list*);  
 myRequestsList.setLayoutManager(new LinearLayoutManager(getContext()));  
 return RequestFragmentView;  
 }  
  
 @Override  
 public void onStart() {  
 super.onStart();  
  
 FirebaseRecyclerOptions<Contacts> options = new FirebaseRecyclerOptions.Builder<Contacts>()  
 .setQuery(ChatRequestRef.child(currentUserId), Contacts.class)  
 .build();  
  
 FirebaseRecyclerAdapter<Contacts, RequestViewHolder> adapter =  
 new FirebaseRecyclerAdapter<Contacts, RequestViewHolder>(options) {  
 @Override  
 protected void onBindViewHolder(@NonNull final RequestViewHolder holder, int position, @NonNull Contacts model) {  
 holder.itemView.findViewById(R.id.*request\_accept\_button*).setVisibility(View.*VISIBLE*);  
 holder.itemView.findViewById(R.id.*request\_cancel\_button*).setVisibility(View.*VISIBLE*);  
  
 final String list\_user\_id = getRef(position).getKey();  
 DatabaseReference getTypeRef = getRef(position).child("request\_type").getRef();  
 getTypeRef.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 String type =dataSnapshot.getValue().toString();  
 if(type.equals("received")){  
 UsersRef.child(list\_user\_id).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
  
 final String requestUserName = dataSnapshot.child("name").getValue().toString();  
 final String requestUserStatus = dataSnapshot.child("status").getValue().toString();  
  
 holder.userName.setText(requestUserName);  
 holder.userStatus.setText("wants to connect with you");  
  
 if(dataSnapshot.hasChild("image")){  
 final String requestProfileImage = dataSnapshot.child("image").getValue().toString();  
 Picasso.*get*().load(requestProfileImage).placeholder(R.drawable.*profile\_image*).into(holder.profileImage);  
 }  
  
 holder.itemView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 // alert dialogue  
 CharSequence options[] = new CharSequence[]  
 {  
 "Accept",  
 "Cancel"  
 };  
 AlertDialog.Builder builder = new AlertDialog.Builder(getContext());  
 builder.setTitle(requestUserName + " Chat Request");  
 builder.setItems(options, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 if(which == 0){  
 ContactsRef.child(currentUserId).child(list\_user\_id).child("Contact")  
 .setValue("Saved").addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 ContactsRef.child(list\_user\_id).child(currentUserId).child("Contact")  
 .setValue("Saved").addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 // remove request  
 String message = "New Contact Saved";  
 removeRequest(currentUserId, list\_user\_id, message);  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
 else if(which == 1){  
 // remove request  
 String message = "Contact Deleted";  
 removeRequest(currentUserId, list\_user\_id, message);  
 }  
 }  
 });  
  
 builder.show();  
 // end alert dialogue  
 }  
 });  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 @NonNull  
 @Override  
 public RequestViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*users\_display\_layout*, parent, false);  
 RequestViewHolder holder = new RequestViewHolder(view);  
 return holder;  
 }  
 };  
  
 myRequestsList.setAdapter(adapter);  
 adapter.startListening();  
 }  
  
 private void removeRequest(String currentUserId, final String list\_user\_id, final String message) {  
 ChatRequestRef.child(this.currentUserId).child(list\_user\_id)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 ChatRequestRef.child(list\_user\_id).child(RequestFragment.this.currentUserId)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(getContext(), message, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
  
 public static class RequestViewHolder extends RecyclerView.ViewHolder{  
  
 TextView userName, userStatus;  
 CircleImageView profileImage;  
 Button AcceptButton, CancelButton;  
  
 public RequestViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 userName = (TextView) itemView.findViewById(R.id.*user\_profile\_name*);  
 userStatus = (TextView) itemView.findViewById(R.id.*user\_status*);  
 profileImage = (CircleImageView) itemView.findViewById(R.id.*users\_profile\_image*);  
 AcceptButton = (Button) itemView.findViewById(R.id.*request\_accept\_button*);  
 CancelButton = (Button) itemView.findViewById(R.id.*request\_cancel\_button*);  
 }  
 }  
}

1. TabAccessorAdapter.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentManager;  
import androidx.fragment.app.FragmentPagerAdapter;  
  
public class TabsAccessorAdapter extends FragmentPagerAdapter {  
 public TabsAccessorAdapter(@NonNull FragmentManager fm) {  
 super(fm);  
 }  
  
 @NonNull  
 @Override  
 public Fragment getItem(int position) {  
  
 switch (position){  
 case 0:  
 ChatsFragment chatsFragment = new ChatsFragment();  
 return chatsFragment;  
 case 1:  
 GroupsFragment groupsFragment = new GroupsFragment();  
 return groupsFragment;  
 case 2:  
 ContactsFragment contactsFragment = new ContactsFragment();  
 return contactsFragment;  
 case 3:  
 RequestFragment requestFragment = new RequestFragment();  
 return requestFragment;  
 default:  
 return null;  
 }  
  
  
 }  
  
 @Override  
 public int getCount() {  
 return 4;  
 }  
  
 @Nullable  
 @Override  
 public CharSequence getPageTitle(int position) {  
 switch (position){  
 case 0:  
 return "Chats";  
 case 1:  
 return "Groups";  
 case 2:  
 return "Contacts";  
 case 3:  
 return "Requests";  
 default:  
 return null;  
 }  
 }  
}

1. User\_display\_layout.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="15dp"  
 android:orientation="horizontal"  
 >  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/users\_profile\_image"  
 android:layout\_width="85dp"  
 android:layout\_height="85dp"  
 android:src="@drawable/profile\_image"  
 />  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginLeft="10dp"  
 android:orientation="vertical">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:weightSum="5"  
 >  
 <TextView  
 android:id="@+id/user\_profile\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/username"  
 android:textAlignment="textStart"  
 android:textSize="18dp"  
 android:textStyle="bold"  
 android:layout\_weight="4"  
 />  
 <ImageView  
 android:id="@+id/user\_online\_status"  
 android:layout\_width="18dp"  
 android:layout\_height="18dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_gravity="right"  
 android:layout\_weight="1"  
 android:src="@drawable/online"  
 android:visibility="invisible"  
 />  
 </LinearLayout>  
 <TextView  
 android:id="@+id/user\_status"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/status"  
 android:textAlignment="textStart"  
 android:textSize="16dp"  
 />  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 >  
 <Button  
 android:id="@+id/request\_accept\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="30dp"  
 android:text="@string/accept"  
 android:textSize="16dp"  
 android:textAllCaps="false"  
 android:background="@android:color/holo\_green\_light"  
 android:visibility="invisible"  
 />  
 <Button  
 android:id="@+id/request\_cancel\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="30dp"  
 android:text="@string/cancel"  
 android:layout\_marginStart="10dp"  
 android:textSize="16dp"  
 android:textAllCaps="false"  
 android:background="@android:color/holo\_red\_dark"  
 android:textColor="@android:color/background\_light"  
 android:visibility="invisible"  
 />  
 </LinearLayout>  
 </LinearLayout>  
 </LinearLayout>  
  
</RelativeLayout>

## BugFix

1. SettingsActivity.java

public void setUpdateAccountSettings(View view){  
 String setUserName = userName.getText().toString();  
 String setUserStatus = userStatus.getText().toString();  
 if(TextUtils.*isEmpty*(setUserName)){  
 Toast.*makeText*(this, "Please enter username", Toast.*LENGTH\_SHORT*).show();  
 }  
 if(TextUtils.*isEmpty*(setUserStatus)){  
 Toast.*makeText*(this, "Please enter status", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 HashMap<String, Object> profileMap = new HashMap<>();  
 profileMap.put("uid", currentUserId);  
 profileMap.put("name", setUserName);  
 profileMap.put("status", setUserStatus);  
 RootRef.child("Users").child(currentUserId).updateChildren(profileMap)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(SettingsActivity.this, "Profile updated successfully", Toast.*LENGTH\_SHORT*).show();  
 SendUserToMainActivity();  
 }  
 else {  
 String message = task.getException().toString();  
 Toast.*makeText*(SettingsActivity.this, "Error: "+ message, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
}

## ChatActivity

1. Create new empty activity – ChatActivity
2. Create new resource file – custom\_chat\_bar.xml
3. Create new drawable resource file – receiver\_message\_layout.xml
4. Create new drawable resource file – sender\_message\_layout.xml
5. Create new layout resource file – custom\_message\_layout.xml
6. Create new Java class – Messages.java
7. Create new Java class – MessagesAdapter.java

## Push Notification

1. Gradle

implementation 'com.google.firebase:firebase-messaging:20.0.1'

1. Open cmd
2. >firebase login
3. >firebase init
4. > select function using arrow key – then press space -> enter
5. > select project
6. > select javascript
7. > N
8. > install dependency – Y
9. Index.js file

'use strict'

const functions = require('firebase-functions');

const admin = require('firebase-admin');

admin.initializeApp(functions.config().firebase);

exports.sendNotification = functions.database.ref("/Notifications/{receiver\_user\_id}/{notification\_id}")

.onWrite((data, context) => {

    const receiver\_user\_id = context.params.receiver\_user\_id;

    const notification\_id = context.params.notification\_id;

    console.log('We have a notification to send: ', receiver\_user\_id);

    if(!data.after.val()){

        console.log('A notification has been deleted: ', notification\_id);

        return null;

    }

    const sender\_user\_id = admin.database().ref(`/Notifications/${receiver\_user\_id}/${notification\_id}`).once("value");

    return sender\_user\_id.then(fromUserResult => {

        const from\_sender\_user\_id = fromUserResult.val().from;

        const notification\_type = fromUserResult.val().type;

        console.log('You have a notification from: ', from\_sender\_user\_id);

        console.log('Notification type: ', notification\_type);

        if(notification\_type==="request"){

            const userQuery = admin.database().ref(`/Users/${from\_sender\_user\_id}/name`).once("value");

            return userQuery.then(userResult =>{

                const senderUserName = userResult.val();

                const DeviceToken = admin.database().ref(`/Users/${receiver\_user\_id}/device\_token`).once('value');

                return DeviceToken.then(result => {

                    const token\_id = result.val();

                    const payload = {

                        notification:{

                            from\_sender\_user\_id : from\_sender\_user\_id,

                            title: "New Chat Request",

                            body: `${senderUserName} wants to connect with you.`,

                            icon: "default"

                        }

                    };

                    return admin.messaging().sendToDevice(token\_id, payload)

                    .then( response => {

                        console.log('This was a notification feature');

                    })

                });

            });

        }else if(notification\_type==="chat"){

            const userQuery = admin.database().ref(`/Users/${from\_sender\_user\_id}/name`).once("value");

            return userQuery.then(userResult =>{

                const senderUserName = userResult.val();

                const DeviceToken = admin.database().ref(`/Users/${receiver\_user\_id}/device\_token`).once('value');

                return DeviceToken.then(result => {

                    const token\_id = result.val();

                    const payload = {

                        notification:{

                            from\_sender\_user\_id : from\_sender\_user\_id,

                            title: "New Message",

                            body: `${senderUserName} send a new message to you.`,

                            icon: "default"

                        }

                    };

                    return admin.messaging().sendToDevice(token\_id, payload)

                    .then( response => {

                        console.log('This was a notification feature');

                    })

                });

            });

        }

    });

});

1. ChatActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.Context;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.ChildEventListener;  
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.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ChatActivity extends AppCompatActivity {  
  
 private String messageReceiverId, messageReceiverName, messageReceiverImage, messageSenderId;  
 private TextView userName, userLastSeen;  
 private CircleImageView userImage;  
 private Toolbar ChatToolBar;  
 private ImageButton SendMessageButton;  
 private EditText MessageInputText;  
  
 // firebase  
 private FirebaseAuth mAuth;  
 private DatabaseReference RootRef, NotificationRef;  
  
 private final List<Messages> messagesList = new ArrayList<>();  
 private LinearLayoutManager linearLayoutManager;  
 private MessageAdapter messageAdapter;  
  
 private RecyclerView userMessagesList;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_chat*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 messageSenderId = mAuth.getCurrentUser().getUid();  
 RootRef = FirebaseDatabase.*getInstance*().getReference();  
 NotificationRef = FirebaseDatabase.*getInstance*().getReference().child("Notifications");  
  
 messageReceiverId = getIntent().getExtras().get("visit\_user\_id").toString();  
 messageReceiverName = getIntent().getExtras().get("visit\_user\_name").toString();  
 messageReceiverImage = getIntent().getExtras().get("visit\_image").toString();  
  
 InitializeControllers();  
  
 userName.setText(messageReceiverName);  
 if(!messageReceiverImage.equals("default\_image")){  
 Picasso.*get*().load(messageReceiverImage).placeholder(R.drawable.*profile\_image*).into(userImage);  
 }  
  
 }  
  
 private void InitializeControllers() {  
  
 ChatToolBar = (Toolbar) findViewById(R.id.*chat\_toolbar*);  
 setSupportActionBar(ChatToolBar);  
 ActionBar actionBar = getSupportActionBar();  
 actionBar.setDisplayHomeAsUpEnabled(true);  
 actionBar.setDisplayShowCustomEnabled(true);  
  
 LayoutInflater layoutInflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 View actionBarView = layoutInflater.inflate(R.layout.*custom\_chat\_bar*, null);  
 actionBar.setCustomView(actionBarView);  
  
 userImage = (CircleImageView) findViewById(R.id.*custom\_profile\_image*);  
 userName = (TextView) findViewById(R.id.*custom\_profile\_name*);  
 userLastSeen = (TextView) findViewById(R.id.*custom\_user\_last\_seen*);  
  
 SendMessageButton = (ImageButton) findViewById(R.id.*send\_message\_button*);  
 MessageInputText = (EditText) findViewById(R.id.*input\_message*);  
  
 messageAdapter = new MessageAdapter(messagesList);  
 userMessagesList = (RecyclerView) findViewById(R.id.*private\_messages\_list\_of\_users*);  
 linearLayoutManager = new LinearLayoutManager(this);  
 userMessagesList.setLayoutManager(linearLayoutManager);  
 userMessagesList.setAdapter(messageAdapter);  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 RootRef.child("Messages").child(messageSenderId).child(messageReceiverId)  
 .addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
 Messages messages = dataSnapshot.getValue(Messages.class);  
 messagesList.add(messages);  
 messageAdapter.notifyDataSetChanged();  
 userMessagesList.smoothScrollToPosition(userMessagesList.getAdapter().getItemCount());  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void setSendMessageButton(View view){  
 String messageText = MessageInputText.getText().toString();  
 if(!TextUtils.*isEmpty*(messageText)){  
 String messageSenderRef = "Messages/" + messageSenderId + "/" + messageReceiverId;  
 String messageReceiverRef = "Messages/" + messageReceiverId + "/" + messageSenderId;  
  
 DatabaseReference userMessageKey = RootRef.child("Messages")  
 .child(messageSenderId).child(messageReceiverId).push();  
 String messagePushId = userMessageKey.getKey();  
  
 Map messageTextBody = new HashMap();  
 messageTextBody.put("message", messageText);  
 messageTextBody.put("type", "text");  
 messageTextBody.put("from", messageSenderId);  
  
 Map messageBodyDetails = new HashMap();  
 messageBodyDetails.put(messageSenderRef + "/" + messagePushId, messageTextBody);  
 messageBodyDetails.put(messageReceiverRef + "/" + messagePushId, messageTextBody);  
  
 RootRef.updateChildren(messageBodyDetails).addOnCompleteListener(new OnCompleteListener() {  
 @Override  
 public void onComplete(@NonNull Task task) {  
 if(task.isSuccessful()){  
 HashMap<String, String> chatNotificationMap = new HashMap<>();  
 chatNotificationMap.put("from", messageSenderId);  
 chatNotificationMap.put("type", "chat");  
  
 NotificationRef.child(messageReceiverId).push()  
 .setValue(chatNotificationMap)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(ChatActivity.this, "Notification send", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
// Toast.makeText(ChatActivity.this, "Message Sent Successfully", Toast.LENGTH\_SHORT).show();  
 }else {  
 Toast.*makeText*(ChatActivity.this, "Message Not sent", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 MessageInputText.setText("");  
 }  
 });  
 }else {  
 Toast.*makeText*(this, "Please enter a message", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

1. ProfileActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
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;  
import com.squareup.picasso.Picasso;  
  
import java.util.HashMap;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ProfileActivity extends AppCompatActivity {  
  
 private String receiverUserID, senderUserId, Current\_State;  
  
 private CircleImageView userProfileImage;  
 private TextView userProfileName, userProfileStatus;  
 private Button SendMessageRequestButton, DeclineMessageRequestButton;  
 private DatabaseReference UserRef, ChatRequestRef, ContactsRef, NotificationRef;  
 private FirebaseAuth mAuth;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_profile*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 UserRef = FirebaseDatabase.*getInstance*().getReference().child("Users");  
 ChatRequestRef = FirebaseDatabase.*getInstance*().getReference().child("Chat Requests");  
 ContactsRef = FirebaseDatabase.*getInstance*().getReference().child("Contacts");  
 NotificationRef = FirebaseDatabase.*getInstance*().getReference().child("Notifications");  
  
 receiverUserID = getIntent().getExtras().get("visit\_user\_id").toString();  
 senderUserId = mAuth.getCurrentUser().getUid();  
  
// Toast.makeText(this, "User id: "+ receiverUserID, Toast.LENGTH\_SHORT).show();  
  
 InitializeFields();  
  
 Current\_State = "new";  
 RetrieveUserInfo();  
 }  
  
 private void InitializeFields() {  
 userProfileImage = (CircleImageView) findViewById(R.id.*visit\_profile\_image*);  
 userProfileName = (TextView) findViewById(R.id.*visit\_user\_name*);  
 userProfileStatus = (TextView) findViewById(R.id.*visit\_profile\_status*);  
 SendMessageRequestButton = (Button) findViewById(R.id.*send\_message\_request\_button*);  
 DeclineMessageRequestButton = (Button) findViewById(R.id.*decline\_message\_request\_button*);  
 }  
 private void RetrieveUserInfo() {  
 UserRef.child(receiverUserID).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if((dataSnapshot.exists()) && (dataSnapshot.hasChild("image"))){  
 String userImage = dataSnapshot.child("image").getValue().toString();  
 String userName = dataSnapshot.child("name").getValue().toString();  
 String userStatus = dataSnapshot.child("status").getValue().toString();  
  
 Picasso.*get*().load(userImage).placeholder(R.drawable.*profile\_image*).into(userProfileImage);  
 userProfileName.setText(userName);  
 userProfileStatus.setText(userStatus);  
  
 ManageChatRequests();  
 }else {  
 String userName = dataSnapshot.child("name").getValue().toString();  
 String userStatus = dataSnapshot.child("status").getValue().toString();  
  
 userProfileName.setText(userName);  
 userProfileStatus.setText(userStatus);  
  
 ManageChatRequests();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 private void ManageChatRequests() {  
 ChatRequestRef.child(senderUserId)  
 .addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild(receiverUserID)){  
 String request\_type = dataSnapshot.child(receiverUserID).child("request\_type").getValue().toString();  
 if(request\_type.equals("sent")){  
 Current\_State = "request\_sent";  
 SendMessageRequestButton.setText("Cancel Chat Request");  
 }  
 else if(request\_type.equals("received")){  
 Current\_State = "request\_received";  
 SendMessageRequestButton.setText("Accept Chat Request");  
 DeclineMessageRequestButton.setVisibility(View.*VISIBLE*);  
 DeclineMessageRequestButton.setEnabled(true);  
 DeclineMessageRequestButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 CancelChatRequest();  
 }  
 });  
 }  
 }  
 else {  
 ContactsRef.child(senderUserId)  
 .addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild(receiverUserID)){  
 Current\_State = "friends";  
 SendMessageRequestButton.setText("Remove this contact");  
  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 if(!senderUserId.equals(receiverUserID)){  
 SendMessageRequestButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 SendMessageRequestButton.setEnabled(false);  
 if(Current\_State.equals("new")){  
 SendChatRequest();  
 }  
 else if(Current\_State.equals("request\_sent")){  
 CancelChatRequest();  
 }  
 else if(Current\_State.equals("request\_received")){  
 AcceptChatRequest();  
 }  
 else if(Current\_State.equals("friends")){  
 RemoveSpecificContact();  
 }  
 }  
 });  
 }else {  
 SendMessageRequestButton.setVisibility(View.*INVISIBLE*);  
 }  
 }  
  
 private void RemoveSpecificContact() {  
 ContactsRef.child(senderUserId).child(receiverUserID)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 ContactsRef.child(receiverUserID).child(senderUserId)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 SendMessageRequestButton.setEnabled(true);  
 Current\_State = "new";  
 SendMessageRequestButton.setText("Sent Chat Request");  
 DeclineMessageRequestButton.setVisibility(View.*INVISIBLE*);  
 DeclineMessageRequestButton.setEnabled(false);  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
  
 private void AcceptChatRequest() {  
 ContactsRef.child(senderUserId).child(receiverUserID).child("Contacts").setValue("Saved")  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 ContactsRef.child(receiverUserID).child(senderUserId).child("Contacts").setValue("Saved")  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 removeRequest(senderUserId, receiverUserID, "friends");  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
  
 private void removeRequest(final String senderUserId, final String receiverUserID, final String curState) {  
 ChatRequestRef.child(senderUserId).child(receiverUserID)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 ChatRequestRef.child(receiverUserID).child(senderUserId)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 SendMessageRequestButton.setEnabled(true);  
 Current\_State = curState;  
 SendMessageRequestButton.setText("Remove this contact");  
 DeclineMessageRequestButton.setVisibility(View.*INVISIBLE*);  
 DeclineMessageRequestButton.setEnabled(false);  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
  
 private void CancelChatRequest() {  
 ChatRequestRef.child(senderUserId).child(receiverUserID)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 ChatRequestRef.child(receiverUserID).child(senderUserId)  
 .removeValue()  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 SendMessageRequestButton.setEnabled(true);  
 Current\_State = "new";  
 SendMessageRequestButton.setText("Sent Chat Request");  
 DeclineMessageRequestButton.setVisibility(View.*INVISIBLE*);  
 DeclineMessageRequestButton.setEnabled(false);  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
  
 private void SendChatRequest() {  
 ChatRequestRef.child(senderUserId).child(receiverUserID)  
 .child("request\_type").setValue("sent")  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 ChatRequestRef.child(receiverUserID).child(senderUserId)  
 .child("request\_type").setValue("received")  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 HashMap<String, String> chatNotificationMap = new HashMap<>();  
 chatNotificationMap.put("from", senderUserId);  
 chatNotificationMap.put("type", "request");  
  
 NotificationRef.child(receiverUserID).push()  
 .setValue(chatNotificationMap)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 SendMessageRequestButton.setEnabled(true);  
 Current\_State = "request\_sent";  
 SendMessageRequestButton.setText("Cancel Chat Request");  
 }  
 }  
 });  
  
  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
}

1. LoginActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.iid.FirebaseInstanceId;  
  
public class LoginActivity extends AppCompatActivity {  
  
 private FirebaseAuth mAuth;  
 private Button LoginButton, PhoneLoginButton;  
 private EditText UserEmail, UserPassword;  
 private TextView NeedNewAccountLink, ForgetPasswordLink;  
 private ProgressDialog loadingBar;  
 private DatabaseReference UsersRef;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 UsersRef = FirebaseDatabase.*getInstance*().getReference().child("Users");  
 InitializeFields();  
 }  
  
 private void InitializeFields() {  
 LoginButton = (Button) findViewById(R.id.*login\_button*);  
 PhoneLoginButton = (Button) findViewById(R.id.*phone\_login\_button*);  
 UserEmail = (EditText) findViewById(R.id.*login\_email*);  
 UserPassword = (EditText) findViewById(R.id.*login\_password*);  
 NeedNewAccountLink = (TextView) findViewById(R.id.*need\_new\_account\_link*);  
 ForgetPasswordLink = (TextView) findViewById(R.id.*forget\_password\_link*);  
 loadingBar = new ProgressDialog(this);  
 }  
  
 public void setNeedNewAccountLink(View view){  
 SendUserToRegisterActivity();  
 }  
 public void setLoginButton(View view){  
 AllowUserToLogin();  
 }  
  
 private void AllowUserToLogin() {  
 String email = UserEmail.getText().toString();  
 String password = UserPassword.getText().toString();  
  
 if(TextUtils.*isEmpty*(email)){  
 Toast.*makeText*(this, "Please enter email", Toast.*LENGTH\_SHORT*).show();  
 }  
 if(TextUtils.*isEmpty*(password)){  
 Toast.*makeText*(this, "Please enter password", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 loadingBar.setTitle("Sign In");  
 loadingBar.setMessage("Please wait while we are authenticating your account");  
 loadingBar.setCanceledOnTouchOutside(true);  
 loadingBar.show();  
 mAuth.signInWithEmailAndPassword(email, password)  
 .addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
 String currentUserId = mAuth.getCurrentUser().getUid();  
 String deveiceToken = FirebaseInstanceId.*getInstance*().getToken();  
 UsersRef.child(currentUserId).child("device\_token")  
 .setValue(deveiceToken)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 SendUserToMainActivity();  
 Toast.*makeText*(LoginActivity.this, "Logged in successful", Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
 }  
 }  
 });  
  
  
 }else {  
 String message = task.getException().toString();  
 Toast.*makeText*(LoginActivity.this, "Error: "+message, Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
 }  
 }  
 });  
 }  
 }  
  
 private void SendUserToMainActivity() {  
 Intent mainIntent = new Intent(LoginActivity.this, MainActivity.class);  
 mainIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(mainIntent);  
 finish();  
 }  
 private void SendUserToRegisterActivity() {  
 Intent mainIntent = new Intent(LoginActivity.this, RegisterActivity.class);  
 startActivity(mainIntent);  
 finish();  
 }  
  
 private void SendUserToPhoneLoginActivity() {  
 Intent mainIntent = new Intent(LoginActivity.this, PhoneLoginActivity.class);  
 startActivity(mainIntent);  
 finish();  
 }  
  
  
 public void setPhoneLoginButton(View view){  
 SendUserToPhoneLoginActivity();  
 }  
}

1. RegisterActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.iid.FirebaseInstanceId;  
  
public class RegisterActivity extends AppCompatActivity {  
  
 private Button CreateAccountButton, PhoneLoginButton;  
 private EditText UserEmail, UserPassword;  
 private TextView AlreadyHavaAccountLink;  
  
 private FirebaseAuth mAuth;  
 private DatabaseReference RootRef;  
 private ProgressDialog loadingBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_register*);  
  
 InitializeFields();  
  
 mAuth = FirebaseAuth.*getInstance*();  
 RootRef = FirebaseDatabase.*getInstance*().getReference();  
 }  
  
 private void InitializeFields() {  
 CreateAccountButton = (Button) findViewById(R.id.*register\_button*);  
 UserEmail = (EditText) findViewById(R.id.*register\_email*);  
 UserPassword = (EditText) findViewById(R.id.*register\_password*);  
 AlreadyHavaAccountLink = (TextView) findViewById(R.id.*already\_have\_account\_link*);  
 loadingBar = new ProgressDialog(this);  
 }  
  
 public void setAlreadyHavaAccountLink(View view){  
 SendUserToLoginActivity();  
 }  
  
 private void SendUserToLoginActivity() {  
 Intent mainIntent = new Intent(RegisterActivity.this, LoginActivity.class);  
 startActivity(mainIntent);  
 finish();  
 }  
  
 public void setCreateAccountButton(View view){  
 CreateNewAccount();  
 }  
  
 private void CreateNewAccount() {  
 String email = UserEmail.getText().toString();  
 String password = UserPassword.getText().toString();  
  
 if(TextUtils.*isEmpty*(email)){  
 Toast.*makeText*(this, "Please enter email", Toast.*LENGTH\_SHORT*).show();  
 }  
 if(TextUtils.*isEmpty*(password)){  
 Toast.*makeText*(this, "Please enter password", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 loadingBar.setTitle("Creating new account");  
 loadingBar.setMessage("Please wait while we are creating new account for you");  
 loadingBar.setCanceledOnTouchOutside(true);  
 loadingBar.show();  
 mAuth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
 String deveiceToken = FirebaseInstanceId.*getInstance*().getToken();  
 String currentUserId = mAuth.getCurrentUser().getUid();  
 RootRef.child("Users").child(currentUserId).child("device\_token").setValue(deveiceToken);  
  
 SendUserToMainActivity();  
 Toast.*makeText*(RegisterActivity.this, "Account Created Successfully", Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
 }else {  
 String message = task.getException().toString();  
 Toast.*makeText*(RegisterActivity.this, "Error: "+message, Toast.*LENGTH\_SHORT*).show();  
 loadingBar.dismiss();  
 }  
 }  
 });  
 }  
 }  
  
 private void SendUserToMainActivity() {  
 Intent mainIntent = new Intent(RegisterActivity.this, MainActivity.class);  
 mainIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(mainIntent);  
 finish();  
 }  
}

## Show Send Request in requests field

1. RequestsFragment.java

@Override  
public void onStart() {  
 super.onStart();  
  
 FirebaseRecyclerOptions<Contacts> options = new FirebaseRecyclerOptions.Builder<Contacts>()  
 .setQuery(ChatRequestRef.child(currentUserId), Contacts.class)  
 .build();  
  
 FirebaseRecyclerAdapter<Contacts, RequestViewHolder> adapter =  
 new FirebaseRecyclerAdapter<Contacts, RequestViewHolder>(options) {  
 @Override  
 protected void onBindViewHolder(@NonNull final RequestViewHolder holder, int position, @NonNull Contacts model) {  
 holder.itemView.findViewById(R.id.*request\_accept\_button*).setVisibility(View.*VISIBLE*);  
 holder.itemView.findViewById(R.id.*request\_cancel\_button*).setVisibility(View.*VISIBLE*);  
  
 final String list\_user\_id = getRef(position).getKey();  
 DatabaseReference getTypeRef = getRef(position).child("request\_type").getRef();  
 getTypeRef.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 String type =dataSnapshot.getValue().toString();  
 if(type.equals("received")){  
 UsersRef.child(list\_user\_id).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
  
 final String requestUserName = dataSnapshot.child("name").getValue().toString();  
 final String requestUserStatus = dataSnapshot.child("status").getValue().toString();  
  
 holder.userName.setText(requestUserName);  
 holder.userStatus.setText("wants to connect with you");  
  
 if(dataSnapshot.hasChild("image")){  
 final String requestProfileImage = dataSnapshot.child("image").getValue().toString();  
 Picasso.*get*().load(requestProfileImage).placeholder(R.drawable.*profile\_image*).into(holder.profileImage);  
 }  
  
 holder.itemView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 // alert dialogue  
 CharSequence options[] = new CharSequence[]  
 {  
 "Accept",  
 "Cancel"  
 };  
 AlertDialog.Builder builder = new AlertDialog.Builder(getContext());  
 builder.setTitle(requestUserName + " Chat Request");  
 builder.setItems(options, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 if(which == 0){  
 ContactsRef.child(currentUserId).child(list\_user\_id).child("Contact")  
 .setValue("Saved").addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 ContactsRef.child(list\_user\_id).child(currentUserId).child("Contact")  
 .setValue("Saved").addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 // remove request  
 String message = "New Contact Saved";  
 removeRequest(currentUserId, list\_user\_id, message);  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
 else if(which == 1){  
 // remove request  
 String message = "Contact Deleted";  
 removeRequest(currentUserId, list\_user\_id, message);  
 }  
 }  
 });  
  
 builder.show();  
 // end alert dialogue  
 }  
 });  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
 else if(type.equals("sent")){  
 Button request\_sent\_button = holder.itemView.findViewById(R.id.*request\_accept\_button*);  
 request\_sent\_button.setText("Req sent");  
  
 holder.itemView.findViewById(R.id.*request\_cancel\_button*).setVisibility(View.*INVISIBLE*);  
  
 UsersRef.child(list\_user\_id).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
  
 final String requestUserName = dataSnapshot.child("name").getValue().toString();  
 final String requestUserStatus = dataSnapshot.child("status").getValue().toString();  
  
 holder.userName.setText(requestUserName);  
 holder.userStatus.setText("You have sent a request to " + requestUserName);  
  
 if(dataSnapshot.hasChild("image")){  
 final String requestProfileImage = dataSnapshot.child("image").getValue().toString();  
 Picasso.*get*().load(requestProfileImage).placeholder(R.drawable.*profile\_image*).into(holder.profileImage);  
 }  
  
 holder.itemView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 // alert dialogue  
 CharSequence options[] = new CharSequence[]  
 {  
 "Cancel Chat Request"  
 };  
 AlertDialog.Builder builder = new AlertDialog.Builder(getContext());  
 builder.setTitle("Already sent request");  
 builder.setItems(options, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 if(which == 0){  
 // remove request  
 String message = "You have cancelled the chat request";  
 removeRequest(currentUserId, list\_user\_id, message);  
 }  
 }  
 });  
  
 builder.show();  
 // end alert dialogue  
 }  
 });  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 @NonNull  
 @Override  
 public RequestViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*users\_display\_layout*, parent, false);  
 RequestViewHolder holder = new RequestViewHolder(view);  
 return holder;  
 }  
 };  
  
 myRequestsList.setAdapter(adapter);  
 adapter.startListening();  
}

## Last Seen

1. ChatActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.Context;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.ChildEventListener;  
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;  
import com.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ChatActivity extends AppCompatActivity {  
  
 private String messageReceiverId, messageReceiverName, messageReceiverImage, messageSenderId;  
 private TextView userName, userLastSeen;  
 private CircleImageView userImage;  
 private Toolbar ChatToolBar;  
 private ImageButton SendMessageButton;  
 private EditText MessageInputText;  
  
 // firebase  
 private FirebaseAuth mAuth;  
 private DatabaseReference RootRef, NotificationRef;  
  
 private final List<Messages> messagesList = new ArrayList<>();  
 private LinearLayoutManager linearLayoutManager;  
 private MessageAdapter messageAdapter;  
  
 private RecyclerView userMessagesList;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_chat*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 messageSenderId = mAuth.getCurrentUser().getUid();  
 RootRef = FirebaseDatabase.*getInstance*().getReference();  
 NotificationRef = FirebaseDatabase.*getInstance*().getReference().child("Notifications");  
  
 messageReceiverId = getIntent().getExtras().get("visit\_user\_id").toString();  
 messageReceiverName = getIntent().getExtras().get("visit\_user\_name").toString();  
 messageReceiverImage = getIntent().getExtras().get("visit\_image").toString();  
  
 InitializeControllers();  
  
 userName.setText(messageReceiverName);  
 if(!messageReceiverImage.equals("default\_image")){  
 Picasso.*get*().load(messageReceiverImage).placeholder(R.drawable.*profile\_image*).into(userImage);  
 }  
  
 }  
  
 private void InitializeControllers() {  
  
 ChatToolBar = (Toolbar) findViewById(R.id.*chat\_toolbar*);  
 setSupportActionBar(ChatToolBar);  
 ActionBar actionBar = getSupportActionBar();  
 actionBar.setDisplayHomeAsUpEnabled(true);  
 actionBar.setDisplayShowCustomEnabled(true);  
  
 LayoutInflater layoutInflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 View actionBarView = layoutInflater.inflate(R.layout.*custom\_chat\_bar*, null);  
 actionBar.setCustomView(actionBarView);  
  
 userImage = (CircleImageView) findViewById(R.id.*custom\_profile\_image*);  
 userName = (TextView) findViewById(R.id.*custom\_profile\_name*);  
 userLastSeen = (TextView) findViewById(R.id.*custom\_user\_last\_seen*);  
  
 SendMessageButton = (ImageButton) findViewById(R.id.*send\_message\_button*);  
 MessageInputText = (EditText) findViewById(R.id.*input\_message*);  
  
 messageAdapter = new MessageAdapter(messagesList);  
 userMessagesList = (RecyclerView) findViewById(R.id.*private\_messages\_list\_of\_users*);  
 linearLayoutManager = new LinearLayoutManager(this);  
 userMessagesList.setLayoutManager(linearLayoutManager);  
 userMessagesList.setAdapter(messageAdapter);  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 displayLastSeen();  
 RootRef.child("Messages").child(messageSenderId).child(messageReceiverId)  
 .addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
 Messages messages = dataSnapshot.getValue(Messages.class);  
 messagesList.add(messages);  
 messageAdapter.notifyDataSetChanged();  
 userMessagesList.smoothScrollToPosition(userMessagesList.getAdapter().getItemCount());  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void setSendMessageButton(View view){  
 String messageText = MessageInputText.getText().toString();  
 if(!TextUtils.*isEmpty*(messageText)){  
 String messageSenderRef = "Messages/" + messageSenderId + "/" + messageReceiverId;  
 String messageReceiverRef = "Messages/" + messageReceiverId + "/" + messageSenderId;  
  
 DatabaseReference userMessageKey = RootRef.child("Messages")  
 .child(messageSenderId).child(messageReceiverId).push();  
 String messagePushId = userMessageKey.getKey();  
  
 Map messageTextBody = new HashMap();  
 messageTextBody.put("message", messageText);  
 messageTextBody.put("type", "text");  
 messageTextBody.put("from", messageSenderId);  
  
 Map messageBodyDetails = new HashMap();  
 messageBodyDetails.put(messageSenderRef + "/" + messagePushId, messageTextBody);  
 messageBodyDetails.put(messageReceiverRef + "/" + messagePushId, messageTextBody);  
  
 RootRef.updateChildren(messageBodyDetails).addOnCompleteListener(new OnCompleteListener() {  
 @Override  
 public void onComplete(@NonNull Task task) {  
 if(task.isSuccessful()){  
 HashMap<String, String> chatNotificationMap = new HashMap<>();  
 chatNotificationMap.put("from", messageSenderId);  
 chatNotificationMap.put("type", "chat");  
  
 NotificationRef.child(messageReceiverId).push()  
 .setValue(chatNotificationMap)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(ChatActivity.this, "Notification send", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
// Toast.makeText(ChatActivity.this, "Message Sent Successfully", Toast.LENGTH\_SHORT).show();  
 }else {  
 Toast.*makeText*(ChatActivity.this, "Message Not sent", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 MessageInputText.setText("");  
 }  
 });  
 }else {  
 Toast.*makeText*(this, "Please enter a message", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private void displayLastSeen(){  
 RootRef.child("Users").child(messageSenderId)  
 .addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if( dataSnapshot.child("userState").hasChild("state")){  
 String state = dataSnapshot.child("userState").child("state").getValue().toString();  
 String date = dataSnapshot.child("userState").child("date").getValue().toString();  
 String time = dataSnapshot.child("userState").child("time").getValue().toString();  
  
 if(state.equals("online")){  
 userLastSeen.setText(state);  
 }  
 else if(state.equals("offline")){  
 userLastSeen.setText("Last seen:" + date + " " + time);  
 }  
 }else{  
 userLastSeen.setText("offline");  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 }  
}

1. ContactsFragment.java

package com.example.whatsapp;  
  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
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;  
import com.squareup.picasso.Picasso;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \*/*public class ContactsFragment extends Fragment {  
  
 private View ContactsView;  
 private RecyclerView myContactList;  
 private DatabaseReference ContactsRef, UsersRef;  
 private FirebaseAuth mAuth;  
 private String currentUserID;  
  
 public ContactsFragment() {  
 // Required empty public constructor  
 }  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 ContactsView = inflater.inflate(R.layout.*fragment\_contacts*, container, false);  
 myContactList = (RecyclerView) ContactsView.findViewById(R.id.*contacts\_list*);  
 myContactList.setLayoutManager(new LinearLayoutManager(getContext()));  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUserID = mAuth.getCurrentUser().getUid();  
 ContactsRef = FirebaseDatabase.*getInstance*().getReference().child("Contacts").child(currentUserID);  
 UsersRef = FirebaseDatabase.*getInstance*().getReference().child("Users");  
  
 return ContactsView;  
 }  
  
 @Override  
 public void onStart() {  
 super.onStart();  
  
 FirebaseRecyclerOptions options = new FirebaseRecyclerOptions.Builder<Contacts>()  
 .setQuery(ContactsRef, Contacts.class)  
 .build();  
 FirebaseRecyclerAdapter<Contacts, ContactsViewHolder> adapter = new FirebaseRecyclerAdapter<Contacts, ContactsViewHolder>(options) {  
 @Override  
 protected void onBindViewHolder(@NonNull final ContactsViewHolder holder, int position, @NonNull Contacts model) {  
 String userIds = getRef(position).getKey();  
 UsersRef.child(userIds).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 // user last seen  
 if(dataSnapshot.child("userState").hasChild("state")){  
 String state = dataSnapshot.child("userState").child("state").getValue().toString();  
 String date = dataSnapshot.child("userState").child("date").getValue().toString();  
 String time = dataSnapshot.child("userState").child("time").getValue().toString();  
  
 if(state.equals("online")){  
 holder.onlineIcon.setVisibility(View.*VISIBLE*);  
 }  
 else if(state.equals("offline")){  
 holder.onlineIcon.setVisibility(View.*INVISIBLE*);  
 }  
 }else{  
 holder.onlineIcon.setVisibility(View.*INVISIBLE*);  
 }  
  
 // user profile  
 if(dataSnapshot.hasChild("image")){  
 String profileImage = dataSnapshot.child("image").getValue().toString();  
 String profileName = dataSnapshot.child("name").getValue().toString();  
 String profileStatus = dataSnapshot.child("status").getValue().toString();  
  
 holder.userName.setText(profileName);  
 holder.userStatus.setText(profileStatus);  
 Picasso.*get*().load(profileImage).placeholder(R.drawable.*profile\_image*).into(holder.profileImage);  
 }  
 else {  
 String profileName = dataSnapshot.child("name").getValue().toString();  
 String profileStatus = dataSnapshot.child("status").getValue().toString();  
  
 holder.userName.setText(profileName);  
 holder.userStatus.setText(profileStatus);  
 }  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 @NonNull  
 @Override  
 public ContactsViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*users\_display\_layout*, parent, false);  
 ContactsViewHolder viewHolder = new ContactsViewHolder(view);  
 return viewHolder;  
 }  
 };  
  
 myContactList.setAdapter(adapter);  
 adapter.startListening();  
 }  
  
 public static class ContactsViewHolder extends RecyclerView.ViewHolder{  
  
 TextView userName, userStatus;  
 CircleImageView profileImage;  
 ImageView onlineIcon;  
  
 public ContactsViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 userName = (TextView) itemView.findViewById(R.id.*user\_profile\_name*);  
 userStatus = (TextView) itemView.findViewById(R.id.*user\_status*);  
 profileImage = (CircleImageView) itemView.findViewById(R.id.*users\_profile\_image*);  
 onlineIcon = (ImageView) itemView.findViewById(R.id.*user\_online\_status*);  
 }  
 }  
}

1. ChatsFrgment.java

package com.example.whatsapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
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;  
import com.squareup.picasso.Picasso;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \*/*public class ChatsFragment extends Fragment {  
  
 private View PrivateChatView;  
 private RecyclerView chatsList;  
 private DatabaseReference ChatRef, UsersRef;  
 private FirebaseAuth mAuth;  
 private String currentUserId;  
  
  
 public ChatsFragment() {  
 // Required empty public constructor  
 }  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 PrivateChatView = inflater.inflate(R.layout.*fragment\_chats*, container, false);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 currentUserId = mAuth.getCurrentUser().getUid();  
 ChatRef = FirebaseDatabase.*getInstance*().getReference().child("Contacts").child(currentUserId);  
 UsersRef = FirebaseDatabase.*getInstance*().getReference().child("Users");  
  
 chatsList = (RecyclerView) PrivateChatView.findViewById(R.id.*chats\_list*);  
 chatsList.setLayoutManager(new LinearLayoutManager(getContext()));  
 return PrivateChatView;  
 }  
  
 @Override  
 public void onStart() {  
 super.onStart();  
  
 FirebaseRecyclerOptions<Contacts> options =  
 new FirebaseRecyclerOptions.Builder<Contacts>()  
 .setQuery(ChatRef, Contacts.class)  
 .build();  
  
 FirebaseRecyclerAdapter<Contacts, ChatsViewHolder> adapter =  
 new FirebaseRecyclerAdapter<Contacts, ChatsViewHolder>(options) {  
 @Override  
 protected void onBindViewHolder(@NonNull final ChatsViewHolder holder, int position, @NonNull Contacts model) {  
 final String usersId = getRef(position).getKey();  
 UsersRef.child(usersId).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 String retImage = "default\_image";  
 if(dataSnapshot.hasChild("image")){  
 retImage = dataSnapshot.child("image").getValue().toString();  
 Picasso.*get*().load(retImage).placeholder(R.drawable.*profile\_image*).into(holder.profileImage);  
 }  
 final String retName = dataSnapshot.child("name").getValue().toString();  
 final String retStatus = dataSnapshot.child("status").getValue().toString();  
  
 holder.userName.setText(retName);  
  
 if(dataSnapshot.child("userState").hasChild("state")){  
 String state = dataSnapshot.child("userState").child("state").getValue().toString();  
 String date = dataSnapshot.child("userState").child("date").getValue().toString();  
 String time = dataSnapshot.child("userState").child("time").getValue().toString();  
  
 if(state.equals("online")){  
 holder.userStatus.setText(state);  
 }  
 else if(state.equals("offline")){  
 holder.userStatus.setText("Last seen:" + date + " " + time);  
 }  
 }else{  
 holder.userStatus.setText("offline");  
 }  
  
  
  
 final String finalRetImage = retImage;  
 holder.itemView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent chatIntent = new Intent(getContext(), ChatActivity.class);  
 chatIntent.putExtra("visit\_user\_id", usersId);  
 chatIntent.putExtra("visit\_user\_name", retName);  
 chatIntent.putExtra("visit\_image", finalRetImage);  
 startActivity(chatIntent);  
 }  
 });  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 @NonNull  
 @Override  
 public ChatsViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*users\_display\_layout*, parent, false);  
 return new ChatsViewHolder(view);  
 }  
 };  
 chatsList.setAdapter(adapter);  
 adapter.startListening();  
 }  
  
 public static class ChatsViewHolder extends RecyclerView.ViewHolder{  
 TextView userName, userStatus;  
 CircleImageView profileImage;  
  
 public ChatsViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 userName = (TextView) itemView.findViewById(R.id.*user\_profile\_name*);  
 userStatus = (TextView) itemView.findViewById(R.id.*user\_status*);  
 profileImage = (CircleImageView) itemView.findViewById(R.id.*users\_profile\_image*);  
 }  
 }  
}

## Send Images – PDF, files

1. Custome\_message\_layout.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/message\_profile\_image"  
 android:layout\_width="45dp"  
 android:layout\_height="45dp"  
 android:src="@drawable/profile\_image"  
 android:layout\_marginTop="10dp"  
 />  
  
 <TextView  
 android:id="@+id/receiver\_message\_text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginTop="10dp"  
 android:layout\_toEndOf="@+id/message\_profile\_image"  
 android:background="@drawable/receiver\_message\_layout"  
 android:text="@string/receiver\_messages\_here"  
 android:padding="10dp"  
 android:textSize="14dp"  
 android:textStyle="bold"  
 android:textColor="@android:color/black"  
 />  
 <ImageView  
 android:id="@+id/message\_sender\_image\_view"  
 android:layout\_width="150dp"  
 android:layout\_height="150dp"  
 android:layout\_alignParentEnd="true"  
 android:layout\_margin="5dp"  
 android:scaleType="centerCrop"  
 />  
 <ImageView  
 android:id="@+id/message\_receiver\_image\_view"  
 android:layout\_width="150dp"  
 android:layout\_height="150dp"  
 android:layout\_toEndOf="@+id/message\_profile\_image"  
 android:layout\_margin="5dp"  
 android:scaleType="centerCrop"  
 />  
  
 <TextView  
 android:id="@+id/sender\_message\_text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginTop="10dp"  
 android:layout\_alignParentTop="@+id/receiver\_message\_text"  
 android:background="@drawable/sender\_message\_layout"  
 android:text="@string/sender\_messages"  
 android:textSize="14dp"  
 android:padding="10dp"  
 android:textStyle="bold"  
 android:textColor="@android:color/black"  
 />  
  
</RelativeLayout>

1. Chat\_activity.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@android:color/darker\_gray"  
 tools:context=".ChatActivity">  
  
 <include  
 android:id="@+id/chat\_toolbar"  
 layout="@layout/app\_bar\_layout"  
 >  
 </include>  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/private\_messages\_list\_of\_users"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/chat\_toolbar"  
 android:layout\_above="@+id/chat\_linear\_layout"  
 >  
  
 </androidx.recyclerview.widget.RecyclerView>  
  
 <LinearLayout  
 android:id="@+id/chat\_linear\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentBottom="true"  
 android:background="@android:color/background\_light"  
 android:orientation="horizontal"  
 android:weightSum="10"  
 >  
 <ImageButton  
 android:id="@+id/sent\_files\_button"  
 android:layout\_weight="1"  
 android:layout\_width="60dp"  
 android:layout\_height="50dp"  
 android:src="@drawable/send\_files"  
 android:onClick="setSendFilesButton"  
 android:backgroundTint="@android:color/white"  
 />  
 <EditText  
 android:id="@+id/input\_message"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="8"  
 android:hint="@string/type\_message\_here"  
 android:padding="15dp"  
 />  
 <ImageButton  
 android:id="@+id/sent\_message\_button"  
 android:layout\_weight="1"  
 android:layout\_width="60dp"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/send\_message"  
 android:onClick="setSendMessageButton"  
 android:backgroundTint="@android:color/white"  
 />  
 </LinearLayout>  
</RelativeLayout>

1. MessageAdapter.java

package com.example.whatsapp;  
  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Message;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
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;  
import com.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class MessageAdapter extends RecyclerView.Adapter<MessageAdapter.MessageViewHolder> {  
 private List<Messages> userMessageList;  
  
 private FirebaseAuth mAuth;  
 private DatabaseReference usersRef;  
  
 public MessageAdapter(List<Messages> userMessageList){  
 this.userMessageList = userMessageList;  
 }  
  
 @NonNull  
 @Override  
 public MessageViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*custom\_message\_layout*, parent, false);  
 mAuth = FirebaseAuth.*getInstance*();  
 return new MessageViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull final MessageViewHolder holder, final int position) {  
 String messageSenderId = mAuth.getCurrentUser().getUid();  
 Messages messages = userMessageList.get(position);  
 String fromUserId = messages.getFrom();  
 String fromMessageType = messages.getType();  
 usersRef = FirebaseDatabase.*getInstance*().getReference().child("Users").child(fromUserId);  
 usersRef.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild("image")){  
 String receiverImage = dataSnapshot.child("image").getValue().toString();  
 Picasso.*get*().load(receiverImage).placeholder(R.drawable.*profile\_image*).into(holder.receiverProfileImage);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 holder.receiverMessageText.setVisibility(View.*GONE*);  
 holder.receiverProfileImage.setVisibility(View.*GONE*);  
 holder.senderMessageText.setVisibility(View.*GONE*);  
 holder.messageSenderPicture.setVisibility(View.*GONE*);  
 holder.messageReceiverPicture.setVisibility(View.*GONE*);  
  
 if(fromMessageType.equals("text")){  
  
  
 if(fromUserId.equals(messageSenderId)){  
 holder.senderMessageText.setVisibility(View.*VISIBLE*);  
 holder.senderMessageText.setBackgroundResource(R.drawable.*sender\_message\_layout*);  
 holder.senderMessageText.setText(messages.getMessage() + "\n \n" + messages.getTime() + " - " + messages.getDate());  
 }  
 else {  
  
 holder.receiverProfileImage.setVisibility(View.*VISIBLE*);  
 holder.receiverMessageText.setVisibility(View.*VISIBLE*);  
  
 holder.receiverMessageText.setBackgroundResource(R.drawable.*receiver\_message\_layout*);  
 holder.receiverMessageText.setText(messages.getMessage() + "\n \n" + messages.getTime() + " - " + messages.getDate());  
 }  
 }  
 else if(fromMessageType.equals("image")){  
 if(fromUserId.equals(messageSenderId)){  
 holder.messageSenderPicture.setVisibility(View.*VISIBLE*);  
 Picasso.*get*().load(messages.getMessage()).into(holder.messageSenderPicture);  
 }else {  
 holder.receiverProfileImage.setVisibility(View.*VISIBLE*);  
 holder.messageReceiverPicture.setVisibility(View.*VISIBLE*);  
 Picasso.*get*().load(messages.getMessage()).into(holder.messageReceiverPicture);  
 }  
 }else {  
 if(fromUserId.equals(messageSenderId)){  
 holder.messageSenderPicture.setVisibility(View.*VISIBLE*);  
 holder.messageSenderPicture.setBackgroundResource(R.drawable.*file*);  
  
 holder.itemView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(Intent.*ACTION\_VIEW*, Uri.*parse*(userMessageList.get(position).getMessage()));  
 holder.itemView.getContext().startActivity(intent);  
 }  
 });  
 }  
 else {  
 holder.receiverProfileImage.setVisibility(View.*VISIBLE*);  
 holder.messageReceiverPicture.setVisibility(View.*VISIBLE*);  
  
 holder.messageReceiverPicture.setBackgroundResource(R.drawable.*file*);  
  
 holder.itemView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(Intent.*ACTION\_VIEW*, Uri.*parse*(userMessageList.get(position).getMessage()));  
 holder.itemView.getContext().startActivity(intent);  
 }  
 });  
 }  
 }  
 }  
  
 @Override  
 public int getItemCount() {  
 return userMessageList.size();  
 }  
  
 public class MessageViewHolder extends RecyclerView.ViewHolder{  
  
 public TextView senderMessageText, receiverMessageText;  
 public CircleImageView receiverProfileImage;  
 public ImageView messageSenderPicture, messageReceiverPicture;  
  
 public MessageViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 senderMessageText = (TextView) itemView.findViewById(R.id.*sender\_message\_text*);  
 receiverMessageText = (TextView) itemView.findViewById(R.id.*receiver\_message\_text*);  
 receiverProfileImage = (CircleImageView) itemView.findViewById(R.id.*message\_profile\_image*);  
 messageSenderPicture = (ImageView) itemView.findViewById(R.id.*message\_sender\_image\_view*);  
 messageReceiverPicture = (ImageView) itemView.findViewById(R.id.*message\_receiver\_image\_view*);  
 }  
 }  
}

1. ChatActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.app.ProgressDialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.Continuation;  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnFailureListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.ChildEventListener;  
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;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.OnProgressListener;  
import com.google.firebase.storage.StorageReference;  
import com.google.firebase.storage.StorageTask;  
import com.google.firebase.storage.UploadTask;  
import com.squareup.picasso.Picasso;  
  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ChatActivity extends AppCompatActivity {  
  
 private String messageReceiverId, messageReceiverName, messageReceiverImage, messageSenderId;  
 private TextView userName, userLastSeen;  
 private CircleImageView userImage;  
 private Toolbar ChatToolBar;  
 private ImageButton SendMessageButton, SendFilesButton;  
 private EditText MessageInputText;  
  
 // firebase  
 private FirebaseAuth mAuth;  
 private DatabaseReference RootRef, NotificationRef;  
  
 private final List<Messages> messagesList = new ArrayList<>();  
 private LinearLayoutManager linearLayoutManager;  
 private MessageAdapter messageAdapter;  
  
 private RecyclerView userMessagesList;  
  
 private String saveCurrentTime, saveCurrentDate;  
 private String checker = "", myUrl = "";  
 private StorageTask uploadTask;  
 private Uri fileUri;  
  
 private ProgressDialog loadingBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_chat*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 messageSenderId = mAuth.getCurrentUser().getUid();  
 RootRef = FirebaseDatabase.*getInstance*().getReference();  
 NotificationRef = FirebaseDatabase.*getInstance*().getReference().child("Notifications");  
  
 messageReceiverId = getIntent().getExtras().get("visit\_user\_id").toString();  
 messageReceiverName = getIntent().getExtras().get("visit\_user\_name").toString();  
 messageReceiverImage = getIntent().getExtras().get("visit\_image").toString();  
  
 InitializeControllers();  
  
 userName.setText(messageReceiverName);  
 if(!messageReceiverImage.equals("default\_image")){  
 Picasso.*get*().load(messageReceiverImage).placeholder(R.drawable.*profile\_image*).into(userImage);  
 }  
  
 }  
  
 private void InitializeControllers() {  
  
 ChatToolBar = (Toolbar) findViewById(R.id.*chat\_toolbar*);  
 setSupportActionBar(ChatToolBar);  
 ActionBar actionBar = getSupportActionBar();  
 actionBar.setDisplayHomeAsUpEnabled(true);  
 actionBar.setDisplayShowCustomEnabled(true);  
  
 LayoutInflater layoutInflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 View actionBarView = layoutInflater.inflate(R.layout.*custom\_chat\_bar*, null);  
 actionBar.setCustomView(actionBarView);  
  
 userImage = (CircleImageView) findViewById(R.id.*custom\_profile\_image*);  
 userName = (TextView) findViewById(R.id.*custom\_profile\_name*);  
 userLastSeen = (TextView) findViewById(R.id.*custom\_user\_last\_seen*);  
  
 SendMessageButton = (ImageButton) findViewById(R.id.*send\_message\_button*);  
 MessageInputText = (EditText) findViewById(R.id.*input\_message*);  
  
 messageAdapter = new MessageAdapter(messagesList);  
 userMessagesList = (RecyclerView) findViewById(R.id.*private\_messages\_list\_of\_users*);  
 linearLayoutManager = new LinearLayoutManager(this);  
 userMessagesList.setLayoutManager(linearLayoutManager);  
 userMessagesList.setAdapter(messageAdapter);  
  
 Calendar calendar = Calendar.*getInstance*();  
  
 SimpleDateFormat currentDate = new SimpleDateFormat("MMM dd, yyyy");  
 saveCurrentDate = currentDate.format(calendar.getTime());  
  
 SimpleDateFormat currentTime = new SimpleDateFormat("hh:mm a");  
 saveCurrentTime = currentTime.format(calendar.getTime());  
  
 loadingBar = new ProgressDialog(this);  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 displayLastSeen();  
  
 messagesList.clear();  
 RootRef.child("Messages").child(messageSenderId).child(messageReceiverId)  
 .addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 Messages messages = dataSnapshot.getValue(Messages.class);  
 messagesList.add(messages);  
 messageAdapter.notifyDataSetChanged();  
 userMessagesList.smoothScrollToPosition(userMessagesList.getAdapter().getItemCount());  
  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void setSendMessageButton(View view){  
 String messageText = MessageInputText.getText().toString();  
 if(!TextUtils.*isEmpty*(messageText)){  
 String messageSenderRef = "Messages/" + messageSenderId + "/" + messageReceiverId;  
 String messageReceiverRef = "Messages/" + messageReceiverId + "/" + messageSenderId;  
  
 DatabaseReference userMessageKey = RootRef.child("Messages")  
 .child(messageSenderId).child(messageReceiverId).push();  
 String messagePushId = userMessageKey.getKey();  
  
 Map messageTextBody = new HashMap();  
 messageTextBody.put("message", messageText);  
 messageTextBody.put("type", "text");  
 messageTextBody.put("from", messageSenderId);  
 messageTextBody.put("to", messageReceiverId);  
 messageTextBody.put("messageID", messagePushId);  
 messageTextBody.put("time", saveCurrentTime);  
 messageTextBody.put("date", saveCurrentDate);  
  
 Map messageBodyDetails = new HashMap();  
 messageBodyDetails.put(messageSenderRef + "/" + messagePushId, messageTextBody);  
 messageBodyDetails.put(messageReceiverRef + "/" + messagePushId, messageTextBody);  
  
 RootRef.updateChildren(messageBodyDetails).addOnCompleteListener(new OnCompleteListener() {  
 @Override  
 public void onComplete(@NonNull Task task) {  
 if(task.isSuccessful()){  
 HashMap<String, String> chatNotificationMap = new HashMap<>();  
 chatNotificationMap.put("from", messageSenderId);  
 chatNotificationMap.put("type", "chat");  
  
 NotificationRef.child(messageReceiverId).push()  
 .setValue(chatNotificationMap)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(ChatActivity.this, "Notification send", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
// Toast.makeText(ChatActivity.this, "Message Sent Successfully", Toast.LENGTH\_SHORT).show();  
 }else {  
 Toast.*makeText*(ChatActivity.this, "Message Not sent", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 MessageInputText.setText("");  
 }  
 });  
 }else {  
 Toast.*makeText*(this, "Please enter a message", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 public void setSendFilesButton(View view){  
 CharSequence options[] = new CharSequence[]{  
 "Images",  
 "PDF Files",  
 "MS Word Files"  
 };  
 AlertDialog.Builder builder = new AlertDialog.Builder(ChatActivity.this);  
 builder.setTitle("Select the file");  
 builder.setItems(options, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 if(which == 0){  
 checker = "image";  
  
 Intent intent = new Intent();  
 intent.setAction(Intent.*ACTION\_GET\_CONTENT*);  
 intent.setType("image/\*");  
 startActivityForResult(intent.*createChooser*(intent, "Select Image"), 438);  
 }else if(which == 1){  
 checker = "pdf";  
  
 Intent intent = new Intent();  
 intent.setAction(Intent.*ACTION\_GET\_CONTENT*);  
 intent.setType("application/pdf");  
 startActivityForResult(intent.*createChooser*(intent, "Select PDF File"), 438);  
 }else if(which == 2){  
 checker = "docx";  
  
 Intent intent = new Intent();  
 intent.setAction(Intent.*ACTION\_GET\_CONTENT*);  
 intent.setType("application/msword");  
 startActivityForResult(intent.*createChooser*(intent, "Select Ms Word File"), 438);  
 }  
 }  
 });  
 builder.show();  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 if(requestCode == 438 && resultCode == *RESULT\_OK* && data != null && data.getData() != null){  
 loadingBar.setTitle("Sending File");  
 loadingBar.setMessage("Please wait, your file is sending");  
 loadingBar.setCanceledOnTouchOutside(true);  
 loadingBar.show();  
  
 fileUri = data.getData();  
 if(!checker.equals("image")){  
 StorageReference storageReference = FirebaseStorage.*getInstance*().getReference().child("Document Files");  
 final String messageSenderRef = "Messages/" + messageSenderId + "/" + messageReceiverId;  
 final String messageReceiverRef = "Messages/" + messageReceiverId + "/" + messageSenderId;  
  
 DatabaseReference userMessageKey = RootRef.child("Messages")  
 .child(messageSenderId).child(messageReceiverId).push();  
 final String messagePushId = userMessageKey.getKey();  
  
 final StorageReference filePath = storageReference.child(messagePushId + "."+ checker);  
 filePath.putFile(fileUri).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {  
 if(task.isSuccessful()){  
 filePath.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {  
 @Override  
 public void onSuccess(Uri uri) {  
 String downloadUrl = uri.toString();  
  
  
 Map messageTextBody = new HashMap();  
 messageTextBody.put("message", downloadUrl);  
 messageTextBody.put("name", fileUri.getLastPathSegment());  
 messageTextBody.put("type", checker);  
 messageTextBody.put("from", messageSenderId);  
 messageTextBody.put("to", messageReceiverId);  
 messageTextBody.put("messageID", messagePushId);  
 messageTextBody.put("time", saveCurrentTime);  
 messageTextBody.put("date", saveCurrentDate);  
  
 Map messageBodyDetails = new HashMap();  
 messageBodyDetails.put(messageSenderRef + "/" + messagePushId, messageTextBody);  
 messageBodyDetails.put(messageReceiverRef + "/" + messagePushId, messageTextBody);  
  
 RootRef.updateChildren(messageBodyDetails);  
 loadingBar.dismiss();  
 }  
 });  
 }  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 loadingBar.dismiss();  
 Toast.*makeText*(ChatActivity.this, "Error: " + e.getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 }).addOnProgressListener(new OnProgressListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onProgress(@NonNull UploadTask.TaskSnapshot taskSnapshot) {  
 double p = (100.0 \* taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();  
 loadingBar.setMessage((int) p + " % uploaded");  
 }  
 });  
 }else {  
 StorageReference storageReference = FirebaseStorage.*getInstance*().getReference().child("Image Files");  
 final String messageSenderRef = "Messages/" + messageSenderId + "/" + messageReceiverId;  
 final String messageReceiverRef = "Messages/" + messageReceiverId + "/" + messageSenderId;  
  
 DatabaseReference userMessageKey = RootRef.child("Messages")  
 .child(messageSenderId).child(messageReceiverId).push();  
 final String messagePushId = userMessageKey.getKey();  
  
 final StorageReference filePath = storageReference.child(messagePushId + ".jpg" );  
 uploadTask = filePath.putFile(fileUri);  
 uploadTask.continueWithTask(new Continuation() {  
 @Override  
 public Object then(@NonNull Task task) throws Exception {  
 if(!task.isSuccessful()){  
 throw task.getException();  
 }  
 return filePath.getDownloadUrl();  
 }  
 }).addOnCompleteListener(new OnCompleteListener<Uri>() {  
 @Override  
 public void onComplete(@NonNull Task<Uri> task) {  
 if(task.isSuccessful()){  
 Uri downloadUrl = task.getResult();  
 myUrl = downloadUrl.toString();  
  
 Map messageTextBody = new HashMap();  
 messageTextBody.put("message", myUrl);  
 messageTextBody.put("name", fileUri.getLastPathSegment());  
 messageTextBody.put("type", checker);  
 messageTextBody.put("from", messageSenderId);  
 messageTextBody.put("to", messageReceiverId);  
 messageTextBody.put("messageID", messagePushId);  
 messageTextBody.put("time", saveCurrentTime);  
 messageTextBody.put("date", saveCurrentDate);  
  
 Map messageBodyDetails = new HashMap();  
 messageBodyDetails.put(messageSenderRef + "/" + messagePushId, messageTextBody);  
 messageBodyDetails.put(messageReceiverRef + "/" + messagePushId, messageTextBody);  
  
 RootRef.updateChildren(messageBodyDetails).addOnCompleteListener(new OnCompleteListener() {  
 @Override  
 public void onComplete(@NonNull Task task) {  
 if(task.isSuccessful()){  
 HashMap<String, String> chatNotificationMap = new HashMap<>();  
 chatNotificationMap.put("from", messageSenderId);  
 chatNotificationMap.put("type", "chat");  
  
 NotificationRef.child(messageReceiverId).push()  
 .setValue(chatNotificationMap)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 loadingBar.dismiss();  
 Toast.*makeText*(ChatActivity.this, "Message send successfully", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
// Toast.makeText(ChatActivity.this, "Message Sent Successfully", Toast.LENGTH\_SHORT).show();  
 }else {  
 loadingBar.dismiss();  
 Toast.*makeText*(ChatActivity.this, "Message Not sent", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 MessageInputText.setText("");  
 }  
 });  
  
 }  
 }  
 });  
 }  
 }  
 }  
  
 private void displayLastSeen(){  
 RootRef.child("Users").child(messageReceiverId)  
 .addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if( dataSnapshot.child("userState").hasChild("state")){  
 String state = dataSnapshot.child("userState").child("state").getValue().toString();  
 String date = dataSnapshot.child("userState").child("date").getValue().toString();  
 String time = dataSnapshot.child("userState").child("time").getValue().toString();  
  
 if(state.equals("online")){  
 userLastSeen.setText(state);  
 }  
 else if(state.equals("offline")){  
 userLastSeen.setText("Last seen:" + date + " " + time);  
 }  
 }else{  
 userLastSeen.setText("offline");  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 }  
}

1. Messages.java

package com.example.whatsapp;  
  
public class Messages {  
 private String from, message, type, to, messageID, time, date, name;  
  
 public Messages() {  
 }  
  
 public Messages(String from, String message, String type, String to, String messageID, String time, String date, String name) {  
 this.from = from;  
 this.message = message;  
 this.type = type;  
 this.to = to;  
 this.messageID = messageID;  
 this.time = time;  
 this.date = date;  
 this.name = name;  
 }  
  
 public String getFrom() {  
 return from;  
 }  
  
 public void setFrom(String from) {  
 this.from = from;  
 }  
  
 public String getMessage() {  
 return message;  
 }  
  
 public void setMessage(String message) {  
 this.message = message;  
 }  
  
 public String getType() {  
 return type;  
 }  
  
 public void setType(String type) {  
 this.type = type;  
 }  
  
 public String getTo() {  
 return to;  
 }  
  
 public void setTo(String to) {  
 this.to = to;  
 }  
  
 public String getMessageID() {  
 return messageID;  
 }  
  
 public void setMessageID(String messageID) {  
 this.messageID = messageID;  
 }  
  
 public String getTime() {  
 return time;  
 }  
  
 public void setTime(String time) {  
 this.time = time;  
 }  
  
 public String getDate() {  
 return date;  
 }  
  
 public void setDate(String date) {  
 this.date = date;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
}

## Delete Messages

1. Create new empty activity – ImageViewerActivity
2. MessageAdapter.java

package com.example.whatsapp;  
  
import android.app.AlertDialog;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Message;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
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;  
import com.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class MessageAdapter extends RecyclerView.Adapter<MessageAdapter.MessageViewHolder> {  
 private List<Messages> userMessageList;  
  
 private FirebaseAuth mAuth;  
 private DatabaseReference usersRef;  
  
 public MessageAdapter(List<Messages> userMessageList){  
 this.userMessageList = userMessageList;  
 }  
  
 @NonNull  
 @Override  
 public MessageViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*custom\_message\_layout*, parent, false);  
 mAuth = FirebaseAuth.*getInstance*();  
 return new MessageViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull final MessageViewHolder holder, final int position) {  
 String messageSenderId = mAuth.getCurrentUser().getUid();  
 Messages messages = userMessageList.get(position);  
 String fromUserId = messages.getFrom();  
 String fromMessageType = messages.getType();  
 usersRef = FirebaseDatabase.*getInstance*().getReference().child("Users").child(fromUserId);  
 usersRef.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild("image")){  
 String receiverImage = dataSnapshot.child("image").getValue().toString();  
 Picasso.*get*().load(receiverImage).placeholder(R.drawable.*profile\_image*).into(holder.receiverProfileImage);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 holder.receiverMessageText.setVisibility(View.*GONE*);  
 holder.receiverProfileImage.setVisibility(View.*GONE*);  
 holder.senderMessageText.setVisibility(View.*GONE*);  
 holder.messageSenderPicture.setVisibility(View.*GONE*);  
 holder.messageReceiverPicture.setVisibility(View.*GONE*);  
  
 if(fromMessageType.equals("text")){  
  
  
 if(fromUserId.equals(messageSenderId)){  
 holder.senderMessageText.setVisibility(View.*VISIBLE*);  
 holder.senderMessageText.setBackgroundResource(R.drawable.*sender\_message\_layout*);  
 holder.senderMessageText.setText(messages.getMessage() + "\n \n" + messages.getTime() + " - " + messages.getDate());  
 }  
 else {  
  
 holder.receiverProfileImage.setVisibility(View.*VISIBLE*);  
 holder.receiverMessageText.setVisibility(View.*VISIBLE*);  
  
 holder.receiverMessageText.setBackgroundResource(R.drawable.*receiver\_message\_layout*);  
 holder.receiverMessageText.setText(messages.getMessage() + "\n \n" + messages.getTime() + " - " + messages.getDate());  
 }  
 }  
 else if(fromMessageType.equals("image")){  
 if(fromUserId.equals(messageSenderId)){  
 holder.messageSenderPicture.setVisibility(View.*VISIBLE*);  
 Picasso.*get*().load(messages.getMessage()).into(holder.messageSenderPicture);  
 }else {  
 holder.receiverProfileImage.setVisibility(View.*VISIBLE*);  
 holder.messageReceiverPicture.setVisibility(View.*VISIBLE*);  
 Picasso.*get*().load(messages.getMessage()).into(holder.messageReceiverPicture);  
 }  
 }else if(fromMessageType.equals("pdf") || fromMessageType.equals("docx")){  
 if(fromUserId.equals(messageSenderId)){  
 holder.messageSenderPicture.setVisibility(View.*VISIBLE*);  
 holder.messageSenderPicture.setBackgroundResource(R.drawable.*file*);  
  
 }  
 else {  
 holder.receiverProfileImage.setVisibility(View.*VISIBLE*);  
 holder.messageReceiverPicture.setVisibility(View.*VISIBLE*);  
  
 holder.messageReceiverPicture.setBackgroundResource(R.drawable.*file*);  
  
  
 }  
 }  
  
 if(fromUserId.equals(messageSenderId)){  
 holder.itemView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if(userMessageList.get(position).getType().equals("pdf") || userMessageList.get(position).getType().equals("docx")){  
 CharSequence options[] = new CharSequence[]{  
 "Delete for me",  
 "Download and View this document",  
 "Cancel",  
 "Delete for everyone"  
 };  
 AlertDialog.Builder builder = new AlertDialog.Builder(holder.itemView.getContext());  
 builder.setTitle("Delete Message?");  
 builder.setItems(options, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 if(which == 0){  
 deleteSendMessage(position, holder);  
 Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);  
 holder.itemView.getContext().startActivity(intent);  
 }  
 else if(which == 1){  
 Intent intent = new Intent(Intent.*ACTION\_VIEW*, Uri.*parse*(userMessageList.get(position).getMessage()));  
 holder.itemView.getContext().startActivity(intent);  
 }  
  
 else if(which == 3){  
 deleteMessageForEveryOne(position, holder);  
 Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);  
 holder.itemView.getContext().startActivity(intent);  
 }  
 }  
 });  
 builder.show();  
 }  
 else if(userMessageList.get(position).getType().equals("text")){  
 CharSequence options[] = new CharSequence[]{  
 "Delete for me",  
 "Cancel",  
 "Delete for everyone"  
 };  
 AlertDialog.Builder builder = new AlertDialog.Builder(holder.itemView.getContext());  
 builder.setTitle("Delete Message?");  
 builder.setItems(options, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 if(which == 0){  
 deleteSendMessage(position, holder);  
 Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);  
 holder.itemView.getContext().startActivity(intent);  
 }  
  
 else if(which == 2){  
 deleteMessageForEveryOne(position, holder);  
 Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);  
 holder.itemView.getContext().startActivity(intent);  
 }  
 }  
 });  
 builder.show();  
 }  
 else if(userMessageList.get(position).getType().equals("image")){  
 CharSequence options[] = new CharSequence[]{  
 "Delete for me",  
 "View this image",  
 "Cancel",  
 "Delete for everyone"  
 };  
 AlertDialog.Builder builder = new AlertDialog.Builder(holder.itemView.getContext());  
 builder.setTitle("Delete Message?");  
 builder.setItems(options, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 if(which == 0){  
 deleteSendMessage(position, holder);  
 Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);  
 holder.itemView.getContext().startActivity(intent);  
 }  
 else if(which == 1){  
 Intent intent = new Intent(holder.itemView.getContext(), ImageViewerActivity.class);  
 intent.putExtra("url", userMessageList.get(position).getMessage());  
 holder.itemView.getContext().startActivity(intent);  
 }  
 else if(which == 3){  
 deleteMessageForEveryOne(position, holder);  
 Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);  
 holder.itemView.getContext().startActivity(intent);  
 }  
 }  
 });  
 builder.show();  
 }  
  
 }  
 });  
 }  
 else {  
 holder.itemView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if(userMessageList.get(position).getType().equals("pdf") || userMessageList.get(position).getType().equals("docx")){  
 CharSequence options[] = new CharSequence[]{  
 "Delete for me",  
 "Download and View this document",  
 "Cancel"  
 };  
 AlertDialog.Builder builder = new AlertDialog.Builder(holder.itemView.getContext());  
 builder.setTitle("Delete Message?");  
 builder.setItems(options, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 if(which == 0){  
 deleteReceiveMessage(position, holder);  
 Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);  
 holder.itemView.getContext().startActivity(intent);  
 }  
 else if(which == 1){  
 Intent intent = new Intent(Intent.*ACTION\_VIEW*, Uri.*parse*(userMessageList.get(position).getMessage()));  
 holder.itemView.getContext().startActivity(intent);  
 }  
 }  
 });  
 builder.show();  
 }  
 else if(userMessageList.get(position).getType().equals("text")){  
 CharSequence options[] = new CharSequence[]{  
 "Delete for me",  
 "Cancel"  
 };  
 AlertDialog.Builder builder = new AlertDialog.Builder(holder.itemView.getContext());  
 builder.setTitle("Delete Message?");  
 builder.setItems(options, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 if(which == 0){  
 deleteReceiveMessage(position, holder);  
 Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);  
 holder.itemView.getContext().startActivity(intent);  
 }  
 }  
 });  
 builder.show();  
 }  
 else if(userMessageList.get(position).getType().equals("image")){  
 CharSequence options[] = new CharSequence[]{  
 "Delete for me",  
 "View this image",  
 "Cancel"  
 };  
 AlertDialog.Builder builder = new AlertDialog.Builder(holder.itemView.getContext());  
 builder.setTitle("Delete Message?");  
 builder.setItems(options, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 if(which == 0){  
 deleteReceiveMessage(position, holder);  
 Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);  
 holder.itemView.getContext().startActivity(intent);  
 }  
 else if(which == 1){  
 Intent intent = new Intent(holder.itemView.getContext(), ImageViewerActivity.class);  
 intent.putExtra("url", userMessageList.get(position).getMessage());  
 holder.itemView.getContext().startActivity(intent);  
 }  
 }  
 });  
 builder.show();  
 }  
  
 }  
 });  
 }  
 }  
  
 private void deleteSendMessage(final int position, final MessageViewHolder holder){  
 DatabaseReference rootRef = FirebaseDatabase.*getInstance*().getReference();  
 rootRef.child("Messages").child(userMessageList.get(position).getFrom())  
 .child(userMessageList.get(position).getTo())  
 .child(userMessageList.get(position).getMessageID())  
 .removeValue().addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(holder.itemView.getContext(), "Deleted Successfully", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 Toast.*makeText*(holder.itemView.getContext(), "Error Occurred", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
  
 private void deleteReceiveMessage(final int position, final MessageViewHolder holder){  
 DatabaseReference rootRef = FirebaseDatabase.*getInstance*().getReference();  
 rootRef.child("Messages").child(userMessageList.get(position).getTo())  
 .child(userMessageList.get(position).getFrom())  
 .child(userMessageList.get(position).getMessageID())  
 .removeValue().addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(holder.itemView.getContext(), "Deleted Successfully", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 Toast.*makeText*(holder.itemView.getContext(), "Error Occurred", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
  
 private void deleteMessageForEveryOne(final int position, final MessageViewHolder holder){  
 deleteSendMessage(position, holder);  
 deleteReceiveMessage(position, holder);  
 }  
  
 @Override  
 public int getItemCount() {  
 return userMessageList.size();  
 }  
  
 public class MessageViewHolder extends RecyclerView.ViewHolder{  
  
 public TextView senderMessageText, receiverMessageText;  
 public CircleImageView receiverProfileImage;  
 public ImageView messageSenderPicture, messageReceiverPicture;  
  
 public MessageViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 senderMessageText = (TextView) itemView.findViewById(R.id.*sender\_message\_text*);  
 receiverMessageText = (TextView) itemView.findViewById(R.id.*receiver\_message\_text*);  
 receiverProfileImage = (CircleImageView) itemView.findViewById(R.id.*message\_profile\_image*);  
 messageSenderPicture = (ImageView) itemView.findViewById(R.id.*message\_sender\_image\_view*);  
 messageReceiverPicture = (ImageView) itemView.findViewById(R.id.*message\_receiver\_image\_view*);  
 }  
 }  
}

1. Activity\_image\_viewer.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ImageViewerActivity">  
 <ImageView  
 android:id="@+id/image\_viewer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:src="@drawable/crop\_image\_menu\_flip"  
 />  
  
</RelativeLayout>

1. ImageViewerActivity.java

package com.example.whatsapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.ImageView;  
  
import com.squareup.picasso.Picasso;  
  
public class ImageViewerActivity extends AppCompatActivity {  
  
 private ImageView imageView;  
 private String imageUrl;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_image\_viewer*);  
  
 imageView = findViewById(R.id.*image\_viewer*);  
 imageUrl = getIntent().getStringExtra("url");  
  
 Picasso.*get*().load(imageUrl).into(imageView);  
 }  
}

# Firebase chat

## Send OTP

<https://www.youtube.com/watch?v=w3Dg8LTmtTo>

1. Tools-> firebase-> authentication -> email & pass auth
2. Connect to firebase
3. Firebase -> Add realtime database -> add realtime database to app
4. Firebase -> Add storage
5. In firebase website go to authentication -> enable phone
6. Firebase website -> database -> realtime database -> rules -> true –RW
7. Android studio -> new empty activity – RegisterActivity
8. Android studio -> new empty activity – OTPActivity
9. ChatActivity.java

package com.example.whatsapp;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.Context;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.ChildEventListener;  
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.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ChatActivity extends AppCompatActivity {  
  
 private String messageReceiverId, messageReceiverName, messageReceiverImage, messageSenderId;  
 private TextView userName, userLastSeen;  
 private CircleImageView userImage;  
 private Toolbar ChatToolBar;  
 private ImageButton SendMessageButton;  
 private EditText MessageInputText;  
  
 // firebase  
 private FirebaseAuth mAuth;  
 private DatabaseReference RootRef;  
  
 private final List<Messages> messagesList = new ArrayList<>();  
 private LinearLayoutManager linearLayoutManager;  
 private MessageAdapter messageAdapter;  
  
 private RecyclerView userMessagesList;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_chat*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 messageSenderId = mAuth.getCurrentUser().getUid();  
 RootRef = FirebaseDatabase.*getInstance*().getReference();  
  
 messageReceiverId = getIntent().getExtras().get("visit\_user\_id").toString();  
 messageReceiverName = getIntent().getExtras().get("visit\_user\_name").toString();  
 messageReceiverImage = getIntent().getExtras().get("visit\_image").toString();  
  
 InitializeControllers();  
  
 userName.setText(messageReceiverName);  
 if(!messageReceiverImage.equals("default\_image")){  
 Picasso.*get*().load(messageReceiverImage).placeholder(R.drawable.*profile\_image*).into(userImage);  
 }  
  
 }  
  
 private void InitializeControllers() {  
  
 ChatToolBar = (Toolbar) findViewById(R.id.*chat\_toolbar*);  
 setSupportActionBar(ChatToolBar);  
 ActionBar actionBar = getSupportActionBar();  
 actionBar.setDisplayHomeAsUpEnabled(true);  
 actionBar.setDisplayShowCustomEnabled(true);  
  
 LayoutInflater layoutInflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 View actionBarView = layoutInflater.inflate(R.layout.*custom\_chat\_bar*, null);  
 actionBar.setCustomView(actionBarView);  
  
 userImage = (CircleImageView) findViewById(R.id.*custom\_profile\_image*);  
 userName = (TextView) findViewById(R.id.*custom\_profile\_name*);  
 userLastSeen = (TextView) findViewById(R.id.*custom\_user\_last\_seen*);  
  
 SendMessageButton = (ImageButton) findViewById(R.id.*send\_message\_button*);  
 MessageInputText = (EditText) findViewById(R.id.*input\_message*);  
  
 messageAdapter = new MessageAdapter(messagesList);  
 userMessagesList = (RecyclerView) findViewById(R.id.*private\_messages\_list\_of\_users*);  
 linearLayoutManager = new LinearLayoutManager(this);  
 userMessagesList.setLayoutManager(linearLayoutManager);  
 userMessagesList.setAdapter(messageAdapter);  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 RootRef.child("Messages").child(messageSenderId).child(messageReceiverId)  
 .addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
 Messages messages = dataSnapshot.getValue(Messages.class);  
 messagesList.add(messages);  
 messageAdapter.notifyDataSetChanged();  
 userMessagesList.smoothScrollToPosition(userMessagesList.getAdapter().getItemCount());  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void setSendMessageButton(View view){  
 String messageText = MessageInputText.getText().toString();  
 if(!TextUtils.*isEmpty*(messageText)){  
 String messageSenderRef = "Messages/" + messageSenderId + "/" + messageReceiverId;  
 String messageReceiverRef = "Messages/" + messageReceiverId + "/" + messageSenderId;  
  
 DatabaseReference userMessageKey = RootRef.child("Messages")  
 .child(messageSenderId).child(messageReceiverId).push();  
 String messagePushId = userMessageKey.getKey();  
  
 Map messageTextBody = new HashMap();  
 messageTextBody.put("message", messageText);  
 messageTextBody.put("type", "text");  
 messageTextBody.put("from", messageSenderId);  
  
 Map messageBodyDetails = new HashMap();  
 messageBodyDetails.put(messageSenderRef + "/" + messagePushId, messageTextBody);  
 messageBodyDetails.put(messageReceiverRef + "/" + messagePushId, messageTextBody);  
  
 RootRef.updateChildren(messageBodyDetails).addOnCompleteListener(new OnCompleteListener() {  
 @Override  
 public void onComplete(@NonNull Task task) {  
 if(task.isSuccessful()){  
// Toast.makeText(ChatActivity.this, "Message Sent Successfully", Toast.LENGTH\_SHORT).show();  
 }else {  
 Toast.*makeText*(ChatActivity.this, "Message Not sent", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 MessageInputText.setText("");  
 }  
 });  
 }else {  
 Toast.*makeText*(this, "Please enter a message", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

1. Activity\_chat.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@android:color/darker\_gray"  
 tools:context=".ChatActivity">  
  
 <include  
 android:id="@+id/chat\_toolbar"  
 layout="@layout/app\_bar\_layout"  
 >  
 </include>  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/private\_messages\_list\_of\_users"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/chat\_toolbar"  
 android:layout\_above="@+id/chat\_linear\_layout"  
 >  
  
 </androidx.recyclerview.widget.RecyclerView>  
  
 <LinearLayout  
 android:id="@+id/chat\_linear\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentBottom="true"  
 android:background="@android:color/background\_light"  
 android:orientation="horizontal"  
 android:weightSum="10"  
 >  
 <EditText  
 android:id="@+id/input\_message"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="9"  
 android:hint="@string/type\_message\_here"  
 android:padding="15dp"  
 />  
 <ImageButton  
 android:id="@+id/sent\_message\_button"  
 android:layout\_weight="1"  
 android:layout\_width="60dp"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/send\_message"  
 android:onClick="setSendMessageButton"  
 />  
 </LinearLayout>  
</RelativeLayout>

1. Fragment\_chats.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ChatsFragment">  
  
 <!-- *TODO: Update blank fragment layout* -->  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/chats\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 />  
  
</RelativeLayout>

1. Receiver\_message\_layout.xml

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android">  
 <solid android:color="#FFFFFF">  
  
 </solid>  
  
 <corners android:radius="15dp"></corners>  
</shape>

1. Sender\_message\_layout.xml

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android">  
 <solid android:color="#E0FFC7"></solid>  
 <corners android:radius="15dp"></corners>  
</shape>

1. Custom\_message\_layout.xml
2. <?xml version="1.0" encoding="utf-8"?>  
   <RelativeLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content">  
    <de.hdodenhof.circleimageview.CircleImageView  
    android:id="@+id/message\_profile\_image"  
    android:layout\_width="45dp"  
    android:layout\_height="45dp"  
    android:src="@drawable/profile\_image"  
    android:layout\_marginTop="10dp"  
    />  
     
    <TextView  
    android:id="@+id/receiver\_message\_text"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:layout\_alignParentTop="true"  
    android:layout\_marginTop="10dp"  
    android:layout\_toEndOf="@+id/message\_profile\_image"  
    android:background="@drawable/receiver\_message\_layout"  
    android:text="@string/receiver\_messages\_here"  
    android:padding="10dp"  
    android:textSize="14dp"  
    android:textStyle="bold"  
    android:textColor="@android:color/black"  
    />  
    <TextView  
    android:id="@+id/sender\_message\_text"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:layout\_alignParentEnd="true"  
    android:layout\_marginTop="10dp"  
    android:layout\_alignParentTop="@+id/receiver\_message\_text"  
    android:background="@drawable/sender\_message\_layout"  
    android:text="@string/sender\_messages"  
    android:textSize="14dp"  
    android:padding="10dp"  
    android:textStyle="bold"  
    android:textColor="@android:color/black"  
    />  
     
   </RelativeLayout>
3. Messages.java

package com.example.whatsapp;  
  
public class Messages {  
 private String from, message, type;  
  
 public Messages(String from, String message, String type) {  
 this.from = from;  
 this.message = message;  
 this.type = type;  
 }  
  
 public Messages() {  
 }  
  
 public String getFrom() {  
 return from;  
 }  
  
 public void setFrom(String from) {  
 this.from = from;  
 }  
  
 public String getMessage() {  
 return message;  
 }  
  
 public void setMessage(String message) {  
 this.message = message;  
 }  
  
 public String getType() {  
 return type;  
 }  
  
 public void setType(String type) {  
 this.type = type;  
 }  
}

1. MessageAdapter.java

package com.example.whatsapp;  
  
import android.os.Message;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
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;  
import com.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class MessageAdapter extends RecyclerView.Adapter<MessageAdapter.MessageViewHolder> {  
 private List<Messages> userMessageList;  
  
 private FirebaseAuth mAuth;  
 private DatabaseReference usersRef;  
  
 public MessageAdapter(List<Messages> userMessageList){  
 this.userMessageList = userMessageList;  
 }  
  
 @NonNull  
 @Override  
 public MessageViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*custom\_message\_layout*, parent, false);  
 mAuth = FirebaseAuth.*getInstance*();  
 return new MessageViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull final MessageViewHolder holder, int position) {  
 String messageSenderId = mAuth.getCurrentUser().getUid();  
 Messages messages = userMessageList.get(position);  
 String fromUserId = messages.getFrom();  
 String fromMessageType = messages.getType();  
 usersRef = FirebaseDatabase.*getInstance*().getReference().child("Users").child(fromUserId);  
 usersRef.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild("image")){  
 String receiverImage = dataSnapshot.child("image").getValue().toString();  
 Picasso.*get*().load(receiverImage).placeholder(R.drawable.*profile\_image*).into(holder.receiverProfileImage);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 if(fromMessageType.equals("text")){  
 holder.receiverMessageText.setVisibility(View.*INVISIBLE*);  
 holder.receiverProfileImage.setVisibility(View.*INVISIBLE*);  
 holder.senderMessageText.setVisibility(View.*INVISIBLE*);  
  
 if(fromUserId.equals(messageSenderId)){  
 holder.senderMessageText.setVisibility(View.*VISIBLE*);  
 holder.senderMessageText.setBackgroundResource(R.drawable.*sender\_message\_layout*);  
 holder.senderMessageText.setText(messages.getMessage());  
 }  
 else {  
  
 holder.receiverProfileImage.setVisibility(View.*VISIBLE*);  
 holder.receiverMessageText.setVisibility(View.*VISIBLE*);  
  
 holder.receiverMessageText.setBackgroundResource(R.drawable.*receiver\_message\_layout*);  
 holder.receiverMessageText.setText(messages.getMessage());  
 }  
 }  
 }  
  
 @Override  
 public int getItemCount() {  
 return userMessageList.size();  
 }  
  
 public class MessageViewHolder extends RecyclerView.ViewHolder{  
  
 public TextView senderMessageText, receiverMessageText;  
 public CircleImageView receiverProfileImage;  
  
 public MessageViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 senderMessageText = (TextView) itemView.findViewById(R.id.*sender\_message\_text*);  
 receiverMessageText = (TextView) itemView.findViewById(R.id.*receiver\_message\_text*);  
 receiverProfileImage = (CircleImageView) itemView.findViewById(R.id.*message\_profile\_image*);  
 }  
 }  
}

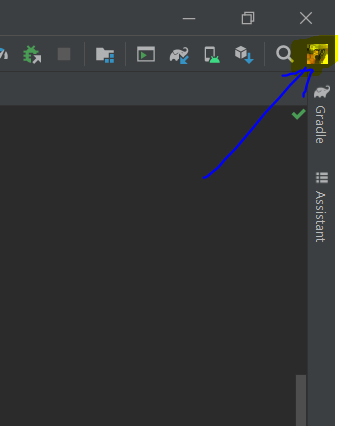
1. Custom\_chat\_bar.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/colorPrimary"  
 >  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/custom\_profile\_image"  
 android:layout\_width="47dp"  
 android:layout\_height="47dp"  
 android:src="@drawable/profile\_image"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentTop="true"  
 android:cropToPadding="true"  
 android:layout\_marginRight="10dp"  
 />  
  
 <TextView  
 android:id="@+id/custom\_profile\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:text="Display Username"  
 android:textColor="@android:color/background\_light"  
 android:textStyle="bold"  
 android:textSize="18dp"  
 />  
 <TextView  
 android:id="@+id/custom\_user\_last\_seen"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/custom\_profile\_name"  
 android:text="Last Seen"  
 android:textColor="@android:color/background\_light"  
 android:textStyle="bold"  
 />  
</RelativeLayout>

## Chat APP

<https://www.youtube.com/watch?v=XJvd8b3-Un4&list=PLGCjwl1RrtcQ3o2jmZtwu2wXEA4OIIq53&index=2>

1. Create project
2. Sign in android studio top right
3. Add firebase auth to app



1. Tools->firebase
2. Authentication-> email & password authentication->connect to firebase
3. It will open firebase in browser create new project
4. Add firebase authentication to app – xlick button in android studio
5. Colors.xml

<color name="colorPrimary">#673bb7</color>  
<color name="colorPrimaryDark">#512da7</color>  
<color name="colorAccent">#ff5521</color>

1. Style.xml – noaction bar

<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">

1. Create new empty activity – AtartActivity
2. Open firebase dashboard in browser
3. Authentication->sign-in method->Email/Password – enable
4. Follow the steps in firebase assistant in android studio
5. 3- check current auth state
6. MainActivity.java

package com.example.firebasechat;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
public class MainActivity extends AppCompatActivity {  
  
 private FirebaseAuth mAuth;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 }  
  
 public void onStart() {  
 super.onStart();  
 // Check if user is signed in (non-null) and update UI accordingly.  
 FirebaseUser currentUser = mAuth.getCurrentUser();  
 if(currentUser==null){  
 Intent intent = new Intent(MainActivity.this, StartActivity.class);  
 startActivity(intent);  
 finish();  
 }  
 }  
}

## Register activity

1. Create new activity RegisterActivity
2. Gradle implement

implementation 'com.android.support:support-compat:28.0.0'  
implementation 'com.android.support:design:28.0.0'

1. Activity\_register.xml

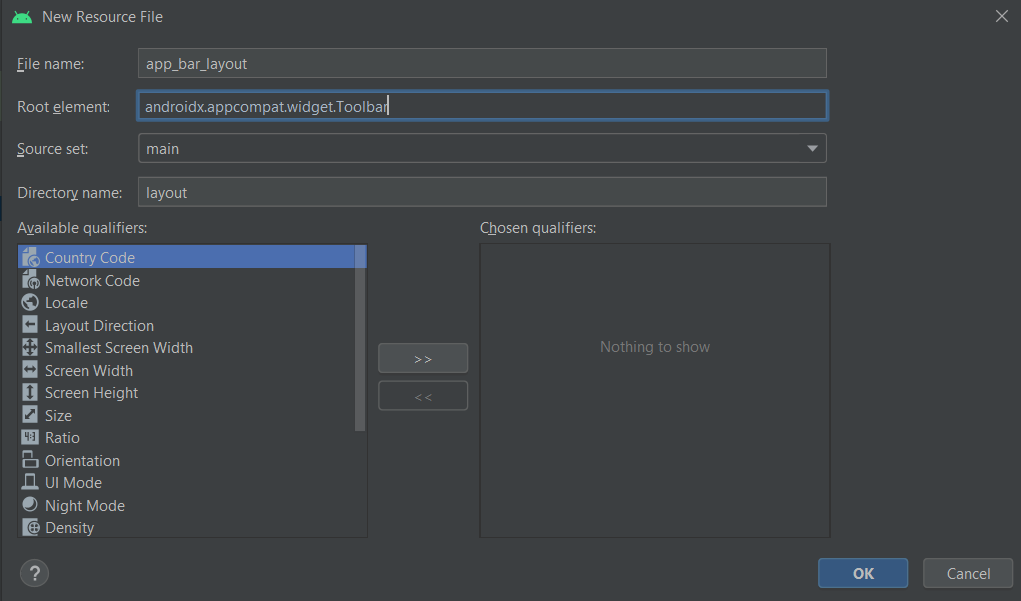
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="10dp"  
 tools:context=".RegisterActivity">  
  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/reg\_display\_name"  
 android:layout\_width="409dp"  
 android:layout\_height="wrap\_content"  
 tools:layout\_editor\_absoluteX="1dp"  
 tools:layout\_editor\_absoluteY="255dp">  
  
 <com.google.android.material.textfield.TextInputEditText  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Display Name" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/reg\_email"  
 android:layout\_width="409dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="@+id/reg\_email"  
 tools:layout\_editor\_absoluteX="1dp">  
  
 <com.google.android.material.textfield.TextInputEditText  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Email"  
 app:layout\_constraintBottom\_toTopOf="@+id/textInputLayout3"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textInputLayout2" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/reg\_password"  
 android:layout\_width="409dp"  
 android:layout\_height="wrap\_content"  
 tools:layout\_editor\_absoluteX="1dp"  
 tools:layout\_editor\_absoluteY="423dp">  
  
 <com.google.android.material.textfield.TextInputEditText  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Password" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <Button  
 android:id="@+id/reg\_btn\_regacc"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="end"  
 android:onClick="reg\_btn\_regaccClick"  
 android:text="@string/create\_account" />  
</LinearLayout>

1. Firebase assistant in android studio
2. Authentication-> 4- signup new user
3. RegisterActivity,java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.textfield.TextInputLayout;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
public class RegisterActivity extends AppCompatActivity {  
 private TextInputLayout mDisplayName, mEmail, mPassword;  
 private FirebaseAuth mAuth;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_register*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 mDisplayName = (TextInputLayout) findViewById(R.id.*reg\_display\_name*);  
 mEmail = (TextInputLayout) findViewById(R.id.*reg\_email*);  
 mPassword = (TextInputLayout) findViewById(R.id.*reg\_password*);  
 }  
  
 public void reg\_btn\_regaccClick(View view){  
 String display\_name = mDisplayName.getEditText().getText().toString();  
 String email = mEmail.getEditText().getText().toString();  
 String password = mPassword.getEditText().getText().toString();  
 register\_user(display\_name, email, password);  
 }  
  
 private void register\_user(String display\_name, String email, String password) {  
 mAuth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 // Sign in success, update UI with the signed-in user's information  
 Intent mainIntent = new Intent(RegisterActivity.this, MainActivity.class);  
 startActivity(mainIntent);  
 finish();  
 } else {  
 // If sign in fails, display a message to the user.  
 Toast.*makeText*(RegisterActivity.this, "Authentication failed.",  
 Toast.*LENGTH\_SHORT*).show();  
  
 }  
  
 // ...  
 }  
 });  
 }  
}

## Set Toolbar

1. Create resource file



<?xml version="1.0" encoding="utf-8"?>  
<androidx.appcompat.widget.Toolbar xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/main\_app\_bar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@color/colorPrimary"  
 android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">  
  
</androidx.appcompat.widget.Toolbar>

1. Activity\_main.xml

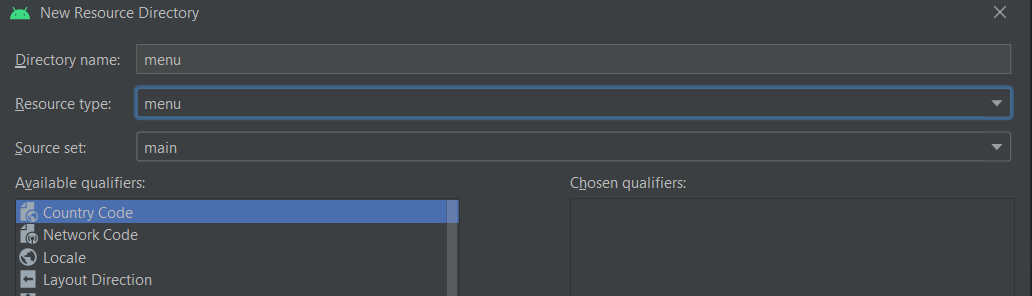
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <include layout="@layout/app\_bar\_layout" android:id="@+id/main\_page\_toolbar" />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

1. MainActivity.java

package com.example.firebasechat;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import androidx.appcompat.widget.Toolbar;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
public class MainActivity extends AppCompatActivity {  
  
 private FirebaseAuth mAuth;  
 private Toolbar mToolbar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolbar = (Toolbar) findViewById(R.id.*main\_page\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Chat");  
 }  
  
 }  
  
  
  
 public void onStart() {  
 super.onStart();  
 // Check if user is signed in (non-null) and update UI accordingly.  
 FirebaseUser currentUser = mAuth.getCurrentUser();  
 if(currentUser==null){  
 Intent intent = new Intent(MainActivity.this, StartActivity.class);  
 startActivity(intent);  
 finish();  
 }  
 }  
}

## Menu items

1. Right click res-> new ->Layout resource file



1. Right click menu folder->new -> menu resource file = menu\_main

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item  
 android:id="@+id/main\_settings\_btn"  
 android:title="Account Settings" />  
 <item  
 android:id="@+id/main\_all\_user\_btn"  
 android:title="All Users" />  
 <item  
 android:id="@+id/main\_logout\_btn"  
 android:title="@string/log\_out" />  
</menu>

1. MainActivity.java

package com.example.firebasechat;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
  
import androidx.appcompat.widget.Toolbar;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
public class MainActivity extends AppCompatActivity {  
  
 private FirebaseAuth mAuth;  
 private Toolbar mToolbar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolbar = (Toolbar) findViewById(R.id.*main\_page\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Chat");  
 }  
  
 }  
  
  
  
 public void onStart() {  
 super.onStart();  
 // Check if user is signed in (non-null) and update UI accordingly.  
 FirebaseUser currentUser = mAuth.getCurrentUser();  
 if(currentUser==null){  
 sendToStart();  
 }  
 }  
  
 private void sendToStart() {  
 Intent intent = new Intent(MainActivity.this, StartActivity.class);  
 startActivity(intent);  
 finish();  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 super.onCreateOptionsMenu(menu);  
 getMenuInflater().inflate(R.menu.*menu\_main*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 super.onOptionsItemSelected(item);  
 if(item.getItemId() == R.id.*main\_logout\_btn*){  
 FirebaseAuth.*getInstance*().signOut();  
 sendToStart();  
 }  
  
 return true;  
 }  
}

## Back toparent activity menu

1. Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".RegisterActivity">  
  
 <include layout="@layout/app\_bar\_layout" android:id="@+id/register\_toolbar" />  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="20dp"  
 android:orientation="vertical"  
 >  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/create\_new\_account"  
 android:textColor="@android:color/black"  
 android:textSize="24dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginBottom="10dp"  
 />  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/reg\_display\_name"  
 android:layout\_width="409dp"  
 android:layout\_height="wrap\_content"  
 tools:layout\_editor\_absoluteX="1dp"  
 tools:layout\_editor\_absoluteY="255dp">  
  
 <com.google.android.material.textfield.TextInputEditText  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Display Name" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/reg\_email"  
 android:layout\_width="409dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="@+id/reg\_email"  
 tools:layout\_editor\_absoluteX="1dp">  
  
 <com.google.android.material.textfield.TextInputEditText  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Email"  
 app:layout\_constraintBottom\_toTopOf="@+id/textInputLayout3"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textInputLayout2" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/reg\_password"  
 android:layout\_width="409dp"  
 android:layout\_height="wrap\_content"  
 tools:layout\_editor\_absoluteX="1dp"  
 tools:layout\_editor\_absoluteY="423dp">  
  
 <com.google.android.material.textfield.TextInputEditText  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Password" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <Button  
 android:id="@+id/reg\_btn\_regacc"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="20dp"  
 android:layout\_gravity="end"  
 android:backgroundTint="@color/colorAccent"  
 android:onClick="reg\_btn\_regaccClick"  
 android:textColor="@android:color/white"  
 android:text="@string/create\_account" />  
 </LinearLayout>  
</LinearLayout>

1. Android manifest.xml

<activity android:name=".RegisterActivity"  
 android:parentActivityName=".StartActivity"></activity>  
<activity android:name=".StartActivity" />

1. RegisterActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.textfield.TextInputLayout;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
public class RegisterActivity extends AppCompatActivity {  
 private TextInputLayout mDisplayName, mEmail, mPassword;  
 private FirebaseAuth mAuth;  
 private Toolbar mToolbar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_register*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolbar = (Toolbar) findViewById(R.id.*register\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Create Account");  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 }  
  
 mDisplayName = (TextInputLayout) findViewById(R.id.*reg\_display\_name*);  
 mEmail = (TextInputLayout) findViewById(R.id.*reg\_email*);  
 mPassword = (TextInputLayout) findViewById(R.id.*reg\_password*);  
 }  
  
 public void reg\_btn\_regaccClick(View view){  
 String display\_name = mDisplayName.getEditText().getText().toString();  
 String email = mEmail.getEditText().getText().toString();  
 String password = mPassword.getEditText().getText().toString();  
 register\_user(display\_name, email, password);  
 }  
  
 private void register\_user(String display\_name, String email, String password) {  
 mAuth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 // Sign in success, update UI with the signed-in user's information  
 Intent mainIntent = new Intent(RegisterActivity.this, MainActivity.class);  
 startActivity(mainIntent);  
 finish();  
 } else {  
 // If sign in fails, display a message to the user.  
 Toast.*makeText*(RegisterActivity.this, "Authentication failed.",  
 Toast.*LENGTH\_SHORT*).show();  
  
 }  
  
 // ...  
 }  
 });  
 }  
}

## Progress bar

1. RegisterActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.View;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.textfield.TextInputLayout;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
public class RegisterActivity extends AppCompatActivity {  
 private TextInputLayout mDisplayName, mEmail, mPassword;  
 private FirebaseAuth mAuth;  
 private Toolbar mToolbar;  
  
 private ProgressDialog mProgressDialog;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_register*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolbar = (Toolbar) findViewById(R.id.*register\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Create Account");  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 }  
  
 mProgressDialog = new ProgressDialog(this);  
  
 mDisplayName = (TextInputLayout) findViewById(R.id.*reg\_display\_name*);  
 mEmail = (TextInputLayout) findViewById(R.id.*reg\_email*);  
 mPassword = (TextInputLayout) findViewById(R.id.*reg\_password*);  
 }  
  
 public void reg\_btn\_regaccClick(View view){  
 String display\_name = mDisplayName.getEditText().getText().toString();  
 String email = mEmail.getEditText().getText().toString();  
 String password = mPassword.getEditText().getText().toString();  
 if(!TextUtils.*isEmpty*(display\_name) || !TextUtils.*isEmpty*(email) || !TextUtils.*isEmpty*(password)) {  
  
 mProgressDialog.setTitle("Registering User");  
 mProgressDialog.setMessage("Please wait while we creating your account!");  
 mProgressDialog.setCanceledOnTouchOutside(false);  
 mProgressDialog.show();  
 register\_user(display\_name, email, password);  
 }  
 }  
  
 private void register\_user(String display\_name, String email, String password) {  
 mAuth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 // Sign in success, update UI with the signed-in user's information  
 mProgressDialog.dismiss();  
 Intent mainIntent = new Intent(RegisterActivity.this, MainActivity.class);  
 startActivity(mainIntent);  
 finish();  
 } else {  
 mProgressDialog.hide();  
 // If sign in fails, display a message to the user.  
 Toast.*makeText*(RegisterActivity.this, "Authentication failed.",  
 Toast.*LENGTH\_SHORT*).show();  
  
 }  
  
 // ...  
 }  
 });  
 }  
}

## LoginActivity

1. Create loginActivity
2. LoginActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.textfield.TextInputLayout;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
  
public class LoginActivity extends AppCompatActivity {  
  
 private TextInputLayout mEmail, mPassword;  
 private Toolbar mToolbar;  
 private FirebaseAuth mAuth;  
 private ProgressDialog mProgressDialog;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolbar = (Toolbar) findViewById(R.id.*login\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Login");  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 }  
 mAuth = FirebaseAuth.*getInstance*();  
  
 mProgressDialog = new ProgressDialog(this);  
  
 mEmail = (TextInputLayout) findViewById(R.id.*reg\_email*);  
 mPassword = (TextInputLayout) findViewById(R.id.*reg\_password*);  
 }  
  
 public void login\_btn\_Click(View view){  
 String email = mEmail.getEditText().getText().toString();  
 String password = mPassword.getEditText().getText().toString();  
 if(!TextUtils.*isEmpty*(email) || !TextUtils.*isEmpty*(password)) {  
  
 mProgressDialog.setTitle("Logging In");  
 mProgressDialog.setMessage("Please wait while check your credentials!");  
 mProgressDialog.setCanceledOnTouchOutside(false);  
 mProgressDialog.show();  
 login\_user(email, password);  
 }  
 }  
  
 private void login\_user(String email, String password) {  
 mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
 mProgressDialog.dismiss();  
 Intent mainIntent = new Intent(LoginActivity.this, MainActivity.class );

mainIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);

startActivity(mainIntent);  
 finish();  
 }else {  
 mProgressDialog.hide();  
 // If sign in fails, display a message to the user.  
 Toast.*makeText*(LoginActivity.this, "Authentication failed.",  
 Toast.*LENGTH\_SHORT*).show();  
  
 }  
 }  
 });  
 }  
}

1. Activity\_login.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".LoginActivity">  
  
 <include layout="@layout/app\_bar\_layout" android:id="@+id/login\_toolbar" />  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="20dp"  
 android:orientation="vertical"  
 >  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/login"  
 android:textColor="@android:color/black"  
 android:textSize="24dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginBottom="10dp"  
 />  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/reg\_email"  
 android:layout\_width="409dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="@+id/reg\_email"  
 tools:layout\_editor\_absoluteX="1dp">  
  
 <com.google.android.material.textfield.TextInputEditText  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Email"  
 android:inputType="textEmailAddress"  
 app:layout\_constraintBottom\_toTopOf="@+id/textInputLayout3"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textInputLayout2" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/reg\_password"  
 android:layout\_width="409dp"  
 android:layout\_height="wrap\_content"  
 tools:layout\_editor\_absoluteX="1dp"  
 tools:layout\_editor\_absoluteY="423dp">  
  
 <com.google.android.material.textfield.TextInputEditText  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="textWebPassword"  
 android:hint="Password" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <Button  
 android:id="@+id/login\_btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="20dp"  
 android:layout\_gravity="end"  
 android:backgroundTint="@color/colorPrimary"  
 android:onClick="login\_btn\_Click"  
 android:textColor="@android:color/white"  
 android:text="@string/login" />  
 </LinearLayout>  
</LinearLayout>

1. Activity\_start.xml

<Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/login"  
 android:id="@+id/start\_btn\_login"  
 android:layout\_centerHorizontal="true"  
 android:onClick="start\_btn\_login\_Click"  
 android:layout\_below="@+id/start\_btn\_register"  
 />

1. startActivity.java

public void start\_btn\_login\_Click(View view){  
 Intent startIntent = new Intent(StartActivity.this, LoginActivity.class);  
 startActivity(startIntent);  
}

## Fragment View

1. create new fragment (RequestsFragment, ChatsFragment, FriendsFragment)– blank fragment
   1. uncheck – include fragment factory methords
2. Create SectionsPageAdapter.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentManager;  
import androidx.fragment.app.FragmentPagerAdapter;  
  
class SectionsPageAdapter extends FragmentPagerAdapter {  
 public SectionsPageAdapter(@NonNull FragmentManager fm) {  
 super(fm);  
 }  
  
 @NonNull  
 @Override  
 public Fragment getItem(int position) {  
 switch (position){  
 case 0:  
 RequestsFragment requestsFragment = new RequestsFragment();  
 return requestsFragment;  
 case 1:  
 ChatsFragment chatsFragment = new ChatsFragment();  
 return chatsFragment;  
 case 2:  
 FriendsFragment friendsFragment = new FriendsFragment();  
 return friendsFragment;  
 default:  
 return null;  
 }  
 }  
  
 @Override  
 public int getCount() {  
 return 3;  
 }

public CharSequence getPageTitle(int position){  
 switch (position){  
 case 0:  
 return "REQUESTS";  
 case 1:  
 return "CHATS";  
 case 2:  
 return "FRIENDS";  
 default:  
 return null;  
 }  
}

}

1. Main Activity.java

package com.example.firebasechat;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
  
import androidx.appcompat.widget.Toolbar;  
import androidx.viewpager.widget.ViewPager;  
  
import com.google.android.material.tabs.TabLayout;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
public class MainActivity extends AppCompatActivity {  
  
 private FirebaseAuth mAuth;  
 private Toolbar mToolbar;  
  
 private ViewPager mViewPager;  
 private SectionsPageAdapter mSectionsPageAdapter;  
 private TabLayout mTabLayout;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolbar = (Toolbar) findViewById(R.id.*main\_page\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Chat");  
 }  
  
 //TABS  
 mViewPager = (ViewPager) findViewById(R.id.*main\_tabPager*);  
 mSectionsPageAdapter = new SectionsPageAdapter(getSupportFragmentManager());  
 mViewPager.setAdapter(mSectionsPageAdapter);  
  
 mTabLayout =(TabLayout) findViewById(R.id.*main\_tabs*);  
 mTabLayout.setupWithViewPager(mViewPager);  
  
 }  
  
  
  
 public void onStart() {  
 super.onStart();  
 // Check if user is signed in (non-null) and update UI accordingly.  
 FirebaseUser currentUser = mAuth.getCurrentUser();  
 if(currentUser==null){  
 sendToStart();  
 }  
 }  
  
 private void sendToStart() {  
 Intent intent = new Intent(MainActivity.this, StartActivity.class);  
 startActivity(intent);  
 finish();  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 super.onCreateOptionsMenu(menu);  
 getMenuInflater().inflate(R.menu.*menu\_main*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 super.onOptionsItemSelected(item);  
 if(item.getItemId() == R.id.*main\_logout\_btn*){  
 FirebaseAuth.*getInstance*().signOut();  
 sendToStart();  
 }  
  
 return true;  
 }  
}

## SettingsActivity

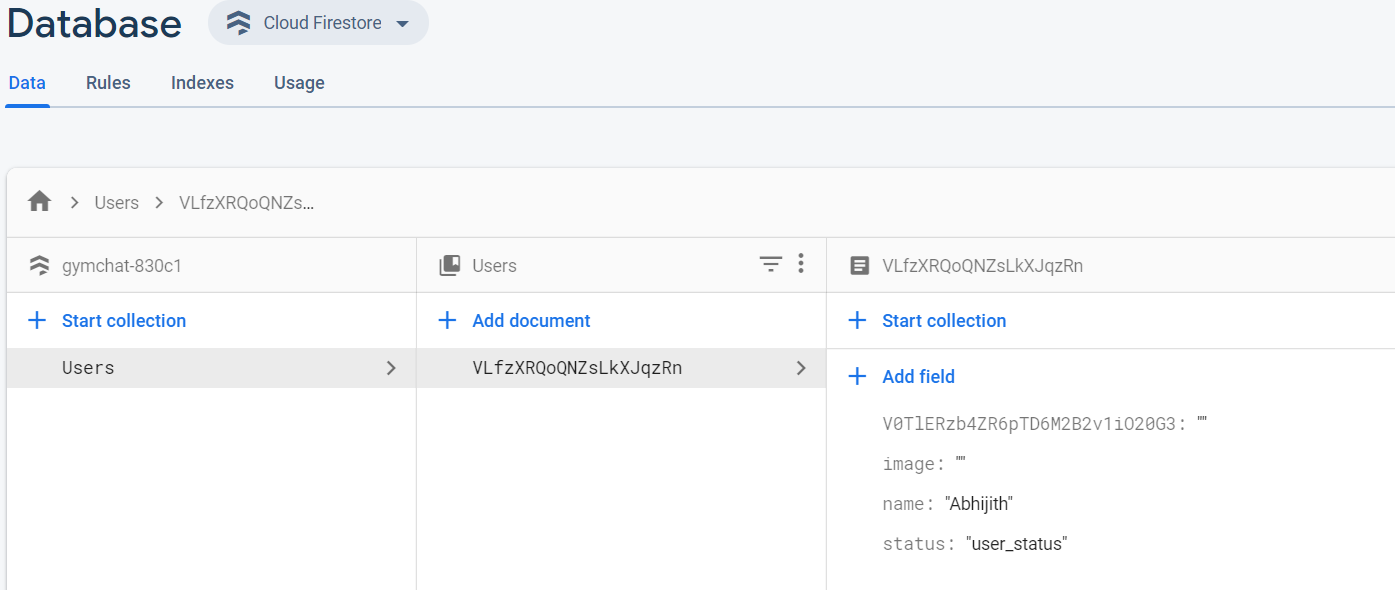
1. Create new empty activity SettingsActivity

Circle image view:- <https://github.com/hdodenhof/CircleImageView>

1. Create mipmap for profile pic
2. Create activity\_settings.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/colorPrimaryDark"  
 tools:context=".SettingsActivity">  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/settings\_image"  
 android:layout\_width="150dp"  
 android:layout\_height="150dp"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="150dp"  
 android:src="@mipmap/ic\_default\_avatar\_round"  
 />  
 <TextView  
 android:id="@+id/settings\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/display\_name"  
 android:textColor="@android:color/white"  
 android:textSize="20dp"  
 android:layout\_below="@+id/settings\_image"  
 android:layout\_centerHorizontal="true"  
 android:layout\_margin="5dp"  
 android:padding="10dp"  
/>  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/settings\_status"  
 android:textColor="@android:color/white"  
 android:textSize="15dp"  
 android:layout\_below="@+id/settings\_name"  
 android:layout\_centerHorizontal="true"  
 android:layout\_margin="5dp"  
 android:padding="10dp"  
 />  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:id="@+id/settings\_img\_btn"  
 android:layout\_marginBottom="150dp"  
 android:layout\_centerHorizontal="true"  
 android:text="@string/change\_image" />  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:backgroundTint="@color/colorAccent"  
 android:id="@+id/settings\_status\_btn"  
 android:textColor="@android:color/white"  
 android:layout\_marginBottom="100dp"  
 android:layout\_centerHorizontal="true"  
 android:text="@string/change\_status" />  
</RelativeLayout>

1. IN firebase create new database



1. In android studio -> tools -> firebase -> realtime database = add realtime database
2. Change rules to true

"rules": {

".read": true,

".write": true

}

1. RegisterActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.View;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.textfield.TextInputLayout;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
import java.util.HashMap;  
  
public class RegisterActivity extends AppCompatActivity {  
 private TextInputLayout mDisplayName, mEmail, mPassword;  
 private FirebaseAuth mAuth;  
 private Toolbar mToolbar;  
  
 private ProgressDialog mProgressDialog;  
  
 // database  
 private DatabaseReference mDatabase;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_register*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolbar = (Toolbar) findViewById(R.id.*register\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Create Account");  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 }  
  
 mProgressDialog = new ProgressDialog(this);  
  
 mDisplayName = (TextInputLayout) findViewById(R.id.*reg\_display\_name*);  
 mEmail = (TextInputLayout) findViewById(R.id.*reg\_email*);  
 mPassword = (TextInputLayout) findViewById(R.id.*reg\_password*);  
 }  
  
 public void reg\_btn\_regaccClick(View view){  
 String display\_name = mDisplayName.getEditText().getText().toString();  
 String email = mEmail.getEditText().getText().toString();  
 String password = mPassword.getEditText().getText().toString();  
 if(!TextUtils.*isEmpty*(display\_name) || !TextUtils.*isEmpty*(email) || !TextUtils.*isEmpty*(password)) {  
  
 mProgressDialog.setTitle("Registering User");  
 mProgressDialog.setMessage("Please wait while we creating your account!");  
 mProgressDialog.setCanceledOnTouchOutside(false);  
 mProgressDialog.show();  
 register\_user(display\_name, email, password);  
 }  
 }  
  
 private void register\_user(final String display\_name, String email, String password) {  
 mAuth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 // Sign in success, update UI with the signed-in user's information  
 FirebaseUser current\_user = FirebaseAuth.*getInstance*().getCurrentUser();  
 String uid = current\_user.getUid();  
 Toast.*makeText*(RegisterActivity.this, "uid"+ uid, Toast.*LENGTH\_SHORT*).show();  
 // Write a message to the database  
 mDatabase = FirebaseDatabase.*getInstance*().getReference();  
  
 HashMap<String, String> userMap = new HashMap<>();  
 userMap.put("name", display\_name);  
 userMap.put("status", "Hi there I'm using chat app");  
 userMap.put("image", "Default");  
 userMap.put("thumb\_image", "Default");  
 mDatabase.child("Users").child(uid).setValue(userMap).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 mProgressDialog.dismiss();  
 Intent mainIntent = new Intent(RegisterActivity.this, MainActivity.class);  
 mainIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(mainIntent);  
 finish();  
 }  
});

} else {  
 mProgressDialog.hide();  
 // If sign in fails, display a message to the user.  
 Toast.*makeText*(RegisterActivity.this, "Authentication failed.",  
 Toast.*LENGTH\_SHORT*).show();  
  
 }  
 }  
 });  
 }  
}

1. Add menu item setting – MainActivity.java

@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
 super.onOptionsItemSelected(item);  
 if(item.getItemId() == R.id.*main\_logout\_btn*){  
 FirebaseAuth.*getInstance*().signOut();  
 sendToStart();  
 }  
 if(item.getItemId() == R.id.*main\_settings\_btn*){  
 Intent intent = new Intent(MainActivity.this, SettingsActivity.class);  
 startActivity(intent);  
 }  
 return true;  
}

## Show user profile – retrieve data from firebase

SettingsActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
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;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class SettingsActivity extends AppCompatActivity {  
  
 private DatabaseReference mUserDatabase;  
 private FirebaseUser mCurrentUser;  
  
 // Android Layout  
 private CircleImageView mDisplayImage;  
 private TextView mName, mStatus;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_settings*);  
 mDisplayImage = (CircleImageView) findViewById(R.id.*settings\_image*);  
 mName = (TextView) findViewById(R.id.*settings\_name*);  
 mStatus = (TextView) findViewById(R.id.*settings\_status*);  
  
 mCurrentUser = FirebaseAuth.*getInstance*().getCurrentUser();  
  
 String current\_uid = mCurrentUser.getUid();  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(current\_uid);  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String name = dataSnapshot.child("name").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
 String status = dataSnapshot.child("status").getValue().toString();  
 String thumb\_image = dataSnapshot.child("thumb\_image").getValue().toString();  
  
 mName.setText(name);  
 mStatus.setText(status);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
}

## Status Activity

1. Create empty Activity - Status Activity
2. ManifestFile Include parentActivity

<activity  
 android:name=".StatusActivity"  
 android:parentActivityName=".SettingsActivity" />

1. Activity\_settings.xml

<Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:backgroundTint="@color/colorAccent"  
 android:id="@+id/settings\_status\_btn"  
 android:onClick="settings\_status\_btnClick"  
 android:textColor="@android:color/white"  
 android:layout\_marginBottom="100dp"  
 android:layout\_centerHorizontal="true"  
 android:text="@string/change\_status" />

1. SettingsActivity.java

public void settings\_status\_btnClick(View view){  
 String status\_value = mStatus.getText().toString();  
 Intent intent = new Intent(SettingsActivity.this, StatusActivity.class);  
 intent.putExtra("status\_value", status\_value);  
 startActivity(intent);  
  
}

1. StatusActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
import androidx.appcompat.widget.Toolbar;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.textfield.TextInputLayout;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
public class StatusActivity extends AppCompatActivity {  
  
 private Toolbar mToolbar;  
  
 private TextInputLayout mStatus;  
// Firebase  
 private DatabaseReference mStatusDatabase;  
 private FirebaseUser mCurrentUser;  
  
 // Progress  
 private ProgressDialog mProgress;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_status*);  
  
 // Firebase  
 mCurrentUser = FirebaseAuth.*getInstance*().getCurrentUser();  
 String current\_uid = mCurrentUser.getUid();  
 mStatusDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(current\_uid);  
  
 // Progress.  
 mProgress = new ProgressDialog(this);  
  
 mToolbar = (Toolbar) findViewById(R.id.*status\_appbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Account Status");  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
  
 mStatus = (TextInputLayout) findViewById(R.id.*status\_input*);  
 String status\_value = getIntent().getStringExtra("status\_value");  
 mStatus.getEditText().setText(status\_value);  
  
 }  
  
 public void status\_save\_btnClick(View view){  
 // Progress  
 mProgress.setTitle("Saving Changes");  
 mProgress.setMessage("Please wait while we saving changes");  
 mProgress.show();  
  
 String status = mStatus.getEditText().getText().toString();  
 mStatusDatabase.child("status").setValue(status).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 mProgress.dismiss();  
 }else {  
 Toast.*makeText*(StatusActivity.this, "There was an error while saving changes.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
}

activity\_status.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".StatusActivity">  
  
 <include layout="@layout/app\_bar\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/status\_appbar" />  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:orientation="vertical">  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:layout\_below="@+id/status\_appbar"  
 android:id="@+id/status\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/your\_status" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/status\_input"  
 android:id="@+id/status\_save\_btn"  
 android:onClick="status\_save\_btnClick"  
 android:text="@string/save\_changes"  
 android:backgroundTint="@color/colorAccent"  
 android:textColor="@android:color/white"  
 android:layout\_gravity="end" />  
 </LinearLayout>  
  
</LinearLayout>

## Pick an Image from Gallery

SettingsActivity.java

public void settings\_image\_btnClick(View view){  
 Intent intent = new Intent();  
 intent.setType("image/\*");  
 intent.setAction(Intent.*ACTION\_GET\_CONTENT*);  
 startActivityForResult(Intent.*createChooser*(intent, "SELECT IMAGE"), *GALLERY\_PICk*);  
  
}

activity\_settings.xml

<Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:id="@+id/settings\_img\_btn"  
 android:layout\_marginBottom="150dp"  
 android:onClick="settings\_image\_btnClick"  
 android:layout\_centerHorizontal="true"  
 android:text="@string/change\_image" />

## Crop Image

<https://github.com/ArthurHub/Android-Image-Cropper>

1. AndroidManifest.xml

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/>  
<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

1. Gradle dependency

api 'com.theartofdev.edmodo:android-image-cropper:2.8.+'

1. Add activity to androidmanifest.xml

<activity android:name="com.theartofdev.edmodo.cropper.CropImageActivity"  
 android:theme="@style/Base.Theme.AppCompat"/>

1. In Firebase web browser -> storage -> Create folder = profile\_images
2. In android studio -> firebase assistant -> add cloud storage to app
3. SettingsActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.preference.PreferenceManager;  
import android.view.View;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
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;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
import com.google.firebase.storage.UploadTask;  
import com.theartofdev.edmodo.cropper.CropImage;  
import com.theartofdev.edmodo.cropper.CropImageView;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class SettingsActivity extends AppCompatActivity {  
  
 private DatabaseReference mUserDatabase;  
 private FirebaseUser mCurrentUser;  
  
 // Android Layout  
 private CircleImageView mDisplayImage;  
 private TextView mName, mStatus;  
 public static final int *GALLERY\_PICk* = 1;  
 private StorageReference mImageStorage;  
 private String current\_uid;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_settings*);  
 mDisplayImage = (CircleImageView) findViewById(R.id.*settings\_image*);  
 mName = (TextView) findViewById(R.id.*settings\_name*);  
 mStatus = (TextView) findViewById(R.id.*settings\_status*);  
  
 mCurrentUser = FirebaseAuth.*getInstance*().getCurrentUser();  
 mImageStorage = FirebaseStorage.*getInstance*().getReference();  
  
 current\_uid = mCurrentUser.getUid();  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(current\_uid);  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String name = dataSnapshot.child("name").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
 String status = dataSnapshot.child("status").getValue().toString();  
 String thumb\_image = dataSnapshot.child("thumb\_image").getValue().toString();  
  
 mName.setText(name);  
 mStatus.setText(status);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void settings\_status\_btnClick(View view){  
 String status\_value = mStatus.getText().toString();  
 Intent intent = new Intent(SettingsActivity.this, StatusActivity.class);  
 intent.putExtra("status\_value", status\_value);  
 startActivity(intent);  
  
 }  
  
 public void settings\_image\_btnClick(View view){  
 /\*CropImage.activity()  
 .setGuidelines(CropImageView.Guidelines.ON)  
 .start(this);\*/  
 Intent intent = new Intent();  
 intent.setType("image/\*");  
 intent.setAction(Intent.*ACTION\_GET\_CONTENT*);  
 startActivityForResult(Intent.*createChooser*(intent, "SELECT IMAGE"), *GALLERY\_PICk*);  
  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 if(requestCode == *GALLERY\_PICk* && resultCode == *RESULT\_OK*){  
 Uri imageUri = data.getData();  
  
 CropImage.*activity*(imageUri)  
 .setAspectRatio(1, 1)  
 .start(this);  
  
 }  
  
 if (requestCode == CropImage.*CROP\_IMAGE\_ACTIVITY\_REQUEST\_CODE*) {  
 CropImage.ActivityResult result = CropImage.*getActivityResult*(data);  
 if (resultCode == *RESULT\_OK*) {  
 Uri resultUri = result.getUri();  
 StorageReference filepath = mImageStorage.child("profile\_images").child(current\_uid+".jpg");  
 filepath.putFile(resultUri).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(SettingsActivity.this, "Success", Toast.*LENGTH\_SHORT*).show();  
 }else {  
 Toast.*makeText*(SettingsActivity.this, "Error in uploading.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
  
 } else if (resultCode == CropImage.*CROP\_IMAGE\_ACTIVITY\_RESULT\_ERROR\_CODE*) {  
 Exception error = result.getError();  
 }  
 }  
 }  
}

## Upload Image

SettingActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.preference.PreferenceManager;  
import android.view.View;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
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;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
import com.google.firebase.storage.UploadTask;  
import com.theartofdev.edmodo.cropper.CropImage;  
import com.theartofdev.edmodo.cropper.CropImageView;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class SettingsActivity extends AppCompatActivity {  
  
 private DatabaseReference mUserDatabase;  
 private FirebaseUser mCurrentUser;  
  
 // Android Layout  
 private CircleImageView mDisplayImage;  
 private TextView mName, mStatus;  
 public static final int *GALLERY\_PICk* = 1;  
 private StorageReference mImageStorage;  
 private String current\_uid;  
  
 private ProgressDialog mProgress;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_settings*);  
 mDisplayImage = (CircleImageView) findViewById(R.id.*settings\_image*);  
 mName = (TextView) findViewById(R.id.*settings\_name*);  
 mStatus = (TextView) findViewById(R.id.*settings\_status*);  
  
 mCurrentUser = FirebaseAuth.*getInstance*().getCurrentUser();  
 mImageStorage = FirebaseStorage.*getInstance*().getReference();  
  
 mProgress = new ProgressDialog(this);  
  
 current\_uid = mCurrentUser.getUid();  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(current\_uid);  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String name = dataSnapshot.child("name").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
 String status = dataSnapshot.child("status").getValue().toString();  
 String thumb\_image = dataSnapshot.child("thumb\_image").getValue().toString();  
  
 mName.setText(name);  
 mStatus.setText(status);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void settings\_status\_btnClick(View view){  
 String status\_value = mStatus.getText().toString();  
 Intent intent = new Intent(SettingsActivity.this, StatusActivity.class);  
 intent.putExtra("status\_value", status\_value);  
 startActivity(intent);  
  
 }  
  
 public void settings\_image\_btnClick(View view){  
 /\*CropImage.activity()  
 .setGuidelines(CropImageView.Guidelines.ON)  
 .start(this);\*/  
 Intent intent = new Intent();  
 intent.setType("image/\*");  
 intent.setAction(Intent.*ACTION\_GET\_CONTENT*);  
 startActivityForResult(Intent.*createChooser*(intent, "SELECT IMAGE"), *GALLERY\_PICk*);  
  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 if(requestCode == *GALLERY\_PICk* && resultCode == *RESULT\_OK*){  
 Uri imageUri = data.getData();  
  
 CropImage.*activity*(imageUri)  
 .setAspectRatio(1, 1)  
 .start(this);  
  
 }  
  
 if (requestCode == CropImage.*CROP\_IMAGE\_ACTIVITY\_REQUEST\_CODE*) {  
 CropImage.ActivityResult result = CropImage.*getActivityResult*(data);  
 if (resultCode == *RESULT\_OK*) {  
  
 // Progress  
 mProgress.setTitle("Uploading Image...");  
 mProgress.setMessage("Please wait while we uploading image");  
 mProgress.setCanceledOnTouchOutside(false);  
 mProgress.show();  
  
  
 Uri resultUri = result.getUri();  
 StorageReference filepath = mImageStorage.child("profile\_images").child(current\_uid+".jpg");  
 filepath.putFile(resultUri).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {  
 if(task.isSuccessful()){  
 String download\_url = task.getResult().getStorage().getDownloadUrl().toString();  
 mUserDatabase.child("image").setValue(download\_url).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 mProgress.dismiss();  
 Toast.*makeText*(SettingsActivity.this, "Image uploaded successfully.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }else {  
 Toast.*makeText*(SettingsActivity.this, "Error in uploading.", Toast.*LENGTH\_SHORT*).show();  
 mProgress.dismiss();  
 }  
 }  
 });  
  
 } else if (resultCode == CropImage.*CROP\_IMAGE\_ACTIVITY\_RESULT\_ERROR\_CODE*) {  
 Exception error = result.getError();  
 }  
 }  
 }  
}

## Retrieve Image from firebase

<https://square.github.io/picasso/>

1. Gradle dependency

implementation 'com.squareup.picasso:picasso:2.71828'

1. SettingActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.preference.PreferenceManager;  
import android.view.View;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
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;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
import com.google.firebase.storage.UploadTask;  
import com.squareup.picasso.Picasso;  
import com.theartofdev.edmodo.cropper.CropImage;  
import com.theartofdev.edmodo.cropper.CropImageView;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class SettingsActivity extends AppCompatActivity {  
  
 private DatabaseReference mUserDatabase;  
 private FirebaseUser mCurrentUser;  
  
 // Android Layout  
 private CircleImageView mDisplayImage;  
 private TextView mName, mStatus;  
 public static final int *GALLERY\_PICk* = 1;  
 private StorageReference mImageStorage;  
 private String current\_uid;  
  
 private ProgressDialog mProgress;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_settings*);  
 mDisplayImage = (CircleImageView) findViewById(R.id.*settings\_image*);  
 mName = (TextView) findViewById(R.id.*settings\_name*);  
 mStatus = (TextView) findViewById(R.id.*settings\_status*);  
  
 mCurrentUser = FirebaseAuth.*getInstance*().getCurrentUser();  
 mImageStorage = FirebaseStorage.*getInstance*().getReference();  
  
 mProgress = new ProgressDialog(this);  
  
 current\_uid = mCurrentUser.getUid();  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(current\_uid);  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String name = dataSnapshot.child("name").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
 String status = dataSnapshot.child("status").getValue().toString();  
 String thumb\_image = dataSnapshot.child("thumb\_image").getValue().toString();  
  
 mName.setText(name);  
 mStatus.setText(status);  
 Picasso.*get*().load(image).into(mDisplayImage);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void settings\_status\_btnClick(View view){  
 String status\_value = mStatus.getText().toString();  
 Intent intent = new Intent(SettingsActivity.this, StatusActivity.class);  
 intent.putExtra("status\_value", status\_value);  
 startActivity(intent);  
  
 }  
  
 public void settings\_image\_btnClick(View view){  
 /\*CropImage.activity()  
 .setGuidelines(CropImageView.Guidelines.ON)  
 .start(this);\*/  
 Intent intent = new Intent();  
 intent.setType("image/\*");  
 intent.setAction(Intent.*ACTION\_GET\_CONTENT*);  
 startActivityForResult(Intent.*createChooser*(intent, "SELECT IMAGE"), *GALLERY\_PICk*);  
  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 if(requestCode == *GALLERY\_PICk* && resultCode == *RESULT\_OK*){  
 Uri imageUri = data.getData();  
  
 CropImage.*activity*(imageUri)  
 .setAspectRatio(1, 1)  
 .start(this);  
  
 }  
  
 if (requestCode == CropImage.*CROP\_IMAGE\_ACTIVITY\_REQUEST\_CODE*) {  
 CropImage.ActivityResult result = CropImage.*getActivityResult*(data);  
 if (resultCode == *RESULT\_OK*) {  
  
 // Progress  
 mProgress.setTitle("Uploading Image...");  
 mProgress.setMessage("Please wait while we uploading image");  
 mProgress.setCanceledOnTouchOutside(false);  
 mProgress.show();  
  
  
 Uri resultUri = result.getUri();  
 final StorageReference filepath = mImageStorage.child("profile\_images").child(current\_uid+".jpg");  
 filepath.putFile(resultUri).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {  
 if(task.isSuccessful()){  
// String download\_url = task.getResult().getStorage().getDownloadUrl().toString();  
 filepath.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {  
 @Override  
 public void onSuccess(Uri uri) {  
 String download\_url=uri.toString();  
 mUserDatabase.child("image").setValue(download\_url).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 mProgress.dismiss();  
 Toast.*makeText*(SettingsActivity.this, "Image uploaded successfully.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 });  
 /\*mUserDatabase.child("image").setValue(download\_url).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 mProgress.dismiss();  
 Toast.makeText(SettingsActivity.this, "Image uploaded successfully.", Toast.LENGTH\_SHORT).show();  
 }  
 }  
 });\*/  
 }else {  
 Toast.*makeText*(SettingsActivity.this, "Error in uploading.", Toast.*LENGTH\_SHORT*).show();  
 mProgress.dismiss();  
 }  
 }  
 });  
  
 } else if (resultCode == CropImage.*CROP\_IMAGE\_ACTIVITY\_RESULT\_ERROR\_CODE*) {  
 Exception error = result.getError();  
 }  
 }  
 }  
}

## User RecycleView

1. Create new activity - UserActivity
2. Android Manifest

<activity android:name=".UsersActivity"  
 android:parentActivityName=".MainActivity"></activity>

1. Create new layout resource file = users\_single\_layout – RelativeLayout

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:layout\_width="64dp"  
 android:layout\_height="64dp"  
 android:layout\_marginTop="15dp"  
 android:layout\_marginBottom="15dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="15dp"  
 android:id="@+id/user\_single\_image"  
 android:src="@mipmap/ic\_default\_avatar\_round"/>  
 <TextView  
 android:layout\_marginTop="15dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@android:color/black"  
 android:id="@+id/user\_single\_name"  
 android:text="Display Name"  
 android:textStyle="bold"  
 android:textSize="18dp"  
 android:layout\_toRightOf="@+id/user\_single\_image"/>  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="User default Status"  
 android:textSize="18dp"  
 android:id="@+id/user\_single\_status"  
 android:layout\_below="@+id/user\_single\_name"  
 android:layout\_toRightOf="@+id/user\_single\_image"/>  
</RelativeLayout>

1. Add gradle firebase ui

<https://github.com/firebase/FirebaseUI-Android>

implementation 'com.firebaseui:firebase-ui-database:6.2.1'

1. Create new Java class = User

package com.example.firebasechat;  
  
public class User {  
 public String name;  
 public String image;  
 public String status;  
  
 public User() {  
 }  
  
 public User(String name, String image, String status) {  
 this.name = name;  
 this.image = image;  
 this.status = status;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getImage() {  
 return image;  
 }  
  
 public void setImage(String image) {  
 this.image = image;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
 public void setStatus(String status) {  
 this.status = status;  
 }  
}

### Compress image

<https://github.com/Tourenathan-G5organisation/SiliCompressor>

<https://github.com/zetbaitsu/Compressor>

<https://github.com/zetbaitsu/Compressor/blob/master/README_v2.md>

Upload Bitmap Image:-

SettingsActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.graphics.drawable.BitmapDrawable;  
import android.net.Uri;  
import android.os.Bundle;  
import android.preference.PreferenceManager;  
import android.view.View;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
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;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
import com.google.firebase.storage.UploadTask;  
import com.squareup.picasso.Picasso;  
import com.theartofdev.edmodo.cropper.CropImage;  
import com.theartofdev.edmodo.cropper.CropImageView;  
  
import java.io.ByteArrayOutputStream;  
import java.io.File;  
import java.io.IOException;  
import java.util.HashMap;  
import java.util.Map;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
import id.zelory.compressor.Compressor;  
  
public class SettingsActivity extends AppCompatActivity {  
  
 private DatabaseReference mUserDatabase;  
 private FirebaseUser mCurrentUser;  
  
 // Android Layout  
 private CircleImageView mDisplayImage;  
 private TextView mName, mStatus;  
 public static final int *GALLERY\_PICk* = 1;  
 private StorageReference mImageStorage;  
 private String current\_uid;  
  
 private ProgressDialog mProgress;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_settings*);  
 mDisplayImage = (CircleImageView) findViewById(R.id.*settings\_image*);  
 mName = (TextView) findViewById(R.id.*settings\_name*);  
 mStatus = (TextView) findViewById(R.id.*settings\_status*);  
  
 mCurrentUser = FirebaseAuth.*getInstance*().getCurrentUser();  
 mImageStorage = FirebaseStorage.*getInstance*().getReference();  
  
 mProgress = new ProgressDialog(this);  
  
 current\_uid = mCurrentUser.getUid();  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(current\_uid);  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String name = dataSnapshot.child("name").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
 String status = dataSnapshot.child("status").getValue().toString();  
 String thumb\_image = dataSnapshot.child("thumb\_image").getValue().toString();  
  
 mName.setText(name);  
 mStatus.setText(status);  
 if(!image.equals("Default")) {  
 Picasso.*get*().load(image).placeholder(R.mipmap.*ic\_default\_avatar\_round*).into(mDisplayImage);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void settings\_status\_btnClick(View view){  
 String status\_value = mStatus.getText().toString();  
 Intent intent = new Intent(SettingsActivity.this, StatusActivity.class);  
 intent.putExtra("status\_value", status\_value);  
 startActivity(intent);  
  
 }  
  
 public void settings\_image\_btnClick(View view){  
 /\*CropImage.activity()  
 .setGuidelines(CropImageView.Guidelines.ON)  
 .start(this);\*/  
 Intent intent = new Intent();  
 intent.setType("image/\*");  
 intent.setAction(Intent.*ACTION\_GET\_CONTENT*);  
 startActivityForResult(Intent.*createChooser*(intent, "SELECT IMAGE"), *GALLERY\_PICk*);  
  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 if(requestCode == *GALLERY\_PICk* && resultCode == *RESULT\_OK*){  
 Uri imageUri = data.getData();  
  
 CropImage.*activity*(imageUri)  
 .setAspectRatio(1, 1)  
 .start(this);  
  
 }  
  
 if (requestCode == CropImage.*CROP\_IMAGE\_ACTIVITY\_REQUEST\_CODE*) {  
 CropImage.ActivityResult result = CropImage.*getActivityResult*(data);  
 if (resultCode == *RESULT\_OK*) {  
  
 // Progress  
 mProgress.setTitle("Uploading Image...");  
 mProgress.setMessage("Please wait while we uploading image");  
 mProgress.setCanceledOnTouchOutside(false);  
 mProgress.show();  
  
  
 Uri resultUri = result.getUri();  
 File thumpFilePath = new File(resultUri.getPath());  
 // Compress image  
 Bitmap thumbImage = null;  
 try {  
 thumbImage = new Compressor(this)  
 .setMaxWidth(200)  
 .setMaxHeight(200)  
 .setQuality(75)  
 .compressToBitmap(thumpFilePath);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 //Upload bit map to server  
 ByteArrayOutputStream baos = new ByteArrayOutputStream();  
 thumbImage.compress(Bitmap.CompressFormat.*JPEG*, 100, baos);  
 final byte[] thumb\_byte = baos.toByteArray();  
  
 final StorageReference filepath = mImageStorage.child("profile\_images").child(current\_uid+".jpg");  
 final StorageReference thumb\_filepath = mImageStorage.child("profile\_images").child("thumbs").child(current\_uid+".jpg");  
  
 filepath.putFile(resultUri).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {  
 if(task.isSuccessful()){  
  
// String download\_url = task.getResult().getStorage().getDownloadUrl().toString();  
 filepath.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {  
 @Override  
 public void onSuccess(Uri uri) {  
 final String download\_url=uri.toString();  
  
 final UploadTask uploadTask = thumb\_filepath.putBytes(thumb\_byte);  
 uploadTask.addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> thumpb\_task) {  
// String thumb\_download\_url = thumpb\_task.getResult().getStorage().getDownloadUrl().toString();  
 if(thumpb\_task.isSuccessful()){  
 thumb\_filepath.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {  
 @Override  
 public void onSuccess(Uri uri) {  
 String thumb\_download\_url = uri.toString();  
 Map updateHashmap = new HashMap();  
 updateHashmap.put("image", download\_url);  
 updateHashmap.put("thumb\_image", thumb\_download\_url);  
  
 mUserDatabase.updateChildren(updateHashmap).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 mProgress.dismiss();  
 Toast.*makeText*(SettingsActivity.this, "Image uploaded successfully.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 });  
  
 }else{  
 Toast.*makeText*(SettingsActivity.this, "Error in uploading thumbnail.", Toast.*LENGTH\_SHORT*).show();  
 mProgress.dismiss();  
 }  
 }  
 });  
  
 }  
 });  
  
 }else {  
 Toast.*makeText*(SettingsActivity.this, "Error in uploading.", Toast.*LENGTH\_SHORT*).show();  
 mProgress.dismiss();  
 }  
 }  
 });  
  
 } else if (resultCode == CropImage.*CROP\_IMAGE\_ACTIVITY\_RESULT\_ERROR\_CODE*) {  
 Exception error = result.getError();  
 }  
 }  
 }  
}

UsersActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
  
import android.content.Context;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.firebase.ui.database.FirebaseListOptions;  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.Query;  
import com.squareup.picasso.Picasso;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class UsersActivity extends AppCompatActivity {  
  
 private Toolbar mToolbar;  
 private RecyclerView mUsersList;  
 private DatabaseReference mUserDatabase;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_users*);  
  
 mToolbar = (Toolbar) findViewById(R.id.*users\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("All Users");  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users");  
  
 mUsersList = (RecyclerView) findViewById(R.id.*users\_list*);  
 mUsersList.setHasFixedSize(true);  
 mUsersList.setLayoutManager(new LinearLayoutManager(this));  
  
  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 Query query = FirebaseDatabase.*getInstance*()  
 .getReference()  
 .child("Users")  
 .limitToLast(50);  
 FirebaseRecyclerOptions<User> options =  
 new FirebaseRecyclerOptions.Builder<User>()  
 .setQuery(mUserDatabase, User.class)  
 .build();  
 FirebaseRecyclerAdapter<User, UsersViewHolder> adapter = new FirebaseRecyclerAdapter<User, UsersViewHolder>(options) {  
 @Override  
 protected void onBindViewHolder(@NonNull UsersViewHolder holder, int position, @NonNull User model) {  
 holder.setName(model.getName());  
 holder.setStatus(model.getStatus());  
 holder.setImage(model.getThumb\_image(), getApplicationContext());  
 }  
  
 @NonNull  
 @Override  
 public UsersViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*users\_single\_layout*, parent, false);  
 UsersViewHolder viewHolder = new UsersViewHolder(view);  
 return viewHolder;  
 }  
 };  
  
 mUsersList.setAdapter(adapter);  
 adapter.startListening();  
  
 }  
  
  
  
  
 public static class UsersViewHolder extends RecyclerView.ViewHolder{  
 View mView;  
  
 public UsersViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 mView = itemView;  
 }  
  
 public void setName(String name) {  
 TextView userNameView = (TextView) mView.findViewById(R.id.*user\_single\_name*);  
 userNameView.setText(name);  
 }  
  
 public void setStatus(String status) {  
 TextView userStatusView = (TextView) mView.findViewById(R.id.*user\_single\_status*);  
 userStatusView.setText(status);  
 }  
  
 public void setImage(String thumb\_image, Context applicationContext) {  
 CircleImageView circleImageView = (CircleImageView) mView.findViewById(R.id.*user\_single\_image*);  
 Picasso.*get*().load(thumb\_image).placeholder(R.mipmap.*ic\_default\_avatar\_round*).into(circleImageView);  
 }  
 }  
}

User.java

package com.example.firebasechat;  
  
public class User {  
 public String name;  
 public String image;  
 public String thumb\_image;  
 public String status;  
  
 public User() {  
 }  
  
 public User(String name, String image, String status, String thumb\_image) {  
 this.name = name;  
 this.image = image;  
 this.status = status;  
 this.thumb\_image = thumb\_image;  
 }  
  
 public String getThumb\_image() {  
 return thumb\_image;  
 }  
  
 public void setThumb\_image(String thumb\_image) {  
 this.thumb\_image = thumb\_image;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getImage() {  
 return image;  
 }  
  
 public void setImage(String image) {  
 this.image = image;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
 public void setStatus(String status) {  
 this.status = status;  
 }  
}

singleuserlayout.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:layout\_width="64dp"  
 android:layout\_height="64dp"  
 android:layout\_marginTop="15dp"  
 android:layout\_marginBottom="15dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="15dp"  
 android:id="@+id/user\_single\_image"  
 android:src="@mipmap/ic\_default\_avatar\_round"/>  
 <TextView  
 android:layout\_marginTop="15dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@android:color/black"  
 android:id="@+id/user\_single\_name"  
 android:text="Display Name"  
 android:textStyle="bold"  
 android:textSize="18dp"  
 android:layout\_toRightOf="@+id/user\_single\_image"/>  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="User default Status"  
 android:textSize="18dp"  
 android:id="@+id/user\_single\_status"  
 android:layout\_below="@+id/user\_single\_name"  
 android:layout\_toRightOf="@+id/user\_single\_image"/>  
</RelativeLayout>

## ProfileActivity

UsersActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.firebase.ui.database.FirebaseListOptions;  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.Query;  
import com.squareup.picasso.Picasso;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class UsersActivity extends AppCompatActivity {  
  
 private Toolbar mToolbar;  
 private RecyclerView mUsersList;  
 private DatabaseReference mUserDatabase;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_users*);  
  
 mToolbar = (Toolbar) findViewById(R.id.*users\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("All Users");  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users");  
  
 mUsersList = (RecyclerView) findViewById(R.id.*users\_list*);  
 mUsersList.setHasFixedSize(true);  
 mUsersList.setLayoutManager(new LinearLayoutManager(this));  
  
  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 Query query = FirebaseDatabase.*getInstance*()  
 .getReference()  
 .child("Users")  
 .limitToLast(50);  
 FirebaseRecyclerOptions<User> options =  
 new FirebaseRecyclerOptions.Builder<User>()  
 .setQuery(mUserDatabase, User.class)  
 .build();  
 FirebaseRecyclerAdapter<User, UsersViewHolder> adapter = new FirebaseRecyclerAdapter<User, UsersViewHolder>(options) {  
 @Override  
 protected void onBindViewHolder(@NonNull UsersViewHolder holder, int position, @NonNull User model) {  
 holder.setName(model.getName());  
 holder.setStatus(model.getStatus());  
 holder.setImage(model.getThumb\_image(), getApplicationContext());  
  
 final String user\_id = getRef(position).getKey();  
  
 holder.mView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(UsersActivity.this, ProfileActivity.class);  
 intent.putExtra("user\_id", user\_id);  
 startActivity(intent);  
 }  
 });  
 }  
  
 @NonNull  
 @Override  
 public UsersViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*users\_single\_layout*, parent, false);  
 UsersViewHolder viewHolder = new UsersViewHolder(view);  
 return viewHolder;  
 }  
 };  
  
 mUsersList.setAdapter(adapter);  
 adapter.startListening();  
  
 }  
  
  
  
  
 public static class UsersViewHolder extends RecyclerView.ViewHolder{  
 View mView;  
  
 public UsersViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 mView = itemView;  
 }  
  
 public void setName(String name) {  
 TextView userNameView = (TextView) mView.findViewById(R.id.*user\_single\_name*);  
 userNameView.setText(name);  
 }  
  
 public void setStatus(String status) {  
 TextView userStatusView = (TextView) mView.findViewById(R.id.*user\_single\_status*);  
 userStatusView.setText(status);  
 }  
  
 public void setImage(String thumb\_image, Context applicationContext) {  
 CircleImageView circleImageView = (CircleImageView) mView.findViewById(R.id.*user\_single\_image*);  
 Picasso.*get*().load(thumb\_image).placeholder(R.mipmap.*ic\_default\_avatar\_round*).into(circleImageView);  
 }  
 }  
}

activity\_profile.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ProfileActivity">  
  
 <TextView  
 android:id="@+id/profile\_display\_id"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 tools:layout\_editor\_absoluteX="173dp"  
 tools:layout\_editor\_absoluteY="354dp" />  
</RelativeLayout>

ProfileActivity.java

package com.example.firebasechat;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.TextView;  
  
public class ProfileActivity extends AppCompatActivity {  
  
 private TextView mProfileID;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_profile*);  
  
 String user\_id = getIntent().getStringExtra("user\_id");  
  
 mProfileID = (TextView) findViewById(R.id.*profile\_display\_id*);  
 mProfileID.setText(user\_id);  
 }  
}

## Send Request

Activity\_profile.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ProfileActivity">  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="250dp"  
 android:id="@+id/profile\_image"  
 android:src="@drawable/prof\_pic\_girl"  
 />  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/profile\_image"  
 android:background="@color/colorProfBg"  
 >  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 >  
 <TextView  
 android:id="@+id/profile\_display\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:padding="10dp"  
 android:text="@string/display\_name"  
 android:textSize="25dp"  
 android:textColor="@android:color/white"  
 />  
  
 <TextView  
 android:id="@+id/profile\_status"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:padding="5dp"  
 android:text="@string/current\_user\_status"  
 android:textColor="@android:color/white"  
 />  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_alignParentBottom="true"  
 android:layout\_gravity="bottom"  
 >  
 <TextView  
 android:id="@+id/profile\_total\_friends"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:padding="5dp"  
 android:text="@string/current\_user\_status"  
 android:textColor="@android:color/white"  
 />  
  
 <Button  
 android:id="@+id/profile\_send\_req\_btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:onClick="profile\_req\_send\_Click"  
 android:backgroundTint="@color/colorAccent"  
 android:textColor="@android:color/white"  
 android:text="@string/send\_friend\_request" />  
 <Button  
 android:id="@+id/profile\_decline\_btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginBottom="20dp"  
 android:backgroundTint="@color/colorAccent"  
 android:textColor="@android:color/white"  
 android:text="@string/decline\_request" />  
 </LinearLayout>  
 </RelativeLayout>  
</RelativeLayout>

ProfileActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
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;  
import com.squareup.picasso.Picasso;  
  
public class ProfileActivity extends AppCompatActivity {  
  
 private ImageView mProfileImage;  
  
 private TextView mProfileName, mProfileStatus, mProfileFriendsCount;  
 private Button mProfileSentBtn;  
 private DatabaseReference mUserDatabase;  
 private ProgressDialog mProgress;  
  
 private String mCurrent\_state;  
 private DatabaseReference mFriendReqDatabase;  
 private FirebaseUser mCurrentUser;  
 private String user\_id;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_profile*);  
  
 user\_id = getIntent().getStringExtra("user\_id");  
  
 mCurrent\_state = "not\_friends";  
 mCurrentUser = FirebaseAuth.*getInstance*().getCurrentUser();  
  
 mFriendReqDatabase = FirebaseDatabase.*getInstance*().getReference().child("Friend\_req");  
 mProgress = new ProgressDialog(this);  
 mProgress.setTitle("Loading user data");  
 mProgress.setMessage("Please wait while we loading user data");  
 mProgress.setCanceledOnTouchOutside(false);  
 mProgress.show();  
  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(user\_id);  
 mProfileImage = (ImageView) findViewById(R.id.*profile\_image*);  
 mProfileName = (TextView) findViewById(R.id.*profile\_display\_name*);  
 mProfileStatus = (TextView) findViewById(R.id.*profile\_status*);  
 mProfileFriendsCount = (TextView) findViewById(R.id.*profile\_total\_friends*);  
 mProfileSentBtn = (Button) findViewById(R.id.*profile\_send\_req\_btn*);  
  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String display\_name = dataSnapshot.child("name").getValue().toString();  
 String status = dataSnapshot.child("status").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
  
 mProfileName.setText(display\_name);  
 mProfileStatus.setText(status);  
 if(!image.equals("Default")) {  
 Picasso.*get*().load(image).placeholder(R.drawable.*prof\_pic\_girl*).into(mProfileImage);  
 }  
  
 // ------------------- FRIENDS LIST / REQUEST FEATURE -------  
 mFriendReqDatabase.child(mCurrentUser.getUid()).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild(user\_id)){  
 String req\_type = dataSnapshot.child(user\_id).child("request\_type").getValue().toString();  
 if(req\_type.equals("received")){  
 mCurrent\_state = "req\_received";  
 mProfileSentBtn.setText("Accept Friend Request");  
 }else if(req\_type.equals("sent")){  
 mCurrent\_state = "req\_sent";  
 mProfileSentBtn.setText("Cancel Friend Request");  
 }  
 }  
 mProgress.dismiss();  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
  
 }  
  
 public void profile\_req\_send\_Click( View view){  
 mProfileSentBtn.setEnabled(false);  
  
 //------------------------NOT FRIENDS STATE -------------------  
  
 if(mCurrent\_state.equals("not\_friends")){  
 mFriendReqDatabase.child(mCurrentUser.getUid()).child(user\_id).child("request\_type")  
 .setValue("sent")  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 mFriendReqDatabase.child(user\_id).child(mCurrentUser.getUid()).child("request\_type")  
 .setValue("received").addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mProfileSentBtn.setEnabled(true);  
 mCurrent\_state = "req\_sent";  
 mProfileSentBtn.setText("Cancel Friend Request");  
// Toast.makeText(ProfileActivity.this, "Request send successfully.", Toast.LENGTH\_SHORT).show();  
 }  
 });  
 }else {  
 Toast.*makeText*(ProfileActivity.this, "Failed sending request", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 //------------------------ CANCEL REQUEST STATE -------------------  
 if(mCurrent\_state.equals("req\_sent")){  
 mFriendReqDatabase.child(mCurrentUser.getUid()).child(user\_id).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mFriendReqDatabase.child(user\_id).child(mCurrentUser.getUid()).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mProfileSentBtn.setEnabled(true);  
 mCurrent\_state = "not\_friends";  
 mProfileSentBtn.setText("Send Friend Request");  
 }  
 });  
 }  
 });  
 }  
 }  
}

## Accept Request

ProfileActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
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;  
import com.squareup.picasso.Picasso;  
  
import java.text.DateFormat;  
import java.util.Date;  
  
public class ProfileActivity extends AppCompatActivity {  
  
 private ImageView mProfileImage;  
  
 private TextView mProfileName, mProfileStatus, mProfileFriendsCount;  
 private Button mProfileSentBtn;  
 private DatabaseReference mUserDatabase;  
 private ProgressDialog mProgress;  
  
 private String mCurrent\_state;  
 private DatabaseReference mFriendReqDatabase;  
 private DatabaseReference mFriendDatabase;  
 private FirebaseUser mCurrentUser;  
 private String user\_id;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_profile*);  
  
 user\_id = getIntent().getStringExtra("user\_id");  
  
 mCurrent\_state = "not\_friends";  
 mCurrentUser = FirebaseAuth.*getInstance*().getCurrentUser();  
  
 mFriendReqDatabase = FirebaseDatabase.*getInstance*().getReference().child("Friend\_req");  
 mFriendDatabase = FirebaseDatabase.*getInstance*().getReference().child("Friends");  
 mProgress = new ProgressDialog(this);  
 mProgress.setTitle("Loading user data");  
 mProgress.setMessage("Please wait while we loading user data");  
 mProgress.setCanceledOnTouchOutside(false);  
 mProgress.show();  
  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(user\_id);  
 mProfileImage = (ImageView) findViewById(R.id.*profile\_image*);  
 mProfileName = (TextView) findViewById(R.id.*profile\_display\_name*);  
 mProfileStatus = (TextView) findViewById(R.id.*profile\_status*);  
 mProfileFriendsCount = (TextView) findViewById(R.id.*profile\_total\_friends*);  
 mProfileSentBtn = (Button) findViewById(R.id.*profile\_send\_req\_btn*);  
  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String display\_name = dataSnapshot.child("name").getValue().toString();  
 String status = dataSnapshot.child("status").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
  
 mProfileName.setText(display\_name);  
 mProfileStatus.setText(status);  
 if(!image.equals("Default")) {  
 Picasso.*get*().load(image).placeholder(R.drawable.*prof\_pic\_girl*).into(mProfileImage);  
 }  
  
 // ------------------- FRIENDS LIST / REQUEST FEATURE -------  
 mFriendReqDatabase.child(mCurrentUser.getUid()).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild(user\_id)){  
 String req\_type = dataSnapshot.child(user\_id).child("request\_type").getValue().toString();  
 if(req\_type.equals("received")){  
 mCurrent\_state = "req\_received";  
 mProfileSentBtn.setText("Accept Friend Request");  
 }else if(req\_type.equals("sent")){  
 mCurrent\_state = "req\_sent";  
 mProfileSentBtn.setText("Cancel Friend Request");  
 }  
 }else {  
 mFriendDatabase.child(mCurrentUser.getUid()).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild(user\_id)){  
 mCurrent\_state = "friends";  
 mProfileSentBtn.setText("UnFriend this person");  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
 mProgress.dismiss();  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
  
 }  
  
 public void profile\_req\_send\_Click( View view){  
 mProfileSentBtn.setEnabled(false);  
  
 //------------------------NOT FRIENDS STATE -------------------  
  
 if(mCurrent\_state.equals("not\_friends")){  
 mFriendReqDatabase.child(mCurrentUser.getUid()).child(user\_id).child("request\_type")  
 .setValue("sent")  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 mFriendReqDatabase.child(user\_id).child(mCurrentUser.getUid()).child("request\_type")  
 .setValue("received").addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
  
 mCurrent\_state = "req\_sent";  
 mProfileSentBtn.setText("Cancel Friend Request");  
// Toast.makeText(ProfileActivity.this, "Request send successfully.", Toast.LENGTH\_SHORT).show();  
 }  
 });  
 }else {  
 Toast.*makeText*(ProfileActivity.this, "Failed sending request", Toast.*LENGTH\_SHORT*).show();  
 }  
 mProfileSentBtn.setEnabled(true);  
 }  
 });  
 }  
 //------------------------ CANCEL REQUEST STATE -------------------  
 if(mCurrent\_state.equals("req\_sent")){  
 mFriendReqDatabase.child(mCurrentUser.getUid()).child(user\_id).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mFriendReqDatabase.child(user\_id).child(mCurrentUser.getUid()).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mProfileSentBtn.setEnabled(true);  
 mCurrent\_state = "not\_friends";  
 mProfileSentBtn.setText("Send Friend Request");  
 }  
 });  
 }  
 });  
 }  
  
 // ------------------------- REQ RECEIVED STATE --------------  
 if(mCurrent\_state.equals("req\_received")){  
 final String currentDate = DateFormat.*getDateTimeInstance*().format(new Date());  
 mFriendDatabase.child(mCurrentUser.getUid()).child(user\_id).setValue(currentDate)  
 .addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mFriendDatabase.child(user\_id).child(mCurrentUser.getUid()).setValue(currentDate)  
 .addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mFriendReqDatabase.child(mCurrentUser.getUid()).child(user\_id).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mFriendReqDatabase.child(user\_id).child(mCurrentUser.getUid()).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mProfileSentBtn.setEnabled(true);  
 mCurrent\_state = "friends";  
 mProfileSentBtn.setText("UnFriend this person");  
 }  
 });  
 }  
 });  
 }  
 });  
 }  
 });  
 }  
 }  
}

## Offline Firebase Database

1. Create new class – LapitChat.java

package com.example.firebasechat;  
  
import android.app.Application;  
  
import com.google.firebase.database.FirebaseDatabase;  
import com.squareup.picasso.OkHttp3Downloader;  
import com.squareup.picasso.Picasso;  
  
public class LapitChat extends Application {  
 @Override  
 public void onCreate() {  
 super.onCreate();  
  
 FirebaseDatabase.*getInstance*().setPersistenceEnabled(true);  
  
 // Picasso  
 Picasso.Builder builder = new Picasso.Builder(this);  
 builder.downloader(new OkHttp3Downloader(this, Integer.*MAX\_VALUE*));  
 Picasso build = builder.build();  
 build.setIndicatorsEnabled(true);  
 build.setLoggingEnabled(true);  
 Picasso.*setSingletonInstance*(build);  
 }  
}

1. SettingsActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.graphics.drawable.BitmapDrawable;  
import android.net.Uri;  
import android.os.Bundle;  
import android.preference.PreferenceManager;  
import android.view.View;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
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;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
import com.google.firebase.storage.UploadTask;  
import com.squareup.picasso.Callback;  
import com.squareup.picasso.NetworkPolicy;  
import com.squareup.picasso.Picasso;  
import com.theartofdev.edmodo.cropper.CropImage;  
import com.theartofdev.edmodo.cropper.CropImageView;  
  
import java.io.ByteArrayOutputStream;  
import java.io.File;  
import java.io.IOException;  
import java.util.HashMap;  
import java.util.Map;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
import id.zelory.compressor.Compressor;  
  
public class SettingsActivity extends AppCompatActivity {  
  
 private DatabaseReference mUserDatabase;  
 private FirebaseUser mCurrentUser;  
  
 // Android Layout  
 private CircleImageView mDisplayImage;  
 private TextView mName, mStatus;  
 public static final int *GALLERY\_PICk* = 1;  
 private StorageReference mImageStorage;  
 private String current\_uid;  
  
 private ProgressDialog mProgress;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_settings*);  
 mDisplayImage = (CircleImageView) findViewById(R.id.*settings\_image*);  
 mName = (TextView) findViewById(R.id.*settings\_name*);  
 mStatus = (TextView) findViewById(R.id.*settings\_status*);  
  
 mCurrentUser = FirebaseAuth.*getInstance*().getCurrentUser();  
 mImageStorage = FirebaseStorage.*getInstance*().getReference();  
  
 mProgress = new ProgressDialog(this);  
  
 current\_uid = mCurrentUser.getUid();  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(current\_uid);  
 mUserDatabase.keepSynced(true);  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String name = dataSnapshot.child("name").getValue().toString();  
 final String image = dataSnapshot.child("image").getValue().toString();  
 String status = dataSnapshot.child("status").getValue().toString();  
 String thumb\_image = dataSnapshot.child("thumb\_image").getValue().toString();  
  
 mName.setText(name);  
 mStatus.setText(status);  
 if(!image.equals("Default")) {  
// Picasso.get().load(image).placeholder(R.mipmap.ic\_default\_avatar\_round).into(mDisplayImage);  
 Picasso.*get*().load(image).networkPolicy(NetworkPolicy.*OFFLINE*)  
 .placeholder(R.mipmap.*ic\_default\_avatar\_round*)  
 .into(mDisplayImage, new Callback() {  
 @Override  
 public void onSuccess() {  
// Toast.makeText(SettingsActivity.this, "Success loading local image", Toast.LENGTH\_SHORT).show();  
 }  
  
 @Override  
 public void onError(Exception e) {  
 Toast.*makeText*(SettingsActivity.this, "Error loading local image", Toast.*LENGTH\_SHORT*).show();  
 Picasso.*get*().load(image).placeholder(R.mipmap.*ic\_default\_avatar\_round*).into(mDisplayImage);  
 }  
 });  
  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void settings\_status\_btnClick(View view){  
 String status\_value = mStatus.getText().toString();  
 Intent intent = new Intent(SettingsActivity.this, StatusActivity.class);  
 intent.putExtra("status\_value", status\_value);  
 startActivity(intent);  
  
 }  
  
 public void settings\_image\_btnClick(View view){  
 /\*CropImage.activity()  
 .setGuidelines(CropImageView.Guidelines.ON)  
 .start(this);\*/  
 Intent intent = new Intent();  
 intent.setType("image/\*");  
 intent.setAction(Intent.*ACTION\_GET\_CONTENT*);  
 startActivityForResult(Intent.*createChooser*(intent, "SELECT IMAGE"), *GALLERY\_PICk*);  
  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 if(requestCode == *GALLERY\_PICk* && resultCode == *RESULT\_OK*){  
 Uri imageUri = data.getData();  
  
 CropImage.*activity*(imageUri)  
 .setAspectRatio(1, 1)  
 .start(this);  
  
 }  
  
 if (requestCode == CropImage.*CROP\_IMAGE\_ACTIVITY\_REQUEST\_CODE*) {  
 CropImage.ActivityResult result = CropImage.*getActivityResult*(data);  
 if (resultCode == *RESULT\_OK*) {  
  
 // Progress  
 mProgress.setTitle("Uploading Image...");  
 mProgress.setMessage("Please wait while we uploading image");  
 mProgress.setCanceledOnTouchOutside(false);  
 mProgress.show();  
  
  
 Uri resultUri = result.getUri();  
 File thumpFilePath = new File(resultUri.getPath());  
 // Compress image  
 Bitmap thumbImage = null;  
 try {  
 thumbImage = new Compressor(this)  
 .setMaxWidth(200)  
 .setMaxHeight(200)  
 .setQuality(75)  
 .compressToBitmap(thumpFilePath);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 //Upload bit map to server  
 ByteArrayOutputStream baos = new ByteArrayOutputStream();  
 thumbImage.compress(Bitmap.CompressFormat.*JPEG*, 100, baos);  
 final byte[] thumb\_byte = baos.toByteArray();  
  
 final StorageReference filepath = mImageStorage.child("profile\_images").child(current\_uid+".jpg");  
 final StorageReference thumb\_filepath = mImageStorage.child("profile\_images").child("thumbs").child(current\_uid+".jpg");  
  
 filepath.putFile(resultUri).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {  
 if(task.isSuccessful()){  
  
// String download\_url = task.getResult().getStorage().getDownloadUrl().toString();  
 filepath.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {  
 @Override  
 public void onSuccess(Uri uri) {  
 final String download\_url=uri.toString();  
  
 final UploadTask uploadTask = thumb\_filepath.putBytes(thumb\_byte);  
 uploadTask.addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> thumpb\_task) {  
// String thumb\_download\_url = thumpb\_task.getResult().getStorage().getDownloadUrl().toString();  
 if(thumpb\_task.isSuccessful()){  
 thumb\_filepath.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {  
 @Override  
 public void onSuccess(Uri uri) {  
 String thumb\_download\_url = uri.toString();  
 Map updateHashmap = new HashMap();  
 updateHashmap.put("image", download\_url);  
 updateHashmap.put("thumb\_image", thumb\_download\_url);  
  
 mUserDatabase.updateChildren(updateHashmap).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 mProgress.dismiss();  
 Toast.*makeText*(SettingsActivity.this, "Image uploaded successfully.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 });  
  
 }else{  
 Toast.*makeText*(SettingsActivity.this, "Error in uploading thumbnail.", Toast.*LENGTH\_SHORT*).show();  
 mProgress.dismiss();  
 }  
 }  
 });  
  
 }  
 });  
  
 }else {  
 Toast.*makeText*(SettingsActivity.this, "Error in uploading.", Toast.*LENGTH\_SHORT*).show();  
 mProgress.dismiss();  
 }  
 }  
 });  
  
 } else if (resultCode == CropImage.*CROP\_IMAGE\_ACTIVITY\_RESULT\_ERROR\_CODE*) {  
 Exception error = result.getError();  
 }  
 }  
 }  
}

1. Implement okhttp

implementation 'com.squareup.okhttp:okhttp:2.5.0'

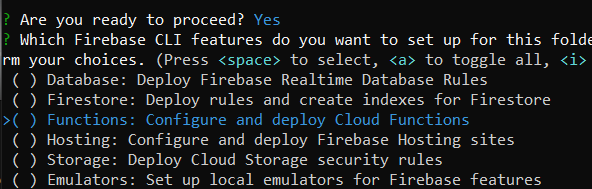
1. AndroidManifest.xml – add lapitchat as name in application field

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.firebasechat">  
  
 <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="android.permission.INTERNET" />  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"  
 android:name=".LapitChat">  
 <activity android:name=".ProfileActivity"></activity>  
 <activity  
 android:name=".UsersActivity"  
 android:parentActivityName=".MainActivity" />  
 <activity  
 android:name="com.theartofdev.edmodo.cropper.CropImageActivity"  
 android:theme="@style/Base.Theme.AppCompat" />  
 <activity  
 android:name=".StatusActivity"  
 android:parentActivityName=".SettingsActivity" />  
 <activity android:name=".SettingsActivity" />  
 <activity android:name=".LoginActivity" />  
 <activity  
 android:name=".RegisterActivity"  
 android:parentActivityName=".StartActivity" />  
 <activity android:name=".StartActivity" />  
 <activity android:name=".MainActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

## PushDown Notification

<https://www.youtube.com/watch?v=mQeHcgWxAFo&list=PLGCjwl1RrtcQ3o2jmZtwu2wXEA4OIIq53&index=21>

1. tools -> firebase -> Set Firebase Cloud Messaging
2. Add fcm to app
3. Install Node js and npm
4. Open cmd
5. > npm install -g firebase-tools
6. Create a folder in D:\fireabase\_chat\notification\_function
7. Cd D:\fireabase\_chat\notification\_function
8. > firebase login
9. > firebase init
10. Select functions



1. Press enter to select option
2. Select existing project
3. Select project
4. Select javascript
5. >y
6. >y
7. Open fuctions/index.js file

'use strict'

const functions = require('firebase-functions');

const admin = require('firebase-admin');

admin.initializeApp();

 exports.sendNotification = functions.database.ref('/notifications/{userId}/{notificationId}').onWrite((change, context) => {

const userId = context.params.userId;

const notificationId = context.params.notificationId;

console.log('The user id : ',userId);

return true;

 });

1. > firebase deploy
2. Save device id in server – LoginActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.textfield.TextInputLayout;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.iid.FirebaseInstanceId;  
  
public class LoginActivity extends AppCompatActivity {  
  
 private TextInputLayout mEmail, mPassword;  
 private Toolbar mToolbar;  
 private FirebaseAuth mAuth;  
 private ProgressDialog mProgressDialog;  
  
 private DatabaseReference mUserDatabase;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolbar = (Toolbar) findViewById(R.id.*login\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Login");  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 }  
 mAuth = FirebaseAuth.*getInstance*();  
  
 mProgressDialog = new ProgressDialog(this);  
  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users");  
  
 mEmail = (TextInputLayout) findViewById(R.id.*reg\_email*);  
 mPassword = (TextInputLayout) findViewById(R.id.*reg\_password*);  
 }  
  
 public void login\_btn\_Click(View view){  
 String email = mEmail.getEditText().getText().toString();  
 String password = mPassword.getEditText().getText().toString();  
 if(!TextUtils.*isEmpty*(email) || !TextUtils.*isEmpty*(password)) {  
  
 mProgressDialog.setTitle("Logging In");  
 mProgressDialog.setMessage("Please wait while check your credentials!");  
 mProgressDialog.setCanceledOnTouchOutside(false);  
 mProgressDialog.show();  
 login\_user(email, password);  
 }  
 }  
  
 private void login\_user(String email, String password) {  
 mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
 mProgressDialog.dismiss();  
  
 String currentUserId = mAuth.getCurrentUser().getUid();  
 String deviceToken = FirebaseInstanceId.*getInstance*().getToken();  
  
 mUserDatabase.child(currentUserId).child("device\_token").setValue(deviceToken).addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 Intent mainIntent = new Intent(LoginActivity.this, MainActivity.class );  
 mainIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(mainIntent);  
 finish();  
 }  
 });  
  
 }else {  
 mProgressDialog.hide();  
 // If sign in fails, display a message to the user.  
 Toast.*makeText*(LoginActivity.this, "Authentication failed.",  
 Toast.*LENGTH\_SHORT*).show();  
  
 }  
 }  
 });  
 }  
}

1. Index.js for sending notification

'use-strict'

const functions = require('firebase-functions');

const admin = require('firebase-admin');

admin.initializeApp(functions.config().firebase);

exports.sendNotification = functions.database.ref('/notifications/{user\_id}/{notification\_id}').onWrite((change, context) => {

  /\*

   \* You can store values as variables from the 'database.ref'

   \* Just like here, I've done for 'user\_id' and 'notification'

   \*/

   const user\_id = context.params.user\_id;

   const notification\_id = context.params.notification\_id;

   console.log('We have a notification to send to : ', context.params.user\_id);

   // if(!context.data.val()){

   //  return console.log('A Notification has been deleted from the database: ',notification\_id);

   // }

   const deviceToken = admin.database().ref('/Users/'+ user\_id +'/device\_token').once('value');

   return deviceToken.then(result => {

    const token\_id = result.val();

    const payload = {

     notification: {

      title: "Friend Request",

      body: "You've received a new Friend Request",

      icon: "default"

     }

    };

    return admin.messaging().sendToDevice(token\_id,payload).then(response => {

     return console.log('This was the notification feature');

    });

   });

});

1. Update RegisterActivity.java – save device token

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.View;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.textfield.TextInputLayout;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.iid.FirebaseInstanceId;  
  
import java.util.HashMap;  
  
public class RegisterActivity extends AppCompatActivity {  
 private TextInputLayout mDisplayName, mEmail, mPassword;  
 private FirebaseAuth mAuth;  
 private Toolbar mToolbar;  
  
 private ProgressDialog mProgressDialog;  
  
 // database  
 private DatabaseReference mDatabase;  
 private DatabaseReference mUserDatabase;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_register*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users");  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolbar = (Toolbar) findViewById(R.id.*register\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Create Account");  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 }  
  
 mProgressDialog = new ProgressDialog(this);  
  
 mDisplayName = (TextInputLayout) findViewById(R.id.*reg\_display\_name*);  
 mEmail = (TextInputLayout) findViewById(R.id.*reg\_email*);  
 mPassword = (TextInputLayout) findViewById(R.id.*reg\_password*);  
 }  
  
 public void reg\_btn\_regaccClick(View view){  
 String display\_name = mDisplayName.getEditText().getText().toString();  
 String email = mEmail.getEditText().getText().toString();  
 String password = mPassword.getEditText().getText().toString();  
 if(!TextUtils.*isEmpty*(display\_name) || !TextUtils.*isEmpty*(email) || !TextUtils.*isEmpty*(password)) {  
  
 mProgressDialog.setTitle("Registering User");  
 mProgressDialog.setMessage("Please wait while we creating your account!");  
 mProgressDialog.setCanceledOnTouchOutside(false);  
 mProgressDialog.show();  
 register\_user(display\_name, email, password);  
 }  
 }  
  
 private void register\_user(final String display\_name, String email, String password) {  
 mAuth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 // Sign in success, update UI with the signed-in user's information  
 FirebaseUser current\_user = FirebaseAuth.*getInstance*().getCurrentUser();  
 final String uid = current\_user.getUid();  
 Toast.*makeText*(RegisterActivity.this, "uid"+ uid, Toast.*LENGTH\_SHORT*).show();  
 // Write a message to the database  
 mDatabase = FirebaseDatabase.*getInstance*().getReference();  
  
 String deviceToken = FirebaseInstanceId.*getInstance*().getToken();  
  
 HashMap<String, String> userMap = new HashMap<>();  
 userMap.put("name", display\_name);  
 userMap.put("status", "Hi there I'm using chat app");  
 userMap.put("image", "Default");  
 userMap.put("thumb\_image", "Default");  
 userMap.put("device\_token", deviceToken);  
  
  
 mDatabase.child("Users").child(uid).setValue(userMap).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
  
 mProgressDialog.dismiss();  
 Intent mainIntent = new Intent(RegisterActivity.this, MainActivity.class);  
 mainIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(mainIntent);  
 finish();  
  
 }  
 });  
  
  
  
 } else {  
 mProgressDialog.hide();  
 // If sign in fails, display a message to the user.  
 Toast.*makeText*(RegisterActivity.this, "Authentication failed.",  
 Toast.*LENGTH\_SHORT*).show();  
  
 }  
 }  
 });  
 }  
}

1. Update LoginActivity.java – save device token

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.textfield.TextInputLayout;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.iid.FirebaseInstanceId;  
  
public class LoginActivity extends AppCompatActivity {  
  
 private TextInputLayout mEmail, mPassword;  
 private Toolbar mToolbar;  
 private FirebaseAuth mAuth;  
 private ProgressDialog mProgressDialog;  
  
 private DatabaseReference mUserDatabase;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolbar = (Toolbar) findViewById(R.id.*login\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Login");  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 }  
 mAuth = FirebaseAuth.*getInstance*();  
  
 mProgressDialog = new ProgressDialog(this);  
  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users");  
  
 mEmail = (TextInputLayout) findViewById(R.id.*reg\_email*);  
 mPassword = (TextInputLayout) findViewById(R.id.*reg\_password*);  
 }  
  
 public void login\_btn\_Click(View view){  
 String email = mEmail.getEditText().getText().toString();  
 String password = mPassword.getEditText().getText().toString();  
 if(!TextUtils.*isEmpty*(email) || !TextUtils.*isEmpty*(password)) {  
  
 mProgressDialog.setTitle("Logging In");  
 mProgressDialog.setMessage("Please wait while check your credentials!");  
 mProgressDialog.setCanceledOnTouchOutside(false);  
 mProgressDialog.show();  
 login\_user(email, password);  
 }  
 }  
  
 private void login\_user(String email, String password) {  
 mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
 mProgressDialog.dismiss();  
  
 String currentUserId = mAuth.getCurrentUser().getUid();  
 String deviceToken = FirebaseInstanceId.*getInstance*().getToken();  
  
 mUserDatabase.child(currentUserId).child("device\_token").setValue(deviceToken).addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 Intent mainIntent = new Intent(LoginActivity.this, MainActivity.class );  
 mainIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(mainIntent);  
 finish();  
 }  
 });  
  
 }else {  
 mProgressDialog.hide();  
 // If sign in fails, display a message to the user.  
 Toast.*makeText*(LoginActivity.this, "Authentication failed.",  
 Toast.*LENGTH\_SHORT*).show();  
  
 }  
 }  
 });  
 }  
}

1. profielActivity.java – save data in realtime database – “notifications”

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
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;  
import com.squareup.picasso.Picasso;  
  
import java.text.DateFormat;  
import java.util.Date;  
import java.util.HashMap;  
  
public class ProfileActivity extends AppCompatActivity {  
  
 private ImageView mProfileImage;  
  
 private TextView mProfileName, mProfileStatus, mProfileFriendsCount;  
 private Button mProfileSentBtn, mDeclineBtn;  
 private DatabaseReference mUserDatabase;  
 private ProgressDialog mProgress;  
  
 private String mCurrent\_state;  
 private DatabaseReference mFriendReqDatabase;  
 private DatabaseReference mFriendDatabase;  
 private DatabaseReference mNotificationDatabase;  
 private FirebaseUser mCurrentUser;  
 private String user\_id;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_profile*);  
  
 user\_id = getIntent().getStringExtra("user\_id");  
  
 mCurrent\_state = "not\_friends";  
 mCurrentUser = FirebaseAuth.*getInstance*().getCurrentUser();  
  
 mFriendReqDatabase = FirebaseDatabase.*getInstance*().getReference().child("Friend\_req");  
 mFriendDatabase = FirebaseDatabase.*getInstance*().getReference().child("Friends");  
 mNotificationDatabase = FirebaseDatabase.*getInstance*().getReference().child("notifications");  
 mProgress = new ProgressDialog(this);  
 mProgress.setTitle("Loading user data");  
 mProgress.setMessage("Please wait while we loading user data");  
 mProgress.setCanceledOnTouchOutside(false);  
 mProgress.show();  
  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(user\_id);  
 mProfileImage = (ImageView) findViewById(R.id.*profile\_image*);  
 mProfileName = (TextView) findViewById(R.id.*profile\_display\_name*);  
 mProfileStatus = (TextView) findViewById(R.id.*profile\_status*);  
 mProfileFriendsCount = (TextView) findViewById(R.id.*profile\_total\_friends*);  
 mProfileSentBtn = (Button) findViewById(R.id.*profile\_send\_req\_btn*);  
 mDeclineBtn = (Button) findViewById(R.id.*profile\_decline\_btn*);  
  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String display\_name = dataSnapshot.child("name").getValue().toString();  
 String status = dataSnapshot.child("status").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
  
 mProfileName.setText(display\_name);  
 mProfileStatus.setText(status);  
 if(!image.equals("Default")) {  
 Picasso.*get*().load(image).placeholder(R.drawable.*prof\_pic\_girl*).into(mProfileImage);  
 }  
  
 // ------------------- FRIENDS LIST / REQUEST FEATURE -------  
 mFriendReqDatabase.child(mCurrentUser.getUid()).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild(user\_id)){  
 String req\_type = dataSnapshot.child(user\_id).child("request\_type").getValue().toString();  
 if(req\_type.equals("received")){  
 mCurrent\_state = "req\_received";  
 mProfileSentBtn.setText("Accept Friend Request");  
  
 mDeclineBtn.setVisibility(View.*VISIBLE*);  
 mDeclineBtn.setEnabled(true);  
 }else if(req\_type.equals("sent")){  
 mCurrent\_state = "req\_sent";  
 mProfileSentBtn.setText("Cancel Friend Request");  
  
 mDeclineBtn.setVisibility(View.*INVISIBLE*);  
 mDeclineBtn.setEnabled(false);  
 }  
 }else {  
 mFriendDatabase.child(mCurrentUser.getUid()).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild(user\_id)){  
 mCurrent\_state = "friends";  
 mProfileSentBtn.setText("UnFriend this person");  
  
 mDeclineBtn.setVisibility(View.*INVISIBLE*);  
 mDeclineBtn.setEnabled(false);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
 mProgress.dismiss();  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
  
 }  
  
 public void profile\_req\_send\_Click( View view){  
 mProfileSentBtn.setEnabled(false);  
  
 //------------------------NOT FRIENDS STATE -------------------  
  
 if(mCurrent\_state.equals("not\_friends")){  
 mFriendReqDatabase.child(mCurrentUser.getUid()).child(user\_id).child("request\_type")  
 .setValue("sent")  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if(task.isSuccessful()){  
 mFriendReqDatabase.child(user\_id).child(mCurrentUser.getUid()).child("request\_type")  
 .setValue("received").addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
  
 HashMap<String, String> notificationData = new HashMap<>();  
 notificationData.put("from", mCurrentUser.getUid());  
 notificationData.put("type", "request");  
 mNotificationDatabase.child(user\_id).push().setValue(notificationData).addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mCurrent\_state = "req\_sent";  
 mProfileSentBtn.setText("Cancel Friend Request");  
  
 mDeclineBtn.setVisibility(View.*INVISIBLE*);  
 mDeclineBtn.setEnabled(false);  
 }  
 });  
  
  
// Toast.makeText(ProfileActivity.this, "Request send successfully.", Toast.LENGTH\_SHORT).show();  
 }  
 });  
 }else {  
 Toast.*makeText*(ProfileActivity.this, "Failed sending request", Toast.*LENGTH\_SHORT*).show();  
 }  
 mProfileSentBtn.setEnabled(true);  
 }  
 });  
 }  
 //------------------------ CANCEL REQUEST STATE -------------------  
 if(mCurrent\_state.equals("req\_sent")){  
 mFriendReqDatabase.child(mCurrentUser.getUid()).child(user\_id).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mFriendReqDatabase.child(user\_id).child(mCurrentUser.getUid()).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mProfileSentBtn.setEnabled(true);  
 mCurrent\_state = "not\_friends";  
 mProfileSentBtn.setText("Send Friend Request");  
  
 mDeclineBtn.setVisibility(View.*INVISIBLE*);  
 mDeclineBtn.setEnabled(false);  
 }  
 });  
 }  
 });  
 }  
  
 // ------------------------- REQ RECEIVED STATE --------------  
 if(mCurrent\_state.equals("req\_received")){  
 final String currentDate = DateFormat.*getDateTimeInstance*().format(new Date());  
 mFriendDatabase.child(mCurrentUser.getUid()).child(user\_id).setValue(currentDate)  
 .addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mFriendDatabase.child(user\_id).child(mCurrentUser.getUid()).setValue(currentDate)  
 .addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mFriendReqDatabase.child(mCurrentUser.getUid()).child(user\_id).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mFriendReqDatabase.child(user\_id).child(mCurrentUser.getUid()).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mProfileSentBtn.setEnabled(true);  
 mCurrent\_state = "friends";  
 mProfileSentBtn.setText("UnFriend this person");  
  
 mDeclineBtn.setVisibility(View.*INVISIBLE*);  
 mDeclineBtn.setEnabled(false);  
 }  
 });  
 }  
 });  
 }  
 });  
 }  
 });  
 }  
 }  
}

### Show Notification in foreground

1. Create new file “FirebaseNotificationServiece.java”

<https://firebase.google.com/docs/cloud-messaging/android/receive>

1. Android manifest include

<service  
    android:name=".java.MyFirebaseMessagingService"  
    android:exported="false">  
    <intent-filter>  
        <action android:name="com.google.firebase.MESSAGING\_EVENT" />  
    </intent-filter>  
</service>

AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.firebasechat">  
  
 <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="android.permission.INTERNET" />  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"  
 android:name=".LapitChat">  
  
 <service  
 android:name=".FirebaseMessagingServiece"  
 android:exported="false">  
 <intent-filter>  
 <action android:name="com.google.firebase.MESSAGING\_EVENT" />  
 </intent-filter>  
 </service>  
  
 <activity android:name=".ProfileActivity"></activity>  
 <activity  
 android:name=".UsersActivity"  
 android:parentActivityName=".MainActivity" />  
 <activity  
 android:name="com.theartofdev.edmodo.cropper.CropImageActivity"  
 android:theme="@style/Base.Theme.AppCompat" />  
 <activity  
 android:name=".StatusActivity"  
 android:parentActivityName=".SettingsActivity" />  
 <activity android:name=".SettingsActivity" />  
 <activity android:name=".LoginActivity" />  
 <activity  
 android:name=".RegisterActivity"  
 android:parentActivityName=".StartActivity" />  
 <activity android:name=".StartActivity" />  
 <activity android:name=".MainActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

### Build Notification

<https://developer.android.com/training/notify-user/build-notification>

1. AndroidManifest.xml

<activity android:name=".ProfileActivity">  
 <intent-filter>  
 <action android:name="com.example.abhi.chat\_TARGET\_NOTIFICATION"></action>  
 <category android:name="android.intent.category.DEFAULT"></category>  
 </intent-filter>  
</activity>

1. Index.js server function

'use-strict'

const functions = require('firebase-functions');

const admin = require('firebase-admin');

admin.initializeApp(functions.config().firebase);

exports.sendNotification = functions.database.ref('/notifications/{user\_id}/{notification\_id}').onWrite((change, context) => {

  /\*

   \* You can store values as variables from the 'database.ref'

   \* Just like here, I've done for 'user\_id' and 'notification'

   \*/

   const user\_id = context.params.user\_id;

   const notification\_id = context.params.notification\_id;

   console.log('We have a notification to send to : ', context.params.user\_id);

   // if(!context.data.val()){

   //  return console.log('A Notification has been deleted from the database: ',notification\_id);

   // }

   const fromUser = admin.database().ref('/notifications/'+ user\_id +'/'+notification\_id).once('value');

    return fromUser.then(fromUserResult => {

        const from\_user\_id = fromUserResult.val().from;

        console.log('We have new notification from : ', from\_user\_id);

        const userQuery = admin.database().ref('/Users/'+ from\_user\_id +'/name').once('value');

        return userQuery.then(UserResult => {

            const userName = UserResult.val();

            const deviceToken = admin.database().ref('/Users/'+ user\_id +'/device\_token').once('value');

            return deviceToken.then(result => {

                const token\_id = result.val();

                const payload = {

                    notification: {

                        title: "Friend Request",

                        body: userName + " has send Request",

                        icon: "default",

                        click\_action : "com.example.abhi.chat\_TARGET\_NOTIFICATION"

                    },

                    data : {

                        from\_user\_id : from\_user\_id

                    }

                };

                return admin.messaging().sendToDevice(token\_id,payload).then(response => {

                    return console.log('This was the notification feature');

                });

            });

        });

    });

});

1. FireBaseMessagingServiece.java

package com.example.firebasechat;  
  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Intent;  
import android.os.Build;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.core.app.NotificationCompat;  
import androidx.core.app.NotificationManagerCompat;  
  
import com.google.firebase.messaging.FirebaseMessagingService;  
import com.google.firebase.messaging.RemoteMessage;  
  
public class FirebaseMessagingServiece extends FirebaseMessagingService {  
 private final String CHANNEL\_ID = "personal\_notifications";  
 @Override  
 public void onMessageReceived(@NonNull RemoteMessage remoteMessage) {  
 super.onMessageReceived(remoteMessage);  
  
 createNotificationChannel();  
  
 String notification\_title = remoteMessage.getNotification().getTitle();  
 String notification\_body = remoteMessage.getNotification().getBody();  
 String click\_action = remoteMessage.getNotification().getClickAction();  
 String from\_user\_id = remoteMessage.getData().get("from\_user\_id");  
  
 Intent intent = new Intent(click\_action);  
 intent.putExtra("user\_id", from\_user\_id);  
 intent.setFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 PendingIntent pendingIntent = PendingIntent.*getActivity*(this, 0, intent, 0);  
  
 NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL\_ID)  
 .setSmallIcon(R.mipmap.*ic\_launcher*)  
 .setContentTitle(notification\_title)  
 .setContentText(notification\_body)  
 .setPriority(NotificationCompat.*PRIORITY\_DEFAULT*)  
 .setContentIntent(pendingIntent);  
 int mNotifiactionId = (int) System.*currentTimeMillis*();  
  
 NotificationManagerCompat notificationManager = NotificationManagerCompat.*from*(this);  
  
 notificationManager.notify(mNotifiactionId, builder.build());  
 }  
  
 private void createNotificationChannel() {  
 // Create the NotificationChannel, but only on API 26+ because  
 // the NotificationChannel class is new and not in the support library  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 CharSequence name = "channel\_name";  
 String description = "channel\_description";  
 int importance = NotificationManager.*IMPORTANCE\_DEFAULT*;  
 NotificationChannel channel = new NotificationChannel(CHANNEL\_ID, name, importance);  
 channel.setDescription(description);  
 // Register the channel with the system; you can't change the importance  
 // or other notification behaviors after this  
 NotificationManager notificationManager = getSystemService(NotificationManager.class);  
 notificationManager.createNotificationChannel(channel);  
 }  
 }  
}

## BugFix – Clean code

1. Server side – index.js

'use-strict'

const functions = require('firebase-functions');

const admin = require('firebase-admin');

admin.initializeApp(functions.config().firebase);

exports.sendNotification = functions.database.ref('/notifications/{user\_id}/{notification\_id}').onWrite((change, context) => {

  /\*

   \* You can store values as variables from the 'database.ref'

   \* Just like here, I've done for 'user\_id' and 'notification'

   \*/

   const user\_id = context.params.user\_id;

   const notification\_id = context.params.notification\_id;

   console.log('We have a notification to send to : ', context.params.user\_id);

   // if(!context.data.val()){

   //  return console.log('A Notification has been deleted from the database: ',notification\_id);

   // }

   const fromUser = admin.database().ref('/notifications/'+ user\_id +'/'+notification\_id).once('value');

    return fromUser.then(fromUserResult => {

        const from\_user\_id = fromUserResult.val().from;

        console.log('We have new notification from : ', from\_user\_id);

        const userQuery = admin.database().ref('/Users/'+ from\_user\_id +'/name').once('value');

        const deviceToken = admin.database().ref('/Users/'+ user\_id +'/device\_token').once('value');

        return Promise.all([userQuery, deviceToken]).then(result => {

            const userName = result[0].val();

            const token\_id = result[1].val();

            const payload = {

                notification: {

                    title: "Friend Request",

                    body: userName + " has send Request",

                    icon: "default",

                    click\_action : "com.example.abhi.chat\_TARGET\_NOTIFICATION"

                },

                data : {

                    from\_user\_id : from\_user\_id

                }

            };

            return admin.messaging().sendToDevice(token\_id,payload).then(response => {

                return console.log('This was the notification feature');

            });

        });

    });

});

1. ProfileActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
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;  
import com.squareup.picasso.Picasso;  
  
import java.text.DateFormat;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.Map;  
  
public class ProfileActivity extends AppCompatActivity {  
  
 private ImageView mProfileImage;  
  
 private TextView mProfileName, mProfileStatus, mProfileFriendsCount;  
 private Button mProfileSentBtn, mDeclineBtn;  
 private DatabaseReference mUserDatabase;  
 private ProgressDialog mProgress;  
  
 private String mCurrent\_state;  
 private DatabaseReference mFriendReqDatabase;  
 private DatabaseReference mFriendDatabase;  
 private DatabaseReference mNotificationDatabase;  
 private DatabaseReference mRootRef;  
 private FirebaseUser mCurrentUser;  
 private String user\_id;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_profile*);  
  
 user\_id = getIntent().getStringExtra("user\_id");  
  
 mCurrent\_state = "not\_friends";  
 mCurrentUser = FirebaseAuth.*getInstance*().getCurrentUser();  
  
 mRootRef = FirebaseDatabase.*getInstance*().getReference();  
 mFriendReqDatabase = FirebaseDatabase.*getInstance*().getReference().child("Friend\_req");  
 mFriendDatabase = FirebaseDatabase.*getInstance*().getReference().child("Friends");  
 mNotificationDatabase = FirebaseDatabase.*getInstance*().getReference().child("notifications");  
 mProgress = new ProgressDialog(this);  
 mProgress.setTitle("Loading user data");  
 mProgress.setMessage("Please wait while we loading user data");  
 mProgress.setCanceledOnTouchOutside(false);  
 mProgress.show();  
  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(user\_id);  
 mProfileImage = (ImageView) findViewById(R.id.*profile\_image*);  
 mProfileName = (TextView) findViewById(R.id.*profile\_display\_name*);  
 mProfileStatus = (TextView) findViewById(R.id.*profile\_status*);  
 mProfileFriendsCount = (TextView) findViewById(R.id.*profile\_total\_friends*);  
 mProfileSentBtn = (Button) findViewById(R.id.*profile\_send\_req\_btn*);  
 mDeclineBtn = (Button) findViewById(R.id.*profile\_decline\_btn*);  
  
 mDeclineBtn.setVisibility(View.*INVISIBLE*);  
 mDeclineBtn.setEnabled(false);  
  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String display\_name = dataSnapshot.child("name").getValue().toString();  
 String status = dataSnapshot.child("status").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
  
 mProfileName.setText(display\_name);  
 mProfileStatus.setText(status);  
 if(!image.equals("Default")) {  
 Picasso.*get*().load(image).placeholder(R.drawable.*prof\_pic\_girl*).into(mProfileImage);  
 }  
  
 // ------------------- FRIENDS LIST / REQUEST FEATURE -------  
 mFriendReqDatabase.child(mCurrentUser.getUid()).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild(user\_id)){  
 String req\_type = dataSnapshot.child(user\_id).child("request\_type").getValue().toString();  
 if(req\_type.equals("received")){  
 mCurrent\_state = "req\_received";  
 mProfileSentBtn.setText("Accept Friend Request");  
  
 mDeclineBtn.setVisibility(View.*VISIBLE*);  
 mDeclineBtn.setEnabled(true);  
 }else if(req\_type.equals("sent")){  
 mCurrent\_state = "req\_sent";  
 mProfileSentBtn.setText("Cancel Friend Request");  
  
 mDeclineBtn.setVisibility(View.*INVISIBLE*);  
 mDeclineBtn.setEnabled(false);  
 }  
 }else {  
 mFriendDatabase.child(mCurrentUser.getUid()).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.hasChild(user\_id)){  
 mCurrent\_state = "friends";  
 mProfileSentBtn.setText("UnFriend this person");  
  
 mDeclineBtn.setVisibility(View.*INVISIBLE*);  
 mDeclineBtn.setEnabled(false);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
 mProgress.dismiss();  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
  
 }  
  
 public void profile\_req\_send\_Click( View view){  
 mProfileSentBtn.setEnabled(false);  
  
 //------------------------NOT FRIENDS STATE -------------------  
  
 if(mCurrent\_state.equals("not\_friends")){  
  
 DatabaseReference newNotificationRef = mRootRef.child("notifications").child(user\_id).push();  
 String newNotificationId = newNotificationRef.getKey();  
  
 HashMap<String, String> notificationData = new HashMap<>();  
 notificationData.put("from", mCurrentUser.getUid());  
 notificationData.put("type", "request");  
  
 Map requestMap = new HashMap();  
 requestMap.put("Friend\_req/" + mCurrentUser.getUid() + "/" +user\_id + "/" + "request\_type", "sent");  
 requestMap.put( "Friend\_req/" + user\_id + "/" +mCurrentUser.getUid() + "/" + "request\_type", "received");  
 requestMap.put("notifications/" + user\_id + "/" + newNotificationId , notificationData);  
  
 mRootRef.updateChildren(requestMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
  
 if(databaseError != null){  
 Toast.*makeText*(ProfileActivity.this, "Their was en error in sending request", Toast.*LENGTH\_SHORT*).show();  
 }  
 mCurrent\_state = "req\_sent";  
 mProfileSentBtn.setText("Cancel Friend Request");  
  
 mProfileSentBtn.setEnabled(true);  
  
 }  
 });  
 }  
 //------------------------ CANCEL REQUEST STATE -------------------  
 if(mCurrent\_state.equals("req\_sent")){  
 mFriendReqDatabase.child(mCurrentUser.getUid()).child(user\_id).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mFriendReqDatabase.child(user\_id).child(mCurrentUser.getUid()).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 mProfileSentBtn.setEnabled(true);  
 mCurrent\_state = "not\_friends";  
 mProfileSentBtn.setText("Send Friend Request");  
  
 mDeclineBtn.setVisibility(View.*INVISIBLE*);  
 mDeclineBtn.setEnabled(false);  
 }  
 });  
 }  
 });  
 }  
  
 // ------------------------- REQ RECEIVED STATE --------------  
 if(mCurrent\_state.equals("req\_received")){  
 final String currentDate = DateFormat.*getDateTimeInstance*().format(new Date());  
 Map friendsMap = new HashMap();  
 friendsMap.put("Friends/" + mCurrentUser.getUid()+"/" + user\_id + "/date", currentDate);  
 friendsMap.put("Friends/" + user\_id+"/" + mCurrentUser.getUid() + "/date", currentDate);  
  
 friendsMap.put("Friend\_req/" + user\_id+"/" + mCurrentUser.getUid() , null);  
 friendsMap.put("Friend\_req/" + mCurrentUser.getUid()+"/" + user\_id , null);  
  
 mRootRef.updateChildren(friendsMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 if(databaseError == null){  
 mProfileSentBtn.setEnabled(true);  
 mCurrent\_state = "friends";  
 mProfileSentBtn.setText("UnFriend this person");  
  
 mDeclineBtn.setVisibility(View.*INVISIBLE*);  
 mDeclineBtn.setEnabled(false);  
 }else {  
 String error = databaseError.getMessage();  
 Toast.*makeText*(ProfileActivity.this, error, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 //----------------------------UNFRIEND-------------------------  
 if(mCurrent\_state.equals("friends")){  
 Map unfriendMap = new HashMap();  
 unfriendMap.put("Friends/" + mCurrentUser.getUid()+"/" + user\_id , null);  
 unfriendMap.put("Friends/" + user\_id+"/" + mCurrentUser.getUid() , null);  
 mRootRef.updateChildren(unfriendMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 if(databaseError == null){  
  
 mCurrent\_state = "not\_friends";  
 mProfileSentBtn.setText("Send Friend Request");  
  
// mDeclineBtn.setVisibility(View.INVISIBLE);  
// mDeclineBtn.setEnabled(false);  
 }else {  
 String error = databaseError.getMessage();  
 Toast.*makeText*(ProfileActivity.this, error, Toast.*LENGTH\_SHORT*).show();  
 }  
 mProfileSentBtn.setEnabled(true);  
 }  
 });  
 }  
 }  
}

## FriendsFragment

1. FriendsFragment.java

package com.example.firebasechat;  
  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AlertDialog;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
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.Query;  
import com.google.firebase.database.ValueEventListener;  
import com.squareup.picasso.Picasso;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \*/*public class FriendsFragment extends Fragment {  
  
 private RecyclerView mFriendsList;  
 private DatabaseReference mFriendsDatabase;  
 private DatabaseReference mUsersDatabase;  
 private FirebaseAuth mAuth;  
 private String mCurrent\_user\_id;  
 private View mMainView;  
 public FriendsFragment() {  
 // Required empty public constructor  
 }  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
  
 mMainView = inflater.inflate(R.layout.*fragment\_friends*, container, false);  
 mFriendsList = mMainView.findViewById(R.id.*friends\_list*);  
 mAuth = FirebaseAuth.*getInstance*();  
 mCurrent\_user\_id = mAuth.getCurrentUser().getUid();  
  
 mFriendsDatabase = FirebaseDatabase.*getInstance*().getReference().child("Friends").child(mCurrent\_user\_id);  
 mFriendsDatabase.keepSynced(true);  
 mUsersDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users");  
 mUsersDatabase.keepSynced(true);  
  
 mFriendsList.setHasFixedSize(true);  
 mFriendsList.setLayoutManager(new LinearLayoutManager(getContext()));  
  
 // Inflate the layout for this fragment  
 return mMainView;  
 }  
 @Override  
 public void onStart() {  
 super.onStart();  
 //  
 FirebaseRecyclerOptions<Friends> options =  
 new FirebaseRecyclerOptions.Builder<Friends>()  
 .setQuery( mFriendsDatabase, Friends.class)  
 .build();  
 FirebaseRecyclerAdapter friendsRecyclerViewAdapter = new FirebaseRecyclerAdapter<Friends, FriendsViewHolder>(options) {  
  
 @Override  
 protected void onBindViewHolder(@NonNull final FriendsViewHolder friendsViewHolder, int i, @NonNull Friends friends) {  
 friendsViewHolder.setDate(friends.getDate());  
  
 final String list\_user\_id = getRef(i).getKey();  
  
 mUsersDatabase.child(list\_user\_id).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
  
 final String userName = dataSnapshot.child("name").getValue().toString();  
 String userThumb = dataSnapshot.child("thumb\_image").getValue().toString();  
  
 if(dataSnapshot.hasChild("online")) {  
  
 String userOnline = dataSnapshot.child("online").getValue().toString();  
 friendsViewHolder.setUserOnline(userOnline);  
  
 }  
  
 friendsViewHolder.setName(userName);  
 friendsViewHolder.setUserImage(userThumb, getContext());  
 // When Click to Friends View  
 friendsViewHolder.mView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 CharSequence options[] = new CharSequence[]{"Open Profile", "Send message"};  
 final AlertDialog.Builder builder = new AlertDialog.Builder(getContext());  
  
 builder.setTitle("Select Options");  
 builder.setItems(options, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 //Click Event for each item.  
 if(i == 0){  
 Intent profileIntent = new Intent(getContext(), ProfileActivity.class);  
 profileIntent.putExtra("user\_id", list\_user\_id);  
 startActivity(profileIntent);  
  
 }  
  
 if(i == 1){  
  
 /\* Intent chatIntent = new Intent(getContext(), ChatActivity.class);  
 chatIntent.putExtra("user\_id", list\_user\_id);  
 chatIntent.putExtra("user\_name", userName);  
 startActivity(chatIntent);\*/  
 }  
 }  
 });  
 builder.show();  
 }  
 });  
 }  
 @Override  
 public void onCancelled(DatabaseError databaseError) {  
  
 }  
 });  
 }  
 @NonNull  
 @Override  
 public FriendsViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int i) {  
 View view = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*users\_single\_layout*, parent, false);  
 return new FriendsViewHolder(view);  
 }  
 };  
  
 mFriendsList.setAdapter(friendsRecyclerViewAdapter);  
 friendsRecyclerViewAdapter.startListening();  
 }  
  
 public class FriendsViewHolder extends RecyclerView.ViewHolder {  
  
 View mView;  
  
 public FriendsViewHolder(View itemView) {  
 super(itemView);  
  
 mView = itemView;  
 }  
  
 public void setDate(String date){  
 TextView userStatusView = mView.findViewById(R.id.*user\_single\_status*);  
 userStatusView.setText(date);  
 }  
  
 public void setName(String name){  
 TextView userNameView = mView.findViewById(R.id.*user\_single\_name*);  
 userNameView.setText(name);  
 }  
 public void setUserImage(String thumb\_image, Context ctx){  
 CircleImageView userImageView = mView.findViewById(R.id.*user\_single\_image*);  
 Picasso.*get*().load(thumb\_image).placeholder(R.mipmap.*ic\_default\_avatar*).into(userImageView);  
 }  
 public void setUserOnline(String online\_status) {  
 /\*ImageView userOnlineView =mView.findViewById(R.id.user\_single\_online\_icon);  
 if(online\_status.equals("true")){  
 userOnlineView.setVisibility(View.VISIBLE);  
 } else {  
 userOnlineView.setVisibility(View.INVISIBLE);  
 }\*/  
 }  
 }  
}

1. Create Class Friends.java

package com.example.firebasechat;  
  
public class Friends {  
 public String date;  
  
 public Friends() {  
 }  
  
 public Friends(String date) {  
 this.date = date;  
 }  
  
 public String getDate() {  
 return date;  
 }  
  
 public void setDate(String date) {  
 this.date = date;  
 }  
}

1. Fragment\_friends.xml

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".FriendsFragment">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/friends\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 </androidx.recyclerview.widget.RecyclerView>  
  
</FrameLayout>

## Online feature

1. LapitChat.java

package com.example.firebasechat;  
  
import android.app.Application;  
  
import androidx.annotation.NonNull;  
  
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;  
import com.squareup.picasso.OkHttp3Downloader;  
import com.squareup.picasso.Picasso;  
  
public class LapitChat extends Application {  
  
 private DatabaseReference mUserDatabase;  
 private FirebaseAuth mAuth;  
  
 @Override  
 public void onCreate() {  
 super.onCreate();  
  
 FirebaseDatabase.*getInstance*().setPersistenceEnabled(true);  
  
 // Picasso  
 Picasso.Builder builder = new Picasso.Builder(this);  
 builder.downloader(new OkHttp3Downloader(this, Integer.*MAX\_VALUE*));  
 Picasso build = builder.build();  
 build.setIndicatorsEnabled(true);  
 build.setLoggingEnabled(true);  
 Picasso.*setSingletonInstance*(build);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(mAuth.getCurrentUser().getUid());  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot != null) {  
 mUserDatabase.child("online").onDisconnect().setValue(false);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
  
}

1. MainActivity.java

package com.example.firebasechat;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
  
import androidx.appcompat.widget.Toolbar;  
import androidx.viewpager.widget.ViewPager;  
  
import com.google.android.material.tabs.TabLayout;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
public class MainActivity extends AppCompatActivity {  
  
 private FirebaseAuth mAuth;  
 private Toolbar mToolbar;  
  
 private ViewPager mViewPager;  
 private SectionsPageAdapter mSectionsPageAdapter;  
 private TabLayout mTabLayout;  
  
 private DatabaseReference mUserRef;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 mUserRef = FirebaseDatabase.*getInstance*().getReference().child("Users").child(mAuth.getCurrentUser().getUid());  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolbar = (Toolbar) findViewById(R.id.*main\_page\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Chat");  
 }  
  
 //TABS  
 mViewPager = (ViewPager) findViewById(R.id.*main\_tabPager*);  
 mSectionsPageAdapter = new SectionsPageAdapter(getSupportFragmentManager());  
 mViewPager.setAdapter(mSectionsPageAdapter);  
  
 mTabLayout =(TabLayout) findViewById(R.id.*main\_tabs*);  
 mTabLayout.setupWithViewPager(mViewPager);  
  
 }  
  
  
  
 public void onStart() {  
 super.onStart();  
 // Check if user is signed in (non-null) and update UI accordingly.  
 FirebaseUser currentUser = mAuth.getCurrentUser();  
 if(currentUser==null){  
 sendToStart();  
 }else {  
 mUserRef.child("online").setValue(true);  
 }  
 }  
  
 @Override  
 protected void onStop() {  
 super.onStop();  
 FirebaseUser currentUser = mAuth.getCurrentUser();  
 if(currentUser!=null) {  
 mUserRef.child("online").setValue(false);  
 }  
 }  
  
 private void sendToStart() {  
 Intent intent = new Intent(MainActivity.this, StartActivity.class);  
 startActivity(intent);  
 finish();  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 super.onCreateOptionsMenu(menu);  
 getMenuInflater().inflate(R.menu.*menu\_main*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 super.onOptionsItemSelected(item);  
 if(item.getItemId() == R.id.*main\_logout\_btn*){  
 FirebaseAuth.*getInstance*().signOut();  
 sendToStart();  
 }  
 if(item.getItemId() == R.id.*main\_settings\_btn*){  
 Intent intent = new Intent(MainActivity.this, SettingsActivity.class);  
 startActivity(intent);  
 }  
  
 if(item.getItemId() == R.id.*main\_all\_user\_btn*){  
 Intent intent = new Intent(MainActivity.this, UsersActivity.class);  
 startActivity(intent);  
 }  
 return true;  
 }  
}

1. FriendsFragment.java

package com.example.firebasechat;  
  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AlertDialog;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
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.Query;  
import com.google.firebase.database.ValueEventListener;  
import com.squareup.picasso.Picasso;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \*/*public class FriendsFragment extends Fragment {  
  
 private RecyclerView mFriendsList;  
 private DatabaseReference mFriendsDatabase;  
 private DatabaseReference mUsersDatabase;  
 private FirebaseAuth mAuth;  
 private String mCurrent\_user\_id;  
 private View mMainView;  
 public FriendsFragment() {  
 // Required empty public constructor  
 }  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
  
 mMainView = inflater.inflate(R.layout.*fragment\_friends*, container, false);  
 mFriendsList = mMainView.findViewById(R.id.*friends\_list*);  
 mAuth = FirebaseAuth.*getInstance*();  
 mCurrent\_user\_id = mAuth.getCurrentUser().getUid();  
  
 mFriendsDatabase = FirebaseDatabase.*getInstance*().getReference().child("Friends").child(mCurrent\_user\_id);  
 mFriendsDatabase.keepSynced(true);  
 mUsersDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users");  
 mUsersDatabase.keepSynced(true);  
  
 mFriendsList.setHasFixedSize(true);  
 mFriendsList.setLayoutManager(new LinearLayoutManager(getContext()));  
  
 // Inflate the layout for this fragment  
 return mMainView;  
 }  
 @Override  
 public void onStart() {  
 super.onStart();  
 //  
 FirebaseRecyclerOptions<Friends> options =  
 new FirebaseRecyclerOptions.Builder<Friends>()  
 .setQuery( mFriendsDatabase, Friends.class)  
 .build();  
 FirebaseRecyclerAdapter friendsRecyclerViewAdapter = new FirebaseRecyclerAdapter<Friends, FriendsViewHolder>(options) {  
  
 @Override  
 protected void onBindViewHolder(@NonNull final FriendsViewHolder friendsViewHolder, int i, @NonNull Friends friends) {  
 friendsViewHolder.setDate(friends.getDate());  
  
 final String list\_user\_id = getRef(i).getKey();  
  
 mUsersDatabase.child(list\_user\_id).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
  
 final String userName = dataSnapshot.child("name").getValue().toString();  
 String userThumb = dataSnapshot.child("thumb\_image").getValue().toString();  
  
 if(dataSnapshot.hasChild("online")) {  
  
 String userOnline = dataSnapshot.child("online").getValue().toString();  
 friendsViewHolder.setUserOnline(userOnline);  
  
 }  
  
 friendsViewHolder.setName(userName);  
 friendsViewHolder.setUserImage(userThumb, getContext());  
 // When Click to Friends View  
 friendsViewHolder.mView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 CharSequence options[] = new CharSequence[]{"Open Profile", "Send message"};  
 final AlertDialog.Builder builder = new AlertDialog.Builder(getContext());  
  
 builder.setTitle("Select Options");  
 builder.setItems(options, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 //Click Event for each item.  
 if(i == 0){  
 Intent profileIntent = new Intent(getContext(), ProfileActivity.class);  
 profileIntent.putExtra("user\_id", list\_user\_id);  
 startActivity(profileIntent);  
  
 }  
  
 if(i == 1){  
  
 /\* Intent chatIntent = new Intent(getContext(), ChatActivity.class);  
 chatIntent.putExtra("user\_id", list\_user\_id);  
 chatIntent.putExtra("user\_name", userName);  
 startActivity(chatIntent);\*/  
 }  
 }  
 });  
 builder.show();  
 }  
 });  
 }  
 @Override  
 public void onCancelled(DatabaseError databaseError) {  
  
 }  
 });  
 }  
 @NonNull  
 @Override  
 public FriendsViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int i) {  
 View view = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*users\_single\_layout*, parent, false);  
 return new FriendsViewHolder(view);  
 }  
 };  
  
 mFriendsList.setAdapter(friendsRecyclerViewAdapter);  
 friendsRecyclerViewAdapter.startListening();  
 }  
  
 public class FriendsViewHolder extends RecyclerView.ViewHolder {  
  
 View mView;  
  
 public FriendsViewHolder(View itemView) {  
 super(itemView);  
  
 mView = itemView;  
 }  
  
 public void setDate(String date){  
 TextView userStatusView = mView.findViewById(R.id.*user\_single\_status*);  
 userStatusView.setText(date);  
 }  
  
 public void setName(String name){  
 TextView userNameView = mView.findViewById(R.id.*user\_single\_name*);  
 userNameView.setText(name);  
 }  
 public void setUserImage(String thumb\_image, Context ctx){  
 CircleImageView userImageView = mView.findViewById(R.id.*user\_single\_image*);  
 Picasso.*get*().load(thumb\_image).placeholder(R.mipmap.*ic\_default\_avatar*).into(userImageView);  
 }  
 public void setUserOnline(String online\_status) {  
 ImageView userOnlineView =mView.findViewById(R.id.*user\_single\_online*);  
 if(online\_status.equals("true")){  
 userOnlineView.setVisibility(View.*VISIBLE*);  
 } else {  
 userOnlineView.setVisibility(View.*INVISIBLE*);  
 }  
 }  
 }  
}

1. User\_single\_layout.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:layout\_width="64dp"  
 android:layout\_height="64dp"  
 android:layout\_marginTop="15dp"  
 android:layout\_marginBottom="15dp"  
 android:layout\_marginLeft="15dp"  
 android:layout\_marginRight="15dp"  
 android:id="@+id/user\_single\_image"  
 android:src="@mipmap/ic\_default\_avatar\_round"/>  
 <TextView  
 android:layout\_marginTop="15dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@android:color/black"  
 android:id="@+id/user\_single\_name"  
 android:text="Display Name"  
 android:textStyle="bold"  
 android:textSize="18dp"  
 android:layout\_toRightOf="@+id/user\_single\_image"/>  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="User default Status"  
 android:textSize="18dp"  
 android:id="@+id/user\_single\_status"  
 android:layout\_below="@+id/user\_single\_name"  
 android:layout\_toRightOf="@+id/user\_single\_image"/>  
 <ImageView  
 android:layout\_width="25dp"  
 android:layout\_height="25dp"  
 android:id="@+id/user\_single\_online"  
 android:src="@drawable/online\_icon"  
 android:visibility="invisible"  
 android:layout\_alignParentRight="true"  
 android:layout\_margin="15dp"  
 />  
</RelativeLayout>

1. Include a green online icon in drawable folder

## ChatActivity

1. Create new empty activity – ChatActivity
2. Create new layout resource file – chat\_custom\_bar.xml – root element - LinearLayout
3. Activity\_chat.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@color/colorGray"  
 tools:context=".ChatActivity">  
  
 <include layout="@layout/app\_bar\_layout" android:id="@+id/chat\_app\_bar" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@android:color/white"  
 android:orientation="horizontal"  
  
 android:layout\_alignParentBottom="true"  
 android:gravity="bottom"  
 android:weightSum="10"  
 >  
 <ImageButton  
 android:layout\_width="55dp"  
 android:layout\_height="55dp"  
 android:background="@android:color/white"  
 android:adjustViewBounds="true"  
 android:scaleType="centerCrop"  
 android:src="@mipmap/ic\_plus"  
 android:layout\_weight="1"  
 />  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_weight="8"  
 />  
 <ImageButton  
 android:layout\_width="55dp"  
 android:layout\_height="55dp"  
 android:layout\_weight="1"  
 android:background="@android:color/white"  
 android:adjustViewBounds="true"  
 android:scaleType="centerCrop"  
 android:padding="5dp"  
 android:layout\_marginRight="5dp"  
 android:src="@mipmap/ic\_sent\_round"  
  
 />  
 </LinearLayout>  
</RelativeLayout>

1. LapitChat.java

package com.example.firebasechat;  
  
import android.app.Application;  
  
import androidx.annotation.NonNull;  
  
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.ServerValue;  
import com.google.firebase.database.ValueEventListener;  
import com.squareup.picasso.OkHttp3Downloader;  
import com.squareup.picasso.Picasso;  
  
public class LapitChat extends Application {  
  
 private DatabaseReference mUserDatabase;  
 private FirebaseAuth mAuth;  
  
 @Override  
 public void onCreate() {  
 super.onCreate();  
  
 FirebaseDatabase.*getInstance*().setPersistenceEnabled(true);  
  
 // Picasso  
 Picasso.Builder builder = new Picasso.Builder(this);  
 builder.downloader(new OkHttp3Downloader(this, Integer.*MAX\_VALUE*));  
 Picasso build = builder.build();  
 build.setIndicatorsEnabled(true);  
 build.setLoggingEnabled(true);  
 Picasso.*setSingletonInstance*(build);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(mAuth.getCurrentUser().getUid());  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot != null) {  
 mUserDatabase.child("online").onDisconnect().setValue(ServerValue.*TIMESTAMP*);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
  
}

1. MainActivity.java

package com.example.firebasechat;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
  
import androidx.appcompat.widget.Toolbar;  
import androidx.viewpager.widget.ViewPager;  
  
import com.google.android.material.tabs.TabLayout;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ServerValue;  
  
public class MainActivity extends AppCompatActivity {  
  
 private FirebaseAuth mAuth;  
 private Toolbar mToolbar;  
  
 private ViewPager mViewPager;  
 private SectionsPageAdapter mSectionsPageAdapter;  
 private TabLayout mTabLayout;  
  
 private DatabaseReference mUserRef;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 mUserRef = FirebaseDatabase.*getInstance*().getReference().child("Users").child(mAuth.getCurrentUser().getUid());  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolbar = (Toolbar) findViewById(R.id.*main\_page\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Chat");  
 }  
  
 //TABS  
 mViewPager = (ViewPager) findViewById(R.id.*main\_tabPager*);  
 mSectionsPageAdapter = new SectionsPageAdapter(getSupportFragmentManager());  
 mViewPager.setAdapter(mSectionsPageAdapter);  
  
 mTabLayout =(TabLayout) findViewById(R.id.*main\_tabs*);  
 mTabLayout.setupWithViewPager(mViewPager);  
  
 }  
  
  
  
 public void onStart() {  
 super.onStart();  
 // Check if user is signed in (non-null) and update UI accordingly.  
 FirebaseUser currentUser = mAuth.getCurrentUser();  
 if(currentUser==null){  
 sendToStart();  
 }else {  
 mUserRef.child("online").setValue(true);  
 }  
 }  
  
 @Override  
 protected void onStop() {  
 super.onStop();  
 FirebaseUser currentUser = mAuth.getCurrentUser();  
 if(currentUser!=null) {  
 mUserRef.child("online").setValue(ServerValue.*TIMESTAMP*);  
 }  
 }  
  
 private void sendToStart() {  
 Intent intent = new Intent(MainActivity.this, StartActivity.class);  
 startActivity(intent);  
 finish();  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 super.onCreateOptionsMenu(menu);  
 getMenuInflater().inflate(R.menu.*menu\_main*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 super.onOptionsItemSelected(item);  
 if(item.getItemId() == R.id.*main\_logout\_btn*){  
 FirebaseAuth.*getInstance*().signOut();  
 sendToStart();  
 }  
 if(item.getItemId() == R.id.*main\_settings\_btn*){  
 Intent intent = new Intent(MainActivity.this, SettingsActivity.class);  
 startActivity(intent);  
 }  
  
 if(item.getItemId() == R.id.*main\_all\_user\_btn*){  
 Intent intent = new Intent(MainActivity.this, UsersActivity.class);  
 startActivity(intent);  
 }  
 return true;  
 }  
}

1. Colors.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="colorPrimary">#673bb7</color>  
 <color name="colorPrimaryDark">#512da7</color>  
 <color name="colorAccent">#ff5521</color>  
 <color name="colorProfBg">#3A3837</color>  
 <color name="colorGray">#AAA7A5</color>  
</resources>

1. ChatActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Context;  
import android.os.Build;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.TextView;  
  
import androidx.appcompat.widget.Toolbar;  
  
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;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ChatActivity extends AppCompatActivity {  
  
 private String mChatUser;  
 private String mChatUsername;  
 private Toolbar mToolBar;  
 private DatabaseReference mRootRef;  
  
 private TextView mTitleView, mLastSeenView;  
 private CircleImageView mProfileImage;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_chat*);  
  
 mRootRef = FirebaseDatabase.*getInstance*().getReference();  
  
 mChatUser = getIntent().getStringExtra("user\_id");  
 mChatUsername = getIntent().getStringExtra("user\_name");  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolBar = (Toolbar) findViewById(R.id.*chat\_app\_bar*);  
 setSupportActionBar(mToolBar);  
 }  
 ActionBar actionBar = getSupportActionBar();  
  
 actionBar.setDisplayHomeAsUpEnabled(true);  
// actionBar.setTitle(mChatUsername);  
 actionBar.setDisplayShowCustomEnabled(true);  
  
 LayoutInflater layoutInflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 View action\_bar\_view = layoutInflater.inflate(R.layout.*chat\_custom\_bar*, null);  
  
 actionBar.setCustomView(action\_bar\_view);  
  
 // -------------- CUSTOM ACTION BAR ITEMS -----------------  
 mTitleView = (TextView) findViewById(R.id.*custom\_bar\_title*);  
 mLastSeenView = (TextView) findViewById(R.id.*custom\_bar\_seen*);  
 mProfileImage = (CircleImageView) findViewById(R.id.*custom\_bar\_image*);  
  
 mTitleView.setText(mChatUsername);  
  
 mRootRef.child("Users").child(mChatUser).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String online = dataSnapshot.child("online").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
  
 if(online.equals("true")){  
 mLastSeenView.setText("Online");  
 }else {  
 mLastSeenView.setText(online);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
}

1. Chat\_custom\_bar.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:background="@color/colorPrimary"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Display Name"  
 android:textColor="@android:color/white"  
 android:textSize="18dp"  
 android:id="@+id/custom\_bar\_title"/>  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Last Seen"  
 android:textSize="13dp"  
 android:textColor="@android:color/white"  
 android:layout\_below="@+id/custom\_bar\_title"  
 android:id="@+id/custom\_bar\_seen"/>  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:layout\_width="36dp"  
 android:layout\_height="36dp"  
 android:src="@mipmap/ic\_default\_avatar"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginRight="10dp"  
 android:id="@+id/custom\_bar\_image"  
 />  
  
</RelativeLayout>

1. FriendsFragment.java

package com.example.firebasechat;  
  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AlertDialog;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
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.Query;  
import com.google.firebase.database.ValueEventListener;  
import com.squareup.picasso.Picasso;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \*/*public class FriendsFragment extends Fragment {  
  
 private RecyclerView mFriendsList;  
 private DatabaseReference mFriendsDatabase;  
 private DatabaseReference mUsersDatabase;  
 private FirebaseAuth mAuth;  
 private String mCurrent\_user\_id;  
 private View mMainView;  
 public FriendsFragment() {  
 // Required empty public constructor  
 }  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
  
 mMainView = inflater.inflate(R.layout.*fragment\_friends*, container, false);  
 mFriendsList = mMainView.findViewById(R.id.*friends\_list*);  
 mAuth = FirebaseAuth.*getInstance*();  
 mCurrent\_user\_id = mAuth.getCurrentUser().getUid();  
  
 mFriendsDatabase = FirebaseDatabase.*getInstance*().getReference().child("Friends").child(mCurrent\_user\_id);  
 mFriendsDatabase.keepSynced(true);  
 mUsersDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users");  
 mUsersDatabase.keepSynced(true);  
  
 mFriendsList.setHasFixedSize(true);  
 mFriendsList.setLayoutManager(new LinearLayoutManager(getContext()));  
  
 // Inflate the layout for this fragment  
 return mMainView;  
 }  
 @Override  
 public void onStart() {  
 super.onStart();  
 //  
 FirebaseRecyclerOptions<Friends> options =  
 new FirebaseRecyclerOptions.Builder<Friends>()  
 .setQuery( mFriendsDatabase, Friends.class)  
 .build();  
 FirebaseRecyclerAdapter friendsRecyclerViewAdapter = new FirebaseRecyclerAdapter<Friends, FriendsViewHolder>(options) {  
  
 @Override  
 protected void onBindViewHolder(@NonNull final FriendsViewHolder friendsViewHolder, int i, @NonNull Friends friends) {  
 friendsViewHolder.setDate(friends.getDate());  
  
 final String list\_user\_id = getRef(i).getKey();  
  
 mUsersDatabase.child(list\_user\_id).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
  
 final String userName = dataSnapshot.child("name").getValue().toString();  
 String userThumb = dataSnapshot.child("thumb\_image").getValue().toString();  
  
 if(dataSnapshot.hasChild("online")) {  
  
 String userOnline = dataSnapshot.child("online").getValue().toString();  
 friendsViewHolder.setUserOnline(userOnline);  
  
 }  
  
 friendsViewHolder.setName(userName);  
 friendsViewHolder.setUserImage(userThumb, getContext());  
 // When Click to Friends View  
 friendsViewHolder.mView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 CharSequence options[] = new CharSequence[]{"Open Profile", "Send message"};  
 final AlertDialog.Builder builder = new AlertDialog.Builder(getContext());  
  
 builder.setTitle("Select Options");  
 builder.setItems(options, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 //Click Event for each item.  
 if(i == 0){  
 Intent profileIntent = new Intent(getContext(), ProfileActivity.class);  
 profileIntent.putExtra("user\_id", list\_user\_id);  
 startActivity(profileIntent);  
  
 }  
  
 if(i == 1){  
  
 Intent chatIntent = new Intent(getContext(), ChatActivity.class);  
 chatIntent.putExtra("user\_id", list\_user\_id);  
 chatIntent.putExtra("user\_name", userName);  
 startActivity(chatIntent);  
 }  
 }  
 });  
 builder.show();  
 }  
 });  
 }  
 @Override  
 public void onCancelled(DatabaseError databaseError) {  
  
 }  
 });  
 }  
 @NonNull  
 @Override  
 public FriendsViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int i) {  
 View view = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*users\_single\_layout*, parent, false);  
 return new FriendsViewHolder(view);  
 }  
 };  
  
 mFriendsList.setAdapter(friendsRecyclerViewAdapter);  
 friendsRecyclerViewAdapter.startListening();  
 }  
  
 public class FriendsViewHolder extends RecyclerView.ViewHolder {  
  
 View mView;  
  
 public FriendsViewHolder(View itemView) {  
 super(itemView);  
  
 mView = itemView;  
 }  
  
 public void setDate(String date){  
 TextView userStatusView = mView.findViewById(R.id.*user\_single\_status*);  
 userStatusView.setText(date);  
 }  
  
 public void setName(String name){  
 TextView userNameView = mView.findViewById(R.id.*user\_single\_name*);  
 userNameView.setText(name);  
 }  
 public void setUserImage(String thumb\_image, Context ctx){  
 CircleImageView userImageView = mView.findViewById(R.id.*user\_single\_image*);  
 Picasso.*get*().load(thumb\_image).placeholder(R.mipmap.*ic\_default\_avatar*).into(userImageView);  
 }  
 public void setUserOnline(String online\_status) {  
 ImageView userOnlineView =mView.findViewById(R.id.*user\_single\_online*);  
 if(online\_status.equals("true")){  
 userOnlineView.setVisibility(View.*VISIBLE*);  
 } else {  
 userOnlineView.setVisibility(View.*INVISIBLE*);  
 }  
 }  
 }  
}

## Add Time Ago – last seen

1. Create new class – GetTimeAgo.java

package com.example.firebasechat;  
  
import android.app.Application;  
import android.content.Context;  
  
public class GetTimeAgo extends Application {  
 private static final int *SECOND\_MILLIS* = 1000;  
 private static final int *MINUTE\_MILLIS* = 60 \* *SECOND\_MILLIS*;  
 private static final int *HOUR\_MILLIS* = 60 \* *MINUTE\_MILLIS*;  
 private static final int *DAY\_MILLIS* = 24 \* *HOUR\_MILLIS*;  
  
  
 public String getTimeAgo(long time, Context ctx) {  
 if (time < 1000000000000L) {  
 // if timestamp given in seconds, convert to millis  
 time \*= 1000;  
 }  
  
 long now = System.*currentTimeMillis*();  
 if (time > now || time <= 0) {  
 return null;  
 }  
  
 // *TODO: localize* final long diff = now - time;  
 if (diff < *MINUTE\_MILLIS*) {  
 return "just now";  
 } else if (diff < 2 \* *MINUTE\_MILLIS*) {  
 return "a minute ago";  
 } else if (diff < 50 \* *MINUTE\_MILLIS*) {  
 return diff / *MINUTE\_MILLIS* + " minutes ago";  
 } else if (diff < 90 \* *MINUTE\_MILLIS*) {  
 return "an hour ago";  
 } else if (diff < 24 \* *HOUR\_MILLIS*) {  
 return diff / *HOUR\_MILLIS* + " hours ago";  
 } else if (diff < 48 \* *HOUR\_MILLIS*) {  
 return "yesterday";  
 } else {  
 return diff / *DAY\_MILLIS* + " days ago";  
 }  
 }  
}

1. ChatActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Context;  
import android.os.Build;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.TextView;  
  
import androidx.appcompat.widget.Toolbar;  
  
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;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ChatActivity extends AppCompatActivity {  
  
 private String mChatUser;  
 private String mChatUsername;  
 private Toolbar mToolBar;  
 private DatabaseReference mRootRef;  
  
 private TextView mTitleView, mLastSeenView;  
 private CircleImageView mProfileImage;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_chat*);  
  
 mRootRef = FirebaseDatabase.*getInstance*().getReference();  
  
 mChatUser = getIntent().getStringExtra("user\_id");  
 mChatUsername = getIntent().getStringExtra("user\_name");  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolBar = (Toolbar) findViewById(R.id.*chat\_app\_bar*);  
 setSupportActionBar(mToolBar);  
 }  
 ActionBar actionBar = getSupportActionBar();  
  
 actionBar.setDisplayHomeAsUpEnabled(true);  
// actionBar.setTitle(mChatUsername);  
 actionBar.setDisplayShowCustomEnabled(true);  
  
 LayoutInflater layoutInflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 View action\_bar\_view = layoutInflater.inflate(R.layout.*chat\_custom\_bar*, null);  
  
 actionBar.setCustomView(action\_bar\_view);  
  
 // -------------- CUSTOM ACTION BAR ITEMS -----------------  
 mTitleView = (TextView) findViewById(R.id.*custom\_bar\_title*);  
 mLastSeenView = (TextView) findViewById(R.id.*custom\_bar\_seen*);  
 mProfileImage = (CircleImageView) findViewById(R.id.*custom\_bar\_image*);  
  
 mTitleView.setText(mChatUsername);  
  
 mRootRef.child("Users").child(mChatUser).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String online = dataSnapshot.child("online").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
  
 if(online.equals("true")){  
 mLastSeenView.setText("Online");  
 }else {  
 GetTimeAgo getTimeAgo = new GetTimeAgo();  
 long lastTime = Long.*parseLong*(online);  
 String lastSeen = getTimeAgo.getTimeAgo(lastTime, getApplicationContext());  
 mLastSeenView.setText(lastSeen);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
}

## Store message in server

1. ChatActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Context;  
import android.os.Build;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.TextView;  
  
import androidx.appcompat.widget.Toolbar;  
  
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.ServerValue;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.HashMap;  
import java.util.Map;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ChatActivity extends AppCompatActivity {  
  
 private String mChatUser;  
 private String mChatUsername;  
 private Toolbar mToolBar;  
 private DatabaseReference mRootRef;  
  
 private TextView mTitleView, mLastSeenView;  
 private CircleImageView mProfileImage;  
 private FirebaseAuth mAuth;  
 private String mCurrentUserId;  
  
 private ImageButton mChatAddBtn;  
 private ImageButton mChatSendBtn;  
 private EditText mChatMessageView;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_chat*);  
  
 mRootRef = FirebaseDatabase.*getInstance*().getReference();  
  
 mAuth = FirebaseAuth.*getInstance*();  
 mCurrentUserId = mAuth.getCurrentUser().getUid();  
  
 mChatUser = getIntent().getStringExtra("user\_id");  
 mChatUsername = getIntent().getStringExtra("user\_name");  
  
 mChatAddBtn = (ImageButton) findViewById(R.id.*chat\_add\_btn*);  
 mChatSendBtn = (ImageButton) findViewById(R.id.*chat\_sent\_btn*);  
 mChatMessageView = (EditText) findViewById(R.id.*chat\_message\_view*);  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolBar = (Toolbar) findViewById(R.id.*chat\_app\_bar*);  
 setSupportActionBar(mToolBar);  
 }  
 ActionBar actionBar = getSupportActionBar();  
  
 actionBar.setDisplayHomeAsUpEnabled(true);  
// actionBar.setTitle(mChatUsername);  
 actionBar.setDisplayShowCustomEnabled(true);  
  
 LayoutInflater layoutInflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 View action\_bar\_view = layoutInflater.inflate(R.layout.*chat\_custom\_bar*, null);  
  
 actionBar.setCustomView(action\_bar\_view);  
  
 // -------------- CUSTOM ACTION BAR ITEMS -----------------  
 mTitleView = (TextView) findViewById(R.id.*custom\_bar\_title*);  
 mLastSeenView = (TextView) findViewById(R.id.*custom\_bar\_seen*);  
 mProfileImage = (CircleImageView) findViewById(R.id.*custom\_bar\_image*);  
  
 mTitleView.setText(mChatUsername);  
  
 mRootRef.child("Users").child(mChatUser).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String online = dataSnapshot.child("online").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
  
 if(online.equals("true")){  
 mLastSeenView.setText("Online");  
 }else {  
 GetTimeAgo getTimeAgo = new GetTimeAgo();  
 long lastTime = Long.*parseLong*(online);  
 String lastSeen = getTimeAgo.getTimeAgo(lastTime, getApplicationContext());  
 mLastSeenView.setText(lastSeen);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 mRootRef.child("Chat").child(mCurrentUserId).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
  
 if(!dataSnapshot.hasChild(mChatUser)){  
 Map chatAddMap = new HashMap();  
 chatAddMap.put("seen", false);  
 chatAddMap.put("timestamp", ServerValue.*TIMESTAMP*);  
  
 Map chatUserMap = new HashMap();  
 chatUserMap.put("Chat/" + mCurrentUserId + "/" + mChatUser, chatAddMap);  
 chatUserMap.put("Chat/" + mChatUser + "/" + mCurrentUserId, chatAddMap);  
  
 mRootRef.updateChildren(chatUserMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 if(databaseError != null){  
 Log.*d*("CHAT\_LOG", databaseError.getMessage().toString());  
 }  
 }  
 });  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void chatSentBtnClick(View view){  
 sendMessage();  
 }  
  
 private void sendMessage() {  
 String message = mChatMessageView.getText().toString();  
 if(!TextUtils.*isEmpty*(message)){  
 String current\_user\_ref = "messages/" + mCurrentUserId + "/" + mChatUser;  
 String chat\_user\_ref = "messages/" + mChatUser + "/" + mCurrentUserId;  
  
 DatabaseReference user\_message\_push = mRootRef.child("message")  
 .child(mCurrentUserId).child(mChatUser).push();  
  
 String push\_id = user\_message\_push.getKey();  
  
 Map messageMap = new HashMap();  
 messageMap.put("message", message);  
 messageMap.put("seen", false);  
 messageMap.put("type", "text");  
 messageMap.put("time", ServerValue.*TIMESTAMP*);  
  
 Map messageUserMap = new HashMap();  
 messageUserMap.put(current\_user\_ref + "/" + push\_id, messageMap);  
 messageUserMap.put(chat\_user\_ref + "/" + push\_id, messageMap);  
  
 mRootRef.updateChildren(messageUserMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 if(databaseError != null){  
 Log.*d*("CHAT\_LOG", databaseError.getMessage().toString());  
 }  
 }  
 });  
 }  
 }  
}

1. Activity\_chat.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@color/colorGray"  
 tools:context=".ChatActivity">  
  
 <include layout="@layout/app\_bar\_layout" android:id="@+id/chat\_app\_bar" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@android:color/white"  
 android:orientation="horizontal"  
  
 android:layout\_alignParentBottom="true"  
 android:gravity="bottom"  
 android:weightSum="10"  
 >  
 <ImageButton  
 android:id="@+id/chat\_add\_btn"  
 android:layout\_width="55dp"  
 android:layout\_height="55dp"  
 android:background="@android:color/white"  
 android:adjustViewBounds="true"  
 android:scaleType="centerCrop"  
 android:src="@mipmap/ic\_plus"  
 android:layout\_weight="1"  
 />  
 <EditText  
 android:id="@+id/chat\_message\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_weight="8"  
 />  
 <ImageButton  
 android:id="@+id/chat\_sent\_btn"  
 android:layout\_width="55dp"  
 android:layout\_height="55dp"  
 android:layout\_weight="1"  
 android:background="@android:color/white"  
 android:adjustViewBounds="true"  
 android:scaleType="centerCrop"  
 android:padding="5dp"  
 android:layout\_marginRight="5dp"  
 android:src="@mipmap/ic\_sent\_round"  
android:onClick="chatSentBtnClick"  
 />  
 </LinearLayout>  
</RelativeLayout>

## AppCrash – update

LapitChat.java

package com.example.firebasechat;  
  
import android.app.Application;  
  
import androidx.annotation.NonNull;  
  
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.ServerValue;  
import com.google.firebase.database.ValueEventListener;  
import com.squareup.picasso.OkHttp3Downloader;  
import com.squareup.picasso.Picasso;  
  
public class LapitChat extends Application {  
  
 private DatabaseReference mUserDatabase;  
 private FirebaseAuth mAuth;  
  
 @Override  
 public void onCreate() {  
 super.onCreate();  
  
 FirebaseDatabase.*getInstance*().setPersistenceEnabled(true);  
  
 // Picasso  
 Picasso.Builder builder = new Picasso.Builder(this);  
 builder.downloader(new OkHttp3Downloader(this, Integer.*MAX\_VALUE*));  
 Picasso build = builder.build();  
 build.setIndicatorsEnabled(true);  
 build.setLoggingEnabled(true);  
 Picasso.*setSingletonInstance*(build);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 if (mAuth.getCurrentUser() != null) {  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(mAuth.getCurrentUser().getUid());  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if (dataSnapshot != null) {  
 mUserDatabase.child("online").onDisconnect().setValue(ServerValue.*TIMESTAMP*);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
 }  
  
  
}

MainActivity.java

package com.example.firebasechat;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
  
import androidx.appcompat.widget.Toolbar;  
import androidx.viewpager.widget.ViewPager;  
  
import com.google.android.material.tabs.TabLayout;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ServerValue;  
  
public class MainActivity extends AppCompatActivity {  
  
 private FirebaseAuth mAuth;  
 private Toolbar mToolbar;  
  
 private ViewPager mViewPager;  
 private SectionsPageAdapter mSectionsPageAdapter;  
 private TabLayout mTabLayout;  
  
 private DatabaseReference mUserRef;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 if(mAuth.getCurrentUser() != null) {  
  
 mUserRef = FirebaseDatabase.*getInstance*().getReference().child("Users").child(mAuth.getCurrentUser().getUid());  
 }  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolbar = (Toolbar) findViewById(R.id.*main\_page\_toolbar*);  
 setSupportActionBar(mToolbar);  
 getSupportActionBar().setTitle("Chat");  
 }  
  
 //TABS  
 mViewPager = (ViewPager) findViewById(R.id.*main\_tabPager*);  
 mSectionsPageAdapter = new SectionsPageAdapter(getSupportFragmentManager());  
 mViewPager.setAdapter(mSectionsPageAdapter);  
  
 mTabLayout =(TabLayout) findViewById(R.id.*main\_tabs*);  
 mTabLayout.setupWithViewPager(mViewPager);  
  
 }  
  
  
  
 public void onStart() {  
 super.onStart();  
 // Check if user is signed in (non-null) and update UI accordingly.  
 FirebaseUser currentUser = mAuth.getCurrentUser();  
 if(currentUser==null){  
 sendToStart();  
 }else {  
 mUserRef.child("online").setValue(true);  
 }  
 }  
  
 @Override  
 protected void onStop() {  
 super.onStop();  
 FirebaseUser currentUser = mAuth.getCurrentUser();  
 if(currentUser!=null) {  
 mUserRef.child("online").setValue(ServerValue.*TIMESTAMP*);  
 }  
 }  
  
 private void sendToStart() {  
 Intent intent = new Intent(MainActivity.this, StartActivity.class);  
 startActivity(intent);  
 finish();  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 super.onCreateOptionsMenu(menu);  
 getMenuInflater().inflate(R.menu.*menu\_main*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 super.onOptionsItemSelected(item);  
 if(item.getItemId() == R.id.*main\_logout\_btn*){  
 FirebaseAuth.*getInstance*().signOut();  
 sendToStart();  
 }  
 if(item.getItemId() == R.id.*main\_settings\_btn*){  
 Intent intent = new Intent(MainActivity.this, SettingsActivity.class);  
 startActivity(intent);  
 }  
  
 if(item.getItemId() == R.id.*main\_all\_user\_btn*){  
 Intent intent = new Intent(MainActivity.this, UsersActivity.class);  
 startActivity(intent);  
 }  
 return true;  
 }  
}

## Retrieve Message

1. Create layout resource file – message\_single\_layout.xml – RelativeLayout
2. Create new drawable resource file –drawable-> right click-> drawable resource file-> message\_text\_background.xml -> root element = shape

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android">  
 <solid android:color="@color/colorPrimary" />  
 <corners android:radius="30dp" />  
</shape>

1. Create new class – Messages.java

package com.example.firebasechat;  
  
public class Messages {  
 private String message, type;  
 private boolean seen;  
 private long time;

public Messages() {  
}

public Messages(String message, boolean seen, long time, String type) {  
 this.message = message;  
 this.seen = seen;  
 this.time = time;  
 this.type = type;  
 }  
  
 public String getMessage() {  
 return message;  
 }  
  
 public void setMessage(String message) {  
 this.message = message;  
 }  
  
 public boolean getSeen() {  
 return seen;  
 }  
  
 public void setSeen(boolean seen) {  
 this.seen = seen;  
 }  
  
 public long getTime() {  
 return time;  
 }  
  
 public void setTime(long time) {  
 this.time = time;  
 }  
  
 public String getType() {  
 return type;  
 }  
  
 public void setType(String type) {  
 this.type = type;  
 }  
}

1. Create new java class – MessageAdapter.java

package com.example.firebasechat;  
  
import android.app.Notification;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.List;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class MessageAdapter extends RecyclerView.Adapter<MessageAdapter.MessageViewHolder> {  
 private List<Messages> mMessageList;  
 public MessageAdapter(List<Messages> mMessageList){  
 this.mMessageList = mMessageList;  
 }  
  
 @NonNull  
 @Override  
 public MessageViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
  
 View v = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*message\_single\_layout*, parent, false);  
 return new MessageViewHolder(v);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull MessageViewHolder holder, int position) {  
 Messages c = mMessageList.get(position);  
 holder.messageText.setText(c.getMessage());  
 }  
  
 @Override  
 public int getItemCount() {  
 return mMessageList.size();  
 }  
  
 public class MessageViewHolder extends RecyclerView.ViewHolder {  
  
 public TextView messageText;  
 public CircleImageView profileImage;  
  
 public MessageViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 messageText = (TextView) itemView.findViewById(R.id.*message\_text\_layout*);  
 profileImage = (CircleImageView) itemView.findViewById(R.id.*message\_profile\_layout*);  
 }  
 }  
}

1. Message\_single\_layout.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/message\_single\_layout"  
 android:padding="10dp">  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/message\_profile\_layout"  
 android:layout\_width="55dp"  
 android:layout\_height="55dp"  
 android:src="@mipmap/ic\_default\_avatar\_round"  
 >  
  
 </de.hdodenhof.circleimageview.CircleImageView>  
  
 <TextView  
 android:id="@+id/message\_text\_layout"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/message\_text\_background"  
 android:layout\_toRightOf="@+id/message\_profile\_layout"  
 android:text="Message"  
 android:padding="10dp"  
 android:textColor="@android:color/white"  
 />  
</RelativeLayout>

1. Activity\_chat.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@color/colorGray"  
 tools:context=".ChatActivity">  
  
 <include layout="@layout/app\_bar\_layout" android:id="@+id/chat\_app\_bar" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/chat\_app\_bar"  
 android:id="@+id/messages\_list"  
 android:layout\_above="@+id/lr\_type\_msg">  
  
 </androidx.recyclerview.widget.RecyclerView>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@android:color/white"  
 android:orientation="horizontal"  
android:id="@+id/lr\_type\_msg"  
 android:layout\_alignParentBottom="true"  
 android:gravity="bottom"  
 android:weightSum="10"  
 >  
 <ImageButton  
 android:id="@+id/chat\_add\_btn"  
 android:layout\_width="55dp"  
 android:layout\_height="55dp"  
 android:background="@android:color/white"  
 android:adjustViewBounds="true"  
 android:scaleType="centerCrop"  
 android:src="@mipmap/ic\_plus"  
 android:layout\_weight="1"  
 />  
 <EditText  
 android:id="@+id/chat\_message\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_weight="8"  
 />  
 <ImageButton  
 android:id="@+id/chat\_sent\_btn"  
 android:layout\_width="55dp"  
 android:layout\_height="55dp"  
 android:layout\_weight="1"  
 android:background="@android:color/white"  
 android:adjustViewBounds="true"  
 android:scaleType="centerCrop"  
 android:padding="5dp"  
 android:layout\_marginRight="5dp"  
 android:src="@mipmap/ic\_sent\_round"  
android:onClick="chatSentBtnClick"  
 />  
 </LinearLayout>  
</RelativeLayout>

1. ChatActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Context;  
import android.os.Build;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.Adapter;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.ChildEventListener;  
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.ServerValue;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ChatActivity extends AppCompatActivity {  
  
 private String mChatUser;  
 private String mChatUsername;  
 private Toolbar mToolBar;  
 private DatabaseReference mRootRef;  
  
 private TextView mTitleView, mLastSeenView;  
 private CircleImageView mProfileImage;  
 private FirebaseAuth mAuth;  
 private String mCurrentUserId;  
  
 private ImageButton mChatAddBtn;  
 private ImageButton mChatSendBtn;  
 private EditText mChatMessageView;  
  
 private RecyclerView mMessagesList;  
 private final List<Messages> messagesList = new ArrayList<>();  
 private LinearLayoutManager mLinearLayout;  
 private MessageAdapter mAdapter;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_chat*);  
  
 mRootRef = FirebaseDatabase.*getInstance*().getReference();  
  
 mAuth = FirebaseAuth.*getInstance*();  
 mCurrentUserId = mAuth.getCurrentUser().getUid();  
  
 mChatUser = getIntent().getStringExtra("user\_id");  
 mChatUsername = getIntent().getStringExtra("user\_name");  
  
 mChatAddBtn = (ImageButton) findViewById(R.id.*chat\_add\_btn*);  
 mChatSendBtn = (ImageButton) findViewById(R.id.*chat\_sent\_btn*);  
  
 mAdapter = new MessageAdapter(messagesList);  
  
 mChatMessageView = (EditText) findViewById(R.id.*chat\_message\_view*);  
 mMessagesList = (RecyclerView) findViewById(R.id.*messages\_list*);  
  
 mLinearLayout = new LinearLayoutManager(this);  
 mMessagesList.setHasFixedSize(true);  
 mMessagesList.setLayoutManager(mLinearLayout);  
  
 mMessagesList.setAdapter(mAdapter);  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolBar = (Toolbar) findViewById(R.id.*chat\_app\_bar*);  
 setSupportActionBar(mToolBar);  
 }  
 ActionBar actionBar = getSupportActionBar();  
  
 actionBar.setDisplayHomeAsUpEnabled(true);  
// actionBar.setTitle(mChatUsername);  
 actionBar.setDisplayShowCustomEnabled(true);  
  
 LayoutInflater layoutInflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 View action\_bar\_view = layoutInflater.inflate(R.layout.*chat\_custom\_bar*, null);  
  
 actionBar.setCustomView(action\_bar\_view);  
  
 // -------------- CUSTOM ACTION BAR ITEMS -----------------  
 mTitleView = (TextView) findViewById(R.id.*custom\_bar\_title*);  
 mLastSeenView = (TextView) findViewById(R.id.*custom\_bar\_seen*);  
 mProfileImage = (CircleImageView) findViewById(R.id.*custom\_bar\_image*);  
  
 mTitleView.setText(mChatUsername);  
  
 loadMessages();  
  
 mRootRef.child("Users").child(mChatUser).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String online = dataSnapshot.child("online").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
  
 if(online.equals("true")){  
 mLastSeenView.setText("Online");  
 }else {  
 GetTimeAgo getTimeAgo = new GetTimeAgo();  
 long lastTime = Long.*parseLong*(online);  
 String lastSeen = getTimeAgo.getTimeAgo(lastTime, getApplicationContext());  
 mLastSeenView.setText(lastSeen);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 mRootRef.child("Chat").child(mCurrentUserId).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
  
 if(!dataSnapshot.hasChild(mChatUser)){  
 Map chatAddMap = new HashMap();  
 chatAddMap.put("seen", false);  
 chatAddMap.put("timestamp", ServerValue.*TIMESTAMP*);  
  
 Map chatUserMap = new HashMap();  
 chatUserMap.put("Chat/" + mCurrentUserId + "/" + mChatUser, chatAddMap);  
 chatUserMap.put("Chat/" + mChatUser + "/" + mCurrentUserId, chatAddMap);  
  
 mRootRef.updateChildren(chatUserMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 if(databaseError != null){  
 Log.*d*("CHAT\_LOG", databaseError.getMessage().toString());  
 }  
 }  
 });  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 private void loadMessages() {  
 mRootRef.child("messages").child(mCurrentUserId).child(mChatUser).addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
 Messages messages = dataSnapshot.getValue(Messages.class);  
  
 messagesList.add(messages);  
 mAdapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void chatSentBtnClick(View view){  
 sendMessage();  
 }  
  
 private void sendMessage() {  
 String message = mChatMessageView.getText().toString();  
 if(!TextUtils.*isEmpty*(message)){  
 String current\_user\_ref = "messages/" + mCurrentUserId + "/" + mChatUser;  
 String chat\_user\_ref = "messages/" + mChatUser + "/" + mCurrentUserId;  
  
 DatabaseReference user\_message\_push = mRootRef.child("message")  
 .child(mCurrentUserId).child(mChatUser).push();  
  
 String push\_id = user\_message\_push.getKey();  
  
 Map messageMap = new HashMap();  
 messageMap.put("message", message);  
 messageMap.put("seen", false);  
 messageMap.put("type", "text");  
 messageMap.put("time", ServerValue.*TIMESTAMP*);  
  
 Map messageUserMap = new HashMap();  
 messageUserMap.put(current\_user\_ref + "/" + push\_id, messageMap);  
 messageUserMap.put(chat\_user\_ref + "/" + push\_id, messageMap);  
  
 mRootRef.updateChildren(messageUserMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 if(databaseError != null){  
 Log.*d*("CHAT\_LOG", databaseError.getMessage().toString());  
 }  
 }  
 });  
 }  
 }  
}

## Scroll Last message

1. MessageActivity.java –update

package com.example.firebasechat;  
  
public class Messages {  
 private String message, type, from;  
 private boolean seen;  
 private long time;  
  
 public Messages() {  
 }  
  
 public Messages(String message, String type, String from, boolean seen, long time) {  
 this.message = message;  
 this.type = type;  
 this.from = from;  
 this.seen = seen;  
 this.time = time;  
 }  
  
 public String getFrom() {  
 return from;  
 }  
  
 public void setFrom(String from) {  
 this.from = from;  
 }  
  
 public boolean isSeen() {  
 return seen;  
 }  
  
 public void setSeen(boolean seen) {  
 this.seen = seen;  
 }  
  
 public String getMessage() {  
 return message;  
 }  
  
 public void setMessage(String message) {  
 this.message = message;  
 }  
  
  
  
 public long getTime() {  
 return time;  
 }  
  
 public void setTime(long time) {  
 this.time = time;  
 }  
  
 public String getType() {  
 return type;  
 }  
  
 public void setType(String type) {  
 this.type = type;  
 }  
}

1. ChatActivity.java –update

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Context;  
import android.os.Build;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.Adapter;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.ChildEventListener;  
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.ServerValue;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ChatActivity extends AppCompatActivity {  
  
 private String mChatUser;  
 private String mChatUsername;  
 private Toolbar mToolBar;  
 private DatabaseReference mRootRef;  
  
 private TextView mTitleView, mLastSeenView;  
 private CircleImageView mProfileImage;  
 private FirebaseAuth mAuth;  
 private String mCurrentUserId;  
  
 private ImageButton mChatAddBtn;  
 private ImageButton mChatSendBtn;  
 private EditText mChatMessageView;  
  
 private RecyclerView mMessagesList;  
 private final List<Messages> messagesList = new ArrayList<>();  
 private LinearLayoutManager mLinearLayout;  
 private MessageAdapter mAdapter;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_chat*);  
  
 mRootRef = FirebaseDatabase.*getInstance*().getReference();  
  
 mAuth = FirebaseAuth.*getInstance*();  
 mCurrentUserId = mAuth.getCurrentUser().getUid();  
  
 mChatUser = getIntent().getStringExtra("user\_id");  
 mChatUsername = getIntent().getStringExtra("user\_name");  
  
 mChatAddBtn = (ImageButton) findViewById(R.id.*chat\_add\_btn*);  
 mChatSendBtn = (ImageButton) findViewById(R.id.*chat\_sent\_btn*);  
  
 mAdapter = new MessageAdapter(messagesList);  
  
 mChatMessageView = (EditText) findViewById(R.id.*chat\_message\_view*);  
 mMessagesList = (RecyclerView) findViewById(R.id.*messages\_list*);  
  
 mLinearLayout = new LinearLayoutManager(this);  
 mMessagesList.setHasFixedSize(true);  
 mMessagesList.setLayoutManager(mLinearLayout);  
  
 mMessagesList.setAdapter(mAdapter);  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolBar = (Toolbar) findViewById(R.id.*chat\_app\_bar*);  
 setSupportActionBar(mToolBar);  
 }  
 ActionBar actionBar = getSupportActionBar();  
  
 actionBar.setDisplayHomeAsUpEnabled(true);  
// actionBar.setTitle(mChatUsername);  
 actionBar.setDisplayShowCustomEnabled(true);  
  
 LayoutInflater layoutInflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 View action\_bar\_view = layoutInflater.inflate(R.layout.*chat\_custom\_bar*, null);  
  
 actionBar.setCustomView(action\_bar\_view);  
  
 // -------------- CUSTOM ACTION BAR ITEMS -----------------  
 mTitleView = (TextView) findViewById(R.id.*custom\_bar\_title*);  
 mLastSeenView = (TextView) findViewById(R.id.*custom\_bar\_seen*);  
 mProfileImage = (CircleImageView) findViewById(R.id.*custom\_bar\_image*);  
  
 mTitleView.setText(mChatUsername);  
  
 loadMessages();  
  
 mRootRef.child("Users").child(mChatUser).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String online = dataSnapshot.child("online").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
  
 if(online.equals("true")){  
 mLastSeenView.setText("Online");  
 }else {  
 GetTimeAgo getTimeAgo = new GetTimeAgo();  
 long lastTime = Long.*parseLong*(online);  
 String lastSeen = getTimeAgo.getTimeAgo(lastTime, getApplicationContext());  
 mLastSeenView.setText(lastSeen);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 mRootRef.child("Chat").child(mCurrentUserId).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
  
 if(!dataSnapshot.hasChild(mChatUser)){  
 Map chatAddMap = new HashMap();  
 chatAddMap.put("seen", false);  
 chatAddMap.put("timestamp", ServerValue.*TIMESTAMP*);  
  
 Map chatUserMap = new HashMap();  
 chatUserMap.put("Chat/" + mCurrentUserId + "/" + mChatUser, chatAddMap);  
 chatUserMap.put("Chat/" + mChatUser + "/" + mCurrentUserId, chatAddMap);  
  
 mRootRef.updateChildren(chatUserMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 if(databaseError != null){  
 Log.*d*("CHAT\_LOG", databaseError.getMessage().toString());  
 }  
 }  
 });  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 private void loadMessages() {  
 mRootRef.child("messages").child(mCurrentUserId).child(mChatUser).addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
 Messages messages = dataSnapshot.getValue(Messages.class);  
  
 messagesList.add(messages);  
 mAdapter.notifyDataSetChanged();  
  
 mMessagesList.scrollToPosition(messagesList.size() - 1 );  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void chatSentBtnClick(View view){  
 sendMessage();  
 }  
  
 private void sendMessage() {  
 String message = mChatMessageView.getText().toString();  
 if(!TextUtils.*isEmpty*(message)){  
 String current\_user\_ref = "messages/" + mCurrentUserId + "/" + mChatUser;  
 String chat\_user\_ref = "messages/" + mChatUser + "/" + mCurrentUserId;  
  
 DatabaseReference user\_message\_push = mRootRef.child("message")  
 .child(mCurrentUserId).child(mChatUser).push();  
  
 String push\_id = user\_message\_push.getKey();  
  
 Map messageMap = new HashMap();  
 messageMap.put("message", message);  
 messageMap.put("seen", false);  
 messageMap.put("type", "text");  
 messageMap.put("time", ServerValue.*TIMESTAMP*);  
 messageMap.put("from", mCurrentUserId);  
  
 mChatMessageView.setText("");  
  
 Map messageUserMap = new HashMap();  
 messageUserMap.put(current\_user\_ref + "/" + push\_id, messageMap);  
 messageUserMap.put(chat\_user\_ref + "/" + push\_id, messageMap);  
  
 mRootRef.updateChildren(messageUserMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 if(databaseError != null){  
 Log.*d*("CHAT\_LOG", databaseError.getMessage().toString());  
 }  
 }  
 });  
 }  
 }  
}

1. MessageAdapter.java – update

package com.example.firebasechat;  
  
import android.app.Notification;  
import android.graphics.Color;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.google.firebase.auth.FirebaseAuth;  
  
import java.util.List;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class MessageAdapter extends RecyclerView.Adapter<MessageAdapter.MessageViewHolder> {  
 private List<Messages> mMessageList;  
 private FirebaseAuth mAuth;  
  
 public MessageAdapter(List<Messages> mMessageList){  
 this.mMessageList = mMessageList;  
 }  
  
 @NonNull  
 @Override  
 public MessageViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
  
 View v = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*message\_single\_layout*, parent, false);  
 return new MessageViewHolder(v);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull MessageViewHolder holder, int position) {  
 mAuth = FirebaseAuth.*getInstance*();  
  
 String current\_user\_id = mAuth.getCurrentUser().getUid();  
  
 Messages c = mMessageList.get(position);  
 String from\_user = c.getFrom();  
  
 if(from\_user.equals(current\_user\_id)){  
 holder.messageText.setBackgroundColor(Color.*WHITE*);  
 holder.messageText.setTextColor(Color.*BLACK*);  
 }else {  
 holder.messageText.setBackgroundResource(R.drawable.*message\_text\_background*);  
 holder.messageText.setTextColor(Color.*WHITE*);  
 }  
 holder.messageText.setText(c.getMessage());  
 }  
  
 @Override  
 public int getItemCount() {  
 return mMessageList.size();  
 }  
  
 public class MessageViewHolder extends RecyclerView.ViewHolder {  
  
 public TextView messageText;  
 public CircleImageView profileImage;  
  
 public MessageViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 messageText = (TextView) itemView.findViewById(R.id.*message\_text\_layout*);  
 profileImage = (CircleImageView) itemView.findViewById(R.id.*message\_profile\_layout*);  
 }  
 }  
}

1. Message\_single\_layout.xml - update

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/message\_single\_layout"  
 android:padding="10dp">  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/message\_profile\_layout"  
 android:layout\_width="35dp"  
 android:layout\_height="35dp"  
 android:src="@mipmap/ic\_default\_avatar\_round"  
 >  
  
 </de.hdodenhof.circleimageview.CircleImageView>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/name\_text\_layout"  
 android:text="@string/display\_name"  
 android:textColor="@android:color/black"  
 android:textSize="20dp"  
 android:layout\_toRightOf="@+id/message\_profile\_layout"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/time\_text\_layout"  
 android:layout\_marginLeft="5dp"  
 android:text="@string/\_00\_00"  
 android:layout\_toRightOf="@+id/name\_text\_layout"  
 />  
  
 <TextView  
 android:id="@+id/message\_text\_layout"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_toRightOf="@+id/message\_profile\_layout"  
 android:layout\_below="@+id/name\_text\_layout"  
 android:text="@string/message\_text"  
 android:layout\_marginLeft="7dp"  
 />  
</RelativeLayout>

1. Activity\_chat.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@color/colorGray"  
 tools:context=".ChatActivity">  
  
 <include layout="@layout/app\_bar\_layout" android:id="@+id/chat\_app\_bar" />  
 <androidx.swiperefreshlayout.widget.SwipeRefreshLayout  
 android:id="@+id/message\_swipe\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/chat\_app\_bar"  
 android:layout\_above="@+id/lr\_type\_msg"  
 >  
 <androidx.recyclerview.widget.RecyclerView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
  
 android:id="@+id/messages\_list"  
 >  
  
 </androidx.recyclerview.widget.RecyclerView>  
 </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@android:color/white"  
 android:orientation="horizontal"  
android:id="@+id/lr\_type\_msg"  
 android:layout\_alignParentBottom="true"  
 android:gravity="bottom"  
 android:weightSum="10"  
 >  
 <ImageButton  
 android:id="@+id/chat\_add\_btn"  
 android:layout\_width="55dp"  
 android:layout\_height="55dp"  
 android:background="@android:color/white"  
 android:adjustViewBounds="true"  
 android:scaleType="centerCrop"  
 android:src="@mipmap/ic\_plus"  
 android:layout\_weight="1"  
 />  
 <EditText  
 android:id="@+id/chat\_message\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_weight="8"  
 />  
 <ImageButton  
 android:id="@+id/chat\_sent\_btn"  
 android:layout\_width="55dp"  
 android:layout\_height="55dp"  
 android:layout\_weight="1"  
 android:background="@android:color/white"  
 android:adjustViewBounds="true"  
 android:scaleType="centerCrop"  
 android:padding="5dp"  
 android:layout\_marginRight="5dp"  
 android:src="@mipmap/ic\_sent\_round"  
android:onClick="chatSentBtnClick"  
 />  
 </LinearLayout>  
</RelativeLayout>

## Remeve Repeated Message Loading

1. ChatActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Context;  
import android.os.Build;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.Adapter;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.ChildEventListener;  
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.Query;  
import com.google.firebase.database.ServerValue;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ChatActivity extends AppCompatActivity {  
  
 private String mChatUser;  
 private String mChatUsername;  
 private Toolbar mToolBar;  
 private DatabaseReference mRootRef;  
  
 private TextView mTitleView, mLastSeenView;  
 private CircleImageView mProfileImage;  
 private FirebaseAuth mAuth;  
 private String mCurrentUserId;  
  
 private ImageButton mChatAddBtn;  
 private ImageButton mChatSendBtn;  
 private EditText mChatMessageView;  
  
 private RecyclerView mMessagesList;  
 private SwipeRefreshLayout mRefreshLayout;  
  
 private final List<Messages> messagesList = new ArrayList<>();  
 private LinearLayoutManager mLinearLayout;  
 private MessageAdapter mAdapter;  
  
 private static final int *TOTAL\_ITEMS\_TO\_LOAD* = 10;  
 private int mCurrentPage = 1;  
  
 // ---new solution  
 private int itemPos = 0;  
 private String mLastKey = "";  
 private String mPrevKey = "";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_chat*);  
  
 mRootRef = FirebaseDatabase.*getInstance*().getReference();  
  
 mAuth = FirebaseAuth.*getInstance*();  
 mCurrentUserId = mAuth.getCurrentUser().getUid();  
  
 mChatUser = getIntent().getStringExtra("user\_id");  
 mChatUsername = getIntent().getStringExtra("user\_name");  
  
 mChatAddBtn = (ImageButton) findViewById(R.id.*chat\_add\_btn*);  
 mChatSendBtn = (ImageButton) findViewById(R.id.*chat\_sent\_btn*);  
  
 mAdapter = new MessageAdapter(messagesList);  
  
 mChatMessageView = (EditText) findViewById(R.id.*chat\_message\_view*);  
 mMessagesList = (RecyclerView) findViewById(R.id.*messages\_list*);  
 mRefreshLayout = (SwipeRefreshLayout) findViewById(R.id.*message\_swipe\_layout*);  
  
 mLinearLayout = new LinearLayoutManager(this);  
 mMessagesList.setHasFixedSize(true);  
 mMessagesList.setLayoutManager(mLinearLayout);  
  
 mMessagesList.setAdapter(mAdapter);  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolBar = (Toolbar) findViewById(R.id.*chat\_app\_bar*);  
 setSupportActionBar(mToolBar);  
 }  
 ActionBar actionBar = getSupportActionBar();  
  
 actionBar.setDisplayHomeAsUpEnabled(true);  
// actionBar.setTitle(mChatUsername);  
 actionBar.setDisplayShowCustomEnabled(true);  
  
 LayoutInflater layoutInflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 View action\_bar\_view = layoutInflater.inflate(R.layout.*chat\_custom\_bar*, null);  
  
 actionBar.setCustomView(action\_bar\_view);  
  
 // -------------- CUSTOM ACTION BAR ITEMS -----------------  
 mTitleView = (TextView) findViewById(R.id.*custom\_bar\_title*);  
 mLastSeenView = (TextView) findViewById(R.id.*custom\_bar\_seen*);  
 mProfileImage = (CircleImageView) findViewById(R.id.*custom\_bar\_image*);  
  
 mTitleView.setText(mChatUsername);  
  
 loadMessages();  
  
 mRootRef.child("Users").child(mChatUser).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String online = dataSnapshot.child("online").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
  
 if(online.equals("true")){  
 mLastSeenView.setText("Online");  
 }else {  
 GetTimeAgo getTimeAgo = new GetTimeAgo();  
 long lastTime = Long.*parseLong*(online);  
 String lastSeen = getTimeAgo.getTimeAgo(lastTime, getApplicationContext());  
 mLastSeenView.setText(lastSeen);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 mRootRef.child("Chat").child(mCurrentUserId).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
  
 if(!dataSnapshot.hasChild(mChatUser)){  
 Map chatAddMap = new HashMap();  
 chatAddMap.put("seen", false);  
 chatAddMap.put("timestamp", ServerValue.*TIMESTAMP*);  
  
 Map chatUserMap = new HashMap();  
 chatUserMap.put("Chat/" + mCurrentUserId + "/" + mChatUser, chatAddMap);  
 chatUserMap.put("Chat/" + mChatUser + "/" + mCurrentUserId, chatAddMap);  
  
 mRootRef.updateChildren(chatUserMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 if(databaseError != null){  
 Log.*d*("CHAT\_LOG", databaseError.getMessage().toString());  
 }  
 }  
 });  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 mRefreshLayout.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {  
 @Override  
 public void onRefresh() {  
 mCurrentPage ++;  
 itemPos = 0;  
 loadMoreMessages();  
  
 }  
 });  
 }  
  
 private void loadMoreMessages() {  
 DatabaseReference messageRef = mRootRef.child("messages").child(mCurrentUserId).child(mChatUser);  
 Query mMessageQuery = messageRef.orderByKey().endAt(mLastKey).limitToLast(*TOTAL\_ITEMS\_TO\_LOAD*);  
  
 mMessageQuery.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
 Messages messages = dataSnapshot.getValue(Messages.class);  
  
 String messageKey = dataSnapshot.getKey();  
// itemPos++;  
 if(!mPrevKey.equals(messageKey)){  
 messagesList.add(itemPos++,messages);  
 }else {  
 mPrevKey = mLastKey;  
 }  
  
 if(itemPos == 1){  
 mLastKey = messageKey;  
 }  
  
  
 mAdapter.notifyDataSetChanged();  
  
 mRefreshLayout.setRefreshing(false);  
 mLinearLayout.scrollToPositionWithOffset(10, 0);  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 private void loadMessages() {  
  
 DatabaseReference messageRef = mRootRef.child("messages").child(mCurrentUserId).child(mChatUser);  
 Query mMessageQuery = messageRef.limitToLast(mCurrentPage \* *TOTAL\_ITEMS\_TO\_LOAD*);  
   
 mMessageQuery.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
 Messages messages = dataSnapshot.getValue(Messages.class);  
  
 itemPos++;  
  
 if(itemPos == 1){  
 mLastKey = dataSnapshot.getKey();  
 mPrevKey = mLastKey;  
 }  
 messagesList.add(messages);  
 mAdapter.notifyDataSetChanged();  
  
 mMessagesList.scrollToPosition(messagesList.size() - 1 );  
  
 mRefreshLayout.setRefreshing(false);  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void chatSentBtnClick(View view){  
 sendMessage();  
 }  
  
 private void sendMessage() {  
 String message = mChatMessageView.getText().toString();  
 if(!TextUtils.*isEmpty*(message)){  
 String current\_user\_ref = "messages/" + mCurrentUserId + "/" + mChatUser;  
 String chat\_user\_ref = "messages/" + mChatUser + "/" + mCurrentUserId;  
  
 DatabaseReference user\_message\_push = mRootRef.child("message")  
 .child(mCurrentUserId).child(mChatUser).push();  
  
 String push\_id = user\_message\_push.getKey();  
  
 Map messageMap = new HashMap();  
 messageMap.put("message", message);  
 messageMap.put("seen", false);  
 messageMap.put("type", "text");  
 messageMap.put("time", ServerValue.*TIMESTAMP*);  
 messageMap.put("from", mCurrentUserId);  
  
 mChatMessageView.setText("");  
  
 Map messageUserMap = new HashMap();  
 messageUserMap.put(current\_user\_ref + "/" + push\_id, messageMap);  
 messageUserMap.put(chat\_user\_ref + "/" + push\_id, messageMap);  
  
 mRootRef.updateChildren(messageUserMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 if(databaseError != null){  
 Log.*d*("CHAT\_LOG", databaseError.getMessage().toString());  
 }  
 }  
 });  
 }  
 }  
}

1. Message\_single\_layout.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/message\_single\_layout"  
 android:padding="10dp">  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/message\_profile\_layout"  
 android:layout\_width="35dp"  
 android:layout\_height="35dp"  
 android:src="@mipmap/ic\_default\_avatar\_round"  
 >  
  
 </de.hdodenhof.circleimageview.CircleImageView>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/name\_text\_layout"  
 android:text="@string/display\_name"  
 android:textColor="@android:color/black"  
 android:textSize="20dp"  
 android:layout\_toRightOf="@+id/message\_profile\_layout"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/time\_text\_layout"  
 android:layout\_marginLeft="5dp"  
 android:text="@string/\_00\_00"  
 android:layout\_toRightOf="@+id/name\_text\_layout"  
 />  
  
 <TextView  
 android:id="@+id/message\_text\_layout"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_toRightOf="@+id/message\_profile\_layout"  
 android:layout\_below="@+id/name\_text\_layout"  
 android:text="@string/message\_text"  
 android:layout\_marginLeft="7dp"  
 />  
</RelativeLayout>

## Share images chat

1. MessageAdapter

package com.example.firebasechat;  
  
import android.app.Notification;  
import android.graphics.Color;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
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;  
import com.squareup.picasso.Picasso;  
  
import java.util.List;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class MessageAdapter extends RecyclerView.Adapter<MessageAdapter.MessageViewHolder> {  
 private List<Messages> mMessageList;  
 private FirebaseAuth mAuth;  
 private DatabaseReference mUserDatabase;  
  
 public MessageAdapter(List<Messages> mMessageList){  
 this.mMessageList = mMessageList;  
 }  
  
 @NonNull  
 @Override  
 public MessageViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
  
 View v = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*message\_single\_layout*, parent, false);  
 return new MessageViewHolder(v);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull final MessageViewHolder holder, int position) {  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
  
 String current\_user\_id = mAuth.getCurrentUser().getUid();  
  
  
 Messages c = mMessageList.get(position);  
 String from\_user = c.getFrom();  
 String message\_type = c.getType();  
 mUserDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users").child(from\_user);  
 mUserDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String name = dataSnapshot.child("name").getValue().toString();  
 String image = dataSnapshot.child("thumb\_image").getValue().toString();  
  
 holder.displayName.setText(name);  
 if(!image.equals("Default")) {  
 Picasso.*get*().load(image).placeholder(R.mipmap.*ic\_default\_avatar\_round*).into(holder.profileImage);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 if(message\_type.equals("text")){  
 holder.messageText.setText(c.getMessage());  
 holder.messageImage.setVisibility(View.*INVISIBLE*);  
  
 if(from\_user.equals(current\_user\_id)){  
 holder.messageText.setBackgroundColor(Color.*WHITE*);  
 holder.messageText.setTextColor(Color.*BLACK*);  
 }else {  
 holder.messageText.setBackgroundResource(R.drawable.*message\_text\_background*);  
 holder.messageText.setTextColor(Color.*WHITE*);  
 }  
 }else {  
 holder.messageImage.setVisibility(View.*VISIBLE*);  
 if(!c.getMessage().equals("Default")) {  
 Picasso.*get*().load(c.getMessage()).placeholder(R.mipmap.*ic\_default\_avatar\_round*).into(holder.messageImage);  
 }  
 }  
  
  
// holder.displayName.setText(c.getFrom());  
 }  
  
 @Override  
 public int getItemCount() {  
 return mMessageList.size();  
 }  
  
 public class MessageViewHolder extends RecyclerView.ViewHolder {  
  
 public TextView messageText, displayName;  
 public CircleImageView profileImage;  
 public ImageView messageImage;  
  
 public MessageViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 messageText = (TextView) itemView.findViewById(R.id.*message\_text\_layout*);  
 displayName = (TextView) itemView.findViewById(R.id.*name\_text\_layout*);  
 profileImage = (CircleImageView) itemView.findViewById(R.id.*message\_profile\_layout*);  
 messageImage = (ImageView) itemView.findViewById(R.id.*message\_image\_layout*);  
 }  
 }  
}

1. Activity\_chat.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@color/colorGray"  
 tools:context=".ChatActivity">  
  
 <include layout="@layout/app\_bar\_layout" android:id="@+id/chat\_app\_bar" />  
 <androidx.swiperefreshlayout.widget.SwipeRefreshLayout  
 android:id="@+id/message\_swipe\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/chat\_app\_bar"  
 android:layout\_above="@+id/lr\_type\_msg"  
 >  
 <androidx.recyclerview.widget.RecyclerView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
  
 android:id="@+id/messages\_list"  
 >  
  
 </androidx.recyclerview.widget.RecyclerView>  
 </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@android:color/white"  
 android:orientation="horizontal"  
android:id="@+id/lr\_type\_msg"  
 android:layout\_alignParentBottom="true"  
 android:gravity="bottom"  
 android:weightSum="10"  
 >  
 <ImageButton  
 android:id="@+id/chat\_add\_btn"  
 android:layout\_width="55dp"  
 android:layout\_height="55dp"  
 android:background="@android:color/white"  
 android:adjustViewBounds="true"  
 android:scaleType="centerCrop"  
 android:src="@mipmap/ic\_plus"  
 android:onClick="plusBtnClick"  
 android:layout\_weight="1"  
 />  
 <EditText  
 android:id="@+id/chat\_message\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_weight="8"  
 />  
 <ImageButton  
 android:id="@+id/chat\_sent\_btn"  
 android:layout\_width="55dp"  
 android:layout\_height="55dp"  
 android:layout\_weight="1"  
 android:background="@android:color/white"  
 android:adjustViewBounds="true"  
 android:scaleType="centerCrop"  
 android:padding="5dp"  
 android:layout\_marginRight="5dp"  
 android:src="@mipmap/ic\_sent\_round"  
android:onClick="chatSentBtnClick"  
 />  
 </LinearLayout>  
</RelativeLayout>

1. Message\_single\_layout.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/message\_single\_layout"  
 android:padding="10dp">  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/message\_profile\_layout"  
 android:layout\_width="35dp"  
 android:layout\_height="35dp"  
 android:src="@mipmap/ic\_default\_avatar\_round"  
 >  
  
 </de.hdodenhof.circleimageview.CircleImageView>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/name\_text\_layout"  
 android:text="@string/display\_name"  
 android:textColor="@android:color/black"  
 android:textSize="20dp"  
 android:layout\_toRightOf="@+id/message\_profile\_layout"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/time\_text\_layout"  
 android:layout\_marginLeft="5dp"  
 android:text="@string/\_00\_00"  
 android:layout\_toRightOf="@+id/name\_text\_layout"  
 />  
  
 <TextView  
 android:id="@+id/message\_text\_layout"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_toRightOf="@+id/message\_profile\_layout"  
 android:layout\_below="@+id/name\_text\_layout"  
 android:text="@string/message\_text"  
 android:layout\_marginLeft="7dp"  
 />  
 <ImageView  
 android:id="@+id/message\_image\_layout"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:scaleType="centerCrop"  
 android:layout\_toRightOf="@+id/message\_profile\_layout"  
 android:layout\_below="@id/name\_text\_layout"  
 />  
</RelativeLayout>

1. ChatActivity.java

package com.example.firebasechat;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Context;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Build;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.Adapter;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.ChildEventListener;  
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.Query;  
import com.google.firebase.database.ServerValue;  
import com.google.firebase.database.ValueEventListener;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
import com.google.firebase.storage.UploadTask;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ChatActivity extends AppCompatActivity {  
  
 private static final int *GALLERY\_PICK* = 1;  
 private String mChatUser;  
 private String mChatUsername;  
 private Toolbar mToolBar;  
 private DatabaseReference mRootRef;  
  
 private TextView mTitleView, mLastSeenView;  
 private CircleImageView mProfileImage;  
 private FirebaseAuth mAuth;  
 private String mCurrentUserId;  
  
 private ImageButton mChatAddBtn;  
 private ImageButton mChatSendBtn;  
 private EditText mChatMessageView;  
  
 private RecyclerView mMessagesList;  
 private SwipeRefreshLayout mRefreshLayout;  
  
 private final List<Messages> messagesList = new ArrayList<>();  
 private LinearLayoutManager mLinearLayout;  
 private MessageAdapter mAdapter;  
  
 private StorageReference mImageStorage;  
  
 private static final int *TOTAL\_ITEMS\_TO\_LOAD* = 10;  
 private int mCurrentPage = 1;  
  
 // ---new solution  
 private int itemPos = 0;  
 private String mLastKey = "";  
 private String mPrevKey = "";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_chat*);  
  
 mRootRef = FirebaseDatabase.*getInstance*().getReference();  
  
 mImageStorage = FirebaseStorage.*getInstance*().getReference();  
  
 mAuth = FirebaseAuth.*getInstance*();  
 mCurrentUserId = mAuth.getCurrentUser().getUid();  
  
 mChatUser = getIntent().getStringExtra("user\_id");  
 mChatUsername = getIntent().getStringExtra("user\_name");  
  
 mChatAddBtn = (ImageButton) findViewById(R.id.*chat\_add\_btn*);  
 mChatSendBtn = (ImageButton) findViewById(R.id.*chat\_sent\_btn*);  
  
 mAdapter = new MessageAdapter(messagesList);  
  
 mChatMessageView = (EditText) findViewById(R.id.*chat\_message\_view*);  
 mMessagesList = (RecyclerView) findViewById(R.id.*messages\_list*);  
 mRefreshLayout = (SwipeRefreshLayout) findViewById(R.id.*message\_swipe\_layout*);  
  
 mLinearLayout = new LinearLayoutManager(this);  
 mMessagesList.setHasFixedSize(true);  
 mMessagesList.setLayoutManager(mLinearLayout);  
  
 mMessagesList.setAdapter(mAdapter);  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 mToolBar = (Toolbar) findViewById(R.id.*chat\_app\_bar*);  
 setSupportActionBar(mToolBar);  
 }  
 ActionBar actionBar = getSupportActionBar();  
  
 actionBar.setDisplayHomeAsUpEnabled(true);  
// actionBar.setTitle(mChatUsername);  
 actionBar.setDisplayShowCustomEnabled(true);  
  
 LayoutInflater layoutInflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 View action\_bar\_view = layoutInflater.inflate(R.layout.*chat\_custom\_bar*, null);  
  
 actionBar.setCustomView(action\_bar\_view);  
  
 // -------------- CUSTOM ACTION BAR ITEMS -----------------  
 mTitleView = (TextView) findViewById(R.id.*custom\_bar\_title*);  
 mLastSeenView = (TextView) findViewById(R.id.*custom\_bar\_seen*);  
 mProfileImage = (CircleImageView) findViewById(R.id.*custom\_bar\_image*);  
  
 mTitleView.setText(mChatUsername);  
  
 loadMessages();  
  
 mRootRef.child("Users").child(mChatUser).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 String online = dataSnapshot.child("online").getValue().toString();  
 String image = dataSnapshot.child("image").getValue().toString();  
  
 if(online.equals("true")){  
 mLastSeenView.setText("Online");  
 }else {  
 GetTimeAgo getTimeAgo = new GetTimeAgo();  
 long lastTime = Long.*parseLong*(online);  
 String lastSeen = getTimeAgo.getTimeAgo(lastTime, getApplicationContext());  
 mLastSeenView.setText(lastSeen);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 mRootRef.child("Chat").child(mCurrentUserId).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
  
 if(!dataSnapshot.hasChild(mChatUser)){  
 Map chatAddMap = new HashMap();  
 chatAddMap.put("seen", false);  
 chatAddMap.put("timestamp", ServerValue.*TIMESTAMP*);  
  
 Map chatUserMap = new HashMap();  
 chatUserMap.put("Chat/" + mCurrentUserId + "/" + mChatUser, chatAddMap);  
 chatUserMap.put("Chat/" + mChatUser + "/" + mCurrentUserId, chatAddMap);  
  
 mRootRef.updateChildren(chatUserMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 if(databaseError != null){  
 Log.*d*("CHAT\_LOG", databaseError.getMessage().toString());  
 }  
 }  
 });  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 mRefreshLayout.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {  
 @Override  
 public void onRefresh() {  
 mCurrentPage ++;  
 itemPos = 0;  
 loadMoreMessages();  
  
 }  
 });  
 }  
  
 private void loadMoreMessages() {  
 DatabaseReference messageRef = mRootRef.child("messages").child(mCurrentUserId).child(mChatUser);  
 Query mMessageQuery = messageRef.orderByKey().endAt(mLastKey).limitToLast(*TOTAL\_ITEMS\_TO\_LOAD*);  
  
 mMessageQuery.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
 Messages messages = dataSnapshot.getValue(Messages.class);  
  
 String messageKey = dataSnapshot.getKey();  
// itemPos++;  
 if(!mPrevKey.equals(messageKey)){  
 messagesList.add(itemPos++,messages);  
 }else {  
 mPrevKey = mLastKey;  
 }  
  
 if(itemPos == 1){  
 mLastKey = messageKey;  
 }  
  
  
 mAdapter.notifyDataSetChanged();  
  
 mRefreshLayout.setRefreshing(false);  
 mLinearLayout.scrollToPositionWithOffset(10, 0);  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 private void loadMessages() {  
  
 DatabaseReference messageRef = mRootRef.child("messages").child(mCurrentUserId).child(mChatUser);  
 Query mMessageQuery = messageRef.limitToLast(mCurrentPage \* *TOTAL\_ITEMS\_TO\_LOAD*);  
   
 mMessageQuery.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
 Messages messages = dataSnapshot.getValue(Messages.class);  
  
 itemPos++;  
  
 if(itemPos == 1){  
 mLastKey = dataSnapshot.getKey();  
 mPrevKey = mLastKey;  
 }  
 messagesList.add(messages);  
 mAdapter.notifyDataSetChanged();  
  
 mMessagesList.scrollToPosition(messagesList.size() - 1 );  
  
 mRefreshLayout.setRefreshing(false);  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
  
 public void chatSentBtnClick(View view){  
 sendMessage();  
 }  
  
 private void sendMessage() {  
 String message = mChatMessageView.getText().toString();  
 if(!TextUtils.*isEmpty*(message)){  
 String current\_user\_ref = "messages/" + mCurrentUserId + "/" + mChatUser;  
 String chat\_user\_ref = "messages/" + mChatUser + "/" + mCurrentUserId;  
  
 DatabaseReference user\_message\_push = mRootRef.child("message")  
 .child(mCurrentUserId).child(mChatUser).push();  
  
 String push\_id = user\_message\_push.getKey();  
  
 Map messageMap = new HashMap();  
 messageMap.put("message", message);  
 messageMap.put("seen", false);  
 messageMap.put("type", "text");  
 messageMap.put("time", ServerValue.*TIMESTAMP*);  
 messageMap.put("from", mCurrentUserId);  
  
 mChatMessageView.setText("");  
  
 Map messageUserMap = new HashMap();  
 messageUserMap.put(current\_user\_ref + "/" + push\_id, messageMap);  
 messageUserMap.put(chat\_user\_ref + "/" + push\_id, messageMap);  
  
 mRootRef.updateChildren(messageUserMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 if(databaseError != null){  
 Log.*d*("CHAT\_LOG", databaseError.getMessage().toString());  
 }  
 }  
 });  
 }  
 }  
  
 public void plusBtnClick(View view){  
 Intent galleryIntent = new Intent();  
 galleryIntent.setType("image/\*");  
 galleryIntent.setAction(Intent.*ACTION\_GET\_CONTENT*);  
  
 startActivityForResult(Intent.*createChooser*(galleryIntent, "SELECT IMAGE"), *GALLERY\_PICK*);  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 if(requestCode == *GALLERY\_PICK* && resultCode == *RESULT\_OK*){  
 Uri imageUri = data.getData();  
  
 final String current\_user\_ref = "messages/" + mCurrentUserId + "/" + mChatUser;  
 final String chat\_user\_ref = "messages/" + mChatUser + "/" + mCurrentUserId;  
  
 DatabaseReference user\_message\_push = mRootRef.child("messages")  
 .child(mCurrentUserId).child(mChatUser).push();  
  
 final String push\_id = user\_message\_push.getKey();  
  
 final StorageReference filepath = mImageStorage.child("message\_images").child(push\_id + ".jpg");  
 filepath.putFile(imageUri).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {  
 if(task.isSuccessful()){  
 filepath.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {  
 @Override  
 public void onSuccess(Uri uri) {  
 String download\_url = uri.toString();  
  
 Map messageMap = new HashMap();  
 messageMap.put("message", download\_url);  
 messageMap.put("seen", false);  
 messageMap.put("type", "image");  
 messageMap.put("time", ServerValue.*TIMESTAMP*);  
 messageMap.put("from", mCurrentUserId);  
  
 Map messageUserMap = new HashMap();  
 messageUserMap.put(current\_user\_ref + "/" + push\_id, messageMap);  
 messageUserMap.put(chat\_user\_ref + "/" + push\_id, messageMap);  
  
 mRootRef.updateChildren(messageUserMap, new DatabaseReference.CompletionListener() {  
 @Override  
 public void onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 if(databaseError != null){  
 Log.*d*("CHAT\_LOG", databaseError.getMessage().toString());  
 }  
 }  
 });  
 }  
 });  
  
 }  
 }  
 });  
  
 }  
 }  
}

## ChatFragment

1. ChatFragment.java

package com.example.firebasechat;  
  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.ChildEventListener;  
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;  
import com.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \*/*public class ChatsFragment extends Fragment {  
 private RecyclerView mChatsList;  
  
 private DatabaseReference mChatsDatabase;  
  
 private DatabaseReference mUsersDatabase;  
  
 private FirebaseAuth mAuth;  
  
 private String mCurrent\_user\_id;  
  
 private View mMainView;  
  
  
 public ChatsFragment() {  
 // Required empty public constructor  
 }  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 mMainView = inflater.inflate(R.layout.*fragment\_chats*, container, false);  
  
 mChatsList = mMainView.findViewById(R.id.*chats\_list*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 mCurrent\_user\_id = mAuth.getCurrentUser().getUid();  
  
 mChatsDatabase = FirebaseDatabase.*getInstance*().getReference().child("Chat").child(mCurrent\_user\_id);  
 mChatsDatabase.keepSynced(true);  
 mUsersDatabase = FirebaseDatabase.*getInstance*().getReference().child("Users");  
 mUsersDatabase.keepSynced(true);  
  
 mChatsList.setHasFixedSize(true);  
 mChatsList.setLayoutManager(new LinearLayoutManager(getContext()));  
  
 // Inflate the layout for this fragment  
 return mMainView;  
 }  
  
 @Override  
 public void onStart() {  
 super.onStart();  
  
 FirebaseRecyclerOptions<Chats> options=  
 new FirebaseRecyclerOptions.Builder<Chats>()  
 .setQuery(mChatsDatabase,Chats.class)  
 .setLifecycleOwner(this)  
 .build();  
  
 FirebaseRecyclerAdapter chatsRecyclerViewAdapter = new FirebaseRecyclerAdapter<Chats, ChatsViewHolder>(options) {  
  
 @NonNull  
 @Override  
 public ChatsViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 return new ChatsViewHolder(LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*users\_single\_layout*, parent, false));  
  
 }  
  
 @Override  
 protected void onBindViewHolder(@NonNull final ChatsViewHolder chatsViewHolder, int position, @NonNull Chats chats) {  
  
 chatsViewHolder.setDate(chats.getDate());  
  
 final String list\_user\_id = getRef(position).getKey();  
  
 mUsersDatabase.child(list\_user\_id).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
  
 final String userName = dataSnapshot.child("name").getValue().toString();  
 String userThumb = dataSnapshot.child("thumb\_image").getValue().toString();  
  
 if(dataSnapshot.hasChild("online")) {  
  
 String userOnline = dataSnapshot.child("online").getValue().toString();  
 chatsViewHolder.setUserOnline(userOnline);  
  
 }  
  
 chatsViewHolder.setName(userName);  
 chatsViewHolder.setUserImage(userThumb, getContext());  
  
 chatsViewHolder.mView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
  
 Intent chatIntent = new Intent(getContext(), ChatActivity.class);  
 chatIntent.putExtra("user\_id", list\_user\_id);  
 chatIntent.putExtra("user\_name", userName);  
 startActivity(chatIntent);  
  
 }  
 });  
  
  
 }  
  
 @Override  
 public void onCancelled(DatabaseError databaseError) {  
  
 }  
 });  
  
 }  
 };  
  
 mChatsList.setAdapter(chatsRecyclerViewAdapter);  
 }  
  
 public interface OnFragmentInteractionListerner {  
  
 }  
  
 public static class ChatsViewHolder extends RecyclerView.ViewHolder{  
  
 View mView;  
  
 public ChatsViewHolder(View itemView) {  
 super(itemView);  
  
 mView = itemView;  
  
 }  
  
 public void setDate(String date){  
  
 TextView userStatusView = mView.findViewById(R.id.*user\_single\_status*);  
 userStatusView.setText(date);  
  
 }  
  
 public void setName(String name){  
  
 TextView userNameView = mView.findViewById(R.id.*user\_single\_name*);  
 userNameView.setText(name);  
  
 }  
  
 public void setUserImage(String thumb\_image, Context ctx){  
  
 CircleImageView userImageView = mView.findViewById(R.id.*user\_single\_image*);  
 Picasso.*get*().load(thumb\_image).placeholder(R.mipmap.*ic\_default\_avatar\_round*).into(userImageView);  
  
 }  
  
 public void setUserOnline(String online\_status) {  
  
 ImageView userOnlineView = mView.findViewById(R.id.*user\_single\_online*);  
  
 if(online\_status.equals("true")){  
  
 userOnlineView.setVisibility(View.*VISIBLE*);  
  
 } else {  
  
 userOnlineView.setVisibility(View.*INVISIBLE*);  
  
 }  
  
 }  
 }  
}

1. Fragment\_chat.xml

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ChatsFragment">  
  
 <!-- *TODO: Update blank fragment layout* -->  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/chats\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"></androidx.recyclerview.widget.RecyclerView>  
  
</FrameLayout>

1. New class – Chats.java

package com.example.firebasechat;  
  
public class Chats {  
  
 public String date;  
  
 public Chats(){  
  
 }  
  
 public Chats(String date){  
 this.date = date;  
 }  
  
 public String getDate(){  
 return date;  
 }  
  
 public void setDate(String date){  
 this.date = date;  
 }  
}

## Git CODE

<https://github.com/krishna-gavas/Android-messaging-app>

# Design

## Create Custom Fonts

<https://www.youtube.com/watch?v=fB17m3kX-go>

1. File -> new -> Folder -> Accet Folder
2. Select asset folder -> new -> directory = fonts
3. Copy font file to folder
4. MainActivity.java

package com.example.customfont;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.graphics.Typeface;  
import android.os.Bundle;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity {  
  
 TextView myTV;  
 Typeface myFont;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 myTV = (TextView) findViewById(R.id.*my\_text\_view*);  
 myFont = Typeface.*createFromAsset*(this.getAssets(), "fonts/Facon.ttf");  
 myTV.setTypeface(myFont);  
 }  
}

# Online UPI payment

<https://www.youtube.com/watch?v=KUsCHlus-Tc&t=442s>

<https://www.blueappsoftware.com/google-pay-integration-android/>