Minor Project Report
on
Android App

RAKT DAAN

Supervised By:-

Dr. Vishal Passricha

Assistant Professor
(B.Tech CSE)

Submitted By:-

Shashwat Shukla

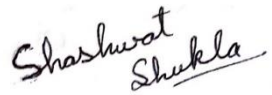
Roll No.:-11514 [Group-2]
(B.Tech CSE)

Department of SOET
Central University Of Haryana
Jant-Pali, Mahendargarh
Haryana,(Pin:-123029)

Acknowledgement

We hereby certify that this work which is being presented in this B.Tech. Minor Project (**BT CSE 705**) report entitled **"Rakt Daan",** in requirements for the award of the **Bachelor of Technology in Computer Engineering** is an authentic record work performed during the period from August 2020 to December 2020 under the supervision of **Dr. Vishal Passricha (**Assistant Professor, CUH) Computer Science and Engineering Department.

The matter presented in this project report has not been submitted for the award of any other degree elsewhere.



Signature of Candidates

SHASHWAT SHUKLA (11514)

This is to certify that the above statements made by the mentioned candidates are correct as per my knowledge.

Date: 23/12/2020

Signature of Supervisor

Dr.Vishal Passricha

(Assistant Professor, CUH)

Index

S.No.	Title	Page No
1.	Rakt Daan - IDEA	1
2.	Technologies Used	1
3.	Main Activity	4
4.	Splash Screen	12
5.	Registration Activity	15
6.	Requesting Activity	27
7.	Your Requests Activity	37
8	Some Essential Utility Files	40

Rakt Daan-IDEA

The Rakt Daan is a Blood Donation and Requesting Android app. Powered by Google's Firebase.

The Idea was to create an app where the users can register them self as a Potential donor and be able to look for other Donors of all the various Blood Groups in there region.

The focus was to create a Simple but efficient app that took little effort to operate and had rock solid utility.

Technologies Used

Firebase Auth

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.

How does it work?

To sign a user into your app, you first get authentication credentials from the user. These credentials can be the user's email address and password, or an OAuth token from a federated identity provider. Then, you pass these credentials to the Firebase Authentication SDK. Firebase's backend services will then verify those credentials and return a response to the client.

After a successful sign in, you can access the user's basic profile information, and you can control the user's access to data stored in other Firebase products. You can also use the provided authentication token to verify the identity of users in your own backend services.

Firebase Realtime Database

Firebase Stores and syncs data with NoSQL cloud database. Data is synced across all clients in realtime, and remains available when your app goes offline.

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

How does it work?

The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, realtime events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how your data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it.

The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly. This enables you to build a great realtime experience that can serve millions of users without compromising on responsiveness. Because of this, it is important to think about how users need to access your data and then structure it accordingly.

Reverse Geocoding

Geocoding is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers on a map, or position the map.

Reverse geocoding is the process of converting geographic coordinates into a human-readable address.

The Geocoding API provides a direct way to access these services via an HTTP request. The following example uses the Geocoding service through the Maps JavaScript API to demonstrate the basic functionality.

FusedLocationProviderClient

Using the Google Play services location APIs, your app can request the last known location of the user's device. In most cases, you are interested in the user's current location, which is usually equivalent to the last known location of the device.

Specifically, use the **fused location provider** to retrieve the device's last known location. The fused location provider is one of the location APIs in Google Play services. It manages the underlying location technology and provides a simple API so that you can specify requirements at a high level, like high accuracy or low power. It also optimizes the device's use of battery power.

But if the user's Last Known Location is not available then Applications can also request Location Updates from the GPS and Internet combined.

The Steps involved in Requesting the User's Locations are:

- Taking Proper Permission from the user.
- Checking is hardware is active or not. If not notify user to Enable it.
- Check if last known Location is available or not if not ,then Request Location Updates .

Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on **IntelliJ** IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for **Google Cloud Platform**, making it easy to integrate Google Cloud Messaging and App Engine

Main Activity

This is the Main Activity that the User Sees as Soon as he/she Opens the App.

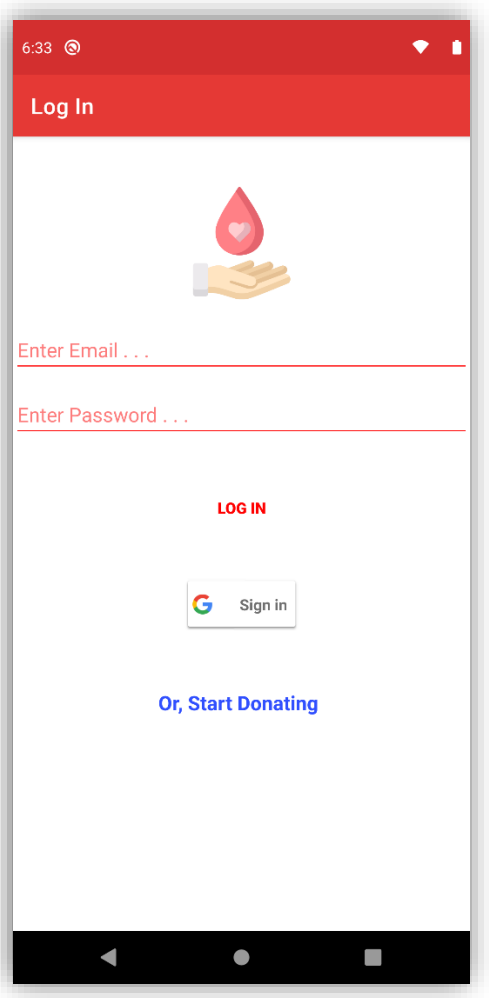
This Activity has two Major Functionalities: -

- **User Sign Up**, or
- **User Login**

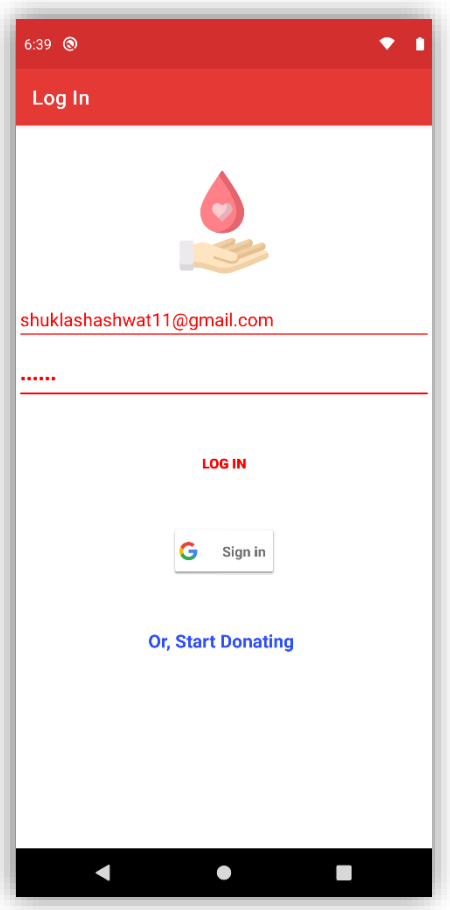
There are Two Possible ways for user to **Sign Up** and **Sign In**:

- **Google Sign-In/Sign-Up Service:**

  Uses **Google Play Services** to Get Access to user's Data and Register him to the Firebase Authentication System.



- **Email + Password Authentication:**
  Uses Firebase Authentication System and Takes User's Email and Password as Input to Register them for the Service.



Or, Start Donation: is a Toggle Button to Switch Between Sign In and Sign Up.

The **SIGN UP** Button takes user to the Registration View, if the User Previously Signed Up but didn't register then **LOG IN** will also redirect the user to the same Registration Activity.

Main Activity – Layout - 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:id="@+id/loginPConstraint"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    tools:context=".MainActivity">

    <TextView
```

```
        android:id="@+id/sinlog"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="swapModes"
        android:text="Or, Start Donating "
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        android:textColor="#304FFE"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.72" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="127dp"
        android:layout_height="0dp"
        android:layout_marginTop="45dp"
        android:layout_marginBottom="23dp"
        app:layout_constraintBottom_toTopOf="@+id/editTextEmail"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/donation" />

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="13dp"
        android:backgroundTint="#FF0000"
        android:ems="10"
        android:hint="Enter Email . . . "
        android:inputType="textPersonName"
        android:textColor="#FF0000"
        android:textColorHint="#FD7C7C"
        app:layout_constraintBottom_toTopOf="@+id/editTextPassword"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView" />

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="37dp"
        android:backgroundTint="#FF0000"
        android:ems="10"
        android:hint="Enter Password . . ."
        android:inputType="textPassword"
        android:textColor="#FF0000"
        android:textColorHint="#FD7C7C"
        app:layout_constraintBottom_toTopOf="@+id/buttonLogin"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
```

```
        app:layout_constraintTop_toBottomOf="@+id/editTextEmail" />

    <Button
        android:id="@+id/buttonLogin"
        style="@android:style/Widget.Material.Light.Button.Borderless.Colored"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="357dp"
        android:onClick="signlog"
        android:text="Log In"
        android:textColor="#FF0000"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextPassword" />

    <ProgressBar
        android:id="@+id/loginProgressBar"
        style="@android:style/Widget.DeviceDefault.Light.ProgressBar.Large"
        android:layout_width="51dp"
        android:layout_height="58dp"
        android:foregroundGravity="fill"
        android:visibility="invisible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <com.google.android.gms.common.SignInButton
        android:id="@+id/google_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_margin="10dp"
        android:layout_marginTop="23dp"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/loginProgressBar" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

Main Activity – Class File

```
package com.example.raktdaan;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
```

```
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.auth.api.signin.GoogleSignIn;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInClient;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthCredential;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.GoogleAuthProvider;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class MainActivity extends AppCompatActivity {

    private FirebaseAuth mAuth ;
    Button buttonLogin ;
    TextView signlog ;
    boolean mode = false ;
    EditText editMail , editPassword ;
    ProgressBar progressBar ;
    Intent regIntent , userIntent ;

    GoogleSignInOptions gso ;
    GoogleSignInClient mGoogleSignInClient ;
    SignInButton mSignInButton ;
    int RC_SIGN_IN = 101 ;

    public void swapModes(View view){
        if( !mode ) {
            setTitle(R.string.signup);
            buttonLogin.setText(R.string.signup);
            signlog.setText("Or, Log In");
            mode = true ;
        }else{
            setTitle(R.string.login);
            buttonLogin.setText(R.string.login);
            signlog.setText("Or, Start Donating");
            mode = false ;
        }
    }

    public void signlog(View view){
        String email = String.valueOf(editMail.getText());
        String password = String.valueOf(editPassword.getText());
```

```
        if( password.length() < 6 ){
            Toast.makeText( MainActivity.this, "Password Length Should be Greater than
Equal to 6" , Toast.LENGTH_SHORT ).show() ;
            return ;
        }

        buttonLogin.setVisibility(View.INVISIBLE);
        progressBar.setVisibility( View.VISIBLE );

        new Handler().postDelayed(() -> {
            buttonLogin.setVisibility( View.VISIBLE );
            progressBar.setVisibility( View.INVISIBLE );
        }, 3000 ) ;

        try {
            Log.i(email, password);
            if (!mode) {
                mAuth.signInWithEmailAndPassword(email, password)
                        .addOnCompleteListener(this, task -> {
                            if (task.isSuccessful()) {
                                FirebaseUser user = mAuth.getCurrentUser();
                                Toast.makeText(MainActivity.this, "Log In Successful",
Toast.LENGTH_SHORT).show() ;
                                logIn();
                                //                                updateUI(user);
                            } else {
                                Toast.makeText(MainActivity.this, "Email or Password
Incorrect.",
                                        Toast.LENGTH_SHORT).show();
                            }
                        });
            } else {
                mAuth.createUserWithEmailAndPassword(email, password)
                        .addOnCompleteListener(this, task -> {
                            if (task.isSuccessful()) {
                                FirebaseUser user = mAuth.getCurrentUser();
                                Toast.makeText(MainActivity.this, "Sign Up
Successful.", Toast.LENGTH_SHORT).show() ;
                                logIn();
                            } else {
                                Toast.makeText(MainActivity.this, "Sign Up
failed.\nPlease use Valid Email and Password.",
                                        Toast.LENGTH_SHORT).show();
                            }
                        });
            }
        }catch( Exception e ){
            Toast.makeText(MainActivity.this , "Please make correct Input" ,
Toast.LENGTH_SHORT  ).show() ;
        }


    }

    public void gSignIn(){
        Intent signInIntent = mGoogleSignInClient.getSignInIntent();
        startActivityForResult(signInIntent, RC_SIGN_IN);
```

```
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);

        if( requestCode == RC_SIGN_IN ){
            Task<GoogleSignInAccount> task =
GoogleSignIn.getSignedInAccountFromIntent(data) ;
            try{
                GoogleSignInAccount account = task.getResult(ApiException.class) ;
                firebaseAuthWithGoogle( account.getIdToken() ) ;
            }catch (ApiException e){
                Toast.makeText(MainActivity.this, "Sign In with Google Failed",
Toast.LENGTH_SHORT).show();
            }
        }
    }

    public void firebaseAuthWithGoogle( String idToken ){
        AuthCredential credential = GoogleAuthProvider.getCredential(idToken,null) ;
        mAuth.signInWithCredential(credential)
                .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            Toast.makeText(MainActivity.this, "SignIn/Up Successful",
Toast.LENGTH_SHORT).show();
                            FirebaseUser user = mAuth.getCurrentUser();
                            logIn();
                        }else{
                            Toast.makeText(MainActivity.this, "SignIn/Up Failed",
Toast.LENGTH_SHORT).show();
                        }
                    }
                });
    }



    public void logIn(){

        FirebaseDatabase.getInstance().getReference().child("user").child(
mAuth.getCurrentUser().getUid() )
                .addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if( snapshot.exists() ) {
                    userIntent = new Intent(MainActivity.this, requests.class);
                    startActivity(userIntent);
                }
                else{
                    regIntent = new Intent( MainActivity.this , registration.class ) ;
                    startActivity(regIntent) ;
                }
            }
            @Override
```

```
            public void onCancelled(@NonNull DatabaseError error) {
                regIntent = new Intent( MainActivity.this , registration.class ) ;
                startActivity(regIntent) ;
            }
        });
    }



    @Override
    protected void onStart() {
        super.onStart();
        FirebaseUser currentUser = mAuth.getCurrentUser() ;
        Log.i("User" , String.valueOf(currentUser)) ;
        if( currentUser != null ) logIn();
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        gso = new GoogleSignInOptions.Builder( GoogleSignInOptions.DEFAULT_SIGN_IN )
                .requestIdToken(getString(R.string.default_web_client_id))
                .requestEmail()
                .build() ;

        mGoogleSignInClient = GoogleSignIn.getClient(this , gso );
        mSignInButton = findViewById( R.id.google_button ) ;
        mSignInButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                gSignIn();
            }
        });

        setTitle( R.string.login );
        mAuth = FirebaseAuth.getInstance();
        buttonLogin = findViewById( R.id.buttonLogin ) ;
        signlog = findViewById( R.id.sinlog ) ;
        editMail = findViewById(R.id.editTextEmail) ;
        editPassword = findViewById(R.id.editTextPassword) ;
        progressBar = findViewById( R.id.loginProgressBar ) ;
    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
        Intent a = new Intent(Intent.ACTION_MAIN);
        a.addCategory(Intent.CATEGORY_HOME);
        a.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(a);
    }

}
```

Splash Screen Activity

More than a Visual aspect of the App the Splash is a necessity of the app. It's the Activity with which the user is greeted with when opening the App .But It's major function is to check weather the user is already Signed In or not and redirect user accordingly .

There are two Major Directs from the Splash Screen .

- **Login/SignUp Activity:** If this is the First time user is opening the App or previously signed out after user.
- **Request Activity:** If the user is already logged in the Splash Screen Takes user Directly into the App.

Splash Screen - Layout - 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:background="#FF0000"
    tools:context=".splash_screen">

    <ImageView
        android:id="@+id/imageView2"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/donation" />

    <TextView
        android:id="@+id/textView8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Rakt Daan"
        android:textAppearance="@style/TextAppearance.AppCompat.Display3"
        android:textColor="#FFFFFF"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView2" />

    <TextView
        android:id="@+id/textView9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="Heros Come in all TYPES \n and SIZE"
        android:textColor="#FFFFFF"
        android:textStyle="italic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView8" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Splash Screen – Class File

```
package com.example.raktdaan;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class splash_screen extends AppCompatActivity {

    Intent regIntent , userIntent ;
    FirebaseAuth mAuth ;
    public void logIn(){

        //Checks if User is Registered or not .
        FirebaseDatabase.getInstance().getReference().child("user").
                child( mAuth.getCurrentUser().getUid() )
                .addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
```

```
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                // If the user is registered then takes him/her into the app .
                if( snapshot.exists() ) {
                    userIntent = new Intent(splash_screen.this, requests.class);
                    startActivity(userIntent);
                }
                else{
                    //If not registered takes the user to
                    regIntent = new Intent( splash_screen.this ,
registration.class ) ;
                    startActivity(regIntent) ;
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                regIntent = new Intent( splash_screen.this , registration.class
) ;
                startActivity(regIntent) ;
            }
        });
    }

    //On Start Checks if user is Logged in or Not.
    @Override
    protected void onStart() {
        super.onStart();

        //If not Logged in Takes user to Login Activity
        if( FirebaseAuth.getInstance().getCurrentUser() == null ){
            Intent i = new Intent(this, MainActivity.class) ;
            startActivity(i);
        }else{
            //If user is Logged int then Checks if User has registered or not .
            logIn();
        }

    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);
        mAuth = FirebaseAuth.getInstance() ;

    }
}
```

Registration Activity

This is the Activity where the user is taken to fill all the Necessary Information required to identify itself as a Unique User.

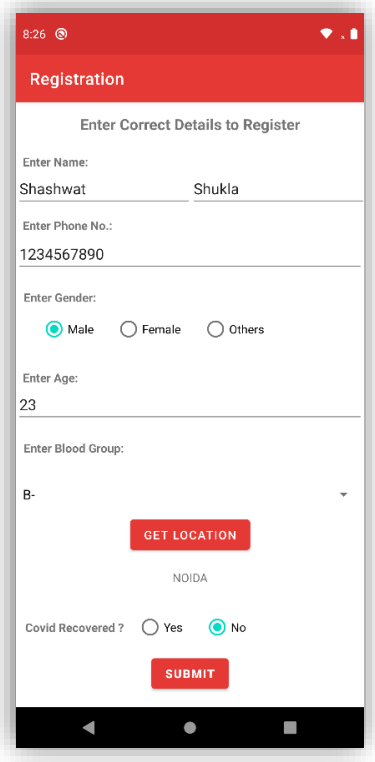
The Registration Fields include:

- First Name,
- Last Name,
- Phone No.
- Age,
- Gender,
- Location and
- Covid Recovery.

All the inputs are regular Text or Radio Button inputs , But the Location Input makes use of two Technologies , provided by Google :

- **FusedLocationProviderClient :** Take s input from GPS and Internet to Pin Point user's Location and access there Coordinates (Latitude and Longitude) .
- **Reverse Geocoding:** Uses Lat-Log Coordinates Provided by the **FusedLocationProviderClient** and Returns a JSON file with the Address of the User and Uses Locality as the User Location for Registration.

All the User Information is stored in a **Realtime Database** Provided by **Google's Firebase** .



The Data is Stored in a **NoSQL Database** in the Form of a **Indexed JSON File** .

Registration Activity – Layout - XML

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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:id="@id/re"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:fillViewport="true">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@id/registerC"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toStartOf="@id/editLname"
        app:layout_constraintStart_toStartOf="@id/editLname">

        <EditText
            android:id="@id/editFname"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="First Name"
            android:inputType="textPersonName"
            app:layout_constraintBaseline_toBaselineOf="@id/editLname"
            app:layout_constraintStart_toStartOf="@id/textView" />

        <EditText
            android:id="@id/editLname"
            android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_marginTop="94.0dip"
        android:ems="10"
        android:hint="Last Name"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@id/editFname"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@id/editTextPhone"
        android:layout_width="0.0dip"
        android:layout_height="wrap_content"
        android:layout_marginTop="2.0dip"
        android:ems="10"
        android:hint="Phone No."
        android:inputType="phone"
        android:maxLength="10"
        android:numeric="integer"
        android:phoneNumber="false"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/textView2" />

    <RadioGroup
        android:id="@id/radioGroup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="28.0dip"
        android:layout_marginTop="4.0dip"
        android:orientation="horizontal"
        app:layout_constraintStart_toStartOf="@id/textView3"
        app:layout_constraintTop_toBottomOf="@id/textView3">

        <RadioButton
            android:id="@id/radioButton"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:checked="true"
            android:paddingRight="25.0dip"
            android:text="Male" />

        <RadioButton
            android:id="@id/radioButton2"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:paddingRight="25.0dip"
            android:text="Female" />

        <RadioButton
            android:id="@id/radioButton3"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Others" />
    </RadioGroup>

    <TextView
        android:id="@id/textView"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="8.0dip"
        android:text="Enter Name:"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@id/editFname"
        app:layout_constraintStart_toStartOf="parent" />

    <TextView
        android:id="@id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="11.0dip"
        android:paddingLeft="8.0dip"
        android:text="Enter Phone No.:"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="@id/editTextPhone"
        app:layout_constraintTop_toBottomOf="@id/editFname" />

    <TextView
        android:id="@id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="1.0dip"
        android:layout_marginTop="19.0dip"
        android:paddingLeft="8.0dip"
        android:text="Enter Gender:"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/editTextPhone" />

    <Button
        android:id="@id/getLoc"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="17.0dip"
        android:onClick="getLocation"
        android:text="Get Location"
        app:layout_constraintBottom_toTopOf="@id/locationTextView"
        app:layout_constraintEnd_toEndOf="@id/locProgressBar"
        app:layout_constraintStart_toStartOf="@id/locProgressBar"
        app:layout_constraintTop_toBottomOf="@id/groupSpinner" />

    <Spinner
        android:id="@id/groupSpinner"
        android:layout_width="0.0dip"
        android:layout_height="wrap_content"
        android:layout_marginTop="21.0dip"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/textView4" />

    <TextView
        android:id="@id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="1.0dip"
        android:layout_marginTop="19.0dip"
```

```
        android:paddingLeft="8.0dip"
        android:text="Enter Blood Group:"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/editTextage" />

    <TextView
        android:id="@id/locationTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="80dp"
        android:text="Get Location"
        android:textAllCaps="true"
        app:layout_constraintBottom_toTopOf="@id/regSubmit"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <TextView
        android:id="@id/textView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24.0dip"
        android:paddingLeft="8.0dip"
        android:text="Enter Age:"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="@id/editTextage"
        app:layout_constraintTop_toBottomOf="@id/radioGroup" />

    <EditText
        android:id="@id/editTextage"
        android:layout_width="0.0dip"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter Age"
        android:inputType="number"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/textView6" />

    <TextView
        android:id="@id/textView7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="28.0dip"
        android:text="Enter Correct Details to Register"
        android:textSize="18.0sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@id/regSubmit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16.0dip"
        android:onClick="commitData"
        android:text="Submit"
```

```
            android:textStyle="bold"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent" />

        <ProgressBar
            android:id="@id/locProgressBar"
            style="?android:progressBarStyle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:visibility="invisible"
            app:layout_constraintBottom_toBottomOf="@id/getLoc"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="@id/getLoc" />

        <TextView
            android:id="@+id/textView10"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="40dp"
            android:layout_marginEnd="10dp"
            android:paddingLeft="8dp"
            android:text="Covid Recovered ?"
            android:textStyle="bold"
            app:layout_constraintEnd_toStartOf="@+id/covidrRadioGroup"
            app:layout_constraintHorizontal_bias="0.372"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/locationTextView" />

        <RadioGroup
            android:id="@+id/covidrRadioGroup"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="25dp"
            android:orientation="horizontal"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="@id/textView10"
            app:layout_constraintTop_toBottomOf="@id/locationTextView">

            <RadioButton
                android:id="@+id/yes"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:paddingRight="25dp"
                android:text="Yes" />

            <RadioButton
                android:id="@+id/no"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:checked="true"
                android:text="No" />
        </RadioGroup>

    </androidx.constraintlayout.widget.ConstraintLayout>
</ScrollView>
```

Registration Activity – Class File

```
package com.example.raktdaan;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.text.TextUtils;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
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.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.Arrays;
import java.util.List;
import java.util.Locale;
import java.util.Objects;

public class registration extends AppCompatActivity {

    Button getLoc , submitB ;
    EditText fname , lname , age , pno ;
    TextView textLocation  ;
    RadioGroup gender ,covid ;
    RadioButton gender1 , gender2 , gender3 ,yes , no ;
    ProgressBar progressBar ;
    Spinner groupSpinner ;
    String loc = "" ,  bGroup = "A+" , covidS = "No" ;
```

```
    FusedLocationProviderClient fusedLocationProviderClient ;
    LocationManager locManager ;
    LocationRequest locationRequest ;
    boolean back = true ; //Track Back Button Press .

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

        MenuInflater menuInflater =new MenuInflater(this ) ;
        menuInflater.inflate( R.menu.reg_menu , menu );
        return super.onCreateOptionsMenu(menu);
    }


    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {

        if( item.getItemId() == R.id.r_log_out ){
            FirebaseAuth.getInstance().signOut();
            Intent i = new Intent( this , MainActivity.class  ) ;
            startActivity( i );
        }

        return super.onOptionsItemSelected(item);
    }

    LocationCallback locationCallback = new LocationCallback(){
        @Override
        public void onLocationResult(LocationResult locationResult) {
            super.onLocationResult(locationResult);
            if( locationResult == null ) {
                return;
            }

            Location location = locationResult.getLastLocation() ;
            Geocoder geoLoc = new Geocoder(getApplicationContext(), Locale.getDefault());
            try {
                List<Address> addList = geoLoc.getFromLocation(location.getLatitude(),
location.getLongitude(), 1);
                if (addList != null && addList.size() > 0) {
                    loc = addList.get(0).getLocality();
                    textLocation.setText(loc);
                    Toast.makeText(registration.this, "Location " + loc,
Toast.LENGTH_SHORT).show();

                }
            } catch (Exception e) {
                Toast.makeText(registration.this, "Unable to Access Location",
Toast.LENGTH_SHORT).show();
                ;
                e.printStackTrace();

            }
        }
    } ;


    public void getLocation( View view ){

        if(ContextCompat.checkSelfPermission( this ,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this , new
```

```
String[]{Manifest.permission.ACCESS_FINE_LOCATION},1);
            Toast.makeText( this , "Repress Get Location after Granting Permission" ,
Toast.LENGTH_SHORT ).show() ;
        }else{

            if( !locManager.isProviderEnabled( LocationManager.GPS_PROVIDER )  ) {
                Toast.makeText( registration.this , "Please Enable Location Services" ,
Toast.LENGTH_SHORT ).show();
                return ;
            }

            fusedLocationProviderClient.requestLocationUpdates( locationRequest ,
locationCallback , Looper.getMainLooper() ) ;

            progressBar.setVisibility(View.VISIBLE);
            getLoc.setVisibility(View.INVISIBLE);


            new Handler().postDelayed(() -> {
                progressBar.setVisibility(View.INVISIBLE);
                getLoc.setVisibility(View.VISIBLE);
//                fusedLocationProviderClient.e
            }, 2000);

        }

    }


    public void commitData( View view ){
        String firstName , lastName , ageS , phoneNo , genderS = "Male" ;

            firstName = fname.getText().toString();
            lastName = lname.getText().toString();
            ageS = age.getText().toString() ;
            phoneNo = pno.getText().toString() ;

            if(TextUtils.isEmpty( firstName )){
                fname.setError("This Field can't be empty");
                return ;
            }

            if(TextUtils.isEmpty( lastName )){
                lname.setError("This Field can't be empty");
                return ;
            }

            if(TextUtils.isEmpty( ageS )){
                age.setError("This Field can't be empty");
                return ;
            }

            if(TextUtils.isEmpty( phoneNo )){
                pno.setError("This Field can't be empty");
                return ;
            }

            if( phoneNo.length() < 10 ){
                pno.setError( "Invalid Phone Number." );
                return ;
            }
```

```
            if( loc.equals("") ){
                getLoc.setError( "Please Provide Location" );
                Toast.makeText(this , "Please give your Location" , Toast.LENGTH_SHORT
).show(); ;
                return ;
            }

            if( gender.getCheckedRadioButtonId() == R.id.radioButton ) genderS = "Male" ;
            else if( gender.getCheckedRadioButtonId() == R.id.radioButton2 ) genderS =
"Female" ;
            else genderS = "Others" ;

            if( covid.getCheckedRadioButtonId() == R.id.yes ) covidS= "Yes" ;
            else covidS = "No" ;

            try {

FirebaseDatabase.getInstance().getReference().child("user").child(FirebaseAuth.getInstance(
).getCurrentUser().getUid()).child("email").setValue(FirebaseAuth.getInstance().getCurrentU
ser().getEmail());

FirebaseDatabase.getInstance().getReference().child("user").child(FirebaseAuth.getInstance(
).getCurrentUser().getUid()).child("fname").setValue(firstName);

FirebaseDatabase.getInstance().getReference().child("user").child(FirebaseAuth.getInstance(
).getCurrentUser().getUid()).child("lname").setValue(lastName);

FirebaseDatabase.getInstance().getReference().child("user").child(FirebaseAuth.getInstance(
).getCurrentUser().getUid()).child("phone").setValue(phoneNo);

FirebaseDatabase.getInstance().getReference().child("user").child(FirebaseAuth.getInstance(
).getCurrentUser().getUid()).child("age").setValue(ageS);

FirebaseDatabase.getInstance().getReference().child("user").child(FirebaseAuth.getInstance(
).getCurrentUser().getUid()).child("gender").setValue(genderS);

FirebaseDatabase.getInstance().getReference().child("user").child(FirebaseAuth.getInstance(
).getCurrentUser().getUid()).child("covid").setValue(covidS);

FirebaseDatabase.getInstance().getReference().child("user").child(FirebaseAuth.getInstance(
).getCurrentUser().getUid()).child("blood").setValue(bGroup);

FirebaseDatabase.getInstance().getReference().child("user").child(FirebaseAuth.getInstance(
).getCurrentUser().getUid()).child("location").setValue( loc );

                //Query Key

FirebaseDatabase.getInstance().getReference().child("user").child(FirebaseAuth.getInstance(
).getCurrentUser().getUid()).child("location_blood_covid").setValue( loc + "_" + bGroup +
"_" + covidS );

                Toast.makeText(registration.this , "Successfully Registered" ,
Toast.LENGTH_SHORT ).show(); ;

                Intent i = new Intent( this , requests.class ) ;
                startActivity(i);
            }catch( Exception e ){
                e.printStackTrace();
                Toast.makeText(registration.this , "Registration Failed" ,
Toast.LENGTH_SHORT ).show(); ;
            }
    }
```

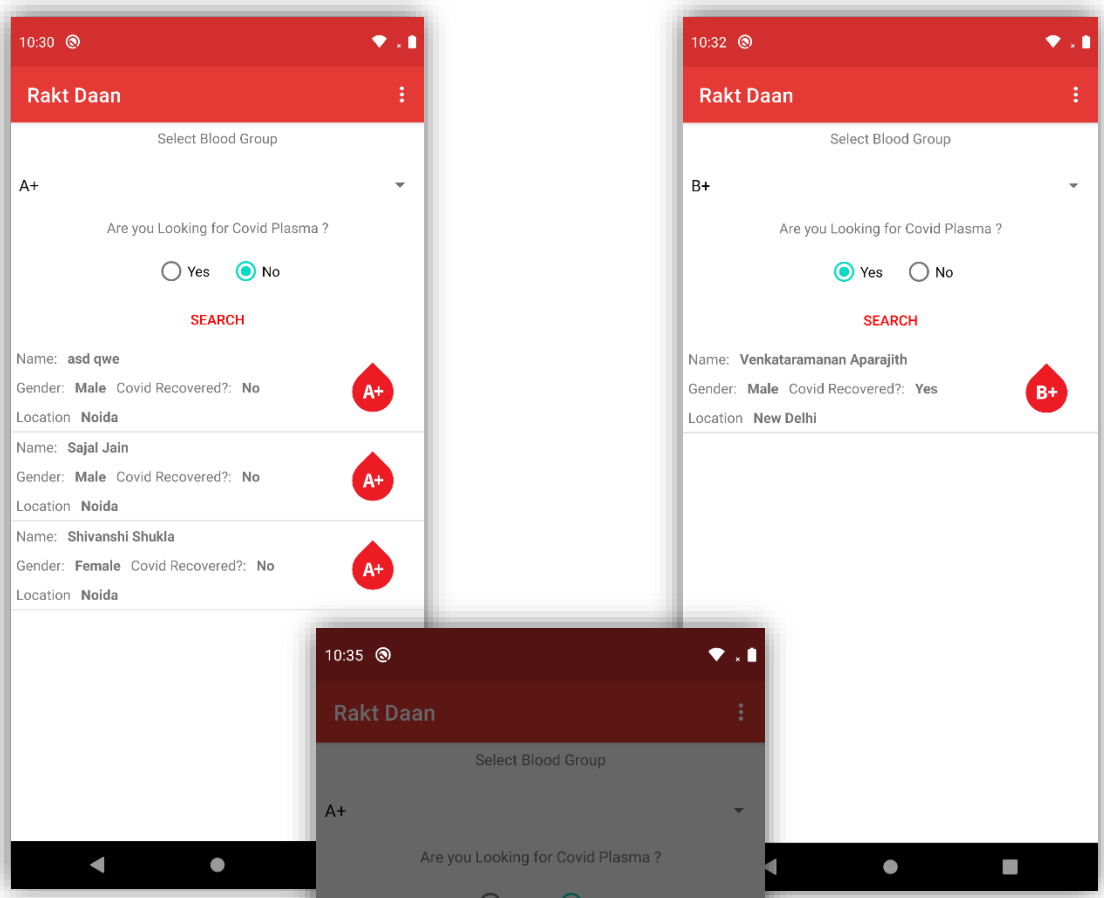
```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration);

        setTitle( "Registration" ) ;

        fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(this)
;
        locManager = (LocationManager) getSystemService(LOCATION_SERVICE) ;
        locationRequest = LocationRequest.create().setInterval( 4000 )
                .setFastestInterval( 2000 )
                .setPriority( LocationRequest.PRIORITY_HIGH_ACCURACY )
                .setNumUpdates(1) ;

        groupSpinner = findViewById(R.id.groupSpinner) ;

        List<String> groups =  Arrays.asList("A+" , "A-" , "B+" , "B-" , "AB+" , "AB-" ,
"O+" , "O-" ) ;

        ArrayAdapter<String> dataAdapter = new ArrayAdapter<>(this ,
android.R.layout.simple_spinner_dropdown_item , groups) ;
        dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item
);
        groupSpinner.setAdapter(    dataAdapter) ;

        fname = findViewById( R.id.editFname ) ;
        lname = findViewById( R.id.editLname ) ;
        age = findViewById( R.id.editTextage ) ;
        pno = findViewById( R.id.editTextPhone ) ;

        gender = findViewById( R.id.radioGroup ) ;
        gender1 = findViewById( R.id.radioButton ) ;
        gender2 = findViewById( R.id.radioButton2 ) ;
        gender3 = findViewById( R.id.radioButton3 ) ;
        progressBar = findViewById( R.id.locProgressBar ) ;
        getLoc = findViewById( R.id.getLoc ) ;
        submitB = findViewById( R.id.regSubmit ) ;

        covid = findViewById(R.id.covidrRadioGroup) ;
        yes = findViewById( R.id.yes ) ;
        no = findViewById( R.id.no ) ;

        textLocation = findViewById( R.id.locationTextView ) ;

        groupSpinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> adapterView, View view, int i, long
l) {
                bGroup = groups.get(i) ;
            }

            @Override
            public void onNothingSelected(AdapterView<?> adapterView) {

            }
        }) ;


FirebaseDatabase.getInstance().getReference().child("user").child(FirebaseAuth.getInstance(
```

```
).getCurrentUser().getUid()).addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if( snapshot.exists() ){

                    back = false ;
                    fname.setText(
Objects.requireNonNull(snapshot.child("fname").getValue()).toString() );
                    lname.setText(
Objects.requireNonNull(snapshot.child("lname").getValue()).toString() );
                    age.setText(
Objects.requireNonNull(snapshot.child("age").getValue()).toString() );
                    pno.setText(
Objects.requireNonNull(snapshot.child("phone").getValue()).toString() );;
                    groupSpinner.setSelection( groups.indexOf(
Objects.requireNonNull(snapshot.child("blood").getValue()).toString() )  );

                    if(
!Objects.requireNonNull(snapshot.child("gender").getValue()).toString().equals( "Male" ) )
                    gender.check( (
Objects.requireNonNull(snapshot.child("gender").getValue()).toString().equals("Female") ?
R.id.radioButton2 : R.id.radioButton3 ) );

                    if(
Objects.requireNonNull(snapshot.child("covid").getValue()).toString().equals( "Yes" ) )
                        covid.check( R.id.yes ) ;

                    loc =
Objects.requireNonNull(snapshot.child("location").getValue()).toString() ;
                    textLocation.setText( loc );
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();

        if( back ){
            Intent a = new Intent(Intent.ACTION_MAIN);
            a.addCategory(Intent.CATEGORY_HOME);
            a.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(a);
        }

    }
}
```

Requesting Activity

This is the Main Operational Activity where the main utility of the App lies. After Proper Registration user gets access to this activity.

This activity makes use of **FusedLocationProviderClient** and **Reverse Geocoding** to seamlessly get user's Location in the Background. And user's just have to select the Type of Blood they required or whether or not they are looking for Covid Plasma . And The Search button will return a List of all the Users in their Locality.



After this, the User can click on any of the user to Request Blood from them , and a Request will be sent to them .

Requesting Activity – Layout - 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=".requests">

    <Spinner
        android:id="@+id/reqSpinner"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="13dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView11" />

    <TextView
        android:id="@+id/textView11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="6dp"
        android:text="Select Blood Group"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView12"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="9dp"
        android:text="Are you Looking for Covid Plasma ?"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/reqSpinner" />

    <RadioGroup
        android:id="@+id/plasmaRequest"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:orientation="horizontal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView12">

        <RadioButton
            android:id="@+id/yes"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:paddingRight="20dp"
            android:text="Yes" />

        <RadioButton
            android:id="@+id/no"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:checked="true"
            android:text="No" />


    </RadioGroup>

    <ListView
        android:id="@+id/donorsList"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/reqButton"
        app:layout_constraintVertical_bias="0.037" />

    <Button
        android:id="@+id/reqButton"
        style="@style/Widget.AppCompat.Button.Borderless.Colored"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="search"
        android:text="Search"
        android:textColor="#FF0000"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/plasmaRequest" />

    <ProgressBar
        android:id="@+id/reqProgressBar"
        style="?android:attr/progressBarStyle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:visibility="invisible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/reqButton" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Requesting Activity – Class File

```
package com.example.raktdaan;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.app.AlertDialog;
import android.content.DialogInterface;
```

```java
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.Toast;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Locale;
import java.util.Objects;

// Code Companion of the Request Activity

public class requests extends AppCompatActivity {

    Spinner spinner ;
    RadioGroup covReq ;
    RadioButton yes , no ;
    Button searchB ;
    ProgressBar reqProgress ;
    ListView avaList ;
    ArrayList<Info> arrayList ;
    InfoAdapter infoAdapter ;
    LocationManager locationManager ;
    LocationRequest locationRequest ;
    FusedLocationProviderClient fusedLocationProviderClient ;
    String loc = "" , bloodG = "A+" , covidS = "No" ;
    Info userInfo ;
```

```
    LocationCallback locationCallback = new LocationCallback(){
        @Override
        public void onLocationResult(LocationResult locationResult) {
            super.onLocationResult(locationResult);

            if( locationResult == null ){
                return ;
            }

            Location location = locationResult.getLastLocation() ;
            Geocoder geocoder = new Geocoder( getApplicationContext() ,
Locale.getDefault() ) ;

            try {
                List<Address> addList = geocoder.getFromLocation(
location.getLatitude() , location.getLongitude() , 1 ) ;

                if( addList.size() > 0 ){
                    loc = addList.get(0).getLocality() ;
                }
            } catch (Exception e) {
                Toast.makeText( requests.this , "Start Location Services First" ,
Toast.LENGTH_SHORT ).show() ;
            }

        }
    };

    //Creates and Add Menu to the UI
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater menuInflater =new MenuInflater(this ) ;
        menuInflater.inflate( R.menu.usermenu , menu );
        return super.onCreateOptionsMenu(menu);
    }

    //Gets the Menu Selected and Provides Proper Response
    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {

        if( item.getItemId() == R.id.editprofile ){
            Intent i = new Intent( this , registration.class ) ;
            startActivity( i );
        }else if( item.getItemId() == R.id.logout ){
            FirebaseAuth.getInstance().signOut();
            Intent i = new Intent( this , MainActivity.class  ) ;
            startActivity( i );
        }
        else if( item.getItemId() == R.id.request ){
            Intent i = new Intent( this , your_requests.class );
            startActivity(i) ;
        }else {
            new MyDialogFragment().show(getSupportFragmentManager(), "My Dialog");
        }
        return super.onOptionsItemSelected(item);
    }
```

```
    // Fetches User Location in the Background for the ease of Searching Blood Donors
in your Location .
    public boolean getLocation(){

        if(ContextCompat.checkSelfPermission( this ,
Manifest.permission.ACCESS_FINE_LOCATION ) != PackageManager.PERMISSION_GRANTED ){
            ActivityCompat.requestPermissions( this , new String[] {
Manifest.permission.ACCESS_FINE_LOCATION } , 1 ) ;
        }else {

            if( !locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER) ){
                Toast.makeText( this , "Start Location Services First" ,
Toast.LENGTH_SHORT ).show();
                return false ;
            }

            fusedLocationProviderClient.requestLocationUpdates( locationRequest ,
locationCallback , Looper.getMainLooper() ) ;

        }
        return true ;
    }

    // Works When Search Button is Pressed
    public void search(View view){

        //Covid Radio Switch
        covidS = ( yes.isChecked() ? "Yes" : "No" ) ;

        reqProgress.setVisibility( View.VISIBLE );

        Log.i("Query" , loc + "_" + bloodG + "_" + covidS ) ;

        int resID = 0 ;

        switch (bloodG) {
            case "A+": resID = R.drawable.ap; break;
            case "A-": resID = R.drawable.an; break;
            case "AB+": resID = R.drawable.abp; break;
            case "AB-": resID = R.drawable.abn; break;
            case "O+": resID = R.drawable.op; break;
            case "O-": resID = R.drawable.on; break;
            case "B+": resID = R.drawable.bp; break;
            default: resID = R.drawable.bn; break;
        }

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                reqProgress.setVisibility(View.INVISIBLE) ;
            }
        } , 3000 ) ;

        if( !getLocation() ) return ;

        int finalResID = resID;
```

```
FirebaseDatabase.getInstance().getReference().child("user").orderByChild("location_blo
od_covid").equalTo( loc + "_" + bloodG + "_" + covidS ).addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                arrayList.clear();
                for( DataSnapshot dataSnapshot : snapshot.getChildren() ){
                    arrayList.add( new
Info(Objects.requireNonNull(dataSnapshot.getKey()),

Objects.requireNonNull(dataSnapshot.child("fname").getValue()).toString() ,

Objects.requireNonNull(dataSnapshot.child("lname").getValue()).toString() ,

Objects.requireNonNull(dataSnapshot.child("gender").getValue()).toString() ,

Objects.requireNonNull(dataSnapshot.child("location").getValue()).toString() ,

Objects.requireNonNull(dataSnapshot.child("blood").getValue()).toString() ,

Objects.requireNonNull(dataSnapshot.child("covid").getValue()).toString() ,
                            finalResID) ) ;
                }
                infoAdapter.notifyDataSetChanged();
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        }) ;


    }

    //As Your Info has to be Sent to the Donor from whom you Request Blood , Your Info
is Featched and Saved from Firebase RealTime Database .
    public void getCurrentUserInfo(){
        FirebaseDatabase.getInstance().getReference().child("user")
                .child(FirebaseAuth.getInstance().getCurrentUser().getUid())
                .addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot snapshot) {
                        userInfo = new Info(
Objects.requireNonNull(snapshot.getKey()),

Objects.requireNonNull(snapshot.child("fname").getValue()).toString() ,

Objects.requireNonNull(snapshot.child("lname").getValue()).toString() ,

Objects.requireNonNull(snapshot.child("email").getValue()).toString() ,

Objects.requireNonNull(snapshot.child("phone").getValue()).toString(),

Objects.requireNonNull(snapshot.child("age").getValue()).toString()) ;

                        Toast.makeText(requests.this , "Welcome,"+userInfo.getFname()
, Toast.LENGTH_SHORT ).show();
```

```
                    }

                    @Override
                    public void onCancelled(@NonNull DatabaseError error) {

                    }
                });
    }

    @Override
    protected void onStart() {
        super.onStart();
        if( FirebaseAuth.getInstance().getCurrentUser() == null ){
            Intent i = new Intent(this, MainActivity.class) ;
            startActivity(i);
        }
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_requests);

        locationManager = ( LocationManager ) getSystemService( LOCATION_SERVICE ) ;
        fusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this) ;
        locationRequest = LocationRequest.create().setInterval(1000)
                .setFastestInterval(500)
                .setPriority( LocationRequest.PRIORITY_HIGH_ACCURACY )
                .setNumUpdates(1) ;

        List<String> groups =  Arrays.asList("A+" , "A-" , "B+" , "B-" , "AB+" , "AB-"
, "O+" , "O-" ) ;

        covReq = findViewById( R.id.covidrRadioGroup ) ;
        yes = findViewById( R.id.yes ) ;
        no = findViewById( R.id.no ) ;

        searchB = findViewById( R.id.reqButton ) ;
        avaList = findViewById( R.id.donorsList ) ;
        reqProgress = findViewById( R.id.reqProgressBar ) ;

        spinner = findViewById( R.id.reqSpinner) ;
        ArrayAdapter<String> dataAdapter = new ArrayAdapter<>(this ,
android.R.layout.simple_spinner_dropdown_item , groups) ;

dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item );
        spinner.setAdapter(dataAdapter) ;


        arrayList = new ArrayList<>() ;
        infoAdapter = new InfoAdapter (this  , arrayList , R.layout.custom_listview )
;
        avaList.setAdapter( infoAdapter );

        getCurrentUserInfo();
```

```
        spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> adapterView, View view, int i,
long l) {
                bloodG = groups.get( i ) ;
            }

            @Override
            public void onNothingSelected(AdapterView<?> adapterView) {

            }
        });

        avaList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

                new AlertDialog.Builder(requests.this)
                        .setIcon(android.R.drawable.ic_dialog_alert)
                        .setTitle("Do You Want to Request Blood from " +
arrayList.get(i).getFname() + "?" )
                        .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialogInterface, int ii ) {

FirebaseDatabase.getInstance().getReference().child("requests").child(arrayList.get(i).getU
id()).child(userInfo.getUid()).child("fname").setValue(userInfo.getFname()) ;

FirebaseDatabase.getInstance().getReference().child("requests").child(arrayList.get(i).getU
id()).child(userInfo.getUid()).child("lname").setValue(userInfo.getLname()) ;

FirebaseDatabase.getInstance().getReference().child("requests").child(arrayList.get(i).getU
id()).child(userInfo.getUid()).child("email").setValue(userInfo.getEmail()) ;

FirebaseDatabase.getInstance().getReference().child("requests").child(arrayList.get(i).getU
id()).child(userInfo.getUid()).child("phone").setValue(userInfo.getPno()) ;

FirebaseDatabase.getInstance().getReference().child("requests").child(arrayList.get(i).getU
id()).child(userInfo.getUid()).child("age").setValue(userInfo.getAge()) ;

FirebaseDatabase.getInstance().getReference().child("requests").child(arrayList.get(i).getU
id()).child(userInfo.getUid()).child("uid").setValue( arrayList.get(i).getUid() ) ;
                            }
                        })
                        .setNegativeButton("No", new DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialogInterface, int i) {
                                Toast.makeText(requests.this , "Request Cancelled" ,
Toast.LENGTH_SHORT).show();
                            }
                        }).show() ;
            }
        });

        //Gets User Permission to use Location
        if(ContextCompat.checkSelfPermission( this ,
Manifest.permission.ACCESS_FINE_LOCATION ) != PackageManager.PERMISSION_GRANTED ){
            ActivityCompat.requestPermissions( this , new String[] {
Manifest.permission.ACCESS_FINE_LOCATION } , 1 ) ;
        }else{
            getLocation() ;
```

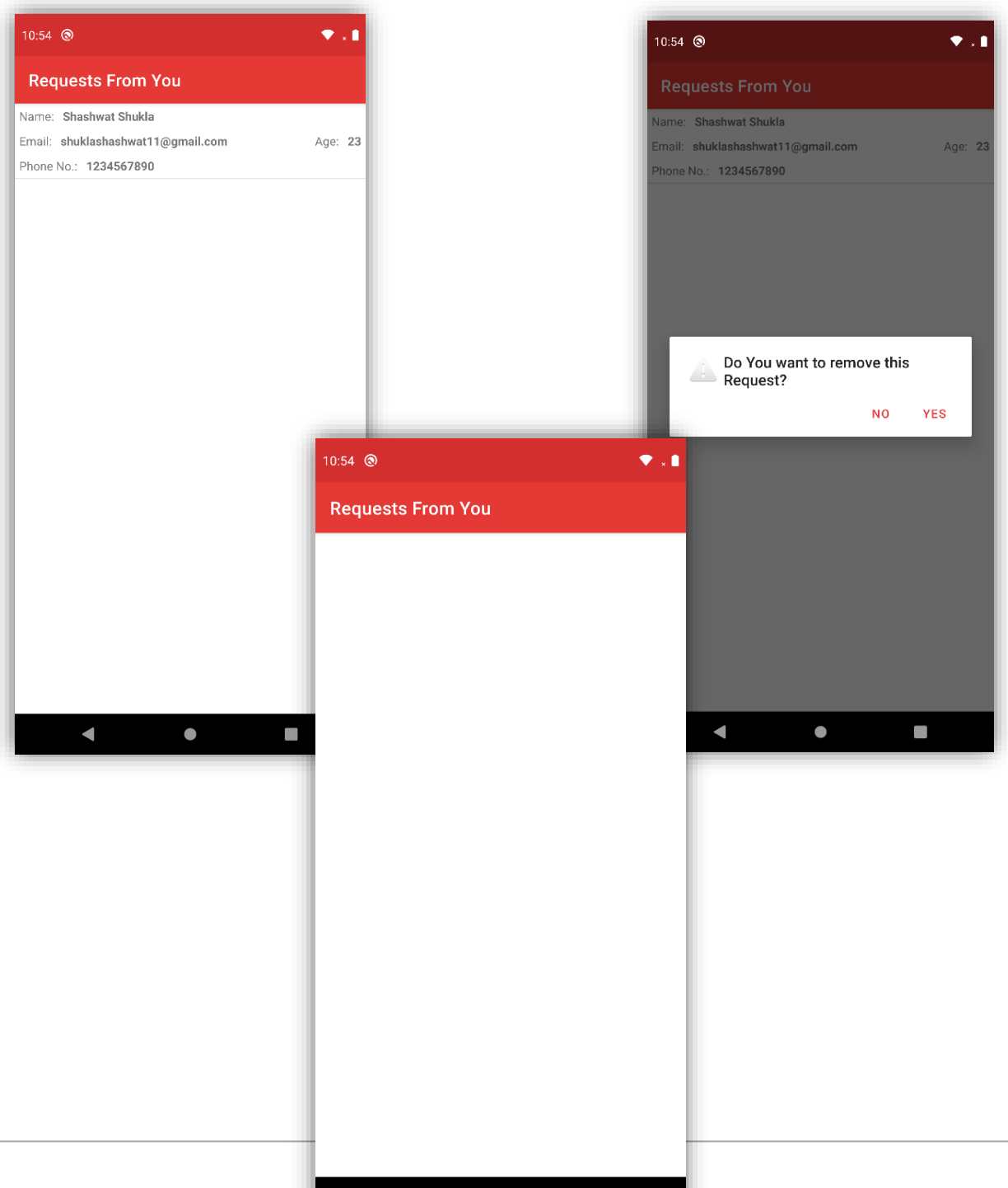
```
        }
    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
        Intent a = new Intent(Intent.ACTION_MAIN);
        a.addCategory(Intent.CATEGORY_HOME);
        a.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(a);
    }
}
```

Your Requests Activity

This is the Activity where you can Fetch and Look at all the Requests that you Received from the People in you Locality. And it's upto the user to decide whether they want to give or not (or able to or not). And to who they want to Donate blood .

If the User Decides to Donate Blood they can Contact the user with the Help of Contacts that the requesting party shared with them willingly.

The user also has the Option to remove requests as they please by Long Pressing on a Request and Selecting Yes from a Dialogue Box .



Your Requests – Layout -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=".your_requests">

    <ListView
        android:id="@+id/reqListView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Your Requests – Class File

```
package com.example.raktdaan;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.util.Log;
import android.widget.ListView;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
import java.util.Objects;

public class your_requests extends AppCompatActivity {

    ArrayList<Info> arrayList ;

    ListView listView ;
    ReqAdapter reqAdapter ;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_your_requests);
```

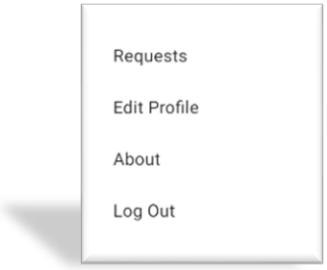
```
        setTitle( "Requests From You" );

        listView = findViewById( R.id.reqListView ) ;

        arrayList = new ArrayList<>() ;

        reqAdapter = new ReqAdapter( this , arrayList , R.layout.custom_req_listview )
;
        listView.setAdapter( reqAdapter );

        FirebaseDatabase.getInstance().getReference().child("requests").child(
FirebaseAuth.getInstance().getCurrentUser().getUid() ).orderByChild("uid").equalTo(
FirebaseAuth.getInstance().getCurrentUser().getUid() ).addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                arrayList.clear();

                for( DataSnapshot dataSnapshot : snapshot.getChildren() ){
                    arrayList.add(new Info( "UID" ,

Objects.requireNonNull(dataSnapshot.child("fname").getValue()).toString(),

Objects.requireNonNull(dataSnapshot.child("lname").getValue()).toString(),

Objects.requireNonNull(dataSnapshot.child("email").getValue()).toString(),

Objects.requireNonNull(dataSnapshot.child("phone").getValue()).toString(),

Objects.requireNonNull(dataSnapshot.child("age").getValue()).toString()));
                }
                reqAdapter.notifyDataSetChanged();
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        }) ;

        listView.setOnItemLongClickListener((adapterView, view, i, l) -> {
            new AlertDialog.Builder(your_requests.this)
                    .setIcon(android.R.drawable.ic_dialog_alert)
                    .setTitle( "Do You want to remove this Request?" )
                    .setPositiveButton("Yes", (dialogInterface, ii) -> {


FirebaseDatabase.getInstance().getReference().child("requests").child(FirebaseAuth.get
Instance().getCurrentUser().getUid()).orderByChild("fname").addListenerForSingleValueE
vent(new ValueEventListener() {
                            @Override
                            public void onDataChange(@NonNull DataSnapshot snapshot) {
                                snapshot.getRef().removeValue() ;
                            }

                            @Override
                            public void onCancelled(@NonNull DatabaseError error) {
```

```
                        }
                    }) ;
                    try {
                        arrayList.remove(i) ;
                    }catch ( Exception e ){
                        Log.i("Index" , "" + i ) ;
                    }
                    reqAdapter.notifyDataSetChanged();
                })
                .setNegativeButton("No", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialogInterface, int ii) {

                    }
                })
                .show() ;
                arrayList.remove(i) ;
            return false;
        });

    }
}
```

Some Essential Utility Files

Menu Files

Separate Menu Files are used in an Android Program and the Menu Files are inflated at the time of Creation of the Activity.



There are 2 Menu Files in this Project:

- Registration Activity Menu.
- Request Activity Menu.

Request Menu – XML

usermenu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/request" android:title="Requests" />
    <item android:id="@+id/editprofile" android:title="Edit Profile"/>
    <item android:id="@+id/about" android:title="About" />
    <item android:id="@+id/logout" android:title="Log Out" />
</menu>
```

Registration Menu - XML

req_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/r_log_out" android:title="Log Out"/>
</menu>
```

Info Class

This class is a Utility class it Performs no major operation on operation of the Application. It was created for Handling Data of Users. It Creates **Objects of the Users** Received on Querying by the User. These Objects can further be used to Inject Data into the **Custom ArrayAdapter**.

It consists of 2 Constructors:

- To Load Users data from who current user want to Requests Blood:
  - UID
  - Name
  - Gender
  - Location
  - Blood Type
  - Covid
- To Store Data of the Current user to send to the user from who they Request Blood:
  - UID
  - Name
  - E-Mail
  - Phone No.
  - Age.

Info – Class File

```
package com.example.raktdaan ;
//Class to Create Objects of Users and Store there Info.
public class Info {

    private  String uid = "" , fname = "" , lname = "" , gender = "" , location = "" ,
blood = "" , covid = "" , email = "" , pno = "" , age = "" ;
    private int imgID = 0 ;

    //Requesting Users
    public Info( String id , String f , String ln , String g , String l , String b ,
String c , int iId ){
        uid = id ; fname = f ; lname = ln ; gender = g ; location = l ; blood = b ;
covid = c ;
        imgID = iId ;
    }

    //Current Users
    public Info( String u , String f , String l , String e , String p ,String a  ){
        uid = u ; fname = f ; lname = l ; email = e ; pno = p ; age = a;
```

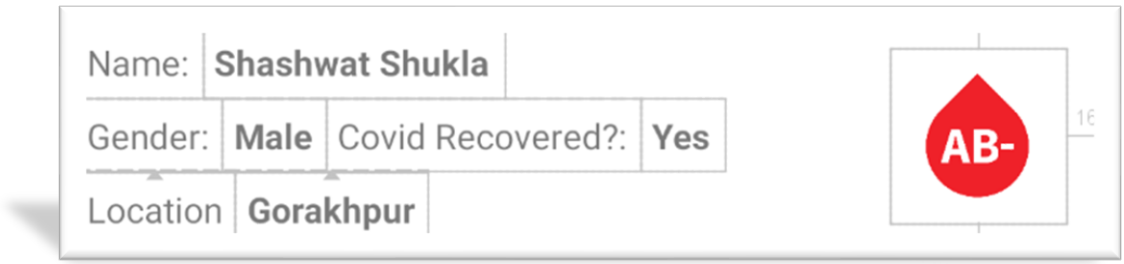
```
    }

    public String getUid() {
        return uid;
    }

    public String getFname() {
        return fname;
    }

    public String getGender() {
        return gender;
    }

    public String getLname() {
        return lname;
    }

    public String getLocation() {
        return location;
    }

    public int getImgID() {
        return imgID;
    }

    public String getCovid() {
        return covid;
    }

    public String getBlood() {
        return blood;
    }

    public String getEmail() {
        return email;
    }

    public String getPno() {
        return pno;
    }

    public String getAge() {
        return age;
    }
}
```

Custom ArrayAdapter

Custom **ArrayAdapter** are Overridden **ArrayAdapter**. Where Custom Layouts are Inflated using **LayoutInflator** and Injects Custom Data into the **View**.

This Apps Makes use of 2 Custom Adapters :

- **InfoAdapter:** To Show List of users into the ListView of the Requesting Activity.



custom_listview – Layout – 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"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:padding="5dp"
        android:text="Name:" />

    <TextView
        android:id="@+id/nameView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@id/name"
        android:padding="5dp"
        android:text="Shashwat Shukla"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/gender"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:padding="5dp"
        android:text="Gender:" />

    <TextView
        android:id="@+id/genderView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_toRightOf="@id/gender"
        android:padding="5dp"
        android:text="Male"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/location"
```

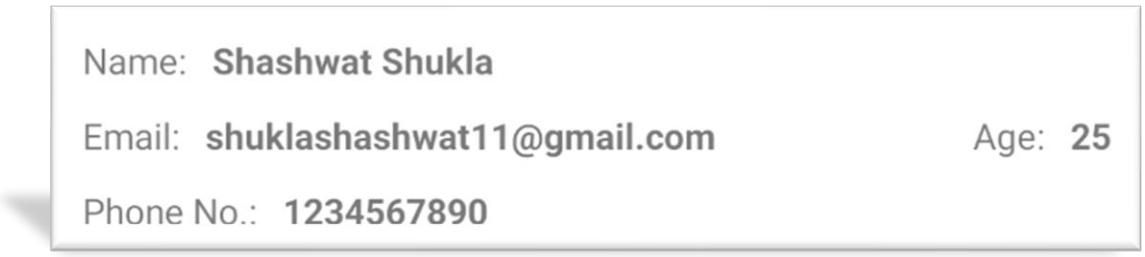
```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/gender"
        android:padding="5dp"
        android:text="Location" />

    <TextView
        android:id="@+id/locationView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/gender"
        android:layout_toRightOf="@id/location"
        android:padding="5dp"
        android:text="Gorakhpur"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/covid"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/nameView"
        android:layout_toRightOf="@id/genderView"
        android:padding="5dp"
        android:text="Covid Recovered?:" />

    <TextView
        android:id="@+id/covidView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/nameView"
        android:layout_toRightOf="@id/covid"
        android:padding="5dp"
        android:text="Yes"
        android:textStyle="bold" />


    <ImageView
        android:id="@+id/groupImgView"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true"
        android:layout_marginRight="16dp"
        app:srcCompat="@drawable/abn" />


</RelativeLayout>
```

InfoAdapter – Class File

```
package com.example.raktdaan ;

import android.app.Activity;
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;

import java.util.ArrayList;

//This Class Overrides Array Adapter and Allows me Implement my Own Vision of
the List View , This One is for the List of Users From whom you Can
//Request Blood
public class InfoAdapter extends ArrayAdapter {

    public InfoAdapter(Activity context , ArrayList<Info> infos , int color ){
        super(context , 0  , infos) ;
    }

    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull
ViewGroup parent) {
        View customView = convertView ;
        if( customView == null ){
            customView =
LayoutInflater.from(getContext()).inflate((R.layout.custom_listview) , parent
,false) ;
        }

        Info currInfo = (Info) getItem(position) ;

        TextView name = customView.findViewById( R.id.nameView ) ;
        TextView gender = customView.findViewById( R.id.genderView ) ;
        TextView location = customView.findViewById( R.id.locationView ) ;
        TextView covid = customView.findViewById(R.id.covidView) ;
        ImageView imageView = customView.findViewById( R.id.groupImgView ) ;

        String fullName = currInfo.getFname() + " " + currInfo.getLname() ;
        name.setText( fullName );
        gender.setText( currInfo.getGender() );
        location.setText( currInfo.getLocation() );
        covid.setText(currInfo.getCovid());

        String bloodGroup = currInfo.getBlood() ;
        imageView.setImageDrawable( customView.getResources().getDrawable(
currInfo.getImgID() ) );

        return customView ;
    }
}
```

- **ReqAdapter:** To Show List of Users into the ListView of Your Request Activity.



custom_req_listview – 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/rRelativeLayout">

    <TextView
        android:id="@+id/rName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:layout_alignParentStart="true"
        android:text="Name:" />

    <TextView
        android:id="@+id/rNameView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:layout_toRightOf="@id/rName"
        android:textStyle="bold"
        android:text="Shashwat Shukla"/>

    <TextView
        android:id="@+id/rEmail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:layout_below="@id/rName"
        android:text="Email:" />

    <TextView
        android:id="@+id/rEmailView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:layout_below="@id/rName"
        android:layout_toRightOf="@id/rEmail"
        android:textStyle="bold"
        android:text="shuklashashwat11@gmail.com" />

    <TextView
        android:id="@+id/rPhone"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:layout_below="@id/rEmail"
        android:text="Phone No.:" />

    <TextView
        android:id="@+id/rPhoneView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:layout_below="@id/rEmail"
        android:layout_toRightOf="@id/rPhone"
        android:textStyle="bold"
        android:text="1234567890" />

    <TextView
        android:id="@+id/rAge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:layout_toLeftOf="@id/rAgeView"
        android:layout_centerVertical="true"
        android:text="Age:" />


    <TextView
        android:id="@+id/rAgeView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:textStyle="bold"
        android:text="25" />

</RelativeLayout>
```

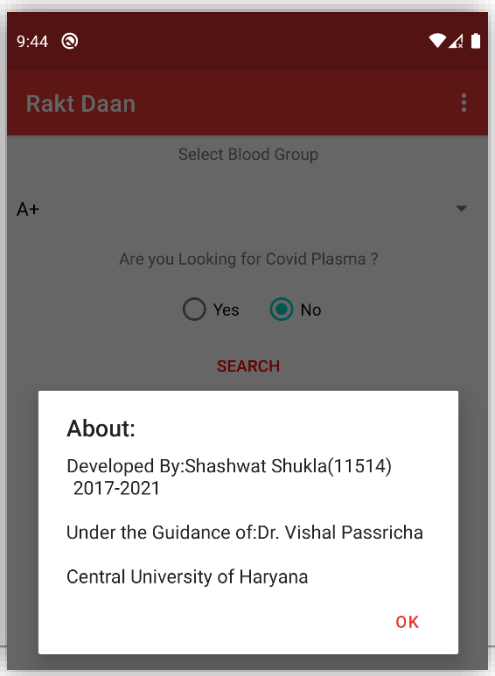
ReqAdapter – Class File

```
package com.example.raktdaan;

import android.app.Activity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;

import java.util.ArrayList;
//This Class Overrides Array Adapter and Allows me Implement my Own Vision of
the List View , This One is for the List of Users
//Who has Requested Blood From You .
public class ReqAdapter extends ArrayAdapter {
```

```
    public ReqAdapter(Activity context , ArrayList<Info> infos , int color ){
        super(context , 0  , infos) ;
    }

    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull
ViewGroup parent) {

        View customView = convertView ;

        if( customView == null ){
            customView =
LayoutInflater.from(getContext()).inflate((R.layout.custom_req_listview) ,
parent ,false) ;
        }

        Info currInfo = (Info) getItem( position ) ;

        TextView rName = customView.findViewById( R.id.rNameView ) ;
        TextView rAge = customView.findViewById( R.id.rAgeView ) ;
        TextView rEmail = customView.findViewById( R.id.rEmailView ) ;
        TextView rPhone = customView.findViewById( R.id.rPhoneView ) ;

        String fullName = currInfo.getFname() + " " + currInfo.getLname() ;
        rName.setText( fullName );
        rAge.setText(currInfo.getAge());
        rEmail.setText(currInfo.getEmail());
        rPhone.setText( currInfo.getPno() );

        return customView ;
    }
}
```

MyDialogueFragment

MyDialogueFragment is a Overloaded **DialogueFragment Class** which is used to Create the **About Section** of the App.



MyDialogueFragment – Class File

```
package com.example.raktdaan;

import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;
import android.os.Bundle;

import androidx.fragment.app.DialogFragment;

public class MyDialogFragment extends DialogFragment {
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {

        // Use the Builder class for convenient dialog construction
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setTitle("About:");
        builder.setMessage("Developed By:Shashwat Shukla(11514)\n\t2017-
2021\n\nUnder the Guidance of:Dr. Vishal Passricha\n\nCentral University of
Haryana");
        builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                // You don't have to do anything here if you just
                // want it dismissed when clicked
            }
        });
        // Create the AlertDialog object and return it
        return builder.create();
    }
}
```

AndroidManifest

Every app project must have an AndroidManifest.xml file (with precisely that name) at the root of the **project source set**. The manifest file describes essential information about your app to the Android build tools, the Android operating system, and Google Play.

Among many other things, the manifest file is required to declare the following:

- The app's package name.
- The components of the app.
- The permissions that the app needs in order to access protected parts of the system or other apps.

AndroidManifest - XML

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.raktdaan">

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

```
    <application
        android:allowBackup="true"
        android:icon="@drawable/donation"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.RaktDaan">
        <activity android:name=".your_requests"></activity>
        <activity android:name=".splash_screen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".requests" />
        <activity android:name=".registration" />
        <activity android:name=".MainActivity" />
    </application>

</manifest>
```


Thank
You