Android_manifest.xml:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.padma.studentchatapp">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.StudentChatApp"
        tools:targetApi="31">

        <!-- Set LoginActivity as launcher -->
        <activity
            android:name=".LoginActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".RegisterActivity" android:exported="true" />
        <activity android:name=".UserListActivity" android:exported="true" />
        <activity android:name=".ChatActivity" android:exported="true" />
```

```
        <activity android:name=".MainActivity" android:exported="false" />
    </application>

</manifest>
```

Activity_register.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="24dp"
    android:orientation="vertical"
    android:gravity="center"
    android:background="#FF6F6F">

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:padding="10dp"
        android:background="@android:drawable/editbox_background" />

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:padding="10dp"
        android:background="@android:drawable/editbox_background"
        android:layout_marginTop="12dp"/>

    <Button
        android:id="@+id/buttonRegister"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Register"
        android:layout_marginTop="20dp"
        android:background="#FF4081"
        android:textColor="#FFFFFF"/>
</LinearLayout>
```

Activity_login.xml:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent"
    android:padding="20dp" android:orientation="vertical">

    <EditText android:id="@+id/editTextEmail"
        android:hint="Email" android:inputType="textEmailAddress"
        android:layout_width="match_parent" android:layout_height="wrap_content"/>

    <EditText android:id="@+id/editTextPassword"
        android:hint="Password" android:inputType="textPassword"
```

```
    android:layout_width="match_parent" android:layout_height="wrap_content"/>

  <Button android:id="@+id/buttonLogin"
    android:text="Login" android:layout_width="match_parent"
android:layout_height="wrap_content"/>

  <Button android:id="@+id/buttonToRegister"
    android:text="Go to Register" android:layout_width="match_parent"
android:layout_height="wrap_content"/>
</LinearLayout>
```

Activity_chat.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical" android:layout_width="match_parent"
  android:layout_height="match_parent">

  <androidx.recyclerview.widget.RecyclerView
    android:id="@+id/recyclerViewMessages"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:padding="10dp" />

  <LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="8dp">
```

```
        <EditText
            android:id="@+id/editTextMessage"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:hint="Type a message" />

        <ImageButton
            android:id="@+id/buttonSend"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@android:drawable/ic_menu_send" />
    </LinearLayout>
</LinearLayout>
```

RegisterActivity.java:

```
package com.padma.studentchatapp;

import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.FirebaseDatabase;
```

```
public class RegisterActivity extends AppCompatActivity {

    EditText editTextEmail, editTextPassword;
    Button buttonRegister;
    FirebaseAuth mAuth;

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

        // Initialize Firebase Authentication
        mAuth = FirebaseAuth.getInstance();

        // Bind UI elements
        editTextEmail = findViewById(R.id.editTextEmail);
        editTextPassword = findViewById(R.id.editTextPassword);
        buttonRegister = findViewById(R.id.buttonRegister);

        // Register button click listener
        buttonRegister.setOnClickListener(view -> {
            String email = editTextEmail.getText().toString().trim();
            String password = editTextPassword.getText().toString().trim();

            // Basic input validation
            if (email.isEmpty() || password.isEmpty()) {
                Toast.makeText(this, "Email and Password cannot be empty",
Toast.LENGTH_SHORT).show();
                return;
```

```
        }

        // Firebase create user
        mAuth.createUserWithEmailAndPassword(email, password)
                .addOnCompleteListener(task -> {
                    if (task.isSuccessful()) {
                        // Get UID and save user to database
                        String uid = mAuth.getCurrentUser().getUid();
                        User newUser = new User(uid, email);

                        FirebaseDatabase.getInstance().getReference("users")
                                .child(uid)
                                .setValue(newUser)
                                .addOnCompleteListener(dbTask -> {
                                    if (dbTask.isSuccessful()) {
                                        Toast.makeText(this, "Registration Successful!",
Toast.LENGTH_SHORT).show();

                                        // ✅ Redirect to UserListActivity
                                        startActivity(new Intent(RegisterActivity.this,
UserListActivity.class));
                                        finish();
                                    } else {
                                        Toast.makeText(this, "Database Error: " +
dbTask.getException().getMessage(), Toast.LENGTH_LONG).show();
                                    }
                                });
                    } else {
                        Toast.makeText(this, "Error: " + task.getException().getMessage(),
Toast.LENGTH_LONG).show();
```

```
                }
            });
        });
    }
}
```

ChatActivity.java:

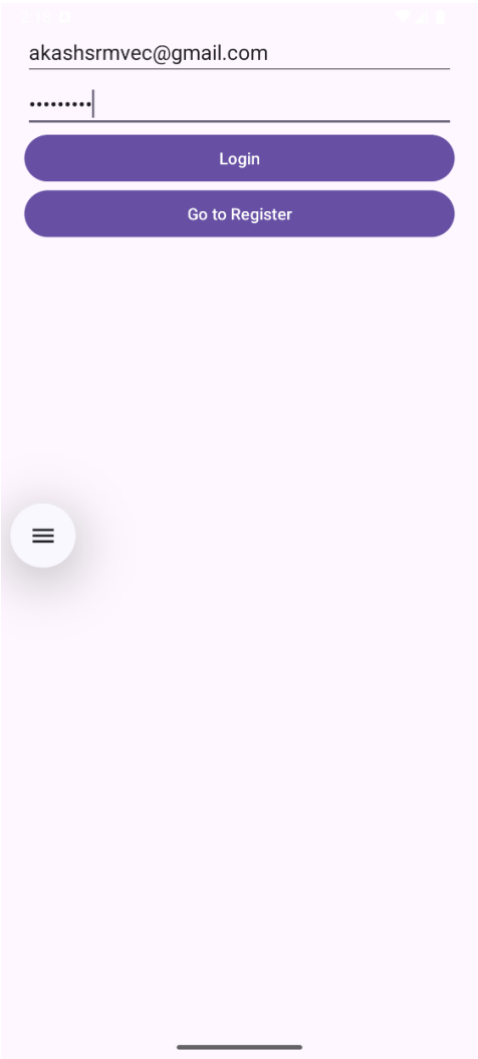
```
package com.padma.studentchatapp;

import android.os.Bundle;
import android.widget.EditText;
import android.widget.ImageButton;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.*;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class ChatActivity extends AppCompatActivity {

    EditText editTextMessage;
    ImageButton buttonSend;
```

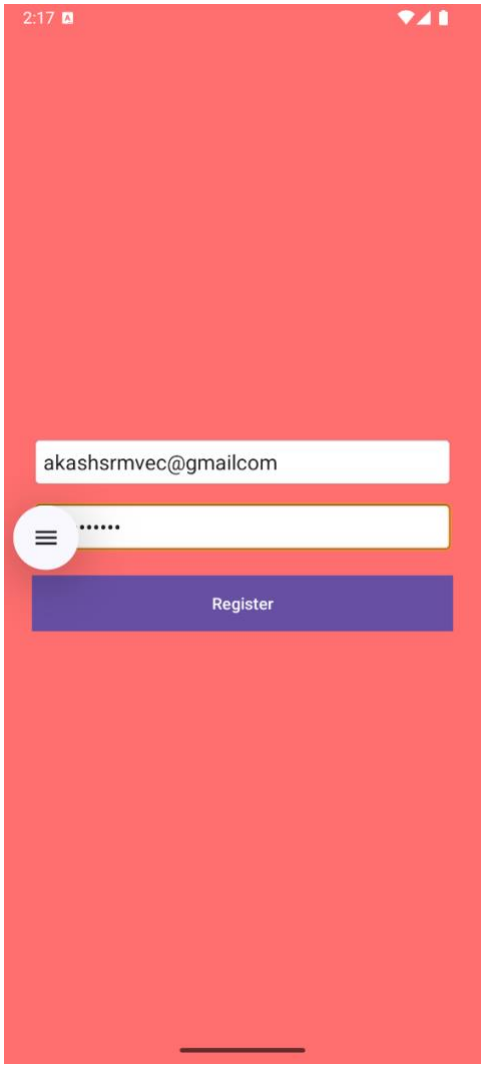
```
    RecyclerView recyclerViewMessages;
    MessageAdapter messageAdapter;
    List<Message> messageList;

    String receiverId, senderId, chatRoomId;
    DatabaseReference chatRef;

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

        editTextMessage = findViewById(R.id.editTextMessage);
        buttonSend = findViewById(R.id.buttonSend);
        recyclerViewMessages = findViewById(R.id.recyclerViewMessages);

        receiverId = getIntent().getStringExtra("receiverId");
        senderId = FirebaseAuth.getInstance().getCurrentUser().getUid();

        // Create common chat room ID regardless of sender or receiver order
        if (senderId.compareTo(receiverId) < 0) {
            chatRoomId = senderId + "_" + receiverId;
        } else {
            chatRoomId = receiverId + "_" + senderId;
        }

        chatRef =
FirebaseDatabase.getInstance().getReference("chats").child(chatRoomId).child("messages");
```

```
    messageList = new ArrayList<>();
    messageAdapter = new MessageAdapter(this, messageList, senderId);

    recyclerViewMessages.setLayoutManager(new LinearLayoutManager(this));
    recyclerViewMessages.setAdapter(messageAdapter);

    buttonSend.setOnClickListener(v -> {
        String messageText = editTextMessage.getText().toString().trim();
        if (!messageText.isEmpty()) {
            sendMessage(senderId, receiverId, messageText);
            editTextMessage.setText("");
        }
    });

    readMessages();
}

private void sendMessage(String sender, String receiver, String message) {
    HashMap<String, String> msg = new HashMap<>();
    msg.put("sender", sender);
    msg.put("receiver", receiver);
    msg.put("message", message);

    chatRef.push().setValue(msg);
}

private void readMessages() {
    chatRef.addValueEventListener(new ValueEventListener() {
        @Override
```

```
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            messageList.clear();
            for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
                Message msg = dataSnapshot.getValue(Message.class);
                messageList.add(msg);
            }
            messageAdapter.notifyDataSetChanged();
            recyclerViewMessages.scrollToPosition(messageList.size() - 1);
        }

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

OUTPUT:


akashsrmvec@gmail.com
Login
Go to Register


2:17
akashsrmvec@gmailcom
Register


alexmcc@gmail.com
akashsrmvec@gmail.com
elumalai@gmail.com


hii akash
how are you?
I'm fine Alex
what are you doing ?
Type a message

Android manifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.ex8"> <!-- Change this to match your actual package name -->

    <!-- Permissions for reading/writing to external storage -->
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"
/>

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Ex8"
        tools:targetApi="31">

        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
```

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

</application>

</manifest>
```

ActivityMain.xml

```
<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="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Saved Data Will Appear Here" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="200dp"
```

```
        android:layout_height="wrap_content"
        android:layout_above="@id/button1"
        android:layout_toRightOf="@id/textView1"
        android:layout_marginBottom="50dp"
        android:hint="Enter data"
        android:ems="10" >
        <requestFocus />
    </EditText>

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView1"
        android:layout_centerVertical="true"
        android:layout_marginLeft="32dp"
        android:text="Save Data" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@id/button1"
        android:layout_centerVertical="true"
        android:layout_marginLeft="36dp"
        android:text="Show Data" />
</RelativeLayout>
```

MainActivity.java

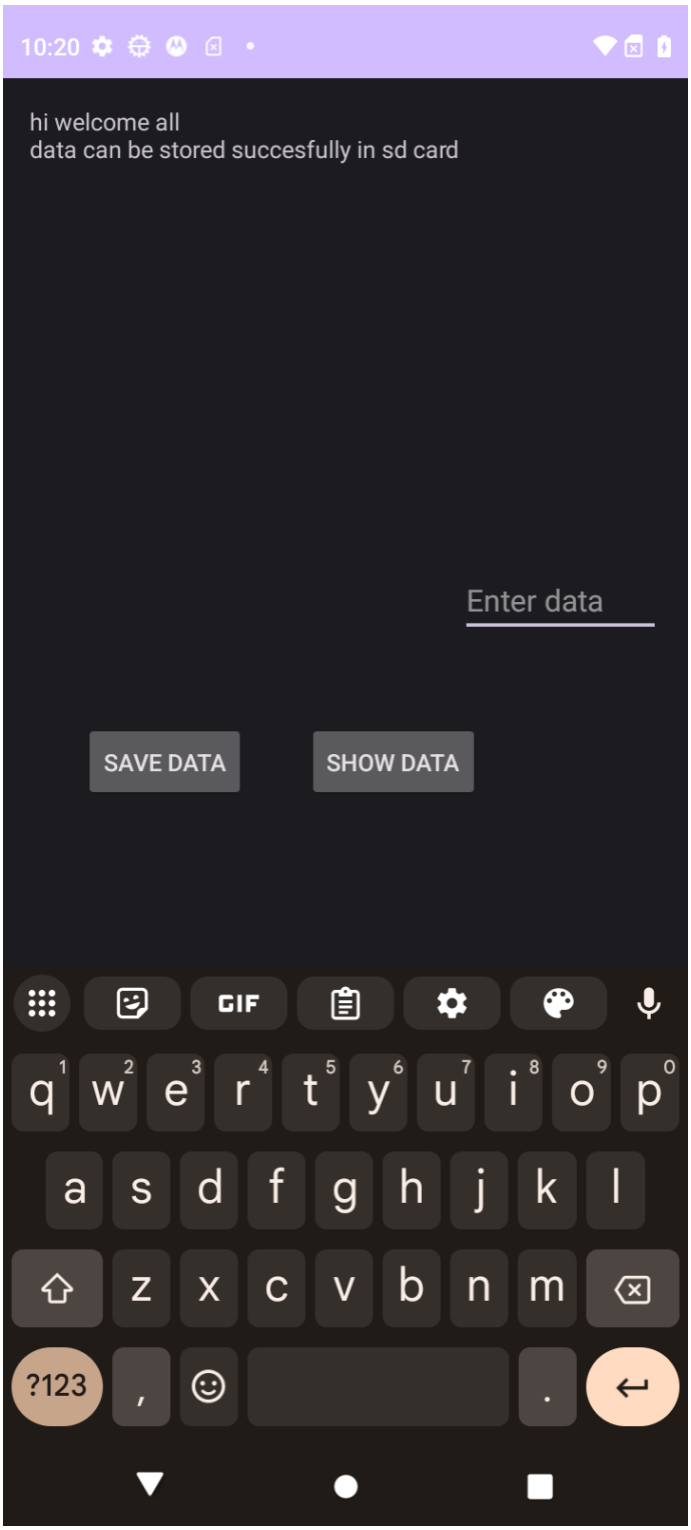
```
package com.example.My Application8;

import android.Manifest;
import android.app.Activity;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class MainActivity extends Activity {

    Button b1, b2;
    EditText e;
    TextView tv;
```

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

        tv = findViewById(R.id.textView1);
        b1 = findViewById(R.id.button1);
        b2 = findViewById(R.id.button2);
        e = findViewById(R.id.editText1);

        // Request storage permissions at runtime
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)
                != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this,
                    new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, 1);
        }

        // Save data to SD card
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String text = e.getText().toString();
                if (isExternalStorageWritable()) {
                    File file = new File(getExternalFilesDir(null), "myfile.txt");
                    try (FileOutputStream fos = new FileOutputStream(file, true)) {
                        fos.write((text + "\n").getBytes());
                        Toast.makeText(getApplicationContext(), "Data saved to SD card",
Toast.LENGTH_SHORT).show();
                        e.setText(""); // Clear after saving
```

```
            } catch (IOException ex) {
                Toast.makeText(getApplicationContext(), "Save failed",
Toast.LENGTH_SHORT).show();
                ex.printStackTrace();
            }
        }
    }
});

// Show data from SD card
b2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        File file = new File(getExternalFilesDir(null), "myfile.txt");
        if (isExternalStorageReadable()) {
            try (FileInputStream fis = new FileInputStream(file)) {
                int ch;
                StringBuilder builder = new StringBuilder();
                while ((ch = fis.read()) != -1) {
                    builder.append((char) ch);
                }
                tv.setText(builder.toString());
                Toast.makeText(getApplicationContext(), "File read",
Toast.LENGTH_SHORT).show();
            } catch (IOException ex) {
                Toast.makeText(getApplicationContext(), "Read failed",
Toast.LENGTH_SHORT).show();
                ex.printStackTrace();
            }
        }
```

```
            }
        });
    }
private boolean isExternalStorageWritable() {
        return
Environment.MEDIA_MOUNTED.equals(Environment.getExternalStorageState());
    }


    private boolean isExternalStorageReadable() {
        String state = Environment.getExternalStorageState();
        return Environment.MEDIA_MOUNTED.equals(state) ||
                Environment.MEDIA_MOUNTED_READ_ONLY.equals(state);
    }
}
```

OUTPUT:


10:20
hi welcome all
data can be stored succesfully in sd card
Enter data
SAVE DATA
SHOW DATA
GIF
?123

Android Manifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.gpstrackingapp">
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.GPSTrackingApp"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>     </application>
```

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="20dp"
    android:gravity="center">

    <Button
        android:id="@+id/btnShowLocation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show My Location"/>
</LinearLayout>
```

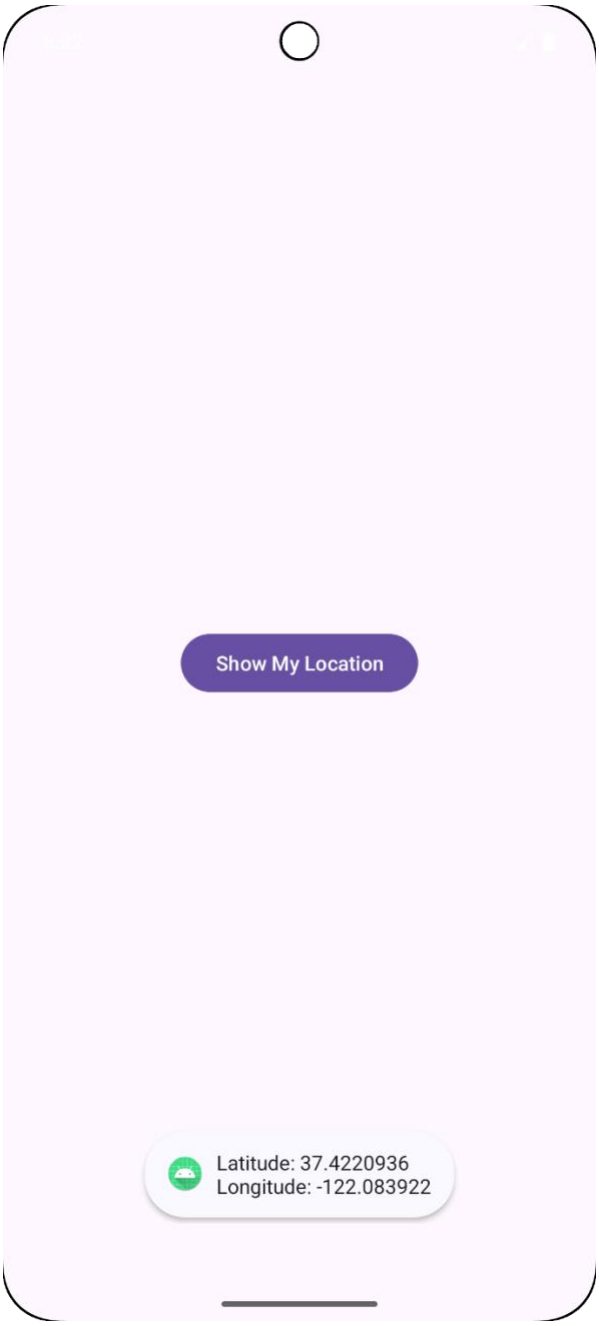
Main Activity.java:

```
package com.example.gpstrackingapp;
import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.widget.Button;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
```

```
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnSuccessListener;
public class MainActivity extends AppCompatActivity {
    Button btnShowLocation;
    FusedLocationProviderClient fusedLocationClient;
    private static final int LOCATION_PERMISSION_CODE = 101;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnShowLocation = findViewById(R.id.btnShowLocation);
        fusedLocationClient = LocationServices.getFusedLocationProviderClient(this);
        btnShowLocation.setOnClickListener(view -> {
            if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION},
LOCATION_PERMISSION_CODE);
            } else {
                getLastLocation();
            }
        });
    }
    private void getLastLocation() {
        fusedLocationClient.getLastLocation()
                .addOnSuccessListener(this, location -> {
                    if (location != null) {
                        double lat = location.getLatitude();
```

```
                double lon = location.getLongitude();
                Toast.makeText(getApplicationContext(),
                        "Latitude: " + lat + "\nLongitude: " + lon,
                        Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(getApplicationContext(), "Location not available",
Toast.LENGTH_SHORT).show();
            }
        });
    }
    // Handle permission result
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
                                           @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == LOCATION_PERMISSION_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                getLastLocation();
            } else {
                Toast.makeText(this, "Permission denied", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

OUTPUT:



Activity.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/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <!-- Drag & Drop UI -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent">

        <!-- Draggable Emoji -->
        <TextView
            android:id="@+id/emoji1"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:text="🍏"
            android:textSize="30sp"
```

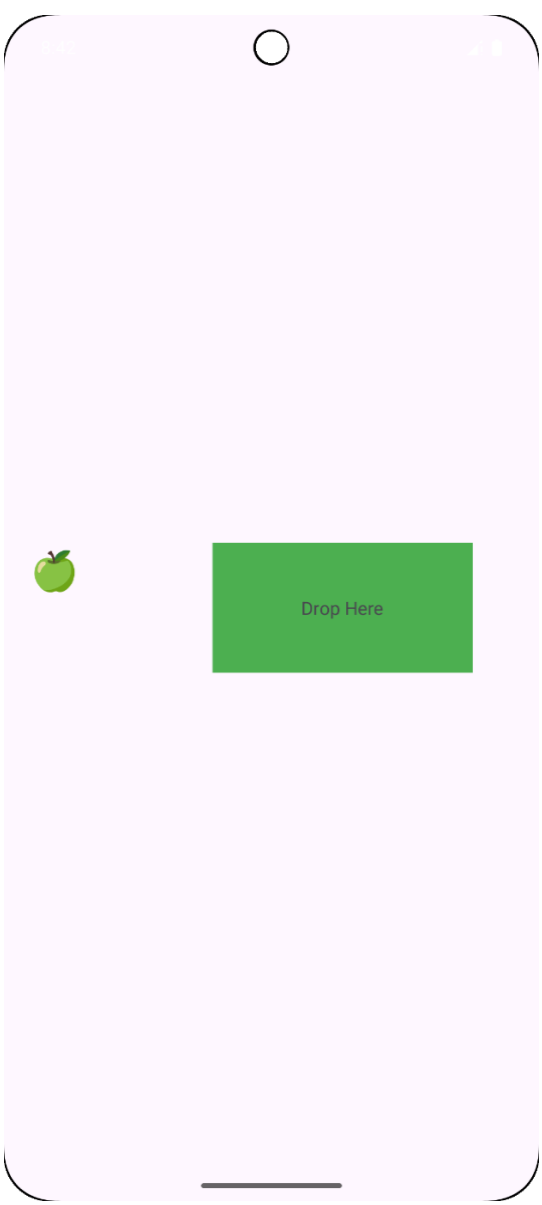
```
        android:layout_gravity="center_vertical"
        android:layout_marginStart="20dp"/>

    <!-- Drop Target -->
    <TextView
        android:id="@+id/target1"
        android:layout_width="200dp"
        android:layout_height="100dp"
        android:background="#4CAF50"
        android:text="Drop Here"
        android:layout_gravity="center_vertical"
        android:layout_marginStart="40dp"
        android:gravity="center"/>
  </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

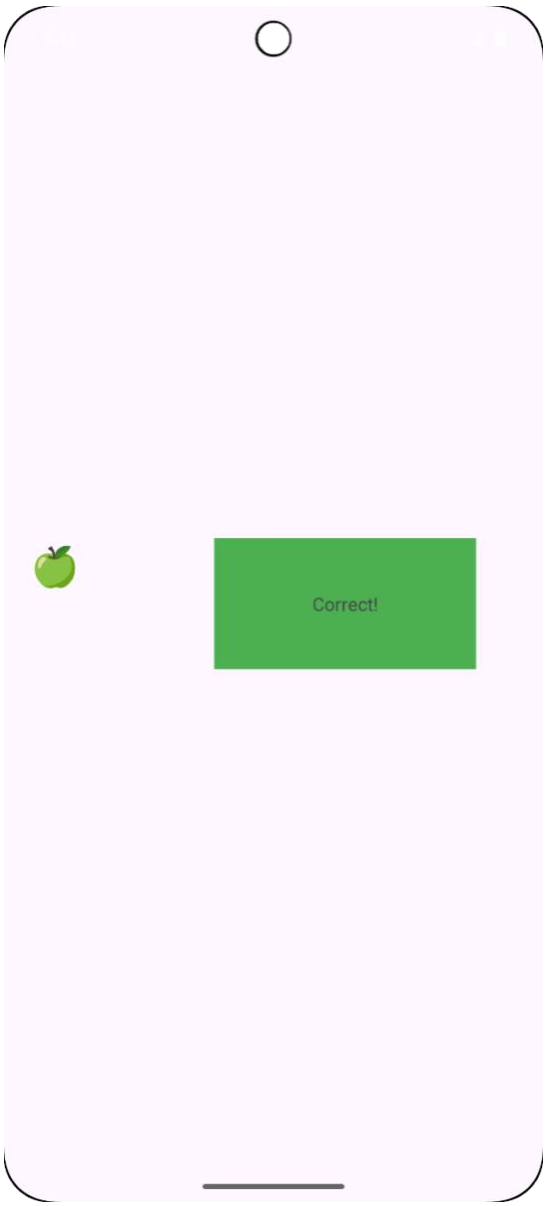
MainActivity.java:

```
package com.example.dragdrop;

import android.os.Bundle;
import android.view.DragEvent;
import android.view.View;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Set drag listeners
        findViewById(R.id.emoji1).setOnLongClickListener(v -> {
            View.DragShadowBuilder shadow = new View.DragShadowBuilder(v);
            v.startDragAndDrop(null, shadow, v, 0);
            return true;
        });

         findViewById(R.id.target1).setOnDragListener((v, event) -> {
            if (event.getAction() == DragEvent.ACTION_DROP) {
                TextView dropped = (TextView) event.getLocalState();
                TextView target = (TextView) v;
                target.setText("Correct!");
            }
            return true;
        });
    }
}
```

OUTPUT:





Main_activity.java:

```
package com.example.alarmclock;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.ToggleButton;
import android.view.View;
import androidx.appcompat.app.AppCompatActivity;
import java.util.Calendar;

public class MainActivity extends AppCompatActivity {

    private AlarmManager alarmManager;
    private PendingIntent pendingIntent;
    private TimePicker alarmTimePicker;
    private TextView alarmText;

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

        alarmTimePicker = findViewById(R.id.alarmTimePicker);
        alarmText = findViewById(R.id.alarmText);
        alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
```

```
    }

    public void onToggleClicked(View view) {
        ToggleButton toggle = (ToggleButton) view;

        if (toggle.isChecked()) {
            int hour = alarmTimePicker.getHour();
            int minute = alarmTimePicker.getMinute();

            Calendar calendar = Calendar.getInstance();
            calendar.set(Calendar.HOUR_OF_DAY, hour);
            calendar.set(Calendar.MINUTE, minute);
            calendar.set(Calendar.SECOND, 0);

            Intent intent = new Intent(this, AlarmReceiver.class);
            pendingIntent = PendingIntent.getBroadcast(this, 0, intent,
PendingIntent.FLAG_IMMUTABLE);
            alarmManager.set(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(),
pendingIntent);

            alarmText.setText("Alarm set for: " + hour + ":" + String.format("%02d", minute));
        } else {
            if (pendingIntent != null) {
                alarmManager.cancel(pendingIntent);
            }
            alarmText.setText("Alarm canceled");
        }
    }
}
```

AlarmService.java:

```
package com.example.alarmclock;
import android.app.IntentService;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import androidx.core.app.NotificationCompat;

public class AlarmService extends IntentService {

    public AlarmService() {
        super("AlarmService");
    }

    @Override
    protected void onHandleIntent(Intent intent) {
        sendNotification("Wake Up! Alarm is ringing!");
    }

    private void sendNotification(String msg) {
        NotificationManager manager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

        Intent intent = new Intent(this, MainActivity.class);
        PendingIntent contentIntent = PendingIntent.getActivity(this, 0, intent,
PendingIntent.FLAG_IMMUTABLE);

        NotificationCompat.Builder builder = new NotificationCompat.Builder(this, "default")
```

```
            .setSmallIcon(R.drawable.ic_launcher_foreground)
            .setContentTitle("Alarm Clock")
            .setContentText(msg)
            .setContentIntent(contentIntent)
            .setAutoCancel(true);

        manager.notify(1, builder.build());
    }
}
```

AlarmReceiver.java:

```
package com.example.alarmclock;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;

public class AlarmReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        Uri alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);
        if (alarmUri == null) {
            alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        }
```

```
        Ringtone ringtone = RingtoneManager.getRingtone(context, alarmUri);
        ringtone.play();

        Intent service = new Intent(context, AlarmService.class);
        context.startService(service);
    }
}
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.WAKE_LOCK" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AlarmClock"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
```

```
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <!-- Register the broadcast receiver for alarm -->
    <receiver android:name=".AlarmReceiver" />
    <!-- Register the service to handle alarm notification -->
    <service android:name=".AlarmService" />
  </application>
</manifest>
```

Activity.xml:

```
<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=".MainActivity">

  <TimePicker
    android:id="@+id/alarmTimePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="50dp"/>

  <ToggleButton
    android:id="@+id/alarmToggle"
    android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="Alarm On/Off"
        android:onClick="onToggleClicked"
        android:layout_below="@id/alarmTimePicker"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="30dp"/>

    <TextView
        android:id="@+id/alarmText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="No Alarm Set"
        android:layout_below="@id/alarmToggle"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"/>
</RelativeLayout>
```

OUTPUT: