

**Lab 09**

CTE306 – Mobile Application Development

Date – 19th September 2022

Divyash Chhetri

02200174

BE 3 IT

Module Tutor: Mr. Pema Galey

Department of Information Technology

**College of Science and Technology**

**Aim**

Perform the task on following topics:

1. Services (Notification, Alarm and Schedule)
2. Storage (Shared Preferences and SQLite Storage)

**Theory**

Services are a unique element in Android that lets an app to run in the background and perform continuous functions. A service's main objective is to maintain an application in the background so that the user can use many applications at once. A user interface is not suited for Android services since it is designed to carry out ongoing tasks without user input. A service can go running in the background even if the program is closed or the user switches to another application. Furthermore, in order to interact with other processes, application components might link themselves to services (IPC).

One of the most intriguing data storage options that Android gives its users is Shared Preferences. Shared Preferences is a technique for saving and retrieving minuscule amounts of basic data, such as String, int, float, and Boolean, which make up your preferences in an XML file inside the software, as key/value pairs to a file on the device storage. Shared Preferences can be compared to a dictionary or a key/value pair.

**Alarm and Notification**

**Program Code**

*activity\_main.xml*

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<Button

android:id="@+id/notify"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/add\_alarm"

android:layout\_marginTop="50sp"

android:layout\_marginStart="150sp"/>

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

*MainActivity.java*

package com.example.alarmandnotificaton;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import androidx.annotation.RequiresApi;

import androidx.core.app.NotificationCompat;

import androidx.core.app.NotificationManagerCompat;

import android.app.NotificationChannel;

import android.app.NotificationManager;

import android.content.Intent;

import android.os.Build;

import android.provider.AlarmClock;

import android.widget.Button;

public class MainActivity extends AppCompatActivity {

@RequiresApi(api = Build.VERSION\_CODES.O)

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

NotificationChannel();

Button notify = (Button) findViewById(R.id.notify);

NotificationCompat.Builder builder = new NotificationCompat.Builder(this, "Hello")

.setSmallIcon(R.drawable.ic\_baseline\_access\_alarm\_24)

.setContentTitle("Alarm").setContentText("Setting Alarm...").setPriority(NotificationCompat.PRIORITY\_DEFAULT);

NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);

notify.setOnClickListener(view -> {

notificationManager.notify(100, builder.build());

Intent i = new Intent(AlarmClock.ACTION\_SHOW\_ALARMS);

startActivity(i);

});

}

@RequiresApi(api = Build.VERSION\_CODES.O)

private void NotificationChannel() {

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {

CharSequence name = "HelloWorldChannel";

String description = "Channel for Hello World Notification";

int importance = NotificationManager.IMPORTANCE\_DEFAULT;

NotificationChannel channel = new NotificationChannel("Hello", name, importance);

channel.setDescription(description);

NotificationManager notificationManager = getSystemService(NotificationManager.class);

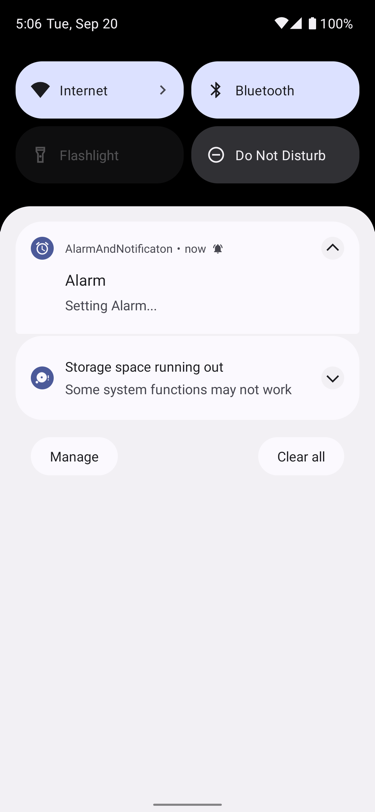
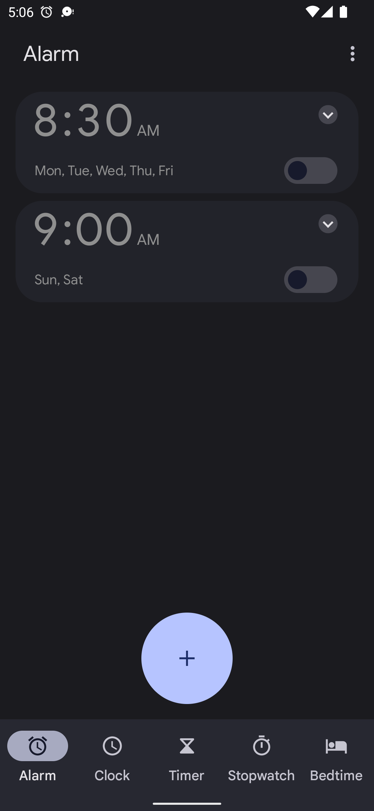
notificationManager.createNotificationChannel(channel);

}

}

}

**Output**

**  **

**Schedule Application**

**Program Code**

*activity\_main.xml*

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<Button

android:id="@+id/button2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="132dp"

android:layout\_marginTop="113dp"

android:layout\_marginEnd="132dp"

android:layout\_marginBottom="41dp"

android:onClick="scheduleJob"

android:text="@string/schedule\_job"

app:layout\_constraintBottom\_toTopOf="@+id/button"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="141dp"

android:layout\_marginEnd="142dp"

android:layout\_marginBottom="481dp"

android:onClick="cancelJob"

android:text="@string/cancel\_job"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/button2" />

</androidx.constraintlayout.widget.ConstraintLayout>

*MainActivity.java*

package com.example.scheduleapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.app.job.JobInfo;

import android.app.job.JobScheduler;

import android.content.ComponentName;

import android.util.Log;

import android.view.View;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

private static final String TAG = "MainActivity";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

public void scheduleJob(View v) {

ComponentName componentName = new ComponentName(this, JobServiceExample.class);

JobInfo info = new JobInfo.Builder(123, componentName)

.setRequiresCharging(true)

.setRequiredNetworkType(JobInfo.NETWORK\_TYPE\_UNMETERED)

.setPersisted(true).setPeriodic(15 \* 60 \* 1000)

.build();

JobScheduler scheduler = (JobScheduler) getSystemService(JOB\_SCHEDULER\_SERVICE);

int resultCode = scheduler.schedule(info);

if (resultCode == JobScheduler.RESULT\_SUCCESS) {

Log.d(TAG, "Job scheduled");

Toast.makeText(getApplicationContext(),"Job Scheduled", Toast.LENGTH\_SHORT).show();

} else {

Log.d(TAG, "Job scheduling failed");

Toast.makeText(getApplicationContext(),"Job Scheduling Failed", Toast.LENGTH\_SHORT).show();

}

}

public void cancelJob(View v) {

JobScheduler scheduler = (JobScheduler) getSystemService(JOB\_SCHEDULER\_SERVICE);

scheduler.cancel(123);

Log.d(TAG, "Job cancelled");

Toast.makeText(getApplicationContext(),"Job Schedule Cancelled", Toast.LENGTH\_SHORT).show();

}

}

*JobServiceExample.java*

package com.example.scheduleapplication;

import android.app.job.JobParameters;

import android.app.job.JobService;

import android.util.Log;

public class JobServiceExample extends JobService {

private static final String TAG = "ExampleJobService";

private boolean jobCancelled = false;

@Override

public boolean onStartJob(JobParameters params) {

Log.d(TAG, "Job started");

doBackgroundWork(params);

return false;

}

private void doBackgroundWork(final JobParameters params) {

new Thread(new Runnable() {

@Override

public void run() {

for (int i = 0; i < 10; i++) {

Log.d(TAG, "run: " + i);

if (jobCancelled) {

return;

}

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

Log.d(TAG, "Job finished");

jobFinished(params, false);

}

}).start();

}

@Override

public boolean onStopJob(JobParameters params) {

Log.d(TAG, "Job cancelled before completion");

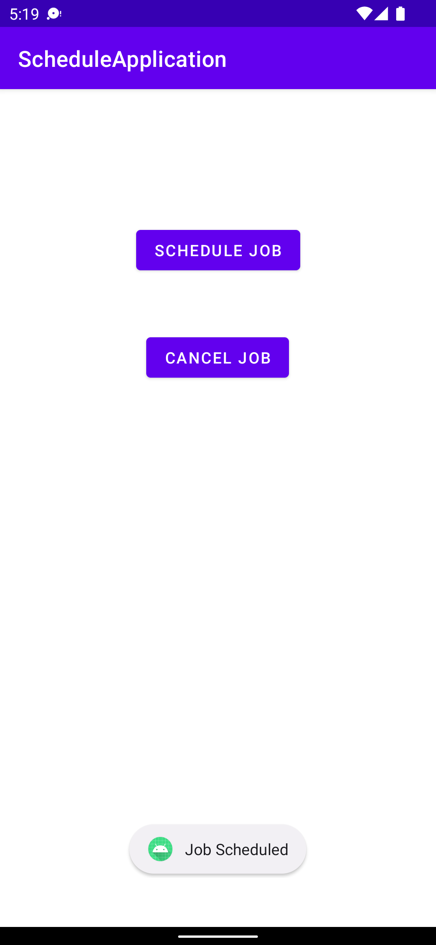
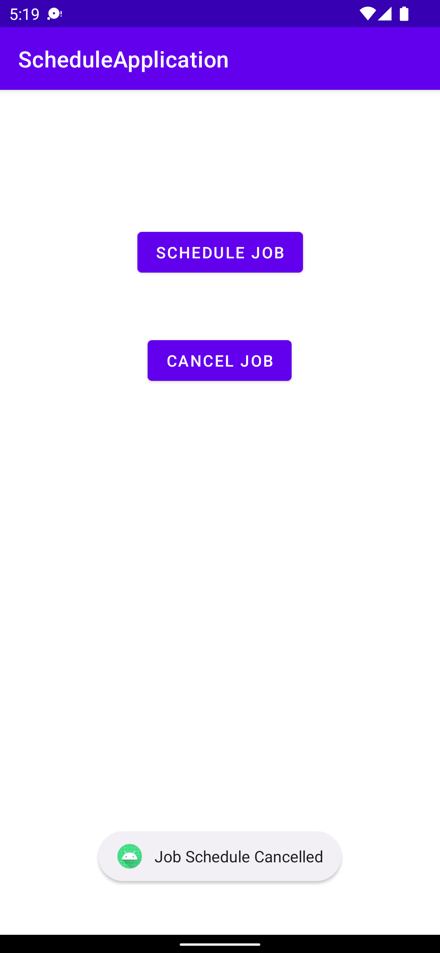
jobCancelled = true;

return true;

}

}

**Output**

** **

**Shared Preferences**

**Program Code**

*activity\_main.xml*

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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:orientation="vertical"

tools:context=".MainActivity">

<TextView

android:id="@+id/textview"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="30dp"

android:fontFamily="sans-serif-medium"

android:text="@string/title"

android:textColor="@android:color/black"

android:textSize="22sp"

android:layout\_marginStart="50sp"/>

<EditText

android:id="@+id/first"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="16dp"

android:layout\_marginTop="41dp"

android:layout\_marginEnd="16dp"

android:hint="@string/first\_name"

android:minHeight="48dp"

android:padding="10dp"

android:inputType="text" />

<EditText

android:id="@+id/last"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="16dp"

android:layout\_marginTop="29dp"

android:layout\_marginEnd="16dp"

android:hint="@string/last\_name"

android:inputType="text"

android:minHeight="48dp"

android:padding="10dp" />

</LinearLayout>

*MainActivity.java*

package com.example.sharedpreference;

import androidx.appcompat.app.AppCompatActivity;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

private EditText first\_name, last\_name;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

first\_name = findViewById(R.id.first);

last\_name = findViewById(R.id.last);

}

@Override

protected void onResume() {

super.onResume();

SharedPreferences sh = getSharedPreferences("MySharedPref", MODE\_PRIVATE);

String fn = sh.getString("first\_name", "");

String ln = sh.getString("last\_name", "");

first\_name.setText(fn);

last\_name.setText(ln);

}

@Override

protected void onPause() {

super.onPause();

SharedPreferences sharedPreferences = getSharedPreferences("MySharedPref", MODE\_PRIVATE);

SharedPreferences.Editor myEdit = sharedPreferences.edit();

myEdit.putString("first\_name", first\_name.getText().toString());

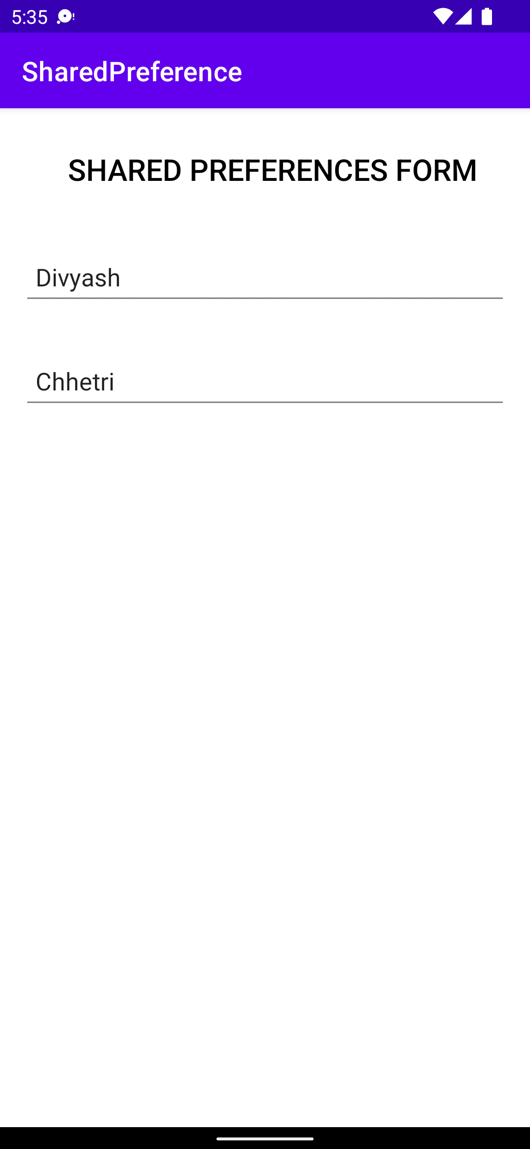
myEdit.putString("last\_name", last\_name.getText().toString());

myEdit.apply();

}

}

**Output**

**

**SQLite**

**Program Code**

*activity\_main.xml*

<?xml version="1.0" encoding="utf-8"?>

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

xmlns:app="http://schemas.android.com/apk/res-auto"

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity"

android:orientation="vertical">

<TextView

android:id="@+id/title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="sans-serif-medium"

android:text="@string/registration\_form"

android:textColor="@android:color/black"

android:textSize="22sp"

android:layout\_marginTop="20sp"

android:layout\_marginStart="100sp"/>

<EditText

android:id="@+id/first\_name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/enter\_first\_name"

android:layout\_marginTop="40sp"

android:layout\_marginStart="25sp"

android:layout\_marginEnd="25sp"

android:inputType="text"/>

<EditText

android:id="@+id/last\_name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/enter\_last\_name"

android:layout\_marginTop="15sp"

android:layout\_marginStart="25sp"

android:layout\_marginEnd="25sp"

android:inputType="text"/>

<EditText

android:id="@+id/phone\_no"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/enter\_phone\_no"

android:layout\_marginTop="15sp"

android:layout\_marginStart="25sp"

android:layout\_marginEnd="25sp"

android:inputType="number"/>

<Button

android:id="@+id/button"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/submit"

android:layout\_marginTop="80sp"

android:layout\_marginStart="150sp"

android:layout\_marginEnd="150sp" />

</LinearLayout>

*DatabaseCon.java*

package com.example.sqlite;

import android.content.ContentValues;

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.view.LayoutInflater;

public class DatabaseCon extends SQLiteOpenHelper {

private static final String DB\_NAME = "registration\_details";

private static final int DB\_VERSION = 1;

private static final String TABLE\_NAME = "users";

private static final String ID\_COL = "id";

private static final String FIRST\_NAME\_COL = "fname";

private static final String LAST\_NAME\_COL = "lname";

private static final String PHONE\_NO\_COL = "phone";

public DatabaseCon(Context context) {

super(context, DB\_NAME, null, DB\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

String query = "CREATE TABLE " + TABLE\_NAME + " ("

+ ID\_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "

+ FIRST\_NAME\_COL + " TEXT,"

+ LAST\_NAME\_COL + " TEXT,"

+ PHONE\_NO\_COL + " TEXT)";

db.execSQL(query);

}

public void addNewUser(String fname, String lname, String phone) {

SQLiteDatabase db = this.getWritableDatabase();

ContentValues values = new ContentValues();

values.put(FIRST\_NAME\_COL, fname);

values.put(LAST\_NAME\_COL, lname);

values.put(PHONE\_NO\_COL, phone);

db.insert(TABLE\_NAME, null, values);

db.close();

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NAME);

onCreate(db);

}

}

*MainActivity.java*

package com.example.sqlite;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View; import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

private EditText fn, ln, ph;

private Button btn;

private DatabaseCon db;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

fn = findViewById(R.id.first\_name);

ln = findViewById(R.id.last\_name);

ph = findViewById(R.id.phone\_no);

btn = findViewById(R.id.button);

db = new DatabaseCon(MainActivity.this);

btn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String fname = fn.getText().toString();

String lname = ln.getText().toString();

String phone = ph.getText().toString();

if (fname.isEmpty() && lname.isEmpty() && phone.isEmpty()) {

Toast.makeText(MainActivity.this, "Please fill up all fields", Toast.LENGTH\_SHORT).show();

return;

}

db.addNewUser(fname, lname, phone);

Toast.makeText(MainActivity.this, "Form has been submitted", Toast.LENGTH\_SHORT).show();

fn.setText("");

ln.setText("");

ph.setText("");

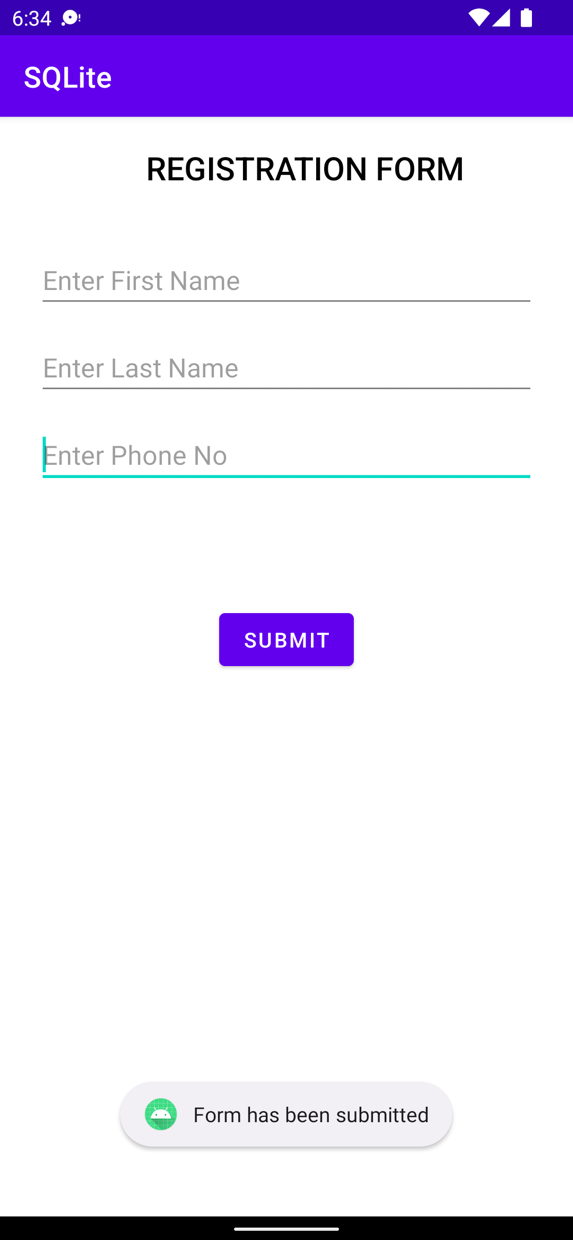
}

});

}

}

**Output**

**

**Conclusion**

I learned about the various services, including notification, alarms, and scheduling, all of which are very helpful to users. An app's service is a part that enables the app to continue in the background and carry out ongoing duties. A service's primary objective is to maintain an application in the background so that the user may use many applications at once. A user interface is inappropriate for android services since they are designed to function continuously without user input. A service can go running in the background even if the program is closed or the user switches to another application. It is possible to build storage services like shared preferences and SQLite Storage. As a result, the user can store information in the database that can subsequently be viewed.

# References

Android. (n.d.). *Android Studio Docs*. Retrieved 2022, from Developers Android: https://developer.android.com/docs

Aggarwal, P. (2022, July 15). *Shared Preferences in Android with Example*. Retrieved Sept 19, 2022, from Geeks for Geeks: https://www.geeksforgeeks.org/shared-preferences-in-android-with-examples/

Chaitanyamunje. (2022, Aug 18). *How to Create and Add Data to SQLite Database in Android?* Retrieved Sept 19, 2022, from Geeks for Geeks: https://www.geeksforgeeks.org/how-to-create-and-add-data-to-sqlite-database-in-android/