ROYAL UNIVERSITY OF BHUTAN
COLLEGE OF SCIENCE AND TECHNOLOGY
PHUENTSHOLING: BHUTAN

PLAGIARISM DECLARATION FORM

Student Name: Divyash Chhetri

Student No: 02200174

Module No and Title of the module: CTE306 - Mobile Application Development
Assignment no and Title of the Assignment: Lab Wok 09

Section H2 of the Royal University of Bhutan's *Wheel of Academic Law* provides the following definition of academic dishonesty:

"Academic dishonesty may be defined as any attempt by a student to gain an unfair advantage in any assessment. It may be demonstrated by one of the following:

Collusion: the representation of a piece of unauthorized group work as the work of a single candidate.

Commissioning: submitting an assignment done by another person as the student's own work.

Duplication: the inclusion in coursework of material identical or substantially similar to material which has already been submitted for any other assessment within the University.

False declaration: making a false declaration in order to receive special consideration by an Examination Board or to obtain extensions to deadlines or exemption from work.

Falsification of data: presentation of data in laboratory reports, projects, etc., based on work purported to have been carried out by the student, which have been invented, altered or copied by the student.

Plagiarism: the unacknowledged use of another's work as if it were one's own.

Examples are:

- verbatim copying of another's work without acknowledgement
- paraphrasing of another's work by simply changing a few words or altering the order of presentation, without acknowledgement
- ideas or intellectual data in any form presented as one's own without acknowledging the source(s)
- making significant use of unattributed digital images such as graphs, tables, photographs, etc. taken from test books, articles, films, plays, handouts, internet, or any other source, whether published or unpublished
- submission of a piece of work which has previously been assessed for a different award or module or at a different institution as if it were new work
- use of any material without prior permission of copyright from appropriate authority or owner of the materials used"

Student Declaration

I confirm that I have read and understood the above definitions of academic dishonesty. I declare that I have not committed any academic dishonesty when completing the attached piece of work.

Signature of Student: Divyash Chhetri

Date: 20 Sept 2022

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
```

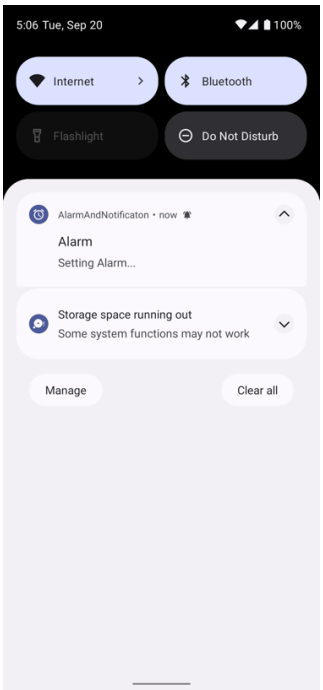
```
            android:id="@+id/notify"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/add_alarm"
            android:layout_marginTop="50sp"
            android:layout_marginStart="150sp"/>

    </LinearLayout>

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

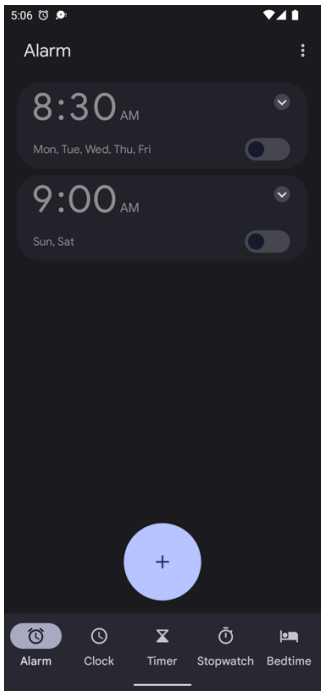
MainActivity.java

```
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();

        }
```

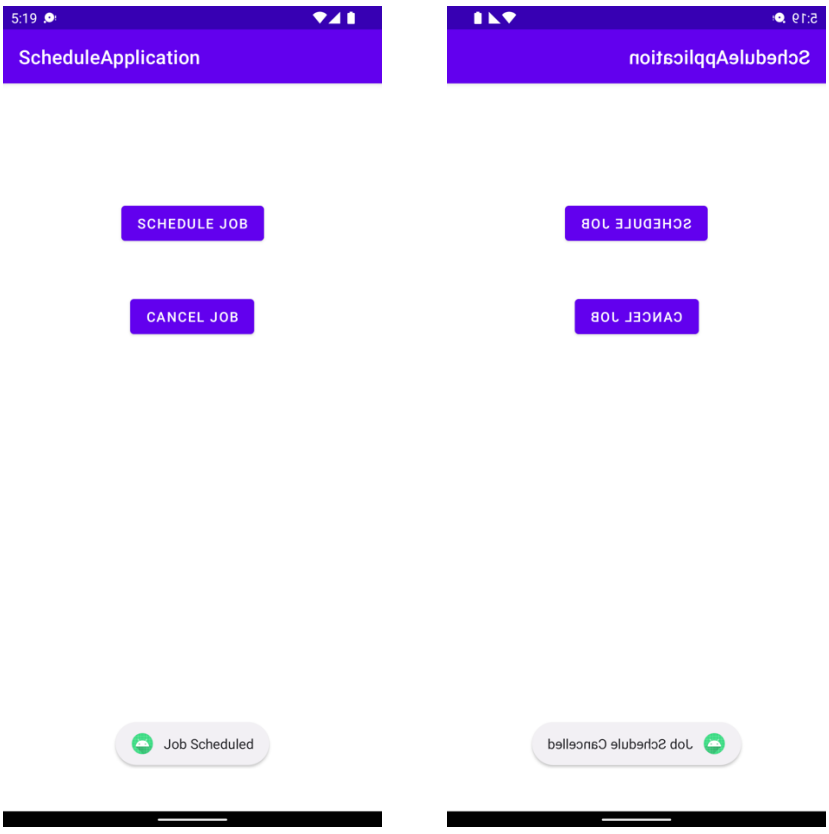
```
    }

    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

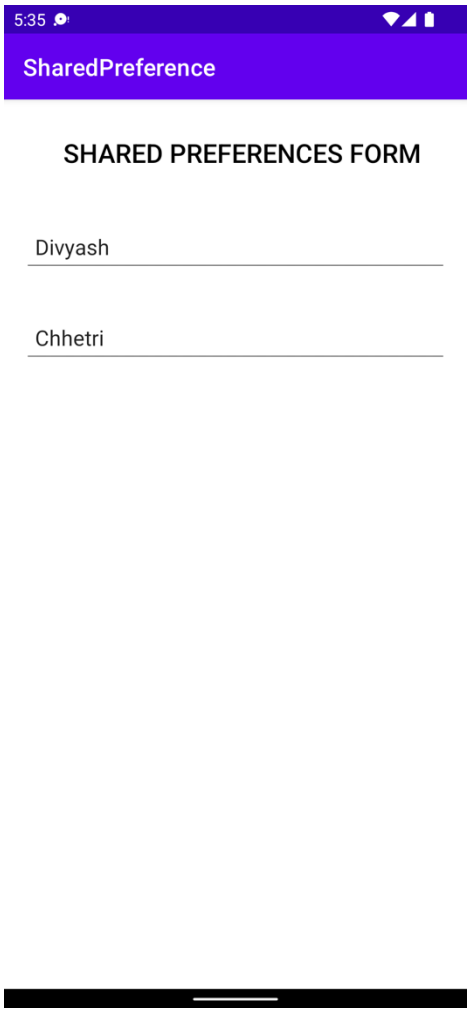
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <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


5:35
SharedPreference
SHARED PREFERENCES FORM
Divyash
Chhetri

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) {
```

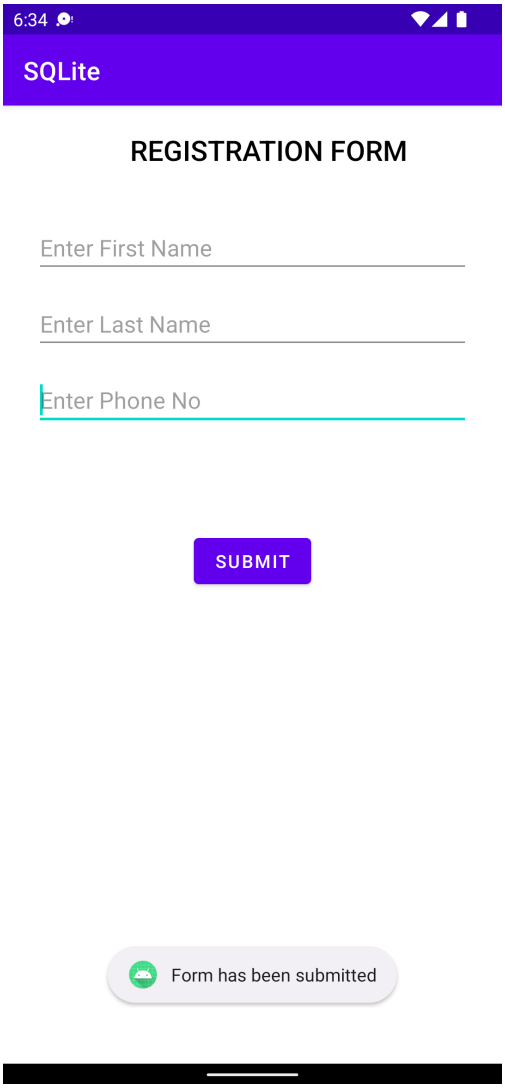
```
        String query = "CREATE TABLE " + TABLE_NAME + " ("
                + ID_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "
                + FIRST_NAME_COL + " TEXT,"
                + LAST_NAME_COL + " TEXT,"
                + PHONE_NO_COL + " TEXT)";

        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


6:34
SQLite
REGISTRATION FORM
Enter First Name
Enter Last Name
Enter Phone No
SUBMIT
Form has been submitted

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/