

RAJALAKSHMI ENGINEERING COLLEGE
RAJALAKSHMI NAGAR, THANDALAM – 602 105



**RAJALAKSHMI
ENGINEERING COLLEGE**

CS19611
Mobile Application Development Laboratory

Laboratory Record Note Book

Name : K.S.JAYASURIYAA

Year / Branch / Section : CSE-D

Register No. 220701332

Semester VI

Academic Year : 2024-2025

RAJALAKSHMI ENGINEERING COLLEGE
RAJALAKSHMI NAGAR, THANDALAM – 602 105

BONAFIDE CERTIFICATE

Name : K.S.JAYASURIYAA

Academic Year : 2025

Semester : VI

Branch : Computer Science And Engineering

Register No.

2116220701332

Certified that this is the bonafide record of work done by the above student in the MOBILE APPLICATION DEVELOPMENT LABORATORY Laboratory during the year 2024 - 2025.

Signature of Faculty in-charge

Submitted for the Practical Examination held on

Internal Examiner

External Examiner

Reg. No. : 220701332

Name :. K.S.JAYASURIYAA

Year : III Branch: Computer Science and Engineering Sec: D

S. No.	Date	Title	Page No.	Teacher's Signature / Remarks
1	20/02/2025	GUI Components	7	
2	27/02/2025	Simple Calculator	12	
3	04/03/2025	Graphical Primitives	21	
4	11/03/2025	Android Fragments	27	
5	18/03/2025	SQLite	36	
6	18/03/2025	Form Validation	45	
7	25/02/2025	SD Card	53	
8	03/02/2025	Alert Dialog Box	58	
9	12/02/2025	Alarm	64	
10	12/02/2025	Telephony Services	70	
11	12/02/2025	Send SMS	76	
12	12/02/2025	Send Email	81	
13	14/02/2025	Text to Speech	86	
14	14/02/2025	Speech to Text	91	
15	14/02/2025	Image Capture	96	

Register No. : 220701508

Name : SHYAM S

GUI Components

Aim

Develop an application to change the font and color of the text and display toast message when the user presses the button.

Procedure

Step 1: Start

Step 2: Initialize the Android application using Kotlin

Step 3: Design the user interface

- Create a **TextView** to display text.
- Create a **Button** for user interaction.

Step 4: Load a custom font

- Place the **.ttf** font file into the **res/font** directory.
- Reference this font in the Kotlin code using **ResourcesCompat**.

Step 5: Link UI elements

- In **onCreate()**, link **TextView** and **Button** using **findViewById**.

Step 6: Set OnClickListener on the Button

- When the button is clicked:
 - a) Change the font of the **TextView** using the custom font.
 - b) Change the color of the **TextView** text using **setTextColor()**.
 - c) Show a **Toast** message indicating the changes were applied

Step 7: End

1€

activity_main.xml

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

    <TextView android:id="@+id/tvText"
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:text="Rajalakshmi Engineering College"
        android:textAlignment="center" android:textSize="16sp" />

    <Button android:id="@+id/btFontSize"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textCapSentences"
        android:text="Change Font Size"
        android:textSize="16sp" />

    <Button android:id="@+id/btFontColor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textCapSentences"
        android:text="Change Font Color"
        android:textSize="16sp" />

    <Button android:id="@+id/btBackgroundColor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textCapSentences"
        android:text="Change Background Color"
        android:textSize="16sp" />
</LinearLayout>
```


Output



Result



Step 5: Display the result

- Convert the result to a string.
- Show it in the TextView.

Step 6: Handle exceptions

- Validate inputs (e.g., non-empty, numeric).
- Handle divide-by-zero and invalid input formats.

Step 7: End


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

    <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:supportsRtl="true"
        android:theme="@style/Theme.SimpleCalculator" 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>
```



```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1" android:text="4" />

    <Button android:id="@+id/btFive"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1" android:text="5" />

    <Button android:id="@+id/btSix"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1" android:text="6" />

    <Button android:id="@+id/btMultiplication"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1" android:text="X" />
</LinearLayout>

<LinearLayout android:layout_width="match_parent"
    android:layout_height="50dp" android:orientation="horizontal">

    <Button android:id="@+id/btOne"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1" android:text="1" />

    <Button android:id="@+id/btTwo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1" android:text="2" />

    <Button android:id="@+id/btThree"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1" android:text="3" />

    <Button android:id="@+id/btSubtraction"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1" android:text="-" />
```



```

package org.rajalakshmi.simplecalculator
import androidx.appcompat.app.AppCompatActivity import
android.os.Bundle import android.widget.Button import
android.widget.TextView
class MainActivity : AppCompatActivity() { var input1
:Double = 0.0 var input2 :Double = 0.0 var addition : Boolean
= false var subtraction : Boolean = false var multiplication :
Boolean = false var division : Boolean = false var decimal :
Boolean = false
override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main)

    val tvExpression : TextView = findViewById(R.id.tvExpression) val tvResult : TextView =
findViewById(R.id.tvResult) val btZero : Button = findViewById(R.id.btZero) val btOne : Button
= findViewById(R.id.btOne) val btTwo : Button = findViewById(R.id.btTwo) val btThree : Button
= findViewById(R.id.btThree) val btFour : Button = findViewById(R.id.btFour) val btFive :
Button = findViewById(R.id.btFive) val btSix : Button = findViewById(R.id.btSix) val btSeven :
Button = findViewById(R.id.btSeven) val btEight : Button = findViewById(R.id.btEight) val
btNine : Button = findViewById(R.id.btNine) val btAddition : Button =
findViewById(R.id.btAddition) val btSubtraction : Button = findViewById(R.id.btSubtraction) val
btMultiplication : Button = findViewById(R.id.btMultiplication) val btDivision : Button =
findViewById(R.id.btDivision) val btDecimal : Button = findViewById(R.id.btDecimal) val
btEqual : Button = findViewById(R.id.btEqual) val btClear : Button = findViewById(R.id.btClear)
btZero.setOnClickListener { tvExpression.setText("${tvExpression.text}0") }
}
btOne.setOnClickListener { tvExpression.setText("${tvExpression.text}1") }
}
btTwo.setOnClickListener { tvExpression.setText("${tvExpression.text}2") }
}
btThree.setOnClickListener { tvExpression.setText("${tvExpression.text}3") }
}
btFour.setOnClickListener { tvExpression.setText("${tvExpression.text}4") }
}
btFive.setOnClickListener { tvExpression.setText("${tvExpression.text}5") }
}
btSix.setOnClickListener { tvExpression.setText("${tvExpression.text}6") }

```

```
}
```

```
btSeven.setOnClickListener {
```

```
    tvExpression.setText("S(tvExpression.text)7")
```



```
if (multiplication) { tvExpression.setText("${input1} * ${input2}") val
```

```
        rmultiplication : Double = input1 * input2
        tvResult.setText("${rmultiplication}") multiplication = false
    }
    if(division) { tvExpression.setText("${input1} / ${input2}") val
        rdivision : Double = input1 / input2
        tvResult.setText("${rdivision}") division = false
    }
}
btClear.setOnClickListener { tvExpression.setText("")  
    tvResult.setText("") input1 = 0.0 input2  
    = 0.0 decimal = false
}
}
```

Register No. : 220701508

Name : SHYAM S

Graphical Primitives

Aim

Develop an android application to draw the circle, ellipse, rectangle and some text using Android Graphical primitives.

Procedure

Step 1: Create a New Android Project

- Open Android Studio.
- Select "New Project" > "Empty Activity."
- Name the project (e.g., **DrawShapesApp**) and choose Kotlin/Java as the language.

Step 2: Create a Custom View for Drawing Shapes

- In the **src** folder, create a new Kotlin file: **CustomView.kt**.
- In this file, create a class that extends **View** to draw shapes using **Canvas**.

Step 3: Override **onDraw** Method

- Use the **Canvas** object to draw the circle, ellipse, rectangle, and text using methods like **drawCircle()**, **drawOval()**, **drawRect()**, and **drawText()**.

Step 6: Update the **MainActivity**

- Set the custom view as the content view in **MainActivity**.

Step 5: Run the App

- Build and run the app to see the shapes and text drawn on the screen

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">

    <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:supportsRtl="true"
        android:theme="@style/Theme.GraphicalPrimitives"
        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>
```


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">

    <org.rajalakshmi.graphicalprimitives.SampleCanvas android:layout_width="match_parent"
        android:layout_height="match_parent">
    </org.rajalakshmi.graphicalprimitives.SampleCanvas> </androidx.constraintlayout.widget.ConstraintLayout>
```


MainActivity.kt

```
package org.rajalakshmi.graphicalprimitives

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```


SampleCanvas.kt

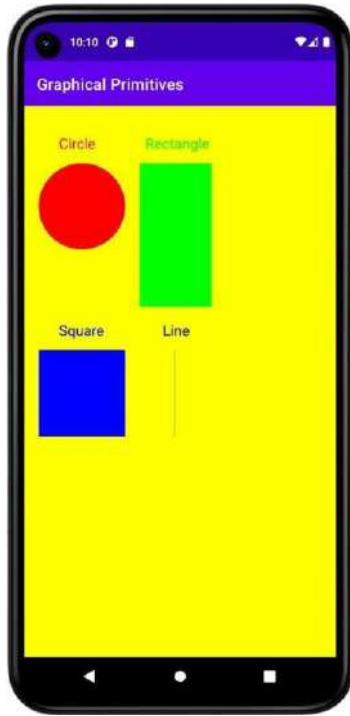
```
package org.rajalakshmi.graphicalprimitives
import android.content.Context import
android.graphics.Canvas import
android.graphics.Color import
android.graphics.Paint import
android.util.AttributeSet import
android.view.View
class SampleCanvas @JvmOverloads constructor( context: Context, attrs: AttributeSet? = null,
defStyleAttr: Int = 0 ) : View(context, attrs, defStyleAttr) {

    override fun onDraw(canvas: Canvas?) { super.onDraw(canvas)

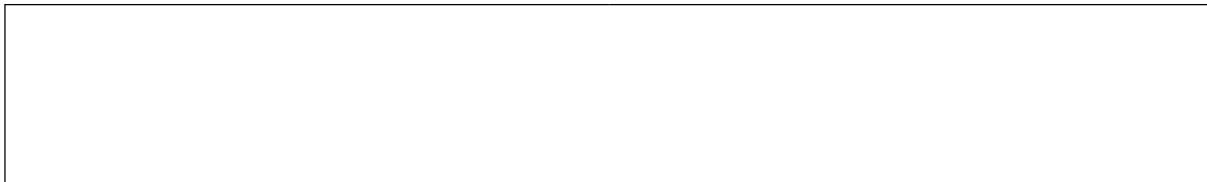
        val paint : Paint = Paint() paint.setColor(Color.YELLOW)
        canvas?.drawPaint(paint) paint.setTextSize(50f);
        paint.setColor(Color.RED);
        canvas?.drawText("Circle", 120f, 150f, paint); canvas?.drawCircle(200f, 350f, 150f, paint);
        paint.setColor(Color.GREEN);
        canvas?.drawText("Rectangle", 420f, 150f, paint); canvas?.drawRect(400f, 200f, 650f, 700f, paint);
        paint.setColor(Color.BLUE);
        canvas?.drawText("Square", 120f, 800f, paint); canvas?.drawRect(50f, 850f, 350f, 1150f, paint);
        paint.setColor(Color.BLACK);
        canvas?.drawText("Line", 480f, 800f, paint); canvas?.drawLine(520f, 850f, 520f, 1150f, paint);

    }
}
```


Output



Result



Register No. : 220701508

Name : SHYAM S

Android Fragments

Aim

Develop an android application to create two activities named as Student Basic Details (Register No., Name, Department) and Student Mark Details (SSLC, HSC, UG). Write an android code to combine these two activities in single screen using android fragment.

Procedure

Step 1: Create a New Android Project:

- Open Android Studio and create a new project.
- Select "Empty Activity" and name it (e.g., **StudentDetailsApp**).
- Choose Kotlin as the programming language.

Step 2: Create Two Fragments:

- One fragment for Student Basic Details.
- One fragment for Student Mark Details.

Step 3: Create a Layout for Each Fragment:

- The first fragment will contain fields for Register Number, Name, and Department.
- The second fragment will contain fields for SSLC, HSC, and UG marks.

Step 4: Combine the Fragments in the Main Activity:

- Use **FragmentTransaction** to display both fragments in a single activity

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">

    <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:supportsRtl="true"
        android:theme="@style/Theme.AndroidFragments"
        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>
```


activity_main.xml

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

    <TextView android:id="@+id/tvTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Student Details"
        android:textAlignment="center"
        android:textSize="24sp" />

    <LinearLayout android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <fragment android:id="@+id/fragmentBasic"
            android:name="org.rajalakshmi.androidfragments.StudentBasicDetails"
            android:layout_width="match_parent" android:layout_height="300dp" />

        <fragment android:id="@+id/fragmentMark"
            android:name="org.rajalakshmi.androidfragments.StudentMarkDetails"
            android:layout_width="match_parent" android:layout_height="300dp" />

    </LinearLayout>
</LinearLayout>
```


fragment_student_basic_details.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".StudentBasicDetails">
    <TextView
        android:id="@+id/tvBasicDetails"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Basic Details"
        android:textAlignment="center"
        android:textSize="24sp" />
    <TextView
        android:id="@+id/tvRegisterNumber"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:text="Register No." />
    <EditText
        android:id="@+id/etRegisterNumber"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="150dp"
        android:layout_marginTop="50dp"
        android:ems="10"
        android:hint="Register Number"
        android:inputType="textPersonName" />
    <TextView
        android:id="@+id/tvName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="125dp"
        android:text="Name" />
    <EditText
        android:id="@+id/etName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="150dp"
        android:layout_marginTop="125dp"
        android:ems="10"
        android:hint="Name"
        android:inputType="textPersonName" />
    <TextView
        android:id="@+id/tvDepartment"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="200dp"
        android:text="Department" />
    <EditText
        android:id="@+id/etDepartment"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="150dp"
        android:layout_marginTop="200dp"
        android:ems="10"
        android:hint="Department"
        android:inputType="textPersonName" />
</FrameLayout>
```


fragment_student_mark_details.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" tools:context=".StudentMarkDetails">
    <TextView android:id="@+id/tvBasicDetails"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Mark Details"
        android:textAlignment="center"
        android:textSize="24sp" />
    <TextView android:id="@+id/tvSSLC"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_marginTop="50dp" android:text="S.S.L.C." />
    <EditText android:id="@+id/etSSLC"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="150dp"
        android:layout_marginTop="50dp" android:ems="10"
        android:hint="S.S.L.C. Mark"
        android:inputType="textPersonName" />
    <TextView android:id="@+id/tvHSc"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_marginTop="125dp" android:text="H.Sc." />
    <EditText android:id="@+id/etHSC"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="150dp"
        android:layout_marginTop="125dp"
        android:ems="10" android:hint="H.Sc. Mark"
        android:inputType="textPersonName" />
    <TextView android:id="@+id/tvUG"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_marginTop="200dp"
        android:text="U.G." />
    <EditText android:id="@+id/etUG"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="150dp"
        android:layout_marginTop="200dp" android:ems="10"
        android:hint="U.G. C.G.P.A."
        android:inputType="textPersonName" />
</FrameLayout>
```

MainActivity.kt

```
package org.rajalakshmi.androidfragments

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```


StudentBasicDetails.kt

```
package org.rajalakshmi.androidfragments import android.os.Bundle
import androidx.fragment.app.Fragment import
android.view.LayoutInflater import android.view.View
import android.view.ViewGroup
// TODO: Rename parameter arguments, choose names that match // the fragment
initialization parameters, e.g. ARG_ITEM_NUMBER private const val ARG_PARAM1 =
"param1" private const val ARG_PARAM2 = "param2"

/**
 * A simple [Fragment] subclass.
 * Use the [StudentBasicDetails.newInstance] factory method to * create an instance of this fragment.
 */
class StudentBasicDetails : Fragment() {
    // TODO: Rename and change types of parameters private var
    param1: String? = null private var param2: String? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState) arguments?.let {
            param1 = it.getString(ARG_PARAM1) param2 =
            it.getString(ARG_PARAM2) }
    }

    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_student_basic_details,
        container, false)
    }

    companion object {
        /**
         * Use this factory method to create a new instance of * this fragment using the provided
         * parameters.
         *
         * @param param1 Parameter 1.
         * @param param2 Parameter 2.
         * @return A new instance of fragment StudentBasicDetails. */
        // TODO: Rename and change types and number of parameters
        @JvmStatic fun newInstance(param1: String, param2: String) =
            StudentBasicDetails().apply { arguments =
                Bundle().apply { putString(ARG_PARAM1,
                    param1) putString(ARG_PARAM2, param2) } }
    }
}
```


Ex. No. : 05

Date : 18/03/2025

Register No. :
220701508

Name : SHYAM S

SQLite

Aim

Create a Database table with the following structure using SQLite: Student (Register Number, Name, CGPA). Develop an android application to perform the following operation using SQLite developer classes. 1. Insert student Details 2. Update the student Record 3. Delete a specified record. 4. View the details.

Procedure

Step 1: Create a new Android project in Android Studio.

Step 2: Design the layout with input fields for Register Number, Name, and CGPA, along with buttons for Insert, Update, Delete, and View.

Step 3: Create a custom **SQLiteOpenHelper** class to define the **Student** table and override the necessary methods for database creation and upgrade.

Step 4: In the **MainActivity**, handle each button click to perform the corresponding database operation (insert, update, delete, or retrieve data).

Step 5: Display the retrieved student details in a **TextView** or using a **ListView/RecyclerView**.

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">

    <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:supportsRtl="true" android:theme="@style/Theme.SQLite"
        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>
```


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:text="View"
    android:textAllCaps="false" />

<Button android:id="@+id/btModify"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Modify" android:textAllCaps="false"
        />

<Button android:id="@+id/btDelete"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Delete" android:textAllCaps="false"
        />

<Button android:id="@+id/btClear"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Clear"
        android:textAllCaps="false" />
</LinearLayout>
```


DBContract.kt

```
package org.rajalakshmi.sqlite
```


UsersDBHelper.kt

```
package org.rajalakshmi.sqlite
import android.annotation.SuppressLint import
android.content.ContentValues import
android.content.Context import android.database.Cursor
import android.database.sqlite.SQLiteConstraintException import
android.database.sqlite.SQLiteDatabase import
android.database.sqlite.SQLiteException import
android.database.sqlite.SQLiteOpenHelper

class UsersDBHelper(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME, null,
DATABASE_VERSION) { override fun onCreate(db: SQLiteDatabase?) { db?.execSQL(SQL_CREATE_ENTRIES)
}

    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
db?.execSQL(SQL_DELETE_ENTRIES) onCreate(db)
}

    @Throws(SQLiteConstraintException::class) fun insertUser(user:
UserModel): Boolean {
    // Gets the data repository in write mode val db =
writableDatabase

    // Create a new map of values, where column names are the keys
    val values = ContentValues()
    values.put(DBContract.UserEntry.COLUMN_REGISTER_NUMBER,
user.registerNumber) values.put(DBContract.UserEntry.COLUMN_NAME, user.name)
    values.put(DBContract.UserEntry.COLUMN_CGPA, user.cgpa)

    // Insert the new row, returning the primary key value of the new row val newRowId =
db.insert(DBContract.UserEntry.TABLE_NAME, null, values)
return true
}

    @SuppressLint("Range")
fun readUser(registerNumber : String): ArrayList<UserModel> { val users =
ArrayList<UserModel>() val db = writableDatabase var cursor: Cursor? = null try {
cursor = db.rawQuery("select * from " +
DBContract.UserEntry.TABLE_NAME + " WHERE " +
DBContract.UserEntry.COLUMN_REGISTER_NUMBER + " = '" + registerNumber + "'", null) }
    catch (e: SQLiteException) {
        db.execSQL(SQL_CREATE_ENTRIES) return
        ArrayList()
    }
    var name : String var cgpa : String if
(cursor!!.moveToFirst()) {
```



```
        while (cursor.isAfterLast == false) {  
            name =  
            cursor.getString(cursor.getColumnIndex(DBContract.UserEntry.COLUMN_NAME))  
            cgpa =
```


MainActivity.kt

```
package org.rajalakshmi.sqlite
import android.database.sqlite.SQLiteConstraintException import
androidx.appcompat.app.AppCompatActivity import android.os.Bundle import
android.widget.Button import android.widget.EditText import
android.widget.Toast
class MainActivity : AppCompatActivity() { lateinit var usersDBHelper :
UsersDBHelper override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState) setContentView(R.layout.activity_main)

    val etRegisterNumber : EditText = findViewById(R.id.etRegisterNumber) val etName : EditText =
findViewById(R.id.etName) val etCGPA : EditText = findViewById(R.id.etCGPA) val btAdd :
Button = findViewById(R.id.btAdd) val btView : Button = findViewById(R.id.btView) val btModify :
Button = findViewById(R.id.btModify) val btDelete : Button = findViewById(R.id.btDelete) val
btClear : Button = findViewById(R.id.btClear) usersDBHelper = UsersDBHelper(this)
btAdd.setOnClickListener { val registerNumber : String = etRegisterNumber.text.toString() val name :
String = etName.text.toString() val cgpa : String = etCGPA.text.toString()
        var result = usersDBHelper.insertUser(UserModel(registerNumber =
registerNumber, name = name, cgpa = cgpa))
        etRegisterNumber.setText("") etName.setText("")
        etCGPA.setText("") }
}
btView.setOnClickListener { var users = usersDBHelper.readUser(etRegisterNumber.text.toString()) users.forEach {
etName.setText(it.name) etCGPA.setText(it.cgpa)
}
}
btDelete.setOnClickListener { var registerNumber = etRegisterNumber.text.toString() val result =
usersDBHelper.deleteUser(registerNumber) if(result)
        Toast.makeText(applicationContext, "User Deleted...!", Toast.LENGTH_LONG).show()
}
}
btClear.setOnClickListener { etRegisterNumber.setText("") etName.setText("") etCGPA.setText("") }
}
}
```

Output



Register No. : 220701508

Name : SHYAM S

Form Validation

Aim

Design an android activity with two text boxes where the user can enter (username and ID) and a button (validate). Validate the entered username and ID field for the following using android code. i) Both the fields should not be empty ii) Name field should have alphabets iii) ID field should have numeric values (only 4-digit).

Procedure

Step 1: Create a New Android Project in Android Studio.

Step 2: Design the Layout with two **EditText** fields (for username and ID) and a **Button** (for validation) in the **activity_main.xml** file.

Step 3: Implement the Validation Logic in **MainActivity.kt** to check if both fields are not empty, validate that the username contains only alphabets, and validate that the ID contains only numeric values and is exactly 4 digits.

Step 4: Run the Application: If the validation fails, show a **Toast** message with an error, and if it succeeds, show a success message.


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

    <EditText android:id="@+id/etUserName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter the userame...!" android:inputType="textPersonName" />

    <EditText android:id="@+id/etPinNumber"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter the pin number...!" android:inputType="textPersonName" />

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

    <Button android:id="@+id/btClear"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Clear"
        android:textAllCaps="false" />
</LinearLayout>
```


activity_main2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity2">

    <TextView
        android:id="@+id/tvLoginSuccess"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Login Success...!"
        android:textAlignment="center"
        android:textSize="24sp" />
</LinearLayout>
```


MainActivity.kt

```
package org.rajalakshmi.formvalidation
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity import
android.os.Bundle import android.widget.Button import
android.widget.EditText import android.widget.Toast
class MainActivity : AppCompatActivity() { override fun
onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main)

    val etUserName : EditText = findViewById(R.id.etUserName) val etPinNumber :
EditText = findViewById(R.id.etPinNumber) val btLogin : Button =
findViewById(R.id.btLogin) val btClear : Button = findViewById(R.id.btClear)
btLogin.setOnClickListener { val checkUserName = "[a-zA-Z]+".toRegex()
        val checkPinNo = "[0-9]{4}+".toRegex()
        if(checkUserName.matches(etUserName.text.toString()) &&
checkPinNo.matches(etPinNumber.text.toString())) { val intent = Intent(this,
            MainActivity2::class.java) startActivity(intent)
        } else {
            Toast.makeText(applicationContext, "Invalid User Name / Pin No.",Toast.LENGTH_LONG).show()
        }
    }

    btClear.setOnClickListener { etUserName.text.clear()
        etPinNumber.text.clear()
    }
}
}
```

College

MainActivity2.kt

```
package org.rajalakshmi.formvalidation

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle

class MainActivity2 : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main2)
    }
}
```

Output





Result



Ex. No. : 07

Date : 25/03/2025

Register No. : 220701508

Name : SHYAM S

SD Card

Aim

Implement an application to write the Register Number, Name and CGPA to SD card in text file format.

Procedure

Step 1: Create a New Android Project in Android Studio

Step 2: Add Necessary Permissions in the **AndroidManifest.xml** to read and write to external storage.

Step 3: Design the Layout with **EditText** fields for Register Number, Name, and CGPA, and a **Button** to trigger the save action.

Step 4: Write Code to Handle Button Click in **MainActivity.kt** to get values from the input fields and save them to a text file on the SD card.

Step 5: Check for Permissions before writing to the SD card, and request them if needed.

Step 6: Write the Data to a text file on the SD card using **FileWriter**.

Step 7: Test the Application to ensure it saves the data to a text file successfully.


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

    <EditText android:id="@+id/etRegisterNumber"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" android:ems="10"
        android:hint="Enter the register number...!" android:inputType="textPersonName" />

    <EditText android:id="@+id/etName"
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter the name...!" android:inputType="textPersonName" />

    <EditText android:id="@+id/etCGPA"
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter the CGPA...!" android:inputType="textPersonName" />

    <Button android:id="@+id/btSave"
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:text="Save"
        android:textAllCaps="false" />

    <Button android:id="@+id/btLoad"
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:text="Load"
        android:textAllCaps="false" />
</LinearLayout>
```

College

MainActivity.kt

```
package org.rajalakshmi.sdcard
```

Output



Ex. No. : 08

Date : 03/0/2025

Register No. : 220701508

Name : SHYAM S

Alert Dialog Box

Aim

Implement an application to display the alert box message.

Procedure

Step 1: Create a New Android Project in Android Studio.

Step 2: Design the Layout with a button to trigger the alert box.

Step 3: In the **MainActivity.kt**, set an **OnClickListener** for the button.

Step 4: Use **AlertDialog.Builder** to create and show the alert dialog with a message.

Step 5: Run the Application to ensure the alert box is displayed when the button is clicked.

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">

    <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:supportsRtl="true"
        android:theme="@style/Theme.AlertDialogBox" 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>
```

College

activity_main.xml

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

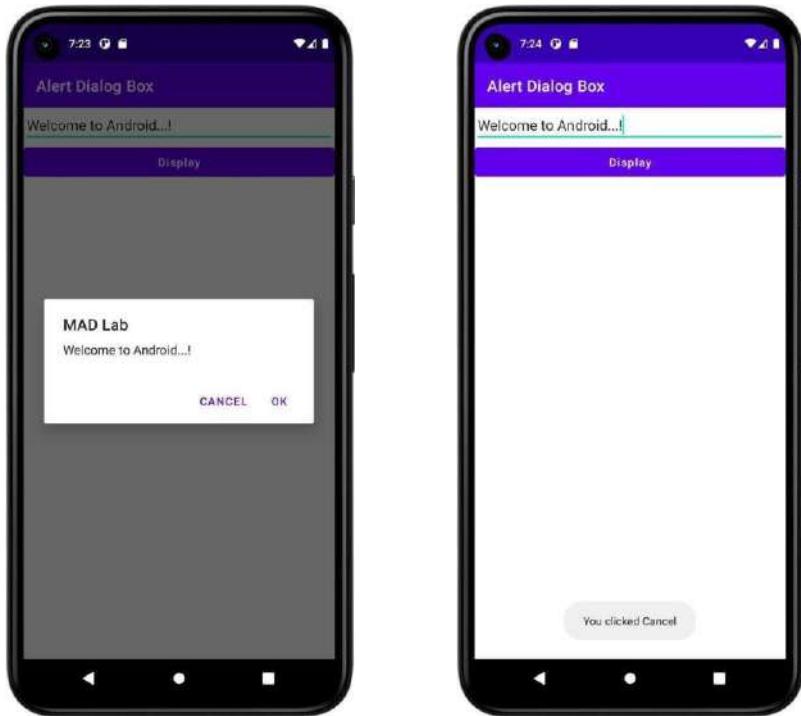
    <EditText
        android:id="@+id/etText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter the text...!"
        android:inputType="textPersonName" />

    <Button
        android:id="@+id/btDisplay"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Display"
        android:textAllCaps="false" />
</LinearLayout>
```


MainActivity.kt

```
package org.rajalakshmi.alertdialogbox
import androidx.appcompat.app.AppCompatActivity import
android.os.Bundle import android.widget.Button import
android.widget.EditText import android.widget.Toast
import androidx.appcompat.app.AlertDialog
class MainActivity : AppCompatActivity() { override fun
onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main)

    val etText : EditText = findViewById(R.id.etText) val btDisplay : Button =
    findViewById(R.id.btDisplay)
    btDisplay.setOnClickListener { val alertDialog = AlertDialog.Builder(this)
        .setTitle("MAD Lab")
        .setMessage(etText.text.toString())
        .setPositiveButton("OK") { dialog, which ->
            Toast.makeText(applicationContext, "You clicked OK",
            Toast.LENGTH_LONG).show()
        }
        .setNegativeButton("Cancel") { dialog, which ->
            Toast.makeText(applicationContext, "You clicked Cancel",
            Toast.LENGTH_LONG).show() }
        .create() alertDialog.show()
    }
}
}
```

Result

Register No. : 220701508

Name : SHYAM S

Alarm

Aim

Write a mobile application to set the alarm using android Alarm Manager class.

Procedure

Step1: Create a New Android Project in Android Studio

Step 2: Add Permissions in the **AndroidManifest.xml** to access the system alarm services.

Step 3: Design the Layout with **EditText** fields to enter the alarm time and a **Button** to set the alarm.

Step 4: Implement Alarm Logic in the **MainActivity.kt** using **AlarmManager** to set an alarm at a specific time.

Step 5: Handle Alarm Trigger: Create a **BroadcastReceiver** to handle when the alarm goes off.

Step 6: Test the Application to ensure the alarm is set and triggered correctly.


```
<?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.SCHEDULE_EXACT_ALARM"/>
    <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:supportsRtl="true"
        android:theme="@style/Theme.MyApplication" 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>
        <receiver android:name=".AlarmReceiver"> </receiver>
    </application>
</manifest>
```



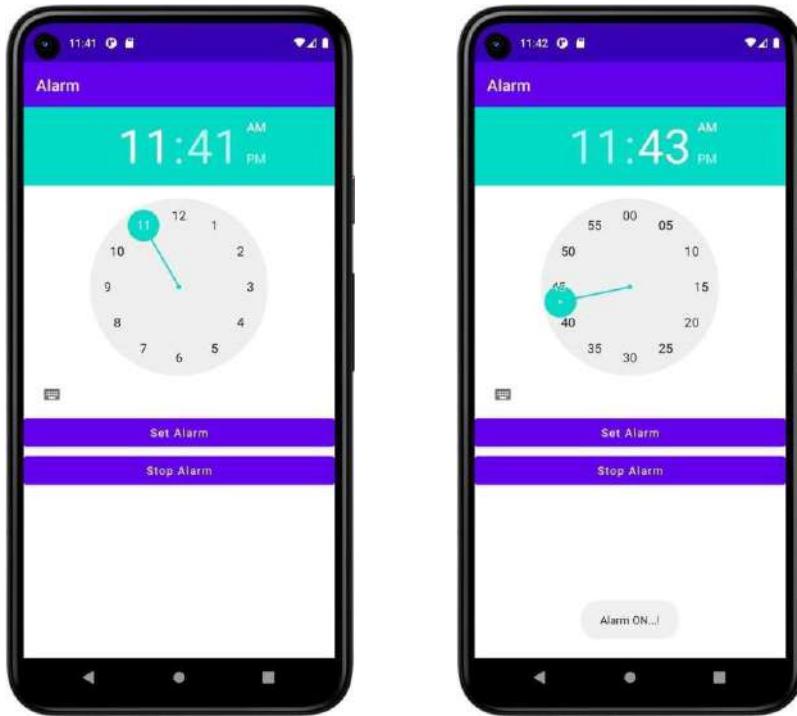
```
package org.rajalakshmi.myapplication

import android.app.AlarmManager import
android.app.PendingIntent import
android.content.Intent
import androidx.appcompat.app.AppCompatActivity import
android.os.Bundle import android.widget.Button import
android.widget.TimePicker import android.widget.Toast import
java.util.*

class MainActivity : AppCompatActivity() { lateinit var pendingIntent:
PendingIntent private lateinit var alarmManager: AlarmManager override
fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState) setContentView(R.layout.activity_main)

    val alarmTimePicker: TimePicker = findViewById(R.id.timePicker) val btSetAlarm : Button
    = findViewById(R.id.btSetAlarm) val btStopAlarm : Button =
    findViewById(R.id.btnStopAlarm) alarmManager = getSystemService(ALARM_SERVICE)
    as AlarmManager
    btSetAlarm.setOnClickListener {
        Toast.makeText(applicationContext, "Alarm ON...!", Toast.LENGTH_LONG).show() val calendar: Calendar = Calendar.getInstance()
        calendar.set(Calendar.HOUR_OF_DAY, alarmTimePicker.hour)
        calendar.set(Calendar.MINUTE, alarmTimePicker.minute) val intent = Intent(this,
        AlarmReceiver::class.java)
        intent.putExtra("alarmTime", alarmTimePicker.time)
        intent.putExtra("alarmType", "set")
        pendingIntent = PendingIntent.getBroadcast(this.applicationContext,
        2, intent, PendingIntent.FLAG_CANCEL_CURRENT) val time:Long = calendar.timeInMillis -
        (calendar.timeInMillis % 60000) alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, time, 10000,
        pendingIntent)
    }
    btStopAlarm.setOnClickListener {
        alarmManager.cancel(pendingIntent)
        Toast.makeText(applicationContext, "Alarm OFF...!", Toast.LENGTH_LONG).show()
    }
}
```


Output




```
<?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.READ_PHONE_STATE"/>
    <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:supportsRtl="true"
        android:theme="@style/Theme.TelephonyServices" 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>
```



```
<TextView android:id="@+id/tvDeviceSoftwareVersion"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" android:text="Device
    Software Version" />

<EditText android:id="@+id/etDeviceSoftwareVersion"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" android:ems="10"
    android:inputType="textPersonName" />

<Button android:id="@+id/btGetTelephonyServices"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" android:text="Get
    Telephony Services" android:textAllCaps="false" />
</LinearLayout>
```


MainActivity.kt

```
package org.rajalakshmi.telephonyservices

import android.content.Context
import android.content.pm.PackageManager
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.telephony.TelephonyManager
import android.widget.Button
import android.widget.EditText
import androidx.core.app.ActivityCompat

class MainActivity : AppCompatActivity() {
    private val REQUEST_CODE_PHONE_STATE = 1000
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val etNetworkOperatorName : EditText =
            findViewById(R.id.etNetworkOperatorName)
        val etPhoneType : EditText = findViewById(R.id.etPhoneType)
        val etNetworkCountryISO : EditText =
            findViewById(R.id.etNetworkCountryISO)
        val etSIMCountryISO : EditText = findViewById(R.id.etSIMCountryISO)
        val etDeviceSoftwareVersion : EditText =
            findViewById(R.id.etDeviceSoftwareVersion)
        val btGetTelephonyServices : Button =
            findViewById(R.id.btGetTelephonyServices)
        val telephonyManager = getSystemService(Context.TELEPHONY_SERVICE) as
            TelephonyManager

        if (ActivityCompat.checkSelfPermission(this,
            android.Manifest.permission.READ_PHONE_STATE) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this,
                arrayOf(android.Manifest.permission.READ_PHONE_STATE), REQUEST_CODE_PHONE_STATE)
        }
        btGetTelephonyServices.setOnClickListener {
            val networkOperatorName = telephonyManager.networkOperatorName
            val phoneType: Int = telephonyManager.getPhoneType()
            var strphoneType : String = ""
            val networkCountryISO: String =
                telephonyManager.getNetworkCountryIso()
            val SIMCountryISO: String = telephonyManager.getSimCountryIso()
            val deviceSoftwareVersion: String? =
                telephonyManager.getDeviceSoftwareVersion()

            when (phoneType) {
                TelephonyManager.PHONE_TYPE_CDMA -> strphoneType = "CDMA"
                TelephonyManager.PHONE_TYPE_GSM -> strphoneType = "GSM"
                TelephonyManager.PHONE_TYPE_NONE -> strphoneType = "NONE"
            }
            etNetworkOperatorName.setText(networkOperatorName)
            etPhoneType.setText(strphoneType)
            etNetworkCountryISO.setText(networkCountryISO)
            etSIMCountryISO.setText(SIMCountryISO)
            etDeviceSoftwareVersion.setText(deviceSoftwareVersion)
        }
    }
}
```


Ex. No. : 11

Date : 12/04/2025

Register No. :
220701508

Name : SHYAM S

Send SMS

Aim

Develop an application to send SMS.

Procedure

Step 1: Create a New Android Project in Android Studio.

Step 2: Add Permissions in the **AndroidManifest.xml** file to allow the app to send SMS.

Step 3: Design the Layout with **EditText** for the phone number and message, and a **Button** to trigger the SMS sending.

Step 4: Handle Button Click in the **MainActivity** to send the SMS.

Step 5: Test the Application to ensure SMS is sent correctly.

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.SEND_SMS"/>
    <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:supportsRtl="true" android:theme="@style/Theme.SendSMS"
        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>
```


activity_main.xml

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

    <EditText
        android:id="@+id/etPhoneNumber"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter the phone number...!"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/etMessage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter the message...!"
        android:inputType="textPersonName" />

    <Button
        android:id="@+id/btSend"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Send"
        android:textAllCaps="false" />
</LinearLayout>
```


MainActivity.kt

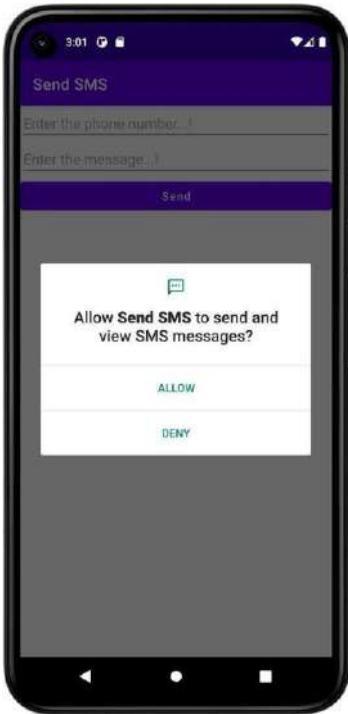
```
package org.rajalakshmi.sendsms
import android.os.Build
import androidx.appcompat.app.AppCompatActivity import android.os.Bundle
import android.telephony.SmsManager import
android.widget.Button import
android.widget.EditText import
android.widget.Toast
import androidx.core.app.ActivityCompat

class MainActivity : AppCompatActivity() { override fun
onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState) setContentView(R.layout.activity_main)

    val etPhoneNumber : EditText = findViewById(R.id.etPhoneNumber) val etMessage :
EditText = findViewById(R.id.etMessage) val btSend : Button = findViewById(R.id.btSend)
    ActivityCompat.requestPermissions(this,
arrayOf(android.Manifest.permission.SEND_SMS),1000)

    btSend.setOnClickListener { val phoneNumber =
etPhoneNumber.text.toString() val message =
etMessage.text.toString() val smsManager: SmsManager
    smsManager = SmsManager.getDefault()
    smsManager.sendTextMessage(phoneNumber, null, message, null, null)
    Toast.makeText(applicationContext, "Message Sent",
Toast.LENGTH_LONG).show()
}
}
}
```


Output



Result



Ex. No. : 12

Date : 12/04/2025

Register No. :
220701508

Name : SHYAM S

Send Email

Aim

Develop an application to send Email.

Procedure

Step 1: Create a new Android project in Android Studio.

Step 2: Design the layout with fields for recipient email, subject, and message, and add a "Send Email" button.

Step 3: Implement the email sending logic in the `MainActivity` using an implicit intent with action `Intent.ACTION_SENDTO`.

Step 4: Use a `mailto:` URI to ensure the email app is launched.

Step 5: Run the app and test by sending an email using the default email client on the device.

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">

    <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:supportsRtl="true"
        android:theme="@style/Theme.SendEmail"
        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>
```


activity_main.xml

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

    <TextView android:id="@+id/tvEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="To" />

    <EditText android:id="@+id/etEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName" />

    <TextView android:id="@+id/tvSubject"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Subject" />

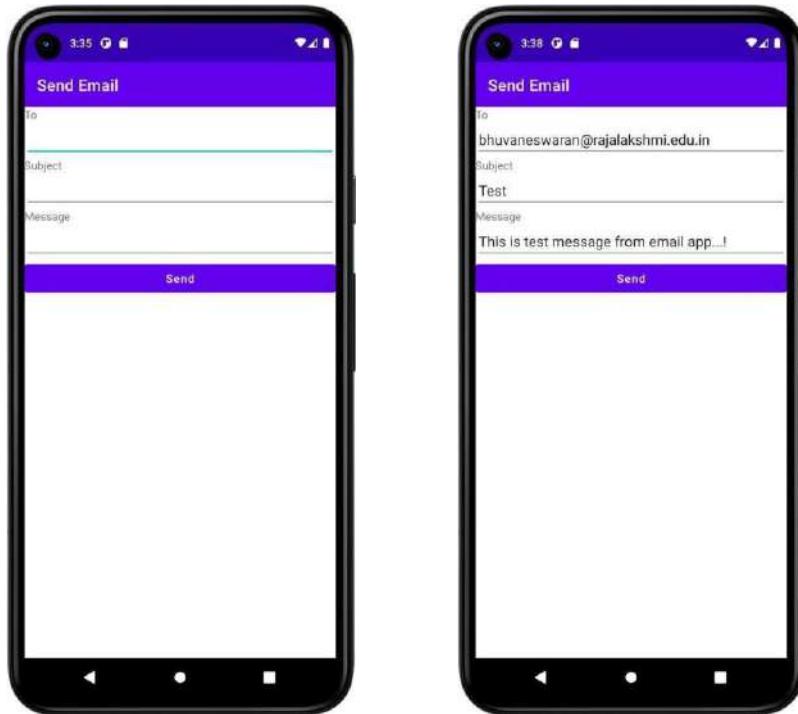
    <EditText android:id="@+id/etSubject"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName" />

    <TextView android:id="@+id/tvMessage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Message" />

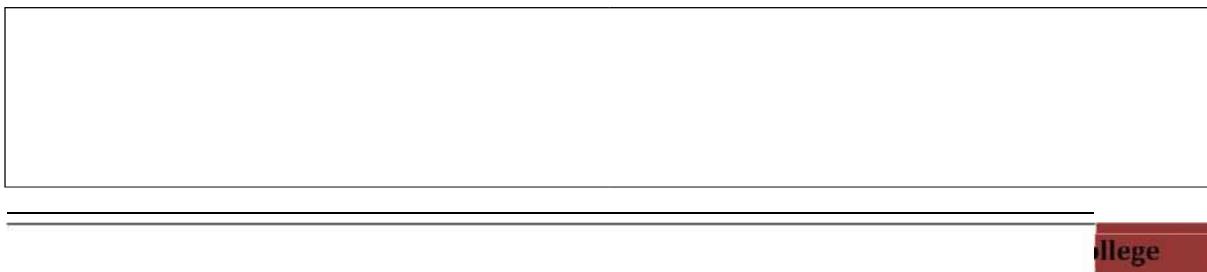
    <EditText android:id="@+id/etMessage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName" />

    <Button android:id="@+id/btSend"
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:text="Send"
        android:textAllCaps="false" />
</LinearLayout>
```


Output



Result



Ex. No. : 13

Date : 14/04/2025

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">

    <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:supportsRtl="true"
        android:theme="@style/Theme.TextToSpeech" 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>
```


activity_main.xml

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

    <EditText
        android:id="@+id/etText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter the text..!"
        android:inputType="textPersonName"
        android:textSize="24sp" />

    <Button
        android:id="@+id/btSpeak"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Speak"
        android:textAllCaps="false"
        android:textSize="24sp" />
</LinearLayout>
```


MainActivity.kt

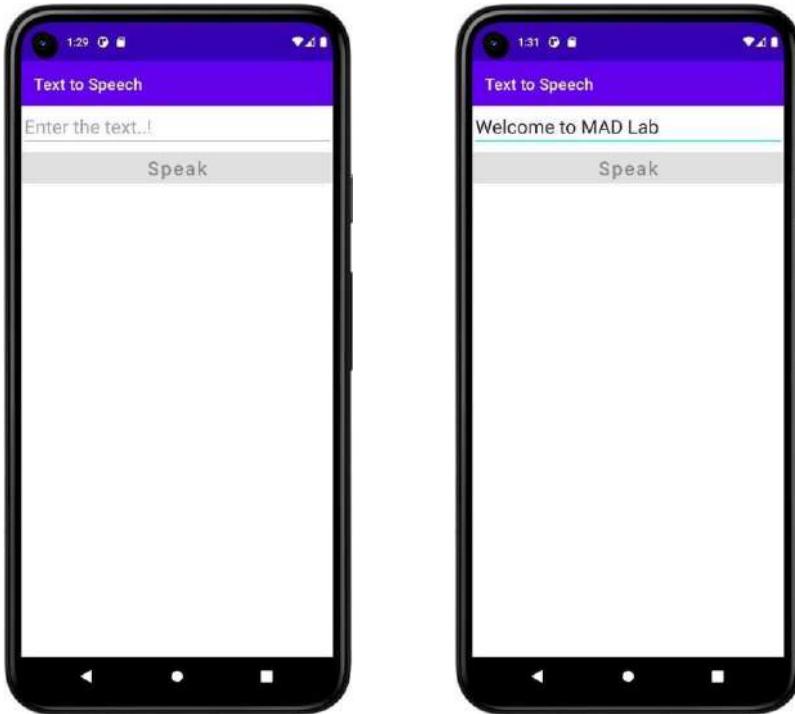
```
package org.rajalakshmi.texttospeech
import androidx.appcompat.app.AppCompatActivity import
android.os.Bundle
import android.speech.tts.TextToSpeech import
android.widget.Button import android.widget.EditText
import android.widget.Toast import java.util.*
class MainActivity : AppCompatActivity(), TextToSpeech.OnInitListener { lateinit var tts : TextToSpeech
lateinit var btSpeak : Button
    override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val etText : EditText = findViewById(R.id.etText) btSpeak =
            findViewById(R.id.btSpeak) btSpeak.isEnabled = false tts =
            TextToSpeech(this, this)

        btSpeak.setOnClickListener { val text =
            etText!!.text.toString() tts!!.speak(text, TextToSpeech.QUEUE_FLUSH, null,"")
        }
    }

    override fun OnInit(status: Int) { if (status == TextToSpeech.SUCCESS)
        { val result = tts!!.setLanguage(Locale.US)

            if (result == TextToSpeech.LANG_MISSING_DATA || result ==
                TextToSpeech.LANG_NOT_SUPPORTED) {
                Toast.makeText(applicationContext, "The Language not
supported...",Toast.LENGTH_LONG).show()
            } else { btSpeak!!.isEnabled = true
            }
        }
    }
}
```

Output



Result



Register No. : 220701508

Name : SHYAM S

Speech to Text

Aim

Develop an android application to perform Speech to Text.

Procedure

Step 1: Create a new Android project in Android Studio.

Step 2: Design the layout with a **Button** to start voice input and a **TextView** to display the recognized text.

Step 3: Use **Intent.ACTION_RECOGNIZE_SPEECH** to launch the speech recognizer.

Step 4: Capture the result in **onActivityResult** and display the recognized speech in the **TextView**.

Step 5: Handle microphone permission for devices running Android 6.0 and above.

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">
    <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:supportsRtl="true"
        android:theme="@style/Theme.SpeechToText"
        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>
```


activity_main.xml

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

    <ImageView android:id="@+id/imgMic"
        android:layout_width="match_parent" android:layout_height="250dp"
        app:srcCompat="@android:drawable/ic_btn_speak_now" />

    <TextView android:id="@+id/tvText"
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:text="Output appears here...!" android:textSize="24sp" />
</LinearLayout>
```

ollege

MainActivity.kt

```
package org.rajalakshmi.speechtotext

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.speech.RecognizerIntent
import android.widget.ImageView
import android.widget.TextView
import java.util.*

class MainActivity : AppCompatActivity() {
    lateinit var tvText : TextView
    private val REQUEST_CODE_SPEECH_INPUT = 1000
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        var imgMic : ImageView = findViewById(R.id.imgMic)
        tvText = findViewById(R.id.tvText)
        imgMic.setOnClickListener {
            val intent = Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH)
            intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
                RecognizerIntent.LANGUAGE_MODEL_FREE_FORM)
            intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE,
                Locale.getDefault())
            intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Speak...!")
            startActivityForResult(intent, REQUEST_CODE_SPEECH_INPUT)
        }
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        super.onActivityResult(requestCode, resultCode, data)
        if(requestCode == REQUEST_CODE_SPEECH_INPUT && resultCode == RESULT_OK
        && data != null) {
            var res : ArrayList<String> =
            data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS) as
            ArrayList<String>
            tvText.setText( Objects.requireNonNull(res)[0] )
        }
    }
}
```

Register No. : 220701508

Name : SHYAM S

Image Capture

Aim

Develop an android application to capture image using camera and displaying the image using ImageView.

Procedure

Step 1: Create a new Android project in Android Studio.

Step 2: Add the camera permission to the **AndroidManifest.xml** file.

Step 3: Design the layout with a **Button** to open the camera and an **ImageView** to display the image.

Step 4: Use an **Intent** with **MediaStore.ACTION_IMAGE_CAPTURE** to launch the camera.

Step 5 : Handle the captured image in **onActivityResult** and display it in the **ImageView**.

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">

    <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:supportsRtl="true"
        android:theme="@style/Theme.ImageCapture" 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>
```

ollege

MainActivity.kt

```
package org.rajalakshmi.imagecapture
import android.content.Intent import
android.graphics.Bitmap
import androidx.appcompat.app.AppCompatActivity import
android.os.Bundle import android.provider.MediaStore import
android.widget.Button import android.widget.ImageView
class MainActivity : AppCompatActivity() { lateinit var
imgImage : ImageView
    private val REQUEST_CODE_IMAGE_CAPTURE = 1000 override fun
onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main)

    imgImage = findViewById(R.id.imgImage)
    val btTakePicture : Button = findViewById(R.id.btTakePicture)
    btTakePicture.setOnClickListener { val intent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
        startActivityForResult(intent, REQUEST_CODE_IMAGE_CAPTURE)
    }
}

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data) if(requestCode == REQUEST_CODE_IMAGE_CAPTURE
&& resultCode == RESULT_OK) {
        val photo = data!![ "data" ] as Bitmap?
        imgImage.setImageBitmap(photo)
    }
}
}
```

Output