SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

FACULTY OF SCIENCE AND HUMANITIES

DEPARTMENT OF COMPUTER APPLICATIONS

PRACTICAL RECORD NOTE

STUDENT NAME :

REGISTER NUMBER :

CLASS : III BCA SECTION:

YEAR & SEMESTER : III YEAR & V SEMESTER

SUBJECT CODE : UCA23G03J

SUBJECT TITLE : BASICS OF ANDROID

OCTOBER 2025

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
FACULTY OF SCIENCE AND HUMANITIES
DEPARTMENT OF COMPUTER APPLICATIONS

SRM Nagar, Kattankulathur – 603 203

CERTIFICATE

Certified to be the bonafide record of practical work done by_____

Register No._____of_____Degree course for UCA23G03J - BASICS OF ANDROID in the computer lab in SRM Institute of Science and Technology during the academic year 2025 – 2026

Staff In-charge Head of the Department

Submitted for Semester Practical Examination held on _____

Internal Examiner External Examiner

INDEX

Source Code:

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:gravity="center"
   android:background="#FFFFFF">

<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Your Name"
android:textSize="40sp"
android:textColor="#000000"
android:layout_gravity="center" />
</LinearLayout>
```

Mainactivity.java

```
package com.example.name1;

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

public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
   }
}
```

Output:

RESULT:

Thus, the Android Application to display a name using TextView is developed and executed successfully.

Source Code:

Activitymain.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:gravity="center"
   android:background="#FFFFFF">

<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Your Name"
android:textSize="40sp"
android:textColor="#000000"
android:layout_gravity="center" />
</LinearLayout>
```

Mainactitvity.java

```
package com.example.name1;
import android.content.pm.ActivityInfo;

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

public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
      setContentView(R.layout.activity_main);     }
}
```

Output:

RESULT:

Thus, the Android Application to change the screen orientation is developed and executed successfully.

Source Code:

Activitymain.XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
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:fitsSystemWindows="true"
   tools:context=".MainActivity">

   <androidx.constraintlayout.widget.ConstraintLayout
      android:id="@+id/constraintLayout"
      android:layout_width="match_parent"
      android:layout_height="match_parent">


      <ToggleButton
         android:id="@+id/toggleButton"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_marginTop="420dp"
         android:onClick="onToggleClick"
         android:text="ToggleButton"
         app:layout_constraintEnd_toEndOf="parent"
         app:layout_constraintHorizontal_bias="0.458"
         app:layout_constraintLeft_toLeftOf="@id/textView"
         app:layout_constraintStart_toStartOf="parent"
         app:layout_constraintTop_toTopOf="parent" />

      <TextView
         android:id="@+id/textView"
         android:layout_width="232dp"
```

```
        android:layout_height="49dp"
        android:layout_marginEnd="88dp"
        android:gravity="center"
        android:text="TextView"
        app:layout_anchor="@+id/constraintLayout"
        app:layout_anchorGravity="center"
        app:layout_constraintBottom_toTopOf="@id/toggleButton"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.574" />
    </androidx.constraintlayout.widget.ConstraintLayout>
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Mainactitvity.java

```
package com.example.toggle1;

import android.os.Bundle;

import android.view.View;

import android.widget.*;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);

// Initialize

ToggleButton togglebutton = findViewById(R.id.toggleButton);

TextView textview = findViewById(R.id.textView);

togglebutton.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {
```

```
if (togglebutton.isChecked()) {
textview.setText("Toggle is ON");
} else {
textview.setText("Toggle is OFF");
}}
});}}
```

Output:

RESULT:

Thus, the Android Application to display a toggle button and change text based on ON/OFF state is developed and executed successfully.

Source Code:

activitymai.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:id="@+id/idRLContainer"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:gravity="center"
tools:context=".MainActivity">
<!--headline text-->
<TextView
android:id="@+id/idTVHeading"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Radio Button in Android"
android:textColor="@color/black"
android:textSize="20sp"
android:textStyle="bold" />
<!--status text-->
<TextView
```

```
android:id="@+id/idTVStatus"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin="32dp"
android:text="Status"
android:textColor="@color/black"
android:textSize="20sp"
android:textStyle="bold" />
<!--radio group-->
<RadioGroup
android:id="@+id/idRadioGroup"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:gravity="center">
<!--radio buttons-->
<RadioButton
android:id="@+id/idBtnJavaRadio"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Java"
android:textSize="20sp" />
<RadioButton
android:id="@+id/idBtnKotlinRadio"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Kotlin"
```

```
android:textSize="20sp" />
</RadioGroup>
</LinearLayout>
```

Mainactitvity.java

```
package com.example.radio1;
import android.os.Bundle;
import android.widget.*;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
RadioGroup radioGroup = findViewById(R.id.rg);
TextView textView = findViewById(R.id.tv1);
// Set listener on RadioGroup
radioGroup.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
@Override
public void onCheckedChanged(RadioGroup group, int checkedId) {
RadioButton radioButton = group.findViewById(checkedId);
if (radioButton != null) {
textView.setText(radioButton.getText());
} } }); }}
```

Output:

RESULT:

Thus, the Android Application to create a radio group and display the selected option is developed and executed successfully.

Source Code:

Activitymain.XML

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/white"
tools:context=".MainActivity">
<TextView
android:id="@+id/textView"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginBottom="32dp"
android:padding="10dp"
android:text="Progress Bar in Android"
android:textAlignment="center"
android:textColor="@color/black"
android:textSize="20sp"
android:textStyle="bold"
app:layout_constraintBottom_toTopOf="@+id/progressBar"
app:layout_constraintEnd_toEndOf="parent"
```

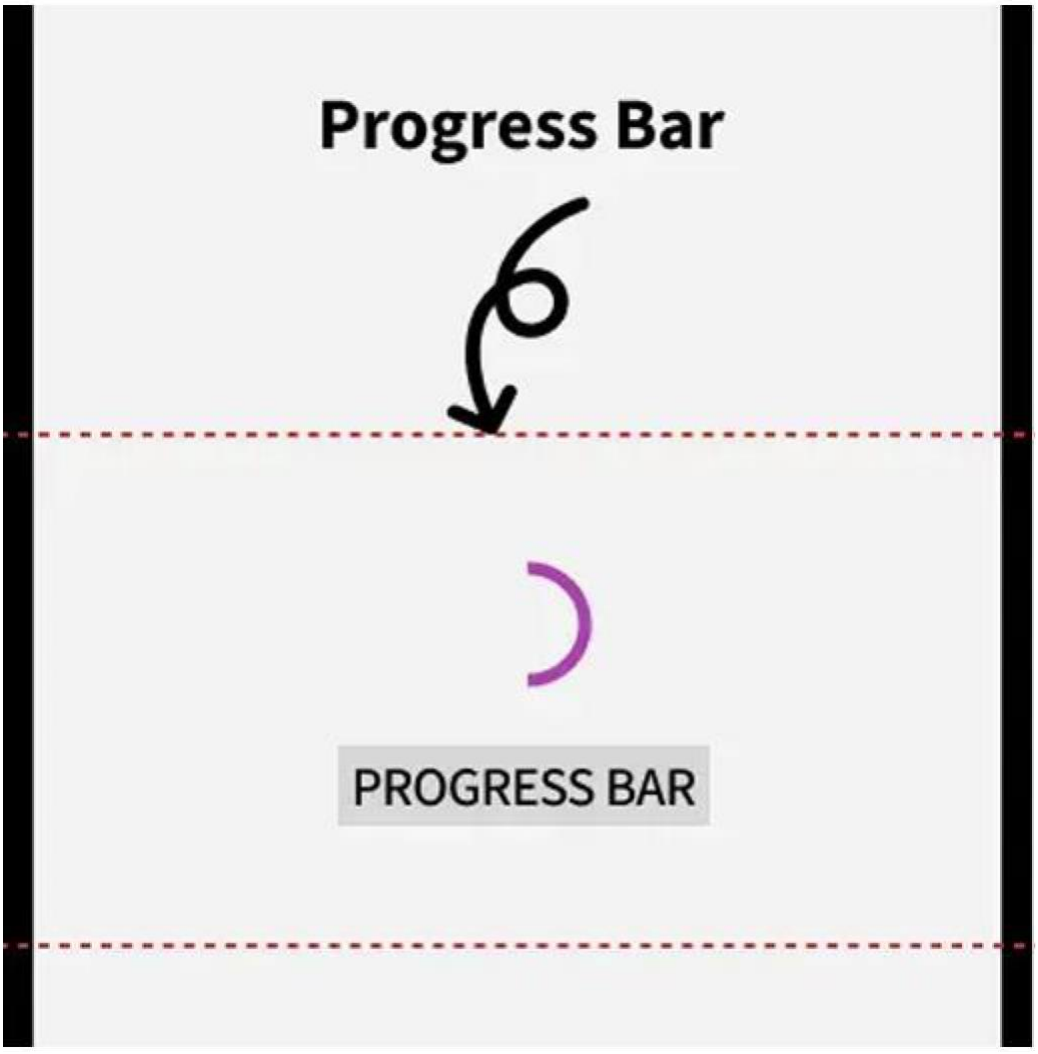
```
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_chainStyle="packed" />
<!--on below line we are creating a progress bar-->
<ProgressBar
android:id="@+id/progressBar"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:visibility="gone"
app:layout_constraintBottom_toTopOf="@+id/button"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView" />
<!--on below line we are creating a button-->
<Button
android:id="@+id/button"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="32dp"
android:text="Show Progress Bar"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/progressBar" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Mainactitvity.java

```
package com.example.progress;
import android.os.Bundle;
import android.view.View;
import android.widget.*;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
private boolean isProgressVisible = false;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
// Initializing the variables
Button Btn = findViewById(R.id.button);
ProgressBar PB = findViewById(R.id.progressBar);
// Click listener for the button
Btn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (isProgressVisible) {
// Hide progress bar and update button text
Btn.setText("Show Progress Bar");
PB.setVisibility(View.GONE);
isProgressVisible = false;
}
else {
// Show progress bar and update button text
```

```
Btn.setText("Hide Progress Bar");

PB.setVisibility(View.VISIBLE);

isProgressVisible = true;} } }); }}
```

Output:



RESULT:

Thus, the Android Application to display and control a progress bar using a button is developed and executed successfully.

Source Code:

Activitymain.XML

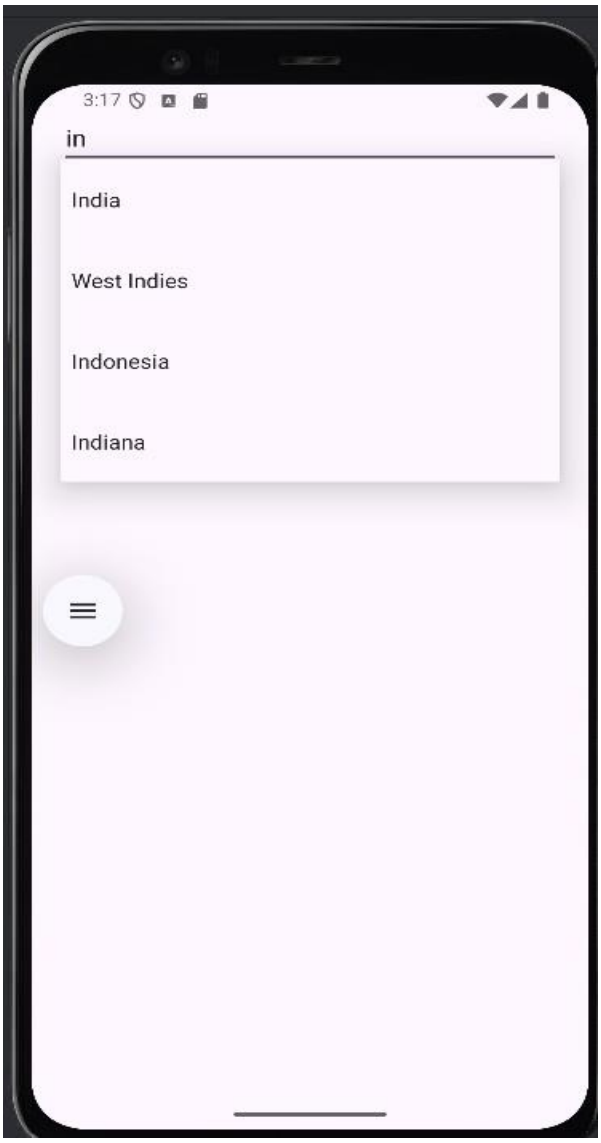
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:padding="20dp">
<AutoCompleteTextView
android:id="@+id/txtcountries"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Enter Country" />
</LinearLayout>
```

Mainactivity.java

```
package com.example.myapplication;
import android.app.Activity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
public class MainActivity extends Activity {
```

```
String[] countries = {
"India", "Australia", "West Indies", "Indonesia", "Indiana",
"South Africa", "England", "Bangladesh", "Sri Lanka", "Singapore"
};
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
AutoCompleteTextView autoTextView = findViewById(R.id.txtcountries);
ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_dropdown_item_1line, countries);
autoTextView.setThreshold(1); // Start suggesting after 1 character
autoTextView.setAdapter(adapter);
}
}
```

Output:



RESULT:

Thus, the Android Application to create an AutoComplete TextView with country suggestions is developed and executed successfully.

Source Code:

actyivitymain.XML

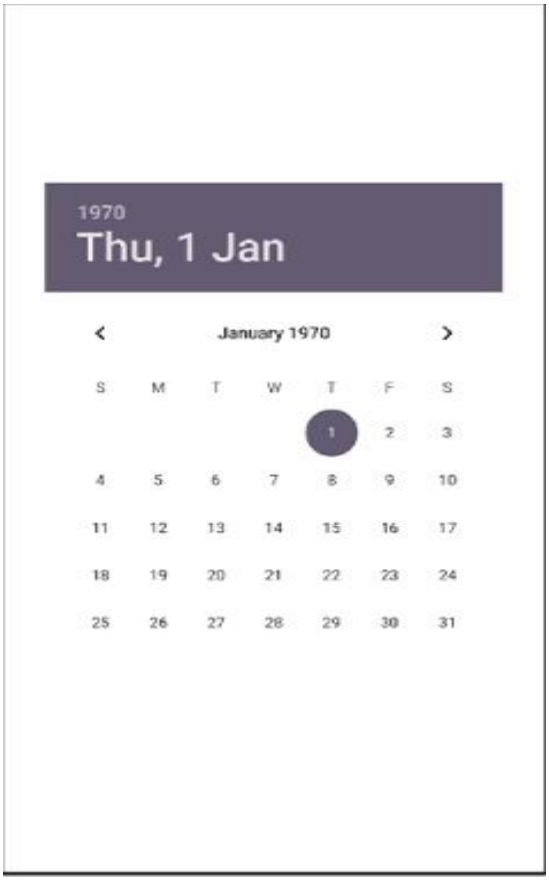
```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/white"
tools:context=".MainActivity">
<DatePicker
android:id="@+id/datePicker"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:datePickerMode="calendar"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Mainactivity.java

```
package org.example.demo;
```

```
import android.os.Bundle;
import android.widget.DatePicker;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import java.util.Calendar;
public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
// Initialize DatePicker from layout
DatePicker datePicker = findViewById(R.id.datePicker);
// Get today's date using Calendar instance
Calendar today = Calendar.getInstance();
// Initialize DatePicker with the current date
datePicker.init(today.get(Calendar.YEAR), today.get(Calendar.MONTH),
today.get(Calendar.DAY_OF_MONTH),new DatePicker.OnDateChangedListener()
{
@Override
public void onDateChanged(DatePicker view, int year, int month, int day) {
// Display selected date in Toast message
String msg = "You Selected: " + day + "/" + (month + 1) + "/" + year;
Toast.makeText(MainActivity.this, msg, Toast.LENGTH_SHORT).show();
} });}}
```

Output:



RESULT:

Thus, the Android Application to create a date picker and show the selected date in a toast is developed and executed successfully.

Source Code:

activitymain.XML

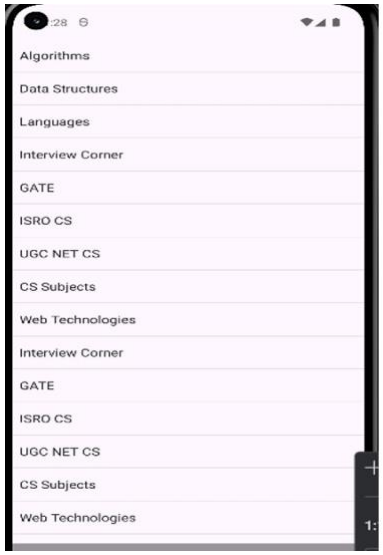
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">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

Mainactitvity.java

```
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
String tutorials[] = { "Algorithms",
"Data Structures",
"Languages",
"Interview Corner",
```

```
"GATE",
"ISRO CS",
"CS Subjects",
"Web Technologies" };
@Override
protected void onCreate(Bundle savedInstanceState){
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
ListView l = findViewById(R.id.list);
ArrayAdapter<String> arr = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_dropdown_item, tutorials);
l.setAdapter(arr);}}
```

Output:



RESULT:

Thus, the Android Application to display a list view of items is developed and executed successfully.

Source Code:

activitymain.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:padding="16dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
<Spinner
android:id="@+id/spinner"
android:layout_width="match_parent"
android:layout_height="wrap_content"/>
</LinearLayout>
```

Mainactitvity.java

```
package com.example.spinner1;
import android.app.Activity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
public class MainActivity extends Activity {
@Override
protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);

Spinner spinner = findViewById(R.id.spinner);

String[] items = {"India", "USA", "Japan"};

ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item,  items);

spinner.setAdapter(adapter);

}}
```

Output:

RESULT:

Thus, the Android Application to display a spinner with a dropdown list is developed and executed successfully.

Source Code:

activitymain.XML

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

Mainactivity.java

```
package com.example.webview1;
import android.os.Bundle;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
```

```
WebView webView = findViewById(R.id.webview);
webView.setWebViewClient(new WebViewClient());//page is loaded inside app's WebView
webView.getSettings().setJavaScriptEnabled(true);//Enables JavaScript for interactive websites.
webView.getSettings().setDomStorageEnabled(true);//sites can store data locally.
webView.loadUrl("https://wikipedia.org");
}}
```

Output:

RESULT:

Thus, the Android Application to display a webpage using WebView is developed and executed successfully.

Source Code:

activitymain.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
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:fitsSystemWindows="true"
tools:context=".MainActivity">
<androidx.constraintlayout.widget.ConstraintLayout
android:layout_width="match_parent"
android:layout_height="match_parent">
<ImageView
android:id="@+id/imageView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:foregroundGravity="center"
app:srcCompat="@drawable/book_publication"
tools:layout_editor_absoluteX="100dp"
tools:layout_editor_absoluteY="321dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Mainactitvity.java

```
package com.example.image1;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);}}
```

Output:

RESULT:

Thus, the Android Application to display an image using ImageView is developed and executed successfully.

Source Code:

activitymain.XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
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:fitsSystemWindows="true"
tools:context=".MainActivity">
<androidx.constraintlayout.widget.ConstraintLayout
android:layout_width="match_parent"
android:layout_height="match_parent">
<TextClock
android:id="@+id/textClock"
android:layout_width="182dp"
android:layout_height="58dp"
android:foregroundGravity="center"
android:gravity="center"
android:textSize="24sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.497"
app:layout_constraintStart_toStartOf="parent"
```

app:layout_constraintTop_toTopOf="parent"

app:layout_constraintVertical_bias="0.644" />

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.coordinatorlayout.widget.CoordinatorLayout>

Mainactitvity.java

package com.example.clock1;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle; public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState){

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);}}

Output:

RESULT:

Thus, the Android Application to display the current time using Digital Clock is developed and executed successfully.

Source Code:

activitymain.XML

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
 android:layout_width="match_parent"
 android:layout_height="match_parent"
 android:gravity="center"
 android:orientation="vertical">
 <Button
 android:id="@+id/startBtn"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="Start Music" />
 <Button
 android:id="@+id/stopBtn"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="Stop Music" />
 </LinearLayout>
```

Mainactitvity.java

```
package com.example.service1;
 import android.app.Service;
 import android.content.Intent;
 import android.media.MediaPlayer;
```

```
import android.os.IBinder;
public class MusicService extends Service {
private MediaPlayer player;
@Override
public void onCreate() {
super.onCreate();
// put your music file inside res/raw/music.mp3
player = MediaPlayer.create(this, R.raw.audio);
player.setLooping(true); // play in loop
}
@Override
   public int onStartCommand(Intent intent, int flags, int startId) {
      player.start();
      return START_STICKY;
   }
   @Override
   public void onDestroy() {
      super.onDestroy();
      if (player != null) {
         player.stop();
         player.release();
      } }
   @Override
   public IBinder onBind(Intent intent) {
      return null; }}
```

MusicService.java

```
package com.example.service1;
import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.IBinder;
public class MusicService extends Service {
   private MediaPlayer player;
   @Override
   public void onCreate() {
      super.onCreate();
      player = MediaPlayer.create(this, R.raw.audio);
      player.setLooping(true); // play in loop
   }
@Override
   public int onStartCommand(Intent intent, int flags, int startId) // intent to start the service,
additional data of start request, unique number for request
{ player.start();
      return START_STICKY; }
   @Override
   public void onDestroy() {
      super.onDestroy();
      if (player != null) {
         player.stop();
         player.release();
      }  }
   @Override
```

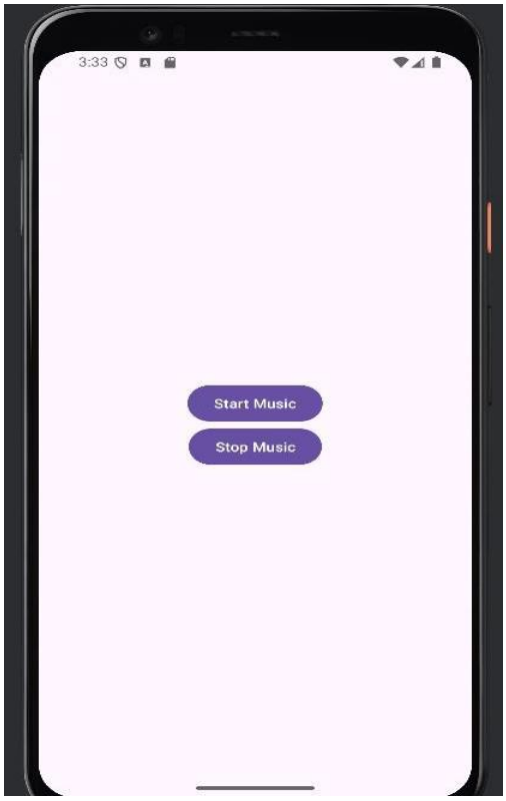
```
    public IBinder onBind(Intent intent) {
        return null;
    }}
```

Register Service in AndroidManifest.xml

Inside <application>...</application> add:

<service android:name=".MusicService" />

Output:



RESULT:

Thus, the Android Application to create a service for playing music is developed and executed successfully.

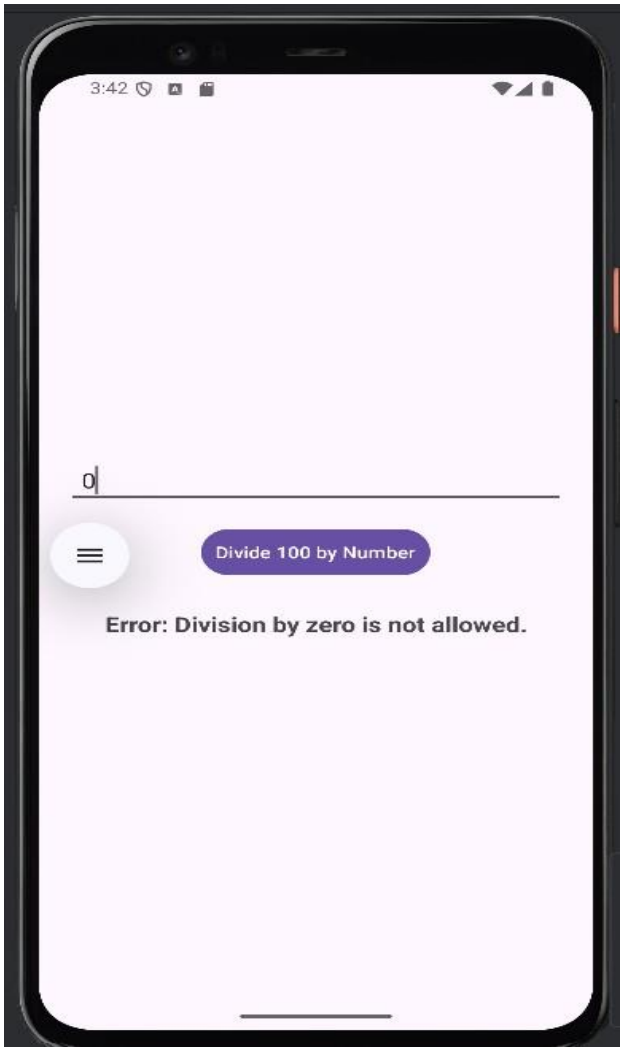
Source Code:

Mainactitvity.java

```
package com.example.exception1;
 import androidx.appcompat.app.AppCompatActivity;
 import android.os.Bundle;
import android.widget.Button;
 import android.widget.EditText;
import android.widget.TextView;
 public class MainActivity extends AppCompatActivity {
 EditText inputNumber;
Button btnDivide;
 TextView resultView;
 @Override
protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
inputNumber = findViewById(R.id.inputNumber);
btnDivide = findViewById(R.id.btnDivide);
resultView = findViewById(R.id.resultView);
 btnDivide.setOnClickListener(v ->
 {
try
{ // Get user input
```

```
String input = inputNumber.getText().toString();
 // May cause NumberFormatException if input is not a number
int number = Integer.parseInt(input);
// May cause ArithmeticException if number = 0
 int result = 100 / number;
 resultView.setText("Result: " + result);
} catch (NumberFormatException e) {
resultView.setText("Error: Please enter a valid number.");
 } catch (ArithmeticException e) {
resultView.setText("Error: Division by zero is not allowed.");
} catch (Exception e) {
resultView.setText("Unexpected error: " + e.getMessage());
 } }); }}
```

Output:



RESULT:

Thus, the Android Application to handle runtime errors using exception handling is developed and executed successfully.

Source Code:

activitymain.XML

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical" android:padding="20dp"
   android:layout_width="match_parent" android:layout_height="match_parent">
   <EditText android:id="@+id/editId"
      android:hint="Enter ID (for update/delete)"
      android:layout_width="match_parent" android:layout_height="wrap_content"/>
   <EditText android:id="@+id/editName"
      android:hint="Enter Name"
      android:layout_width="match_parent" android:layout_height="wrap_content"/>
   <Button android:id="@+id/btnAdd"
      android:text="Add" android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
   <Button android:id="@+id/btnUpdate"
      android:text="Update" android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
   <Button android:id="@+id/btnDelete"
      android:text="Delete" android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
   <TextView android:id="@+id/txtStudents"
      android:layout_width="match_parent" android:layout_height="wrap_content"
      android:paddingTop="10dp"/>
 </LinearLayout>
```

Mainactitvity.java

```
package com.example.database1;

 import android.content.Context;

 import android.database.sqlite.SQLiteDatabase;

 import   android.database.sqlite.SQLiteOpenHelper;

public class database1 extends SQLiteOpenHelper {

public database1(Context context) {

 super(context, "StudentDB", null, 1);

}

@Override

public void onCreate(SQLiteDatabase db) {

 db.execSQL("CREATE TABLE students(id INTEGER PRIMARY KEY AUTOINCREMENT, name
TEXT)");

}

@Override

 public void onUpgrade(SQLiteDatabase db, int oldV, int newV) {

db.execSQL("DROP TABLE IF EXISTS students");

 onCreate(db);

 } }
```

Database1.java

```
package com.example.database1;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;


public class database1 extends SQLiteOpenHelper

{
   public database1(Context context) {       super(context, "StudentDB", null, 1);     }
```

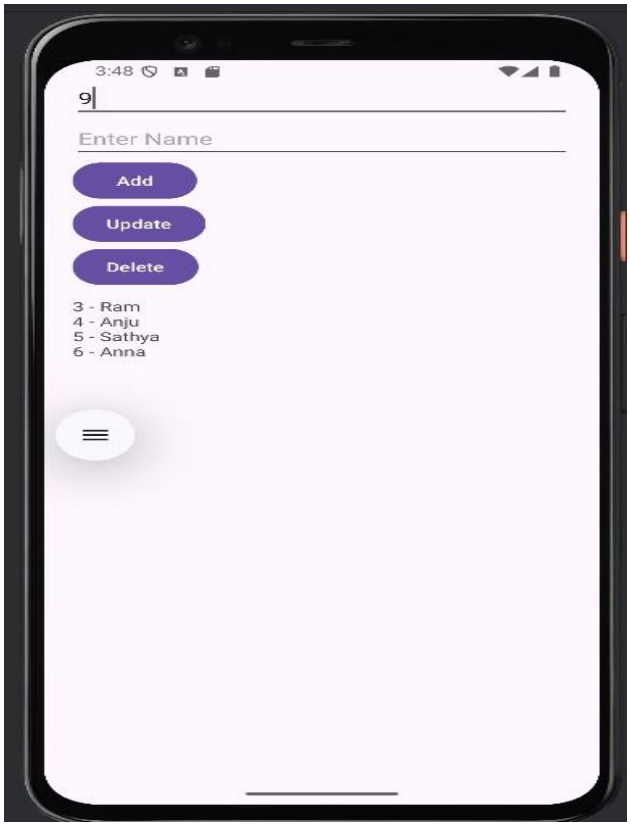
```
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE students(id INTEGER PRIMARY KEY AUTOINCREMENT, name
TEXT)");
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldV, int newV) {
        db.execSQL("DROP TABLE IF EXISTS students");
        onCreate(db);
    }}
```

Student1.java

```
package com.example.database1;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

public class student{

    database1 helper;

public student(Context c)

                    { helper = new database1 (c); }

    // CREATE

    public void addStudent(String name) {

        SQLiteDatabase db = helper.getWritableDatabase();

        ContentValues v = new ContentValues();

        v.put("name", name);

        db.insert("students", null, v);

        db.close();

    }
```

```
// READ
    public Cursor getStudents() {
        return helper.getReadableDatabase().rawQuery("SELECT * FROM students", null);
    }
    // UPDATE
    public void updateStudent(int id, String newName) {
        SQLiteDatabase db = helper.getWritableDatabase();
        ContentValues v = new ContentValues();
        v.put("name", newName);
        db.update("students", v, "id=?", new String[]{String.valueOf(id)});
        db.close();
    }
    // DELETE
    public void deleteStudent(int id) {
        SQLiteDatabase db = helper.getWritableDatabase();
        db.delete("students", "id=?", new String[]{String.valueOf(id)});
        db.close();
    }}
```

Output:



RESULT:

Thus, the Android Application to connect and manage data using SQLite database is developed and executed successfully.