MOBILE APPLICATIONS DEVELOPMENT
LAB RECORD

Submitted By:
Vidhya Lakshmi. B
106117107
CSE-A

EXERCISE : 1 SMS APP WITH ENCRYPTION

AIM:

To develop an application in android that invokes the built-in SMS app to send SMS and encrypt, decrypt the SMS using available built-in algorithm.

PROCEDURE:

- The UI of main activity is created with a textbox for the recipient's phone number, a textbox for the message body, a send button and an open button to read the received messages.
- The activity received is created to read the received messages.
- SMSManager API will directly send SMS from our application. Intent with proper action is used (ACTION_VIEW), to invoke a built-in SMS app to send SMS from our application.
- The Javax.crypto library is used to implement the AES encryption and decryption algorithm, which is an available built-in algorithm to encrypt and decrypt the SMS so as to ensure security. The algorithm is implemented in the Aeshelper. Java class.
- The manifest has the appropriate declarations to allow the use of related features like SMS.

CODE:

activity_main.xml

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

```
tools:context=".MainActivity">

<TextView
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:gravity="center"
   android:textSize="40dp"
   android:text="Enter mobile no. and message to send"
   app:layout_constraintBottom_toBottomOf="parent"
   app:layout_constraintLeft_toLeftOf="parent"
   app:layout_constraintRight_toRightOf="parent"
   app:layout_constraintTop_toTopOf="parent" />

<EditText
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:hint="Enter Phone Number"
   android:id="@+id/phone_number"
   android:layout_gravity="center"
   android:layout_marginTop="20dp"
   android:width="200dp"
   />

<EditText
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:hint="SMS Body"
   android:id="@+id/sms_body"
   android:layout_gravity="center"
   android:layout_marginTop="10dp"
   android:width="200dp"
   />

<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="SEND"
   android:id="@+id/sms_this_app"
   android:layout_gravity="center"
   android:layout_marginTop="30dp"
   />

<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="OPEN"
   android:id="@+id/sms_different_app"
   android:layout_gravity="center"
   android:layout_marginTop="20dp"
```

```
    />

</LinearLayout>
```

activity_received_sms.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"
   tools:context=".ReceivedSMS">

   <com.google.android.material.floatingactionbutton.FloatingActionButton
      android:id="@+id/fab"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="bottom|end"
      android:layout_margin="@dimen/fab_margin"
      app:srcCompat="@android:drawable/ic_dialog_email" />

   <ListView
      android:id="@+id/listView"
      android:layout_width="match_parent"
      android:layout_height="match_parent" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

content_received_sms.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"
   android:layout_width="match_parent"
   android:layout_height="match_parent"

app:layout_behavior="@string/appbar_scrolling_view_behavior"></androidx.constraintlayout.wid
get.ConstraintLayout>
```

fragment_first.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=".FirstFragment">

   <TextView
      android:id="@+id/textview_first"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/hello_first_fragment"
      app:layout_constraintBottom_toTopOf="@id/button_first"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent" />

   <Button
      android:id="@+id/button_first"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/next"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@id/textview_first" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

fragment_second.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=".SecondFragment">

   <TextView
      android:id="@+id/textview_second"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      app:layout_constraintBottom_toTopOf="@id/button_second"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent" />
```

```
    <Button
        android:id="@+id/button_second"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/previous"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/textview_second" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.smsmanagerapp;

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

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.example.smsmanagerapp.AESHelper;

public class MainActivity extends AppCompatActivity {

    Button sendButton, openButton;
    EditText phoneNumber, smsBody;
    String phoneNo, message;
    private static final int MY_SMS_PERMISSION_REQUEST = 0;

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

        sendButton = findViewById(R.id.sms_this_app);
```

```
        openButton = findViewById(R.id.sms_different_app);
        phoneNumber = findViewById(R.id.phone_number);
        smsBody = findViewById(R.id.sms_body);

        sendButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                sendMessage();
            }
        });

        openButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openNextActivity();
            }
        });
    }

    private void openNextActivity() {
        phoneNo = phoneNumber.getText().toString();
        message = smsBody.getText().toString();
        Intent smsIntent = new Intent(MainActivity.this, ReceivedSMS.class);
        smsIntent.setData(Uri.parse("sms:" + phoneNo));
        smsIntent.putExtra("sms_body", message);


        try {
            startActivity(smsIntent);
        } catch(android.content.ActivityNotFoundException ex) {
            Toast.makeText(MainActivity.this, ex.toString(), Toast.LENGTH_LONG).show();
        }
    }

    private void sendMessage() {
        phoneNo = phoneNumber.getText().toString();
        message = smsBody.getText().toString();

        if(ContextCompat.checkSelfPermission(this, Manifest.permission.SEND_SMS) !=
PackageManager.PERMISSION_GRANTED) {
            if(ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.SEND_SMS)){

            } else {
                ActivityCompat.requestPermissions(this, new String[] {Manifest.permission.SEND_SMS,
Manifest.permission.READ_SMS}, MY_SMS_PERMISSION_REQUEST);
            }
        } else {
            try {
```

```
            String enc = AESHelper.encrypt(message, "testseed");
            SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(phoneNo, null, enc, null, null);
            Toast.makeText(getApplicationContext(), "SMS sent", Toast.LENGTH_LONG).show();
        } catch(android.content.ActivityNotFoundException ex) {
            Toast.makeText(getApplicationContext(), "SMS failed", Toast.LENGTH_LONG).show();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
  }


  @Override
  public void onRequestPermissionsResult(int requestCode, String permissions[], int[]
grantResults) {
    if (requestCode == MY_SMS_PERMISSION_REQUEST) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            String enc = AESHelper.encrypt(message, "testseed");
            SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(phoneNo, null, enc, null, null);
            Toast.makeText(getApplicationContext(), "SMS sent", Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(getApplicationContext(), "SMS failed", Toast.LENGTH_LONG).show();
        }
    }
  }
}
```

AESHelper.java

```
package com.example.smsmanagerapp;

import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;
import java.util.Base64;

import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

public class AESHelper {

  private static SecretKeySpec secretKey;
  private static byte[] key;

  public static void setKey(String myKey)
```

```
    {
        MessageDigest sha = null;
        try {
            key = myKey.getBytes("UTF-8");
            sha = MessageDigest.getInstance("SHA-1");
            key = sha.digest(key);
            key = Arrays.copyOf(key, 16);
            secretKey = new SecretKeySpec(key, "AES");
        }
        catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
    }

    public static String encrypt(String strToEncrypt, String secret)
    {
        try
        {
            setKey(secret);
            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
            cipher.init(Cipher.ENCRYPT_MODE, secretKey);
            return Base64.getEncoder().encodeToString(cipher.doFinal(strToEncrypt.getBytes("UTF-
8")));
        }
        catch (Exception e)
        {
            System.out.println("Error while encrypting: " + e.toString());
        }
        return null;
    }

    public static String decrypt(String strToDecrypt, String secret)
    {
        try
        {
            setKey(secret);
            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
            cipher.init(Cipher.DECRYPT_MODE, secretKey);
            return new String(cipher.doFinal(Base64.getDecoder().decode(strToDecrypt)));
        }
        catch (Exception e)
        {
            System.out.println("Error while decrypting: " + e.toString());
        }
        return null;
    }
```

```
}
```

FirstFragment.java

```
package com.example.smsmanagerapp;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.navigation.fragment.NavHostFragment;

public class FirstFragment extends Fragment {

    @Override
    public View onCreateView(
            LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState
    ) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_first, container, false);
    }

    public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        view.findViewById(R.id.button_first).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                NavHostFragment.findNavController(FirstFragment.this)
                        .navigate(R.id.action_FirstFragment_to_SecondFragment);
            }
        });
    }
}
```

ReceivedSMS.java

```
package com.example.smsmanagerapp;

import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
```

```
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

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

public class ReceivedSMS extends AppCompatActivity {

   public List<String> getSMS(){
      List<String> sms = new ArrayList<String>();
      Uri uriSMSURI = Uri.parse("content://sms/sent");
      Cursor cur = getContentResolver().query(uriSMSURI, null, null, null, null);

      while (cur != null && cur.moveToNext()) {
         String address = cur.getString(cur.getColumnIndex("address"));
         String body = cur.getString(cur.getColumnIndexOrThrow("body"));
         sms.add("Number: " + address + "\nMessage: " + body);
      }

      if (cur != null) {
         cur.close();
      }
      return sms;
   }

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_received_s_m_s);
      Intent intent = getIntent();
      String value = intent.getStringExtra("key");

      final ListView list = findViewById(R.id.listView);
      List<String> arrayList = getSMS();
      ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, arrayList);
      list.setAdapter(arrayAdapter);
      list.setOnItemClickListener(new AdapterView.OnItemClickListener() {
         @Override
```

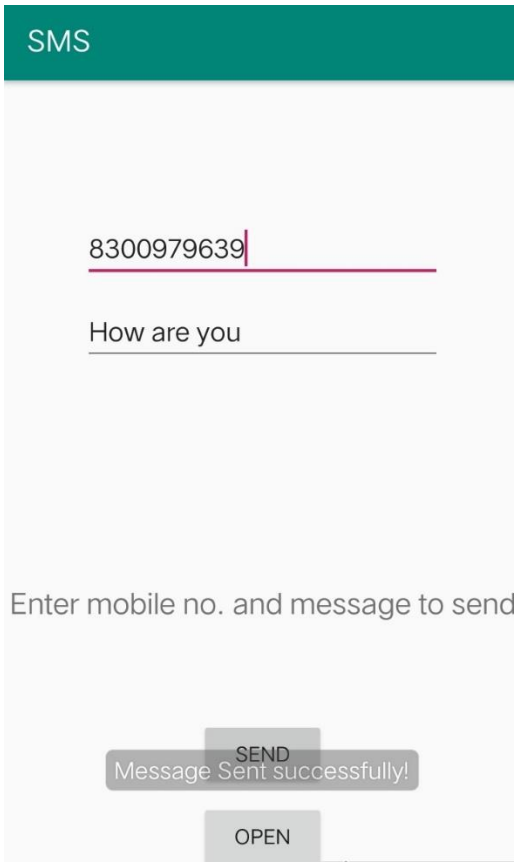
```
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
            String clickedItem = (String) list.getItemAtPosition(position);
            String separator = "Message: ";
            int sepPos = clickedItem.indexOf(separator);
            String mes = clickedItem.substring(sepPos + separator.length());
            mes = AESHelper.decrypt(mes, "testseed");
            Toast.makeText(ReceivedSMS.this, "The decrypted SMS is: " + mes,
Toast.LENGTH_LONG).show();
        }
      });
   }
}
```

SecondFragment.java

```
package com.example.smsmanagerapp;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.navigation.fragment.NavHostFragment;

public class SecondFragment extends Fragment {

   @Override
   public View onCreateView(
         LayoutInflater inflater, ViewGroup container,
         Bundle savedInstanceState
   ) {
      // Inflate the layout for this fragment
      return inflater.inflate(R.layout.fragment_second, container, false);
   }

   public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
      super.onViewCreated(view, savedInstanceState);

      view.findViewById(R.id.button_second).setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            NavHostFragment.findNavController(SecondFragment.this)
                  .navigate(R.id.action_SecondFragment_to_FirstFragment);
         }
      });
   }
```

}

OUTPUT:



INFERENCE:

The created app uses the built-in SMS application to send and receive messages with extra security using bulit-in algorithm to encrypt while sending and decrypt while receiving.

EXERCISE : 2 MEDIA RECORDER

AIM:

To develop an application in android that records and stores media files (audio).

PROCEDURE:

- The UI of the main activity is created using 2 buttons to start and stop recording.
- The buttons respectively call the onClick functions, startRecording() and stopRecording() from the Mainactivity.java class. An object of MediaRecorder class is created to record the audio, specify the output format and the audio source.
- The app mainly uses MediaRecorder APIs to record audio.
- Permission to record audio has to be taken from the user and also MediaMuxer is used to record multiple channels.
- After the recording stops the sound file is created and added to the media library. The audio file is stored on external storage.
- The manifest has the appropriate declarations to allow the use of related features like Microphone and Storage.

CODE:

activity_main.xml

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

   <TextView
      android:id="@+id/textView"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="RECORD AUDIO"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      android:textSize="30dp"
      app:layout_constraintVertical_bias="0.147" />
```

```
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.517"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_constraintVertical_bias="0.273" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Stop"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.517"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button1"
        app:layout_constraintVertical_bias="0.336" />

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

MainActivity.java

```
package com.example.audio;

import androidx.appcompat.app.AppCompatActivity;

import android.content.ContentResolver;
import android.content.ContentValues;
import android.content.Intent;
import android.media.MediaRecorder;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import java.io.File;
import java.io.IOException;
```

```
public class MainActivity extends AppCompatActivity {

    MediaRecorder recorder;
    File audiofile = null;
    static final String TAG = "MediaRecording";
    Button startButton,stopButton;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        startButton = (Button) findViewById(R.id.button1);
        startButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {
                    startRecording(v);
                } catch (IOException e) {
                    Log.e(TAG, e.getMessage(), e);
                }
            }
        });
        stopButton = (Button) findViewById(R.id.button2);
        stopButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                stopRecording(v);
            }
        });
    }

    public void startRecording(View view) throws IOException {
        startButton.setEnabled(false);
        stopButton.setEnabled(true);
        //Creating file
        String file_path=getApplicationContext().getFilesDir().getPath();
//        File dir = new File(file_path);
        File dir = Environment.getExternalStorageDirectory();
        try {
            audiofile = File.createTempFile("sound", ".mp3", dir);
            Toast.makeText(this, audiofile.getAbsolutePath(), Toast.LENGTH_LONG );
            Log.i(TAG, "stored");
            Log.i(TAG, audiofile.getAbsolutePath());
        } catch (IOException e) {
            e.printStackTrace();
            Log.e(TAG, "external storage access error");
            return;
        }
```

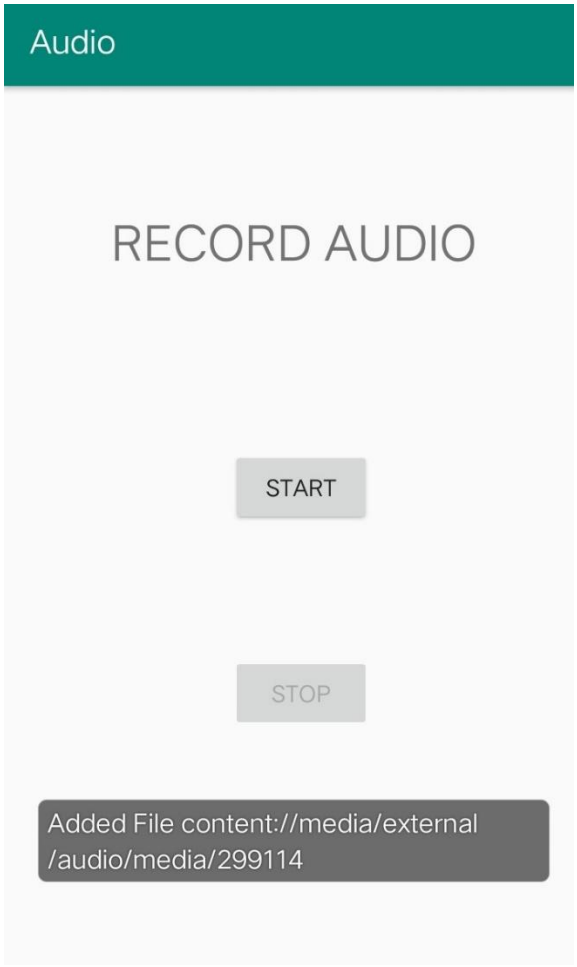
```
        //Creating MediaRecorder and specifying audio source, output format, encoder & output
format
        recorder = new MediaRecorder();
        recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
        recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
        recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
        recorder.setOutputFile(audiofile.getAbsolutePath());
        recorder.prepare();
        recorder.start();
    }

    public void stopRecording(View view) {
        startButton.setEnabled(true);
        stopButton.setEnabled(false);
        //stopping recorder
        recorder.stop();
        recorder.release();
        //after stopping the recorder, create the sound file and add it to media library.
        addRecordingToMediaLibrary();
    }

    protected void addRecordingToMediaLibrary() {
        //creating content values of size 4
        ContentValues values = new ContentValues(4);
        long current = System.currentTimeMillis();
        values.put(MediaStore.Audio.Media.TITLE, "audio" + audiofile.getName());
        values.put(MediaStore.Audio.Media.DATE_ADDED, (int) (current / 1000));
        values.put(MediaStore.Audio.Media.MIME_TYPE, "audio/3gpp");
        values.put(MediaStore.Audio.Media.DATA, audiofile.getAbsolutePath());

        //creating content resolver and storing it in the external content uri
        ContentResolver contentResolver = getContentResolver();
        Uri base = MediaStore.Audio.Media.EXTERNAL_CONTENT_URI;
        Uri newUri = contentResolver.insert(base, values);

        //sending broadcast message to scan the media file so that it can be available
        sendBroadcast(new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE, newUri));
        Toast.makeText(this, "Added File " + newUri, Toast.LENGTH_LONG).show();
    }

 private void muxing() {

        String outputFile = "";

        String TAG="tag";
        try {

            outputFile = getOutputMediaFile(2).toString();
            MediaExtractor videoExtractor = new MediaExtractor();
```

```
        FileInputStream fileInputStream = new FileInputStream(file);
        FileDescriptor afdd = fileInputStream.getFD();

        videoExtractor.setDataSource(afdd);

        MediaExtractor audioExtractor = new MediaExtractor();
        audioExtractor.setDataSource("/storage/emulated/0/a.m4a");

        Log.d(TAG, "Video Extractor Track Count " + videoExtractor.getTrackCount());
        Log.d(TAG, "Audio Extractor Track Count " + audioExtractor.getTrackCount());

        MediaMuxer muxer = new MediaMuxer(outputFile, MediaMuxer.OutputFormat.MUXER_OU
TPUT_MPEG_4);

        videoExtractor.selectTrack(0);
        MediaFormat videoFormat = videoExtractor.getTrackFormat(0);
        int videoTrack = muxer.addTrack(videoFormat);

        audioExtractor.selectTrack(0);
        MediaFormat audioFormat = audioExtractor.getTrackFormat(0);
        int audioTrack = muxer.addTrack(audioFormat);

        Log.d(TAG, "Video Format " + videoFormat.toString());
        Log.d(TAG, "Audio Format " + audioFormat.toString());

        boolean sawEOS = false;
        int frameCount = 0;
        int offset = 100;
        int sampleSize = 256 * 1024;
        ByteBuffer videoBuf = ByteBuffer.allocate(sampleSize);
        ByteBuffer audioBuf = ByteBuffer.allocate(sampleSize);
        MediaCodec.BufferInfo videoBufferInfo = new MediaCodec.BufferInfo();
        MediaCodec.BufferInfo audioBufferInfo = new MediaCodec.BufferInfo();

        videoExtractor.seekTo(0, MediaExtractor.SEEK_TO_CLOSEST_SYNC);
        audioExtractor.seekTo(0, MediaExtractor.SEEK_TO_CLOSEST_SYNC);

        muxer.start();

        while (!sawEOS) {
           videoBufferInfo.offset = offset;
           videoBufferInfo.size = videoExtractor.readSampleData(videoBuf, offset);

           if (videoBufferInfo.size < 0 || audioBufferInfo.size < 0) {
              Log.d(TAG, "saw input EOS.");
              sawEOS = true;
              videoBufferInfo.size = 0;

           } else {
```

```
            videoBufferInfo.presentationTimeUs = videoExtractor.getSampleTime();
            videoBufferInfo.flags = videoExtractor.getSampleFlags();
            muxer.writeSampleData(videoTrack, videoBuf, videoBufferInfo);
            videoExtractor.advance();

            frameCount++;
            Log.d(TAG, "Frame (" + frameCount + ") Video PresentationTimeUs:" + videoBufferInfo.
presentationTimeUs + " Flags:" + videoBufferInfo.flags + " Size(KB) " + videoBufferInfo.size / 1024);
            Log.d(TAG, "Frame (" + frameCount + ") Audio PresentationTimeUs:" + audioBufferInfo.
presentationTimeUs + " Flags:" + audioBufferInfo.flags + " Size(KB) " + audioBufferInfo.size / 1024);

          }
        }

        Toast.makeText(getApplicationContext(), "frame:" + frameCount,          Toast.LENGTH_SHO
RT).show();

        boolean sawEOS2 = false;
        int frameCount2 = 0;
        while (!sawEOS2) {
          frameCount2++;

          audioBufferInfo.offset = offset;
          audioBufferInfo.size = audioExtractor.readSampleData(audioBuf, offset);

          if (videoBufferInfo.size < 0 || audioBufferInfo.size < 0) {
            Log.d(TAG, "saw input EOS.");
            sawEOS2 = true;
            audioBufferInfo.size = 0;
          } else {
            audioBufferInfo.presentationTimeUs = audioExtractor.getSampleTime();
            audioBufferInfo.flags = audioExtractor.getSampleFlags();
            muxer.writeSampleData(audioTrack, audioBuf, audioBufferInfo);
            audioExtractor.advance();

            Log.d(TAG, "Frame (" + frameCount + ") Video PresentationTimeUs:" + videoBufferInfo.
presentationTimeUs + " Flags:" + videoBufferInfo.flags + " Size(KB) " + videoBufferInfo.size / 1024);
            Log.d(TAG, "Frame (" + frameCount + ") Audio PresentationTimeUs:" + audioBufferInfo.
presentationTimeUs + " Flags:" + audioBufferInfo.flags + " Size(KB) " + audioBufferInfo.size / 1024);

          }
        }

        Toast.makeText(getApplicationContext(), "frame:" + frameCount2, Toast.LENGTH_SHORT).sh
ow();

        muxer.stop();
        muxer.release();
```

```
        } catch (IOException e) {
            Log.d(TAG, "Mixer Error 1 " + e.getMessage());
        } catch (Exception e) {
            Log.d(TAG, "Mixer Error 2 " + e.getMessage());
        }
    }

}
```

OUTPUT:



INFERENCE:

An application which records audio using MediaRecorder and muxes it using MediaMuxer was created.

EXERCISE : 3 MEDIA PLAYER

AIM:

To develop an application in android that uses mediaplayer to run audio and video files.

PROCEDURE:

- The UI of the main activity is created such that the user can choose if they want to play video or audio. The options are added as menu items.
- Object of MediaPlayer class is used in Mainactivity.java to set the source path and play the media file.
- The application mainly uses MediaPlayer class is the primary API for playing sound and video, and AudioManager class to manage audio sources and audio output.
- The manifest has the appropriate declarations to allow the use of related features like Internet Permission and Wake Lock Permission.

CODE:

activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools=“http://schemas.android.com/tools”
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   android:paddingBottom="@dimen/activity_vertical_margin"
   tools:context=".MainActivity">

   <TextView android:id="@+id/textview_time"
      android:text="@string/hello_world" android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:layout_alignParentLeft="true"
```

```
    android:layout_alignParentStart="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"/>

    <SurfaceView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/surfaceView"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_below="@+id/textview_time"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"/>
</RelativeLayout>
```

MainActivity.java

```
package com.example.lenovo.mediarecorder;
import androidx.appcompat.app.AppCompatActivity;
import android.content.res.AssetFileDescriptor;
import android.hardware.Camera;
import android.media.CamcorderProfile;
import android.media.MediaCodec;
import android.media.MediaExtractor;
import android.media.MediaFormat;
import android.media.MediaMuxer;
import android.media.MediaRecorder;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
```

```
import android.view.SurfaceView;
import android.view.View;
import android.widget.Button;
import android.widget.FrameLayout;
import android.widget.Toast;

import java.io.File;
import java.io.FileDescriptor;
import java.io.FileInputStream;
import java.io.IOException;
import java.nio.ByteBuffer;
import java.text.SimpleDateFormat;
import java.util.Date;


public class MainActivity extends AppCompatActivity {
    Button btn;
    boolean isrecording = false;
    FrameLayout fl;
    SurfaceView preview;
    private Camera mcamera;
    private String file;
    private MediaRecorder mediaRecorder;
    public static final int MEDIA_TYPE_IMAGE = 1;
    public static final int MEDIA_TYPE_VIDEO = 2;

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

```
    btn = findViewById(R.id.flip);
    fl = findViewById(R.id.fl);
    preview = findViewById(R.id.sv);
    btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (isrecording) {
                mediaRecorder.stop();
                mcamera.lock();
                releaseMediaRecorder();
                btn.setText("Start");
                isrecording = false;
                muxing();

            } else {
                if (prep()) {
                    mediaRecorder.start();
                    btn.setText("Stop");
                    isrecording = true;
                } else {
                    releaseMediaRecorder();
                }
            }
        }
    });
}

private boolean prep() {

    mcamera = Camera.open();
```

```
        mediaRecorder = new MediaRecorder();
        //  preview = new SurfaceView(this);
        mediaRecorder.setPreviewDisplay(preview.getHolder().getSurface());
        mcamera.startPreview();
        mcamera.unlock();

        mediaRecorder.setCamera(mcamera);
        mediaRecorder.setAudioSource(MediaRecorder.AudioSource.DEFAULT);
        mediaRecorder.setVideoSource(MediaRecorder.VideoSource.CAMERA);
//        mediaRecorder.setProfile(CamcorderProfile.get(CamcorderProfile.QUALITY_LOW));
        mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
        mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.DEFAULT);
        mediaRecorder.setVideoEncoder(MediaRecorder.VideoEncoder.DEFAULT);
        file = getOutputMediaFile(MEDIA_TYPE_VIDEO).toString();
        mediaRecorder.setOutputFile(file);
        mediaRecorder.setPreviewDisplay(preview.getHolder().getSurface());


        try {
            mediaRecorder.prepare();
        } catch (IOException e) {
            e.printStackTrace();
            return false;
        }
        return true;
    }

    private static File getOutputMediaFile(int type) {

        File mediaStorageDir = new File(Environment.getExternalStoragePublicDirectory(
```

```
            Environment.DIRECTORY_PICTURES), "MyCameraApp");

    if (!mediaStorageDir.exists()) {
        if (!mediaStorageDir.mkdirs()) {
            Log.d("MyCameraApp", "failed to create directory");
            return null;
        }
    }

    // Create a media file name
    String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
    File mediaFile;
    if (type == MEDIA_TYPE_IMAGE) {
        mediaFile = new File(mediaStorageDir.getPath() + File.separator +
                "IMG_" + timeStamp + ".jpg");
    } else if (type == MEDIA_TYPE_VIDEO) {
        mediaFile = new File(mediaStorageDir.getPath() + File.separator +
                "VID_" + timeStamp + ".mp4");
    } else {
        return null;
    }

    return mediaFile;
}

private void releaseMediaRecorder() {
    if (mediaRecorder != null) {
        mediaRecorder.reset();   // clear recorder configuration
        mediaRecorder.release(); // release the recorder object
        mediaRecorder = null;
```

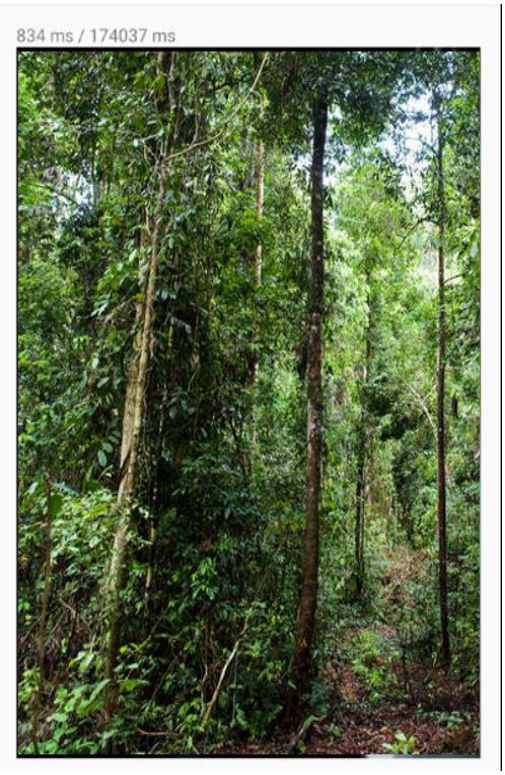
```
            mcamera.lock();         // lock camera for later use
        }
    }

    private void muxing() {

        String outputFile = "";

        String TAG="tag";
        try {

            outputFile = getOutputMediaFile(2).toString();
            MediaExtractor videoExtractor = new MediaExtractor();
            FileInputStream fileInputStream = new FileInputStream(file);
            FileDescriptor afdd = fileInputStream.getFD();

            videoExtractor.setDataSource(afdd);

            MediaExtractor audioExtractor = new MediaExtractor();
            audioExtractor.setDataSource("/storage/emulated/0/a.m4a");



            Log.d(TAG, "Video Extractor Track Count " + videoExtractor.getTrackCount());
            Log.d(TAG, "Audio Extractor Track Count " + audioExtractor.getTrackCount());

            MediaMuxer muxer = new MediaMuxer(outputFile,
MediaMuxer.OutputFormat.MUXER_OUTPUT_MPEG_4);

            videoExtractor.selectTrack(0);
            MediaFormat videoFormat = videoExtractor.getTrackFormat(0);
            int videoTrack = muxer.addTrack(videoFormat);
```

```
audioExtractor.selectTrack(0);
MediaFormat audioFormat = audioExtractor.getTrackFormat(0);
int audioTrack = muxer.addTrack(audioFormat);

Log.d(TAG, "Video Format " + videoFormat.toString());
Log.d(TAG, "Audio Format " + audioFormat.toString());

boolean sawEOS = false;
int frameCount = 0;
int offset = 100;
int sampleSize = 256 * 1024;
ByteBuffer videoBuf = ByteBuffer.allocate(sampleSize);
ByteBuffer audioBuf = ByteBuffer.allocate(sampleSize);
MediaCodec.BufferInfo videoBufferInfo = new MediaCodec.BufferInfo();
MediaCodec.BufferInfo audioBufferInfo = new MediaCodec.BufferInfo();


videoExtractor.seekTo(0, MediaExtractor.SEEK_TO_CLOSEST_SYNC);
audioExtractor.seekTo(0, MediaExtractor.SEEK_TO_CLOSEST_SYNC);

muxer.start();

while (!sawEOS) {
   videoBufferInfo.offset = offset;
   videoBufferInfo.size = videoExtractor.readSampleData(videoBuf, offset);


   if (videoBufferInfo.size < 0 || audioBufferInfo.size < 0) {
      Log.d(TAG, "saw input EOS.");
```

```
                sawEOS = true;
                videoBufferInfo.size = 0;

            } else {
                videoBufferInfo.presentationTimeUs = videoExtractor.getSampleTime();
                videoBufferInfo.flags = videoExtractor.getSampleFlags();
                muxer.writeSampleData(videoTrack, videoBuf, videoBufferInfo);
                videoExtractor.advance();


                frameCount++;
                Log.d(TAG, "Frame (" + frameCount + ") Video PresentationTimeUs:" +
videoBufferInfo.presentationTimeUs + " Flags:" + videoBufferInfo.flags + " Size(KB) " +
videoBufferInfo.size / 1024);
                Log.d(TAG, "Frame (" + frameCount + ") Audio PresentationTimeUs:" +
audioBufferInfo.presentationTimeUs + " Flags:" + audioBufferInfo.flags + " Size(KB) " +
audioBufferInfo.size / 1024);
            }
        }

        Toast.makeText(getApplicationContext(), "frame:" + frameCount,
Toast.LENGTH_SHORT).show();
        boolean sawEOS2 = false;
        int frameCount2 = 0;
        while (!sawEOS2) {
            frameCount2++;

            audioBufferInfo.offset = offset;
            audioBufferInfo.size = audioExtractor.readSampleData(audioBuf, offset);

            if (videoBufferInfo.size < 0 || audioBufferInfo.size < 0) {
                Log.d(TAG, "saw input EOS.");
```

```
                sawEOS2 = true;
                audioBufferInfo.size = 0;
            } else {
                audioBufferInfo.presentationTimeUs = audioExtractor.getSampleTime();
                audioBufferInfo.flags = audioExtractor.getSampleFlags();
                muxer.writeSampleData(audioTrack, audioBuf, audioBufferInfo);
                audioExtractor.advance();


                Log.d(TAG, "Frame (" + frameCount + ") Video PresentationTimeUs:" +
videoBufferInfo.presentationTimeUs + " Flags:" + videoBufferInfo.flags + " Size(KB) " +
videoBufferInfo.size / 1024);
                Log.d(TAG, "Frame (" + frameCount + ") Audio PresentationTimeUs:" +
audioBufferInfo.presentationTimeUs + " Flags:" + audioBufferInfo.flags + " Size(KB) " +
audioBufferInfo.size / 1024);


            }
        }


        Toast.makeText(getApplicationContext(), "frame:" + frameCount2,
Toast.LENGTH_SHORT).show();


        muxer.stop();
        muxer.release();


    } catch (IOException e) {
        Log.d(TAG, "Mixer Error 1 " + e.getMessage());
    } catch (Exception e) {
        Log.d(TAG, "Mixer Error 2 " + e.getMessage());
    }
  }
}
```

OUTPUT:



INFERENCE:

The created app uses MediaPlayer to play both audio and video files.

EXERCISE : 4 TEXT TO SPEECH APP

AIM:

To develop an application in android that converts text to speech.

PROCEDURE:

- The UI of the main activity is created with a textbox to enter text to be converted to speech and also a button on pressing which the text gets converted to speech by calling an onclick function.
- In Mainactivity.java, an object of the TextToSpeech class is created which takes input from the textbox on the mainactivity layout.
- The application mainly uses a TextToSpeech instance to synthesise text. However, it can only be used to synthesize text once it has completed its initialization.
- The application also implements the TextToSpeech.OnInitListener to notify of the completion of the initialization.
- Toasts are used to tell the user what is currently happening and to handle various errors.

CODE:

activity_main.xml

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

   <TextView
      android:id="@+id/textView1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Type the text and press SPEAK button"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      android:textSize="20dp"
      app:layout_constraintVertical_bias="0.143" />
```

```
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="SPEAK"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.744" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="Text"
        app:layout_constraintBottom_toTopOf="@+id/button1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

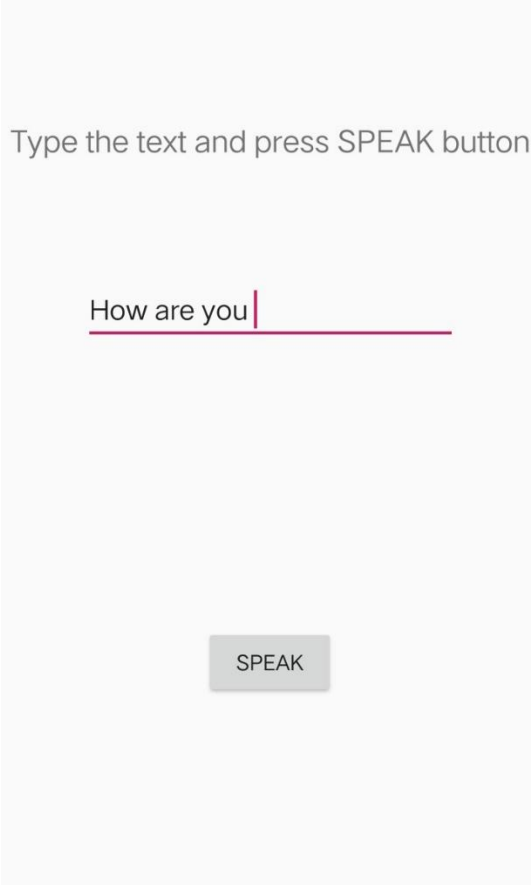
MainActivity.java

```
package com.example.texttospeech;

import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import java.util.Locale;

public class MainActivity extends Activity implements
        TextToSpeech.OnInitListener {
    /** Called when the activity is first created. */

    private TextToSpeech tts;
```

```
    private Button buttonSpeak;
    private EditText editText;

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

        tts = new TextToSpeech(this, this);
        buttonSpeak = (Button) findViewById(R.id.button1);
        editText = (EditText) findViewById(R.id.editText1);

        buttonSpeak.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                speakOut();
            }

        });
    }

    @Override
    public void onDestroy() {
// Don't forget to shutdown tts!
        if (tts != null) {
            tts.stop();
            tts.shutdown();
        }
        super.onDestroy();
    }

    @Override
    public void onInit(int status) {

        if (status == TextToSpeech.SUCCESS) {

            int result = tts.setLanguage(Locale.US);

            if (result == TextToSpeech.LANG_MISSING_DATA
                    || result == TextToSpeech.LANG_NOT_SUPPORTED) {
                Log.e("TTS", "This Language is not supported");
            } else {
                buttonSpeak.setEnabled(true);
                speakOut();
            }

        } else {
            Log.e("TTS", "Initilization Failed!");
        }
```

```
    }

    private void speakOut() {
        String text = editText.getText().toString();
        tts.speak(text, TextToSpeech.QUEUE_FLUSH, null);
    }

}
```

OUTPUT:



INFERENCE:

The app was created which converted the text input typed to speech output.

EXERCISE : 5 IMAGE CAPTURE USING CAMERA

AIM:

To develop an application in android that captures the image using built in camera and store the media file.

PROCEDURE:

- The UI of the main activity is built to allow the user to choose whether they want to capture image or record video.
- There are two buttons, one for capture photo and the other for record video.
- The image can be seen in the gallery, while the video can be directly played from within the app.
- An intent has been implemented in the Mainactivity.java class to access the camera of the device to record videos and capture images.
- The application uses the ACTION_IMAGE_CAPTURE or ACTION_VIDEO_CAPTURE action to open a camera app and receive the resulting photo or video.
- The URI location where the photo or video is to be saved has to be specified, in the EXTRA_OUTPUT extra. The full-size photo is saved using getExternalStoragePublicDirect(), with the DIRECTORY_PICTURES argument.
- The manifest has the appropriate declarations to allow the use of related features like camera.

CODE:

activity_main.xml

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

   <TextView
      android:id="@+id/output"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Captured"
```

```
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.506"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.366" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Capture"
        app:layout_constraintBottom_toTopOf="@+id/output"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="268dp"
        android:layout_height="253dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/output"
        app:srcCompat="@drawable/ic_launcher_background" />

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

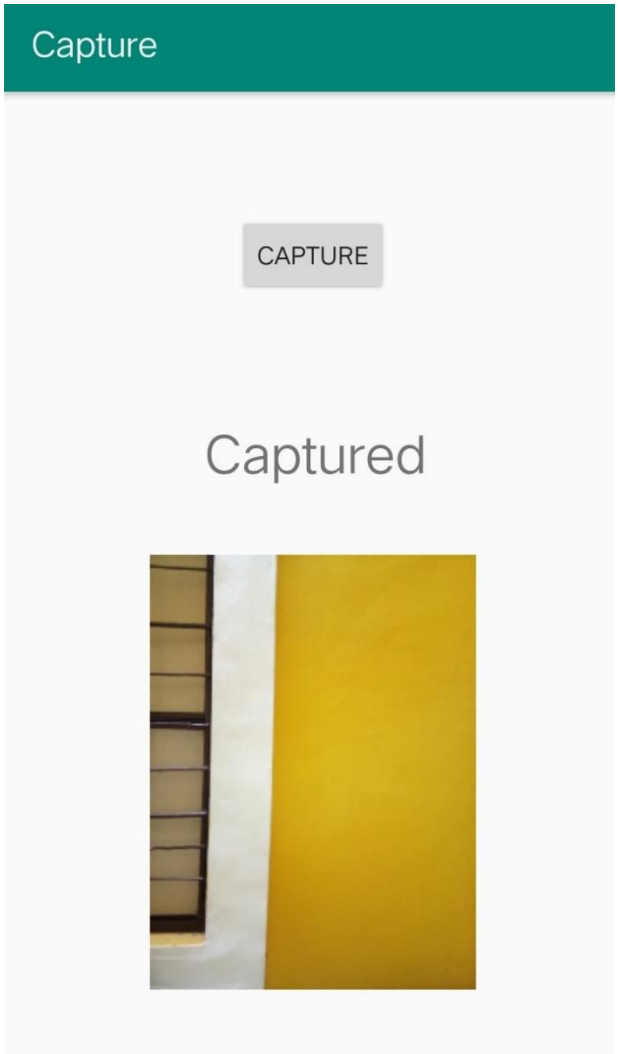
MainActivity.java

```
package com.example.capture;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import java.io.File;
```

```
public class MainActivity extends AppCompatActivity {

   private static final int CAMERA_REQUEST = 1888;
   ImageView imageView;
   public void onCreate(Bundle savedInstanceState) {

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

      imageView = (ImageView) this.findViewById(R.id.imageView1);
      Button photoButton = (Button) this.findViewById(R.id.button1);

      photoButton.setOnClickListener(new View.OnClickListener() {

         @Override
         public void onClick(View v) {
            Intent cameraIntent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
            startActivityForResult(cameraIntent, CAMERA_REQUEST);
         }
      });
   }

   protected void onActivityResult(int requestCode, int resultCode, Intent data) {
      if (requestCode == CAMERA_REQUEST) {
         Bitmap photo = (Bitmap) data.getExtras().get("data");
         imageView.setImageBitmap(photo);
      }
   }

   private void cameraIntent() throws IOException
   {
         if (checkPermissions()){
               final String dir =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES)+ "/picFolder/
";
         File newdir = new File(dir);
            newdir.mkdirs();
            String file = dir+"photo.jpg";
            File newfile = new File(file);
            try {
               newfile.createNewFile();
            }
            catch (IOException e)
            {
                  e.printStackTrace();
            }

            mCurrentPhotoPath = Uri.fromFile(newfile);
```

```
            Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, mCurrentPhotoPath);
            startActivityForResult(takePictureIntent, REQUEST_CAMERA);
        }
    }

    @Override
    public booleanonCreateOptionsMenu(Menu menu){
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

OUTPUT:



INFERENCE:

The created app successfully takes photo using the built-in camera and stores them in a file location.