

# TEXT TO SPEECH CONVERTER

## AIM & ALGORITHM

### AIM :

To convert a text message to an audio file in an Android application.

### ALGORITHM :

Step 1: Set up the TextToSpeech engine and configure it.

Step 2: Implement the logic to convert text to speech.

Step 3: Save the generated audio as a file.

Step 4: Configure the TextToSpeech engine in Android activity.

Step 5: Implement the logic to convert the text to speech using the TextToSpeech API.

Step 6: Save the generated speech as an audio file using Android's MediaRecorder or any appropriate method.

## ACTIVITY\_MAIN.XML

```
<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="16dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Text to Speech Converter"
        android:textSize="24sp"
        android:textStyle="bold"
        android:layout_marginBottom="20dp"/>

    <EditText
        android:id="@+id/editTextInput"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter text to convert to speech"
        android:inputType="textMultiLine"
        android:minHeight="100dp"
```

```

        android:layout_marginBottom="16dp"/>

        <Button
            android:id="@+id/speakButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Speak"
            android:textSize="18sp"
            android:layout_gravity="center"/>

    </LinearLayout>

```

## MAINACTIVITY.JAVA

```

package com.example.texttospeech;

import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import java.util.Locale;

public class MainActivity extends AppCompatActivity implements
TextToSpeech.OnInitListener {

    private TextToSpeech tts;
    private EditText editTextInput;
    private Button speakButton;

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

        editTextInput = findViewById(R.id.editTextInput);
        speakButton = findViewById(R.id.speakButton);

        // Initialize TextToSpeech
        tts = new TextToSpeech(this, this);

        speakButton.setOnClickListener(v -> {
            String text = editTextInput.getText().toString().trim();
            if (!text.isEmpty()) {
                speakText(text);
            }
        });
    }
}

```

```

        } else {
            Toast.makeText(MainActivity.this, "Please enter some text",
Toast.LENGTH_SHORT).show();
        }
    });
}

@Override
public void onInit(int status) {
    if (status == TextToSpeech.SUCCESS) {
        // Set language to US English
        int result = tts.setLanguage(Locale.US);
        if (result == TextToSpeech.LANG_MISSING_DATA || result ==
TextToSpeech.LANG_NOT_SUPPORTED) {
            Toast.makeText(this, "Language not supported",
Toast.LENGTH_SHORT).show();
        } else {
            speakButton.setEnabled(true);
        }
    } else {
        Toast.makeText(this, "Text-to-Speech initialization failed",
Toast.LENGTH_SHORT).show();
    }
}

private void speakText(String text) {
    tts.speak(text, TextToSpeech.QUEUE_FLUSH, null, null);
}

@Override
protected void onDestroy() {
    if (tts != null) {
        tts.stop();
        tts.shutdown();
    }
    super.onDestroy();
}
}

```

## ANDROIDMANIFEST.XML

```

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

```

```
<!-- No special permissions required for Text-to-Speech -->

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
    <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>
```

## OUTPUT

texttospeech

narai

SPEAK

