# ICP11

Venubabu Linga – vl6hw@umsystem.edu
GitHub:https://github.com/VenubabuLinga/WebDevCourse/tree/main/Moblie_dev/ICP11
Sarath Chandra kunisetty – skkh2@umsystem.edu
GitHub: https://github.com/kunisettysarath/WebMobileProgramming-Spring22/tree/main/MobileDevelopment/ICP's/ICP11

## Objective:

The objective of this ICP1 is to learn about TextToSpech library in android by build an android application

## Introduction:

Text to speech is a android library which is used to synthesize a voice playback for the user's text input. This library gives us an option of enabling the voice playback in various languages. It is also known as "Speech Synthesis"

## Implementation:

There are two main files required inorder to implement this task: "activity_main.xml" and "MainActivity.java".

**activity_main.xml**: In this activity file we have added the text view, edit text and material button with corresponding id values to them in the relative layout.

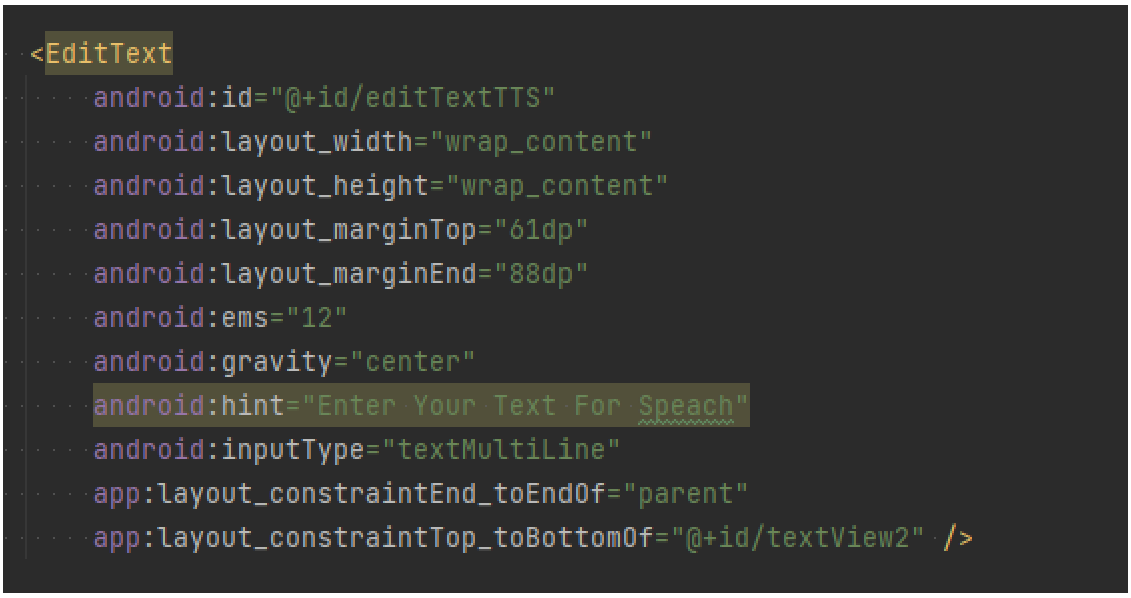
**Text View**: A text-display element that shows text to the user. To provide text that can be edited by the user.

**Button**: This user interface element that allows the user to perform an action by tapping or clicking.

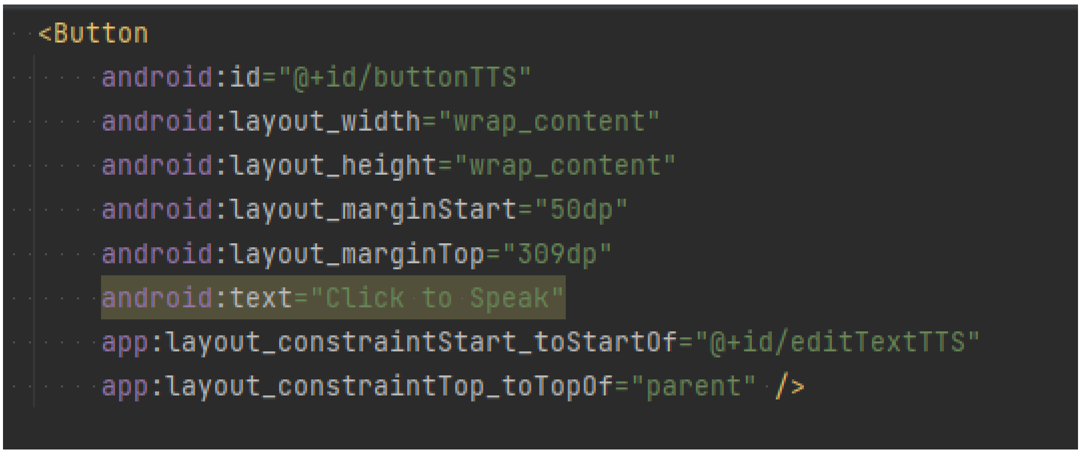
**Edit Text**: This element is used as a text field for the user to enter and input which will be rendered by the application and provide the output as required.

All the necessary styling like margin, size, font for the elements are added in this file itself.

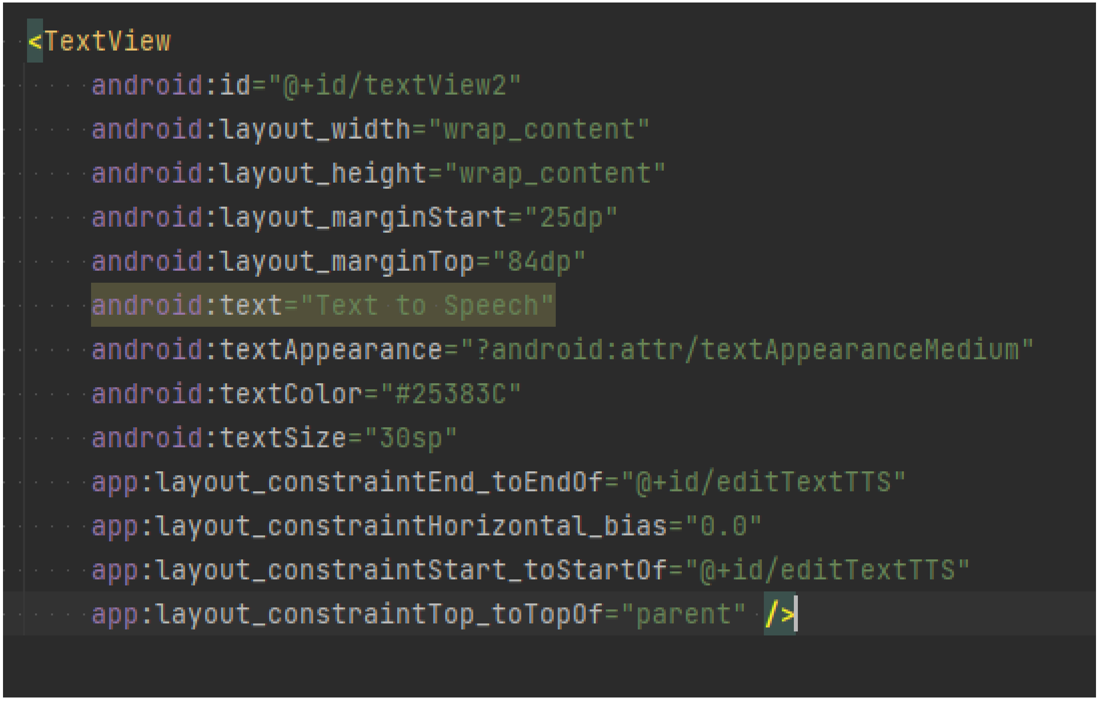
## Input Code:



```xml
<EditText
    android:id="@+id/editTextTTS"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="61dp"
    android:layout_marginEnd="88dp"
    android:ems="12"
    android:gravity="center"
    android:hint="Enter Your Text For Speach"
    android:inputType="textMultiLine"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />
```



```xml
<Button
    android:id="@+id/buttonTTS"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="50dp"
    android:layout_marginTop="309dp"
    android:text="Click to Speak"
    app:layout_constraintStart_toStartOf="@+id/editTextTTS"
    app:layout_constraintTop_toTopOf="parent" />
```



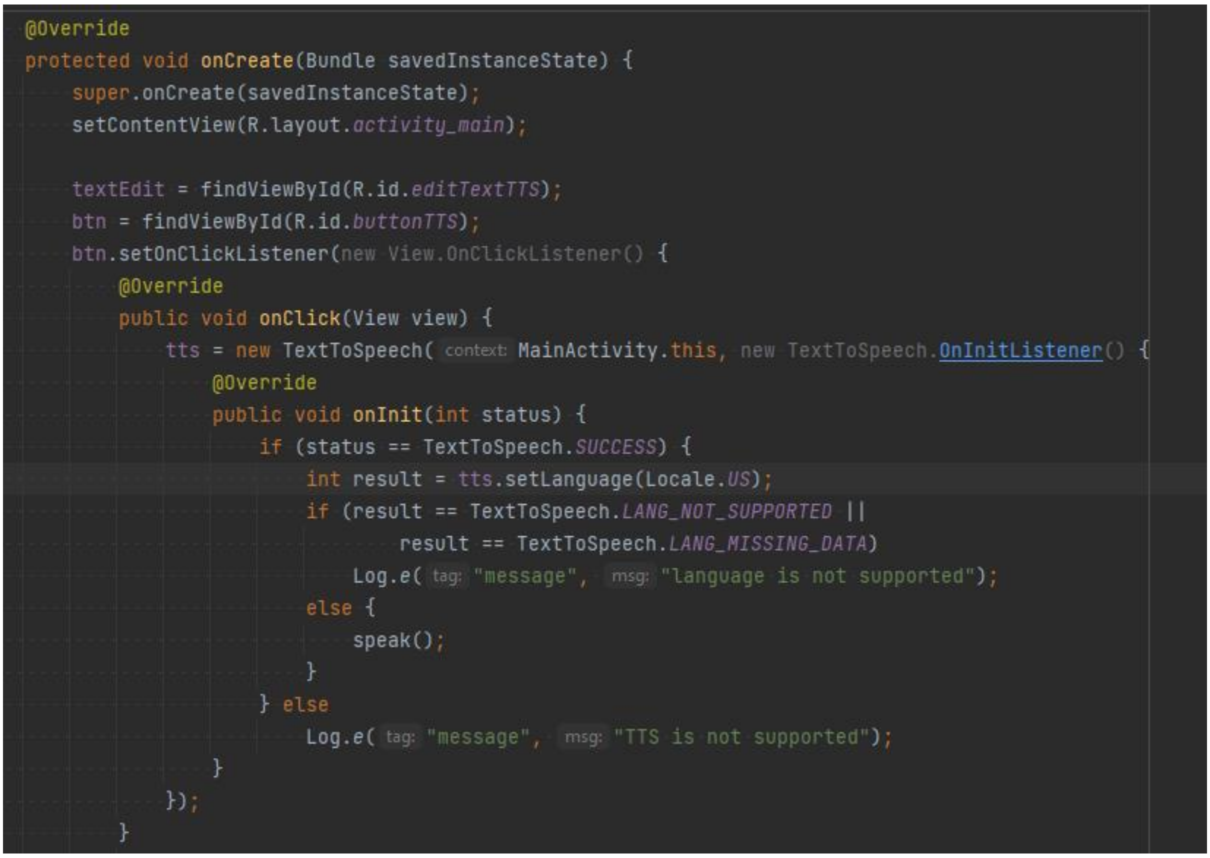
```xml
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="25dp"
    android:layout_marginTop="84dp"
    android:text="Text to Speech"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:textColor="#25383C"
    android:textSize="30sp"
    app:layout_constraintEnd_toEndOf="@+id/editTextTTS"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="@+id/editTextTTS"
    app:layout_constraintTop_toTopOf="parent" />
```

**MainActivity.java:**

We have added the core logic in this file while implementing the "onCreate()" method. In this method 3 variable instances was created for the 3 UI elements by using the ID.

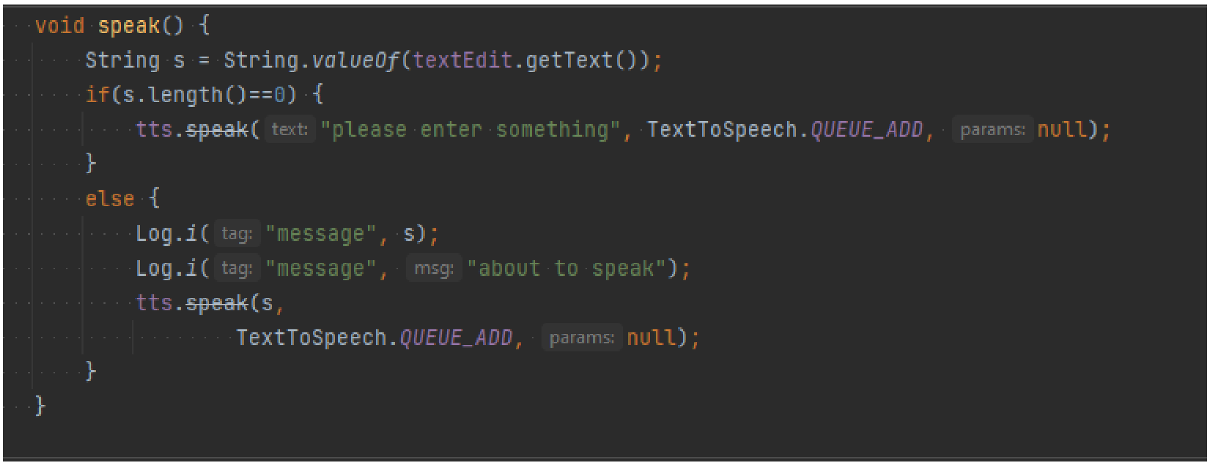
The TTS(Text-To-Speech) will be activated once the user clicks on the "Click to Speak" button. So an event listener was added to that button "setOnClickListener()" which triggers the "onClick()" method once user clicks on the button in which TextToSpeech is implemented.

A new object is created for the TextToSpeech class while implementing OnInit() method. Here we are setting the language as US and verifying whether the user's input as per the requirement and if it is then the "speak()" is invoked.



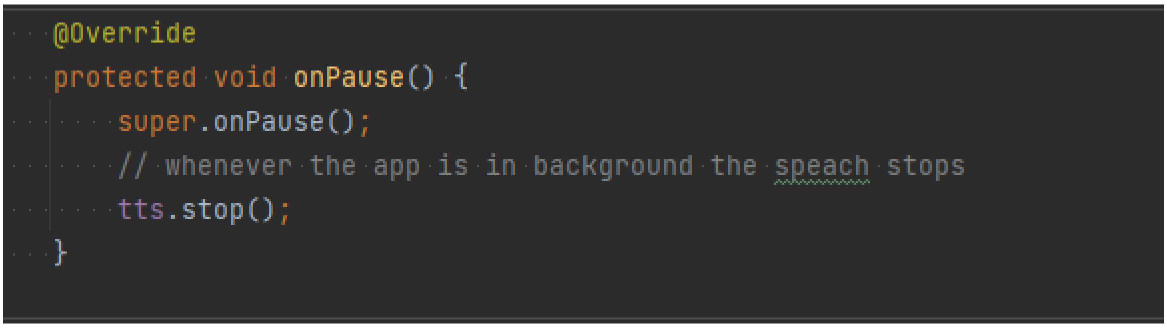
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    textEdit = findViewById(R.id.editTextTTS);
    btn = findViewById(R.id.buttonTTS);
    btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            tts = new TextToSpeech( context: MainActivity.this, new TextToSpeech.OnInitListener() {
                @Override
                public void onInit(int status) {
                    if (status == TextToSpeech.SUCCESS) {
                        int result = tts.setLanguage(Locale.US);
                        if (result == TextToSpeech.LANG_NOT_SUPPORTED ||
                                result == TextToSpeech.LANG_MISSING_DATA)
                            Log.e( tag: "message",  msg: "language is not supported");
                        else {
                            speak();
                        }
                    } else
                        Log.e( tag: "message",  msg: "TTS is not supported");
                }
            });
        }
```

In "speak()" method the user input is stored into the string s, where it is validated if the users input is valid or not. If it is valid then the TalkToSpeech's "speak()" is invoked with the user's input passed as a parameter. If the user's is invalid then a voice play back is initiated prompting the user to input something into the field.

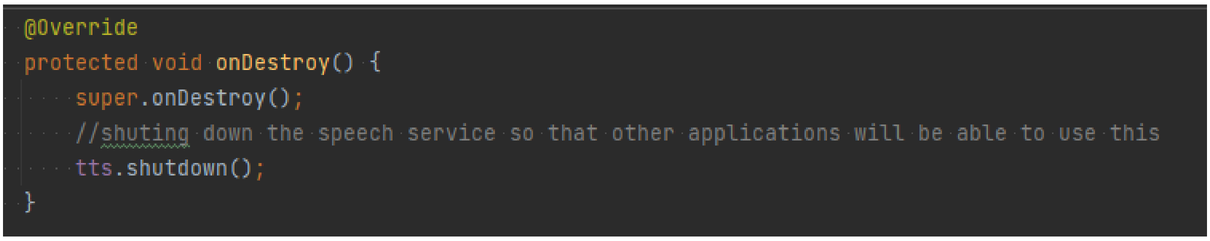


```
void speak() {
    String s = String.valueOf(textEdit.getText());
    if(s.length()==0) {
        tts.~~speak~~( text: "please enter something", TextToSpeech.QUEUE_ADD,  params: null);
    }
    else {
        Log.i( tag: "message", s);
        Log.i( tag: "message",  msg: "about to speak");
        tts.~~speak~~(s,
                TextToSpeech.QUEUE_ADD,  params: null);
    }
}
```

“onPause()” method was implemented inorder to stop the voice playback once the app is returned to background activity and “onDestroy()” method is implemented inorder to shutdown the TextToSpeach service so that other resources will be able to utilize this functionality.

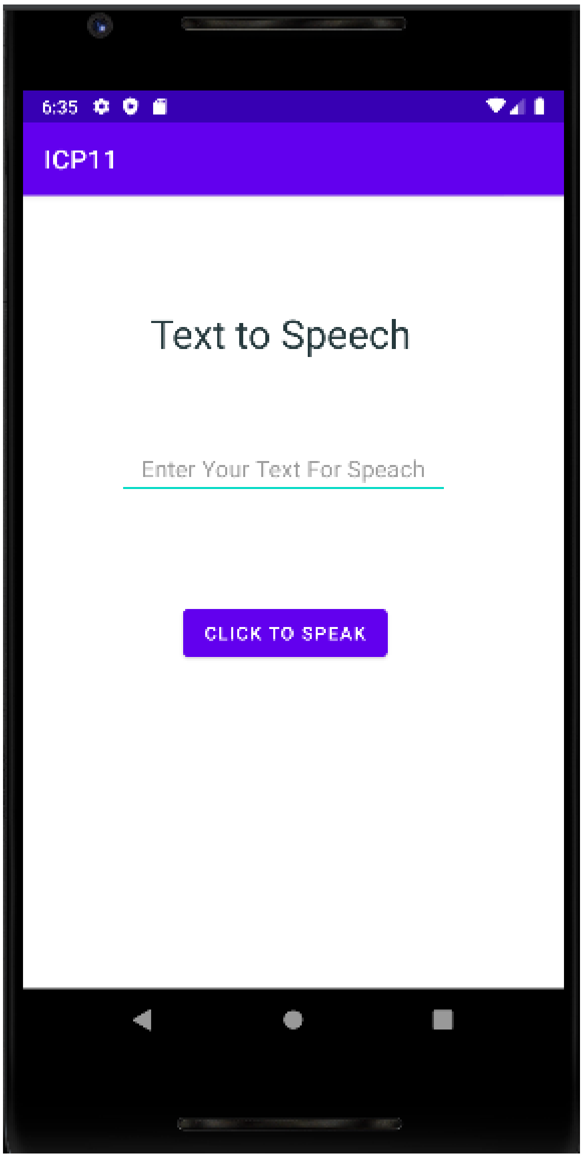


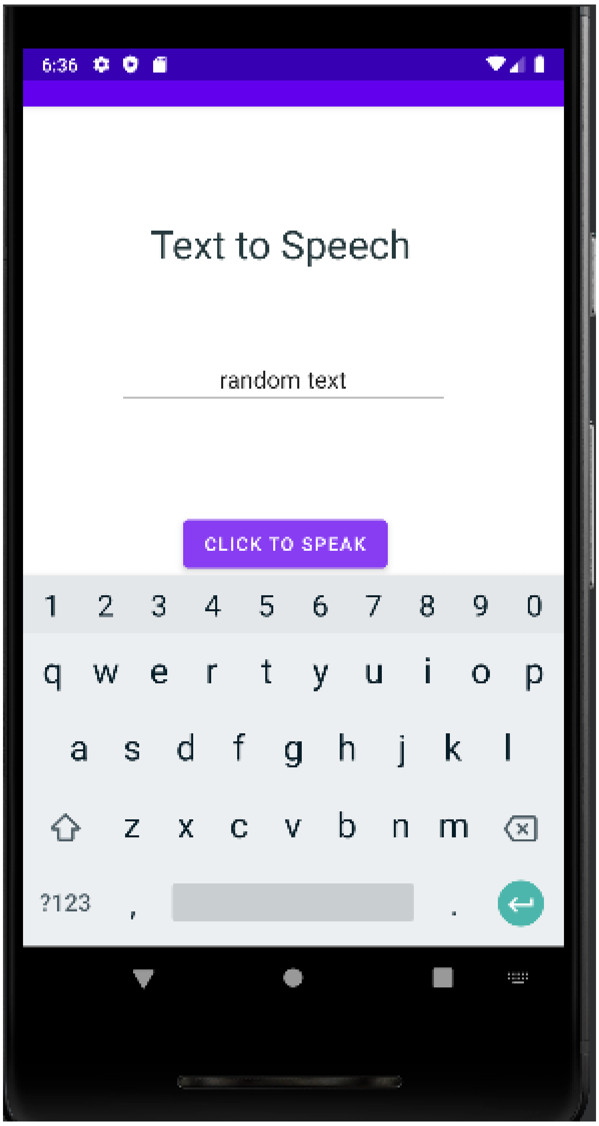
```
@Override
protected void onPause() {
    super.onPause();
    // whenever the app is in background the speach stops
    tts.stop();
}
```



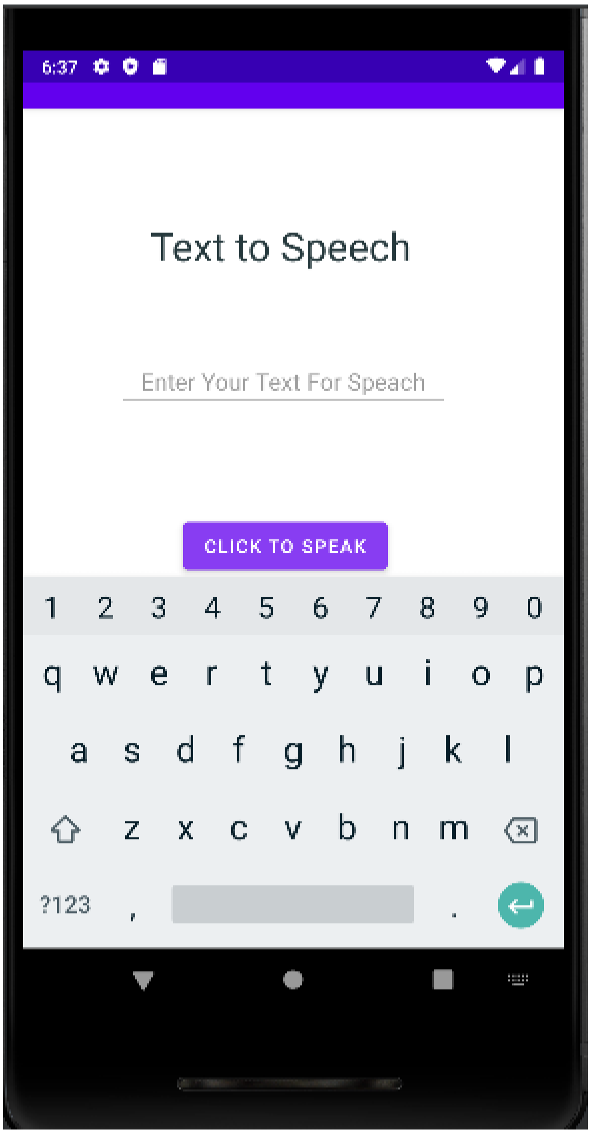
```
@Override
protected void onDestroy() {
    super.onDestroy();
    //shuting down the speech service so that other applications will be able to use this
    tts.shutdown();
}
```

**Output:**


6:35
ICP11
Text to Speech
Enter Your Text For Speach
CLICK TO SPEAK



Once user clicked on the “click to speak” button TextToSpeak is activated and the text “random text” is synthesized for voice playback



As the user tries to click on the “click to speak” button without entering any input in the edit text TextToSpeak is activated with text “please enter something” for voice playback