ICP11 Report

Venkata Naga Sai Srikanth Gollapudi – sggc9@umsystem.edu
github: https://github.com/GollapudiSrikanth/WebMobileSpring2022/tree/main/Mobile/ICP11

Mangi Varun Reddy - vmf9p@umsystem.edu
github: https://github.com/MangiVarunReddy/Webcourse/tree/main/Mobile_part/ICP11

Task Performed

1. Created an empty project in the Android Studio with the name of ICP11.
2. Added textview, edittext and button in the mainacitivity.xml with id provided to each of the tags.
3. Fetching the data based on the id provided in mainactivity.xml and event from the button
4. Onclick of the button an event listener is added where here the text to speech conversion takes place
5. Text to speech is an api provided in android built in package where on Success it takes English locale and then conversion process starts.
6. First step of conversion from text to speech is to get the data from the user which is taken from the edittext field and used as input to TextToSpeech api
7. In this TextToSpeech api , speak method is used to convert the text to speech and results in producing speech as output.
8. In one of the corner case there might be a scenario where the user entered text may not be translated and that error scenario is handled and shown an error message displaying that TextToSpeech cannot be translated.

OutCome
From this task we are able to create an application which converts from text to speech for mobile devices and able to test and develop the application in the android studio.

Task Distribution
Work Contributed was equally distributed together.

Task Screenshots:

Fig-1: Below picture represents, adding EditText.

```xml
<EditText
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Enter your Text"
    android:layout_below="@+id/textView2"
    android:layout_marginTop="50dp"
    tools:ignore="MissingConstraints"
    />
```

Fig-2: Adding Button to mainactivity.xml

```xml
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:text="Speak"
    android:layout_below="@+id/editText"
    android:layout_marginTop="50dp"
    tools:ignore="MissingConstraints"
    />
```

Fig-3: Fetching the data using Id's

```java
editText = findViewById(R.id.editText);
button = findViewById(R.id.button);
```

Fig-4: Text to speech logic which handles error and success scenario's

```java
textToSpeech = new TextToSpeech( context: MainActivity.this, new TextToSpeech.OnInitListener() {
    @Override
    public void onInit(int status) {
        if(status==TextToSpeech.SUCCESS){
            int result =textToSpeech.setLanguage(Locale.ENGLISH);
        if(result==TextToSpeech.LANG_NOT_SUPPORTED || result == TextToSpeech.LANG_MISSING_DATA){
            Log.e( tag: "message", msg: "language is not supported");
        }
        else{
            textToSpeech.speak(String.valueOf(editText.getText()),TextToSpeech.QUEUE_ADD, params: null);
        }


        }
        else{
            Log.e( tag: "message", msg: "TT's is not supported");
        }
    }
```

Output Screenshots

- Below screenshot shows the home page of the application where user has a textbox which is used for entering the text and onclick on speak button it converts text to speech and produces the output.