National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Praise Olamide Abimbola
Student ID: 22101012

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Praise Olamide Abimbola
Student ID:	22101012
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Vikas Sahni
Submission Due Date:	14/08/2023
Project Title:	Configuration Manual
Word Count:	1110
Page Count:	15

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	6th August 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Praise Olamide Abimbola
22101012

1 Introduction

In this configuration manual, a step by step procedure on how to set up the system and replicate what has been done to develop the same application and achieve the same result is presented. The manual follows the following structure. Section 2 presents the Minimum System Requirements, section 3 presents Downloading and Installing Flutter on Android Studio, section 4 presents the speech recognition application and section 5 presents Performance Monitoring.

2 Minimum System Requirements

This is the minimum software and hardware requirements needed to set up the application environment.

Table 1: Minimum System Requirements

Requirements	**Windows operating system**	**Mac operating system**
Operating system	Windows 10 or later (64-bit), x86-64 based	Version 10.14 (Mojave) or later
Disk Space	1.64 GB	2.8 GB
Tools	Android Studio IDE, Android emulator	Xcode , IOS emulator

3 Downloading and Installing Flutter on Android Studio

To download and install flutter SDK, follow these steps;

1. Download and install Flutter from the official Flutter website [1].
2. Once the download is complete, extract the file in the desired location on computer.
3. Add the Flutter bin directory to system PATH. This can be done by running the code in the terminal. “$ export PATH=”$ PATH:‘pwd‘/flutter/bin”
4. To verify that flutter/bin directory is in the PATH successfully, “run $ echo $PATH”

Once flutter SDK has been successfully installed, the next stage is to set up an Emulator or Connect a Physical Device on android studio.

For android, open android studio, go to ”Configure” select ”SDK Manager,” then select ”Android SDK” then select ”SDK platform” then apply and install an android emulator.

[1] `https://flutter.dev/docs/get-started/install`

In Android Virtual Device Manager, click "Run" in the toolbar. The emulator starts up and displays the default canvas for selected OS version and device.

Figure 1: Android Studio Simulator

For iOS, Xcode's built-in simulator can be used, or a physical iOS device can be connected to computer using a cord. macOS supports developing Flutter apps for iOS, Android, macOS itself and the web.

1. Install the latest stable version of Xcode from the apple store.
2. Read through the Xcode license agreement and sign.
3. To open the iOS simulator, on command line run the code "$ open -a Simulator"

Figure 2: iOS Simulator

4 Speech to Text Application

To run the application, the code can be cloned from github [2] or unzip the submitted code titled "SpeechRecognitionOffloadingCode". Once the code is cloned, the Service Account Key JSON file should be added to the assets folder. The next step is to go into the 'pubspec.yaml' file in project directory, and ensure that the required dependencies are added. Once this is done, open the terminal, navigate to project directory and run 'flutter pub get' to fetch the required dependencies.

Start the android or iOS emulator to run the application. At the top bar, click on the green play button to run the application.

[2] `https://github.com/Olamide-1/speech_to_text.git`

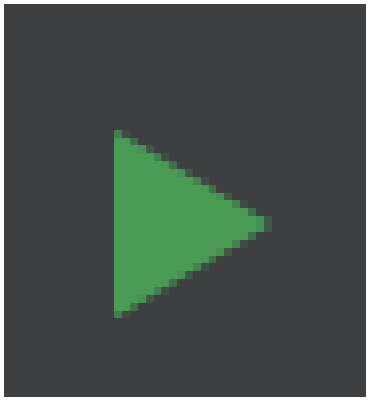


Figure 3: Run the application

The application should launch on the emulator. The application has four screens; the splash screen, the login screen, the sign up screen and the speech to text screen.

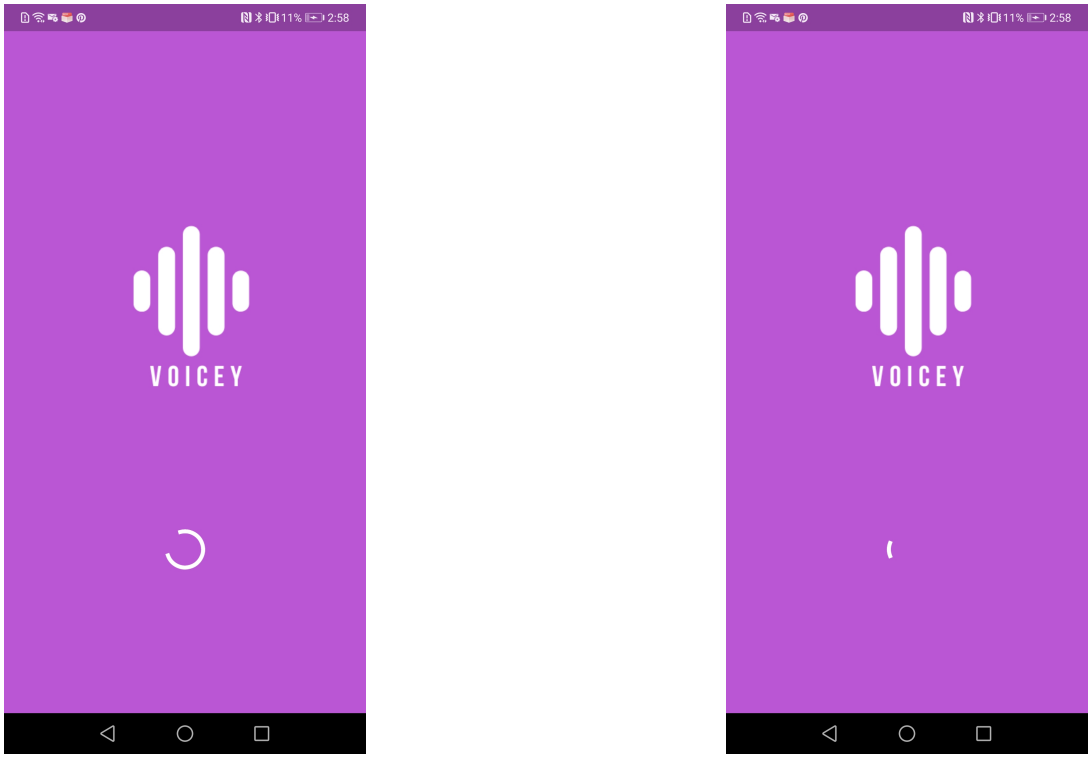


Figure 4: Splash Screen of the application

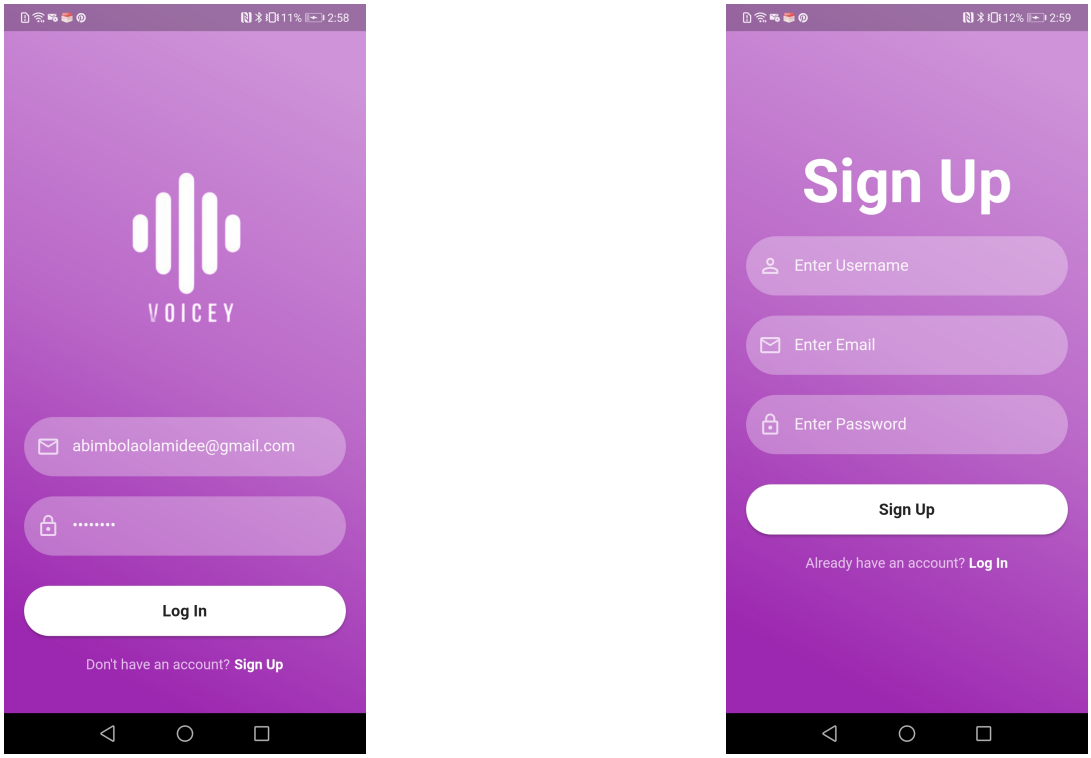


Figure 5: Login Screen and Sign Up Screen

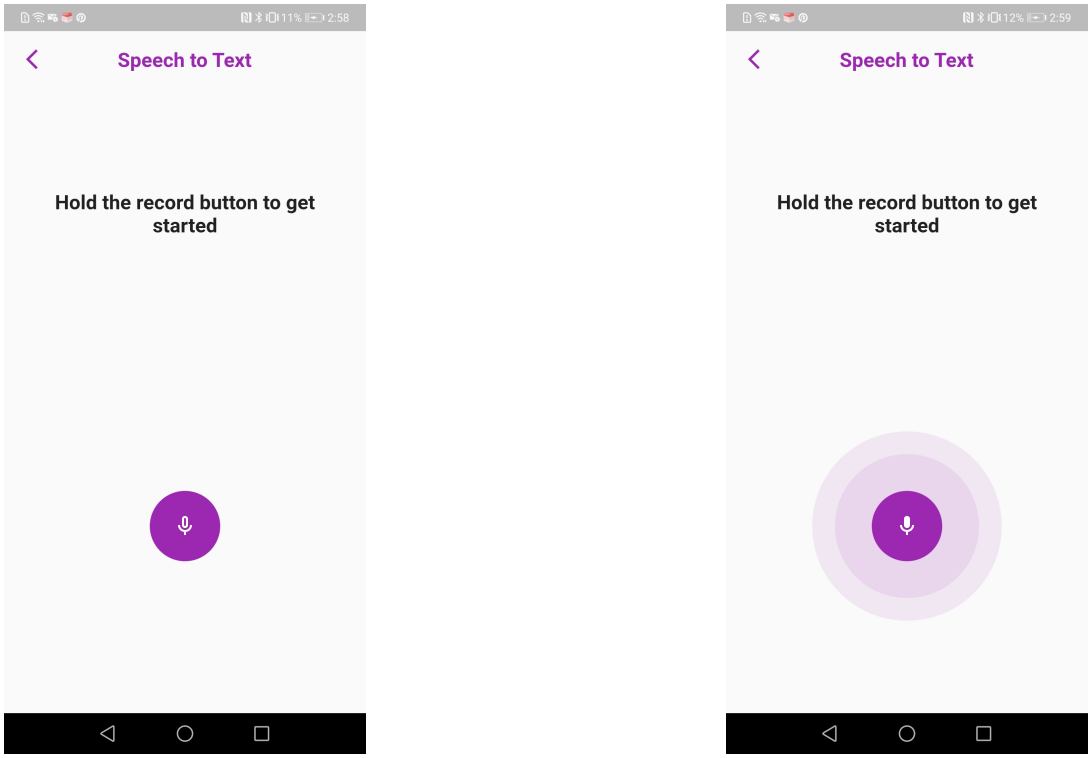


Figure 6: Speech to Text Screen

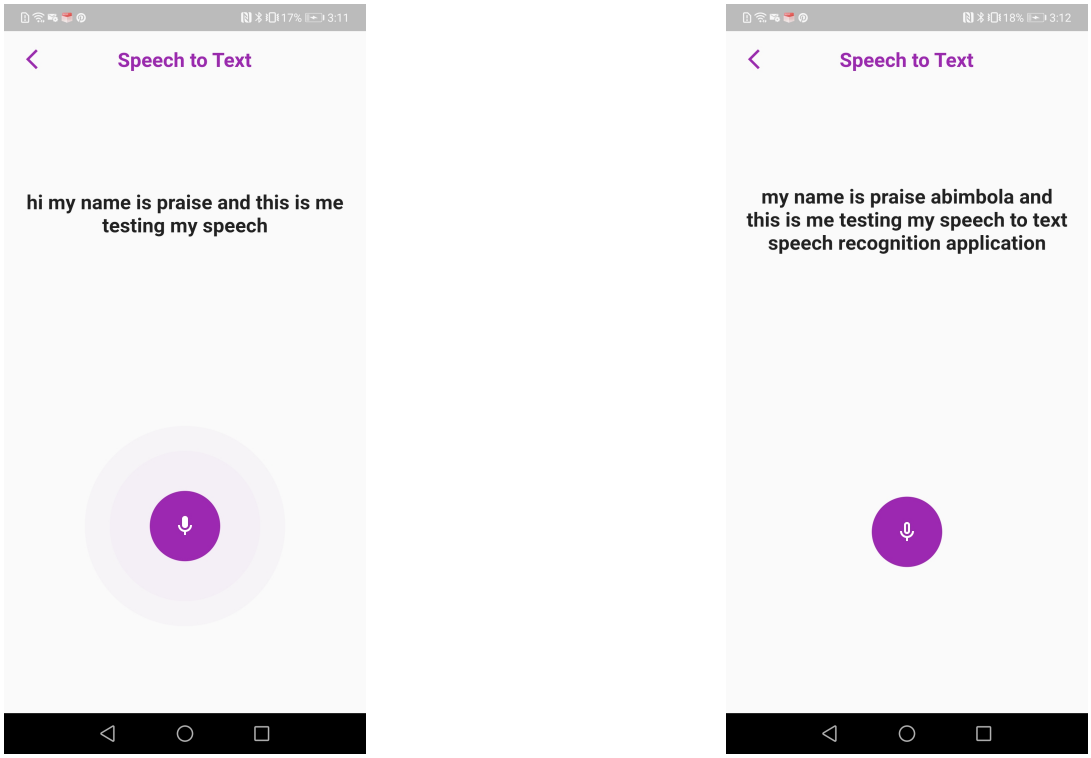


Figure 7: Speech to Text Screen

To test the application, use the microphone button to record speech. The application will transcribe the speech to text and display it on the screen in real-time.

4.1 Firebase for user authentication

The authentication of every registered user is stored on firebase.

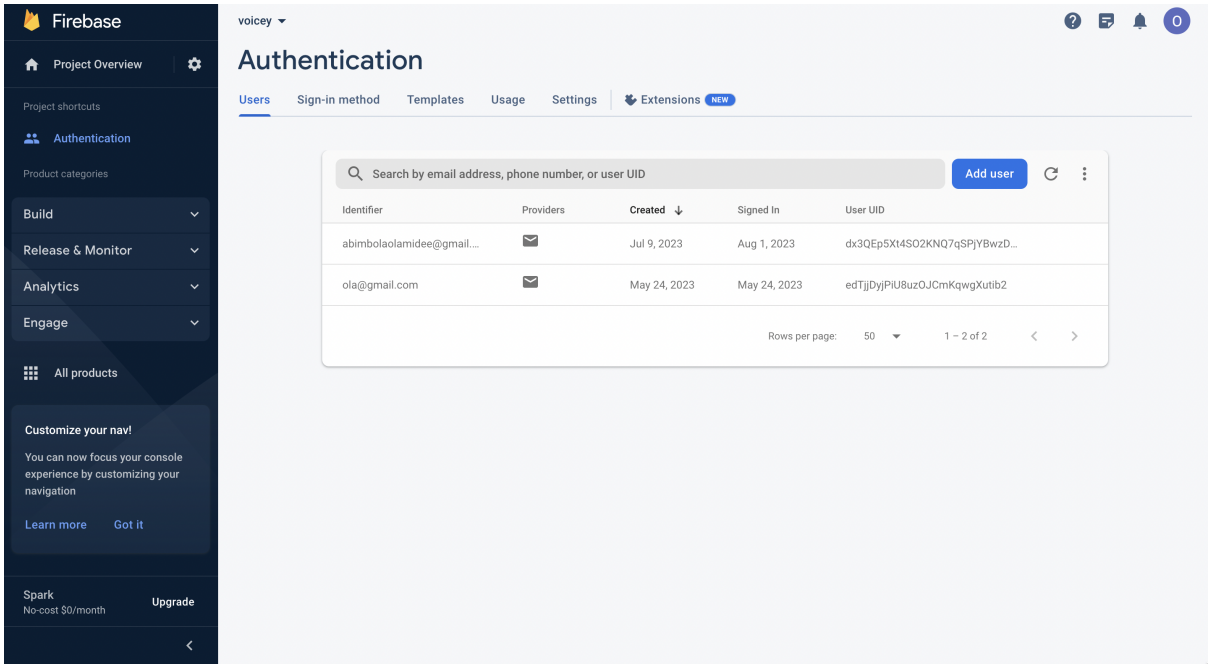
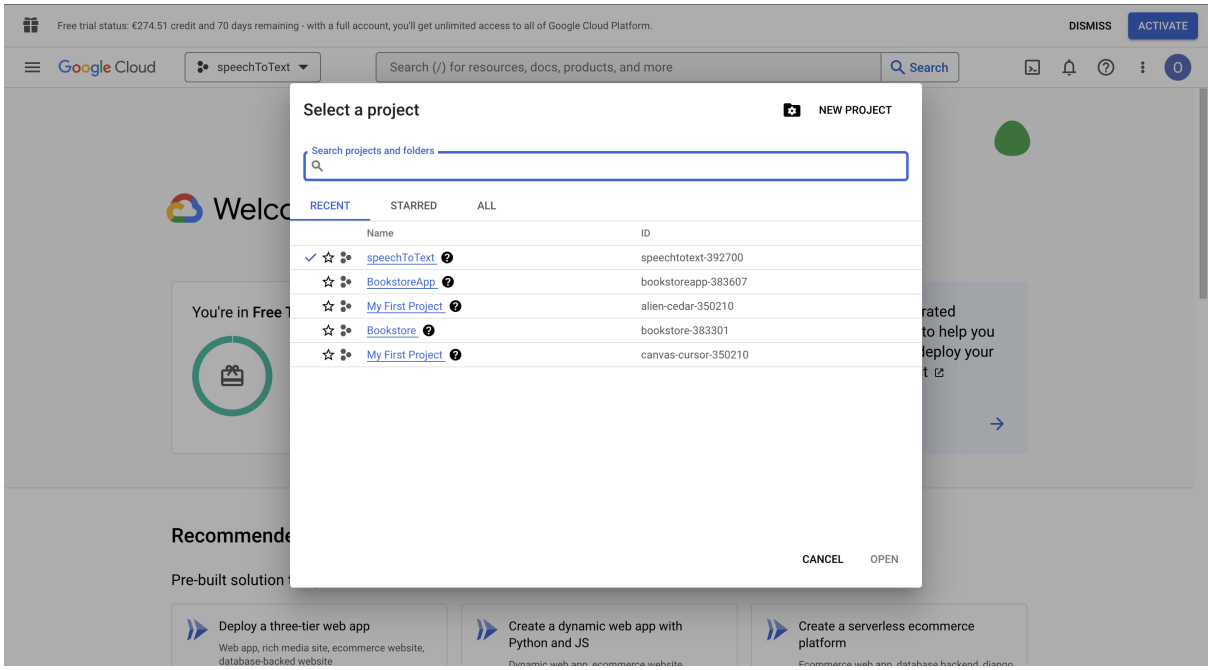


Figure 8: Firebase authentication

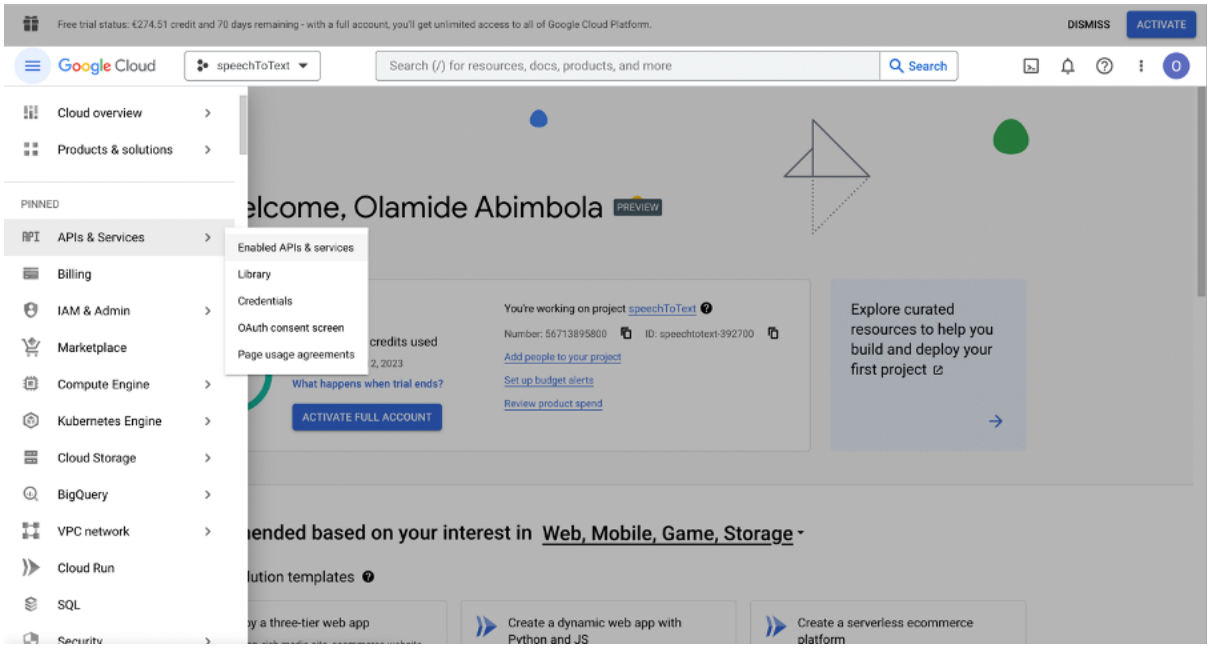
4.2 Google speech API

The google speech API is called from google cloud console. To set this up, follow the steps below;

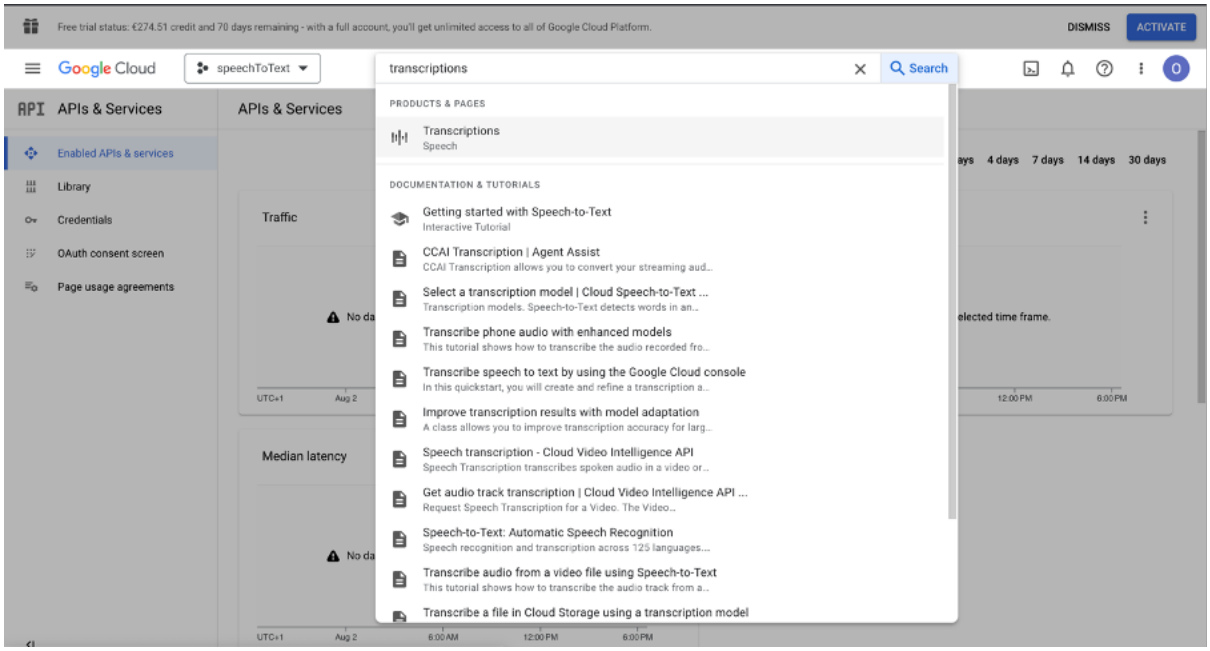
1. Create a new project



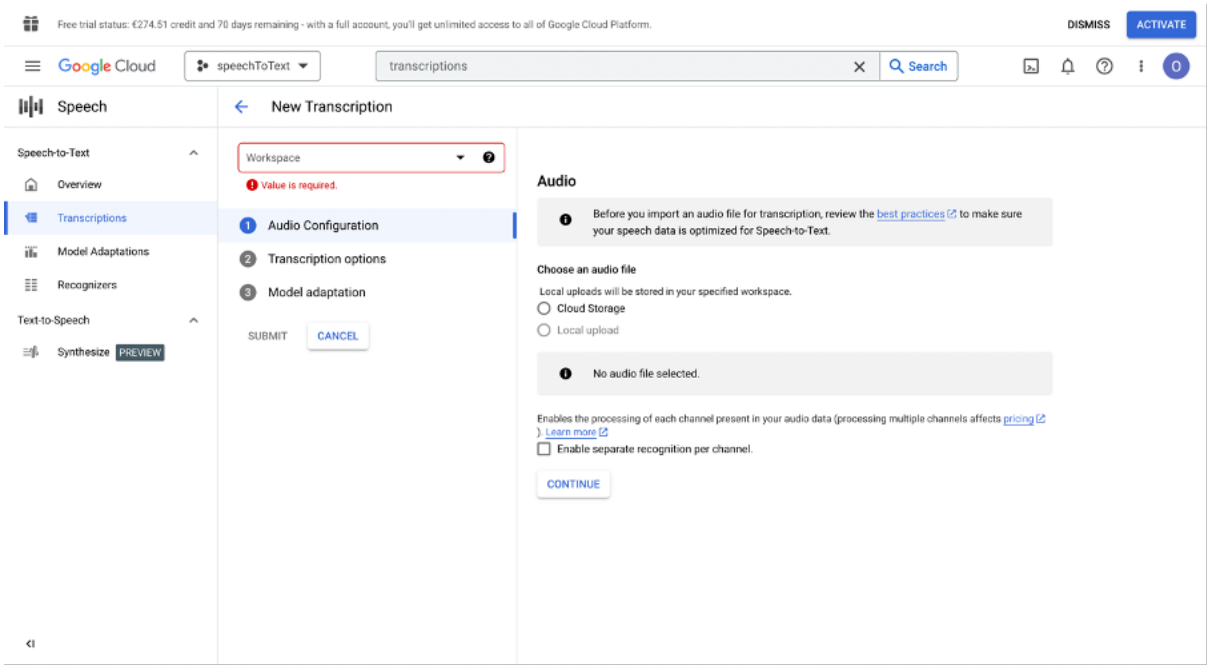
2. Navigate to the APIs and Services and select Enabled APIs and Services to enable the google speech API



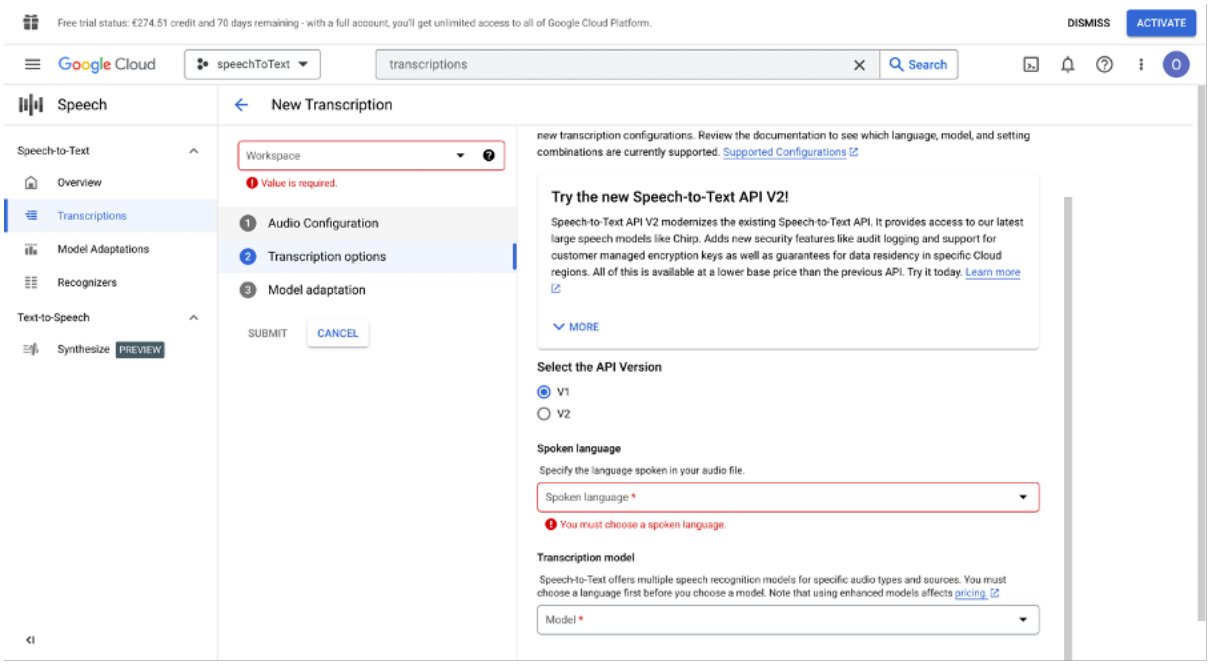
3. Search for the transcription API and select it



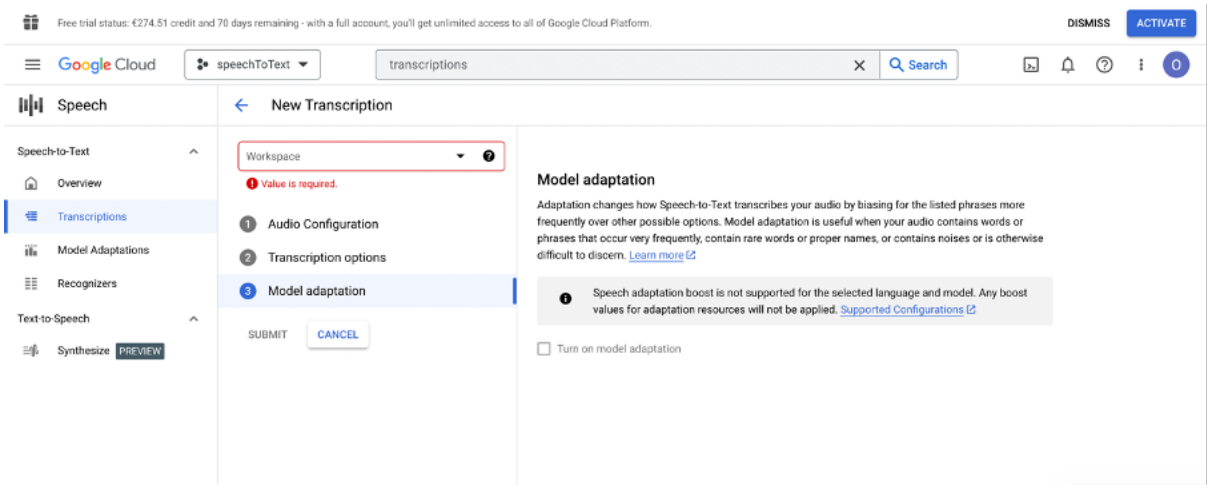
4. Select the preferred option for Audio configuration



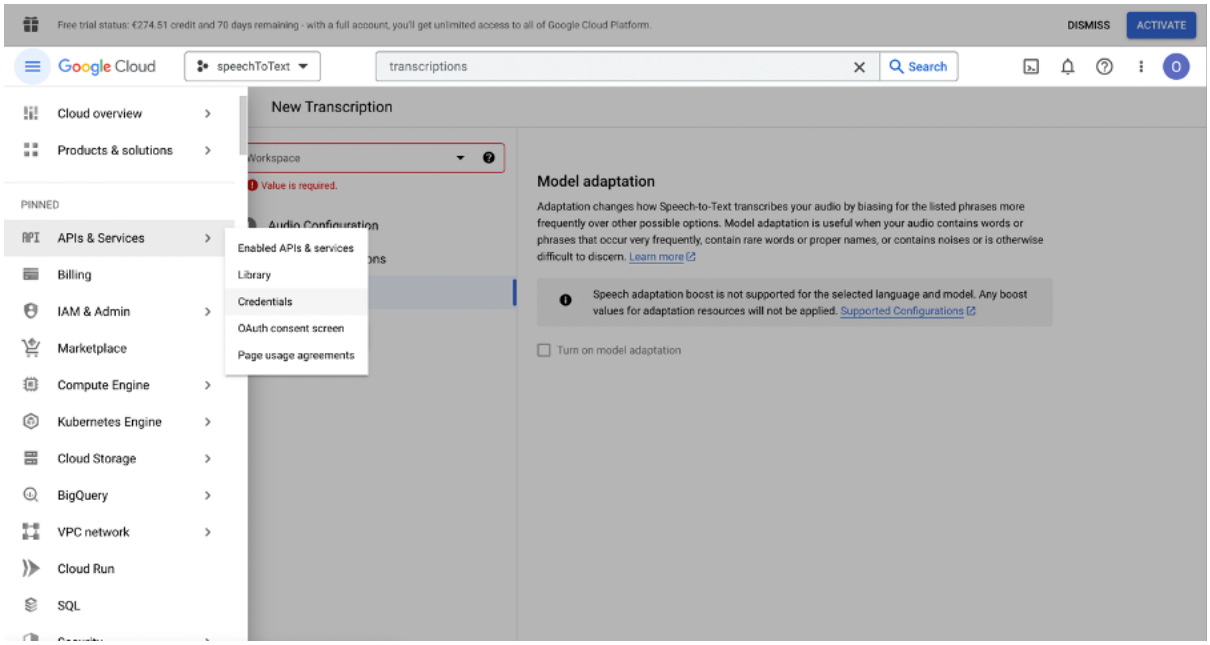
5. Select and fill the forms for Transcription options



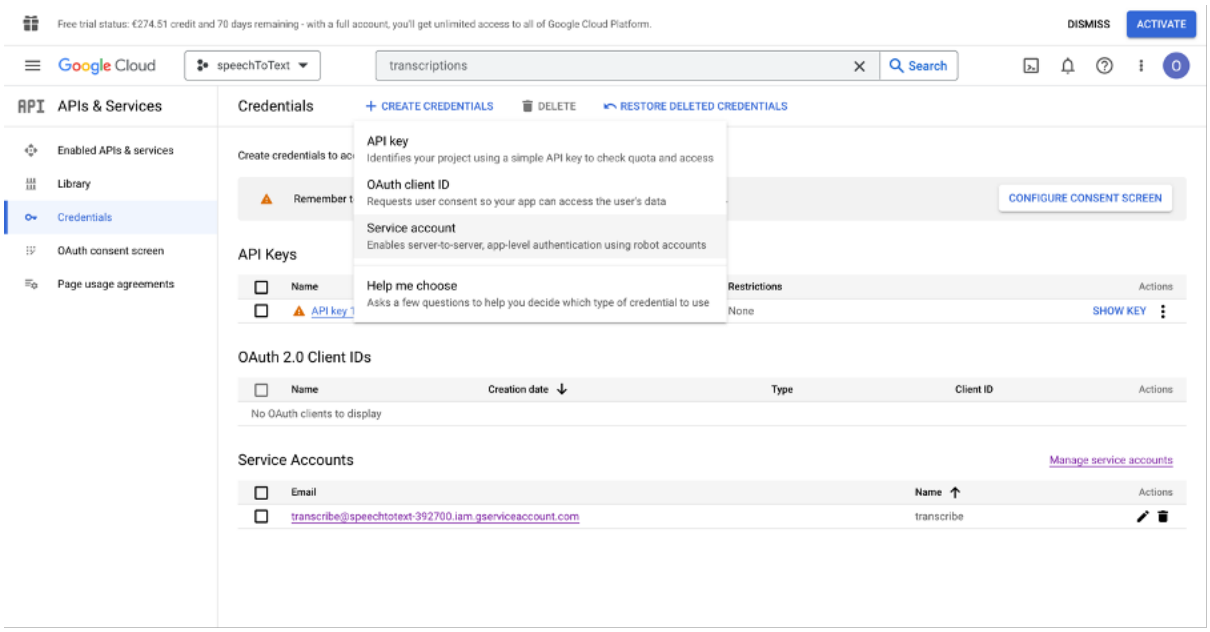
6. Select the preferred option for Model adaptation



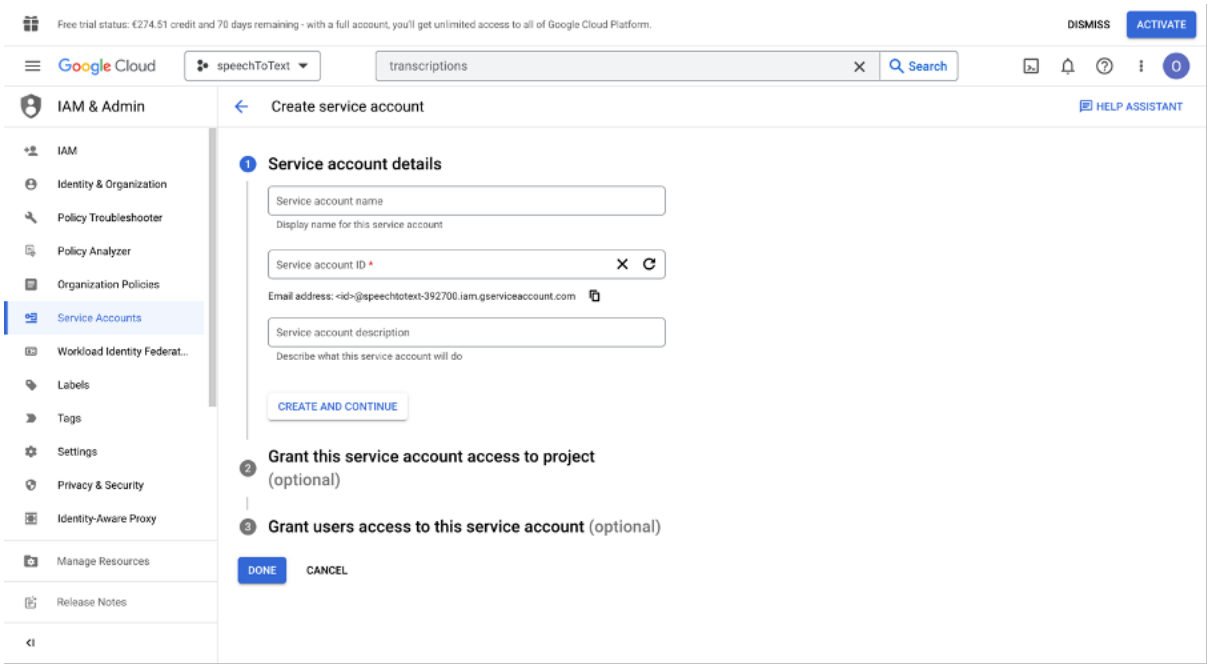
7. Navigate back to the console, select APIs and Services then select credentials



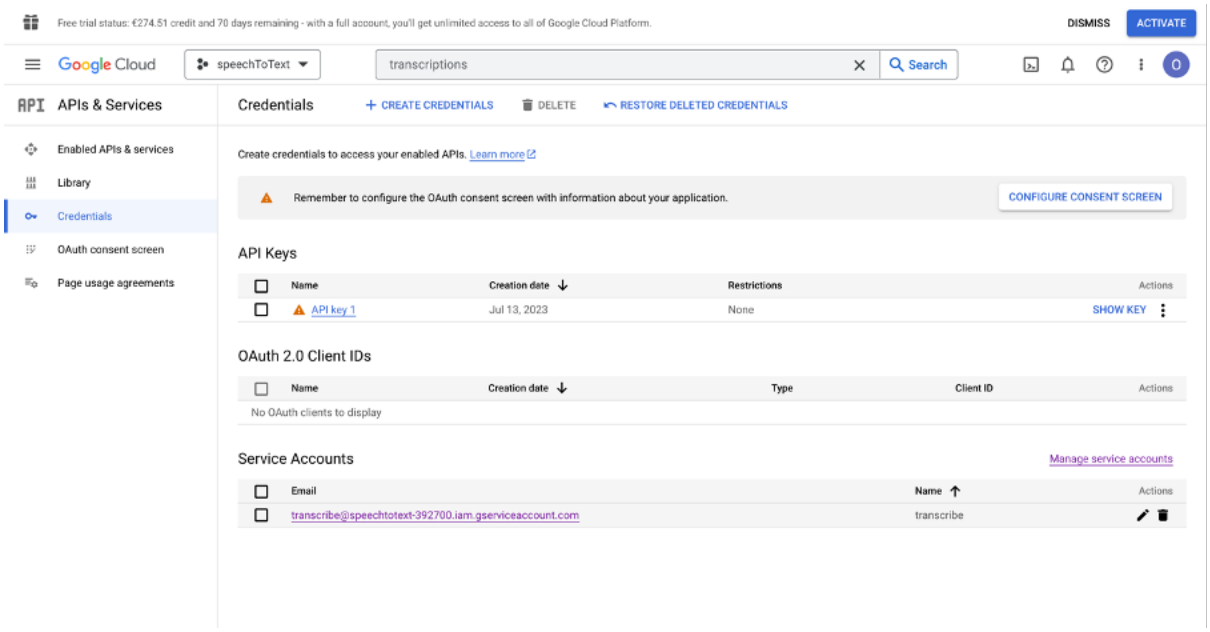
8. Click on Create credentials and select Service account



9. Fill in the details as instructed



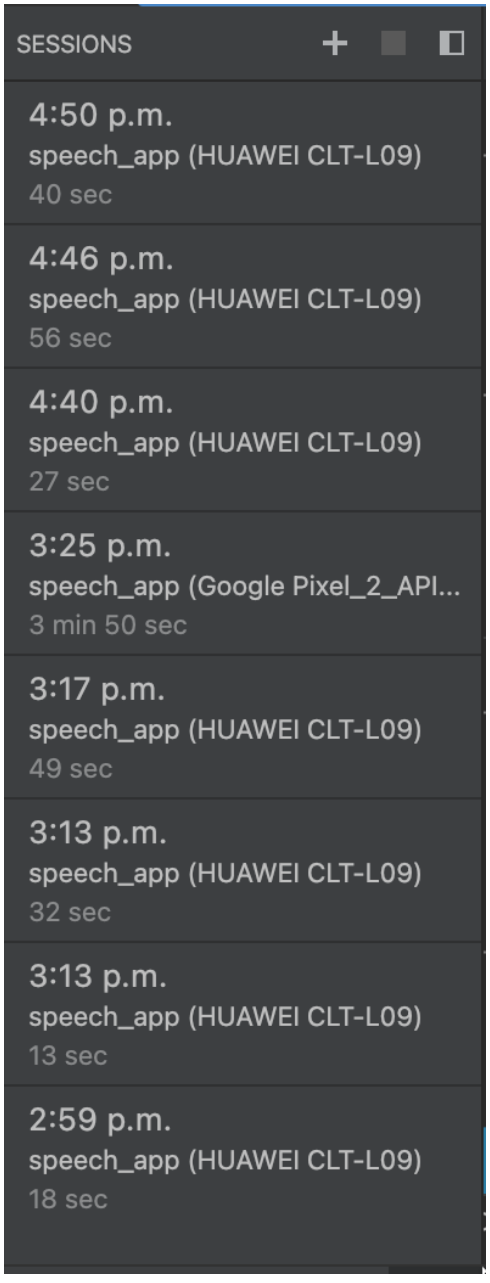
10. The created JSON key will be shown under service accounts.



5 Performance Monitoring

In order to capture performance of experiments, Android Profiler tool is used. The profiler can be opened by navigating to the bottom menu bar and select the device name.

Multiple sessions can be added with multiple devices to test the application and measure the resources used.



The results are shown in Figure 9 and Figure 10 below.

The image Figure 9 illustrates the resource usage (CPU, memory, and energy level) when running the application locally on CLT L09.

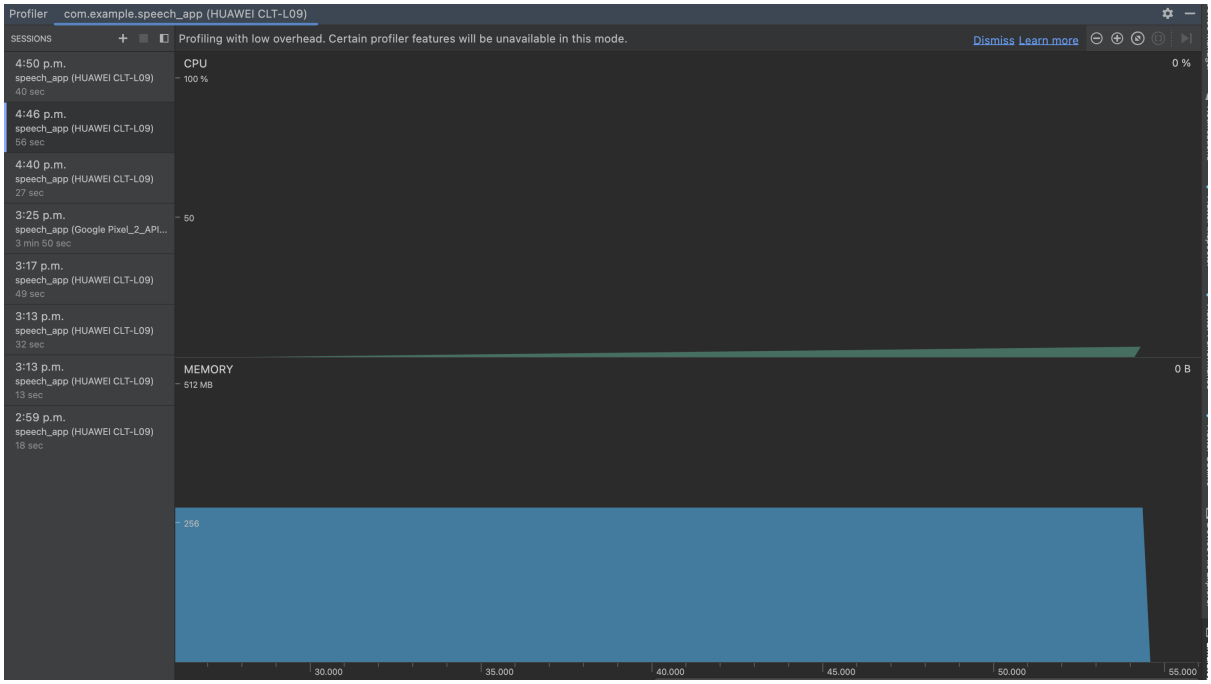


Figure 9: Android Studio Profiler - CPU, Memory and Energy usage

The image in Figure 10 illustrate the resource usage (CPU, memory, and energy level) when running the application locally on Pixel 2 API 33.

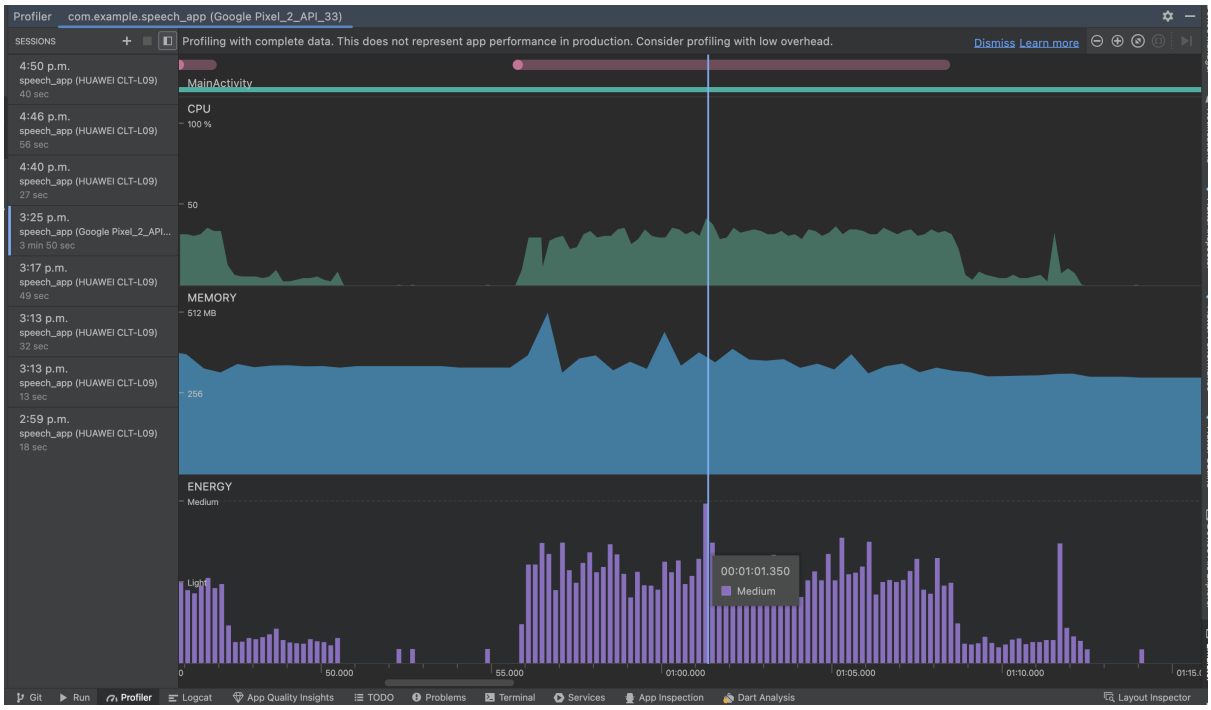


Figure 10: Android Studio Profiler - CPU, Memory and Energy usage