

Projects are Individual

TAs and the Instructor must be granted access to your cloud projects and source for grading.

Project I

The objective of this first project will be focused on options for leveraging spoken language as an interface to computers..

You will create a simple web application in python to record from the user's microphone and upload the recording to the server. And, receive a text input and generate audio from the user's input that can be listened to in the browser. Your python application must:

- present a html to the user with options to record and listen to the audio files
- have a text input field and use the text to generate audio by leveraging Google's text to speech API.
- take the uploaded recording and generate a text transcript by leveraging Google's speech to text API
- recommended (but optional) - your app should be deployed to cloud run and be available in the cloud

Provide a report of your application, architecture, code and design decisions, with a focus on what you learned.

Project II

Build upon your knowledge from project I and make the following adjustments:

- Leverage the Language API in Google Cloud to evaluate the sentiment of the audio or text uploaded
- Display whether the text has a positive/neutral/negative connotation

Provide a report of your application, architecture, code and design decisions, with a focus on what you learned.

Project III

Build upon your knowledge from project II and make the following adjustments:

- Using the multimodal LLM APIs from Google Cloud, replace the previously used APIs with a single LLM API
- Your app should, in a single call, ask for a transcript and the sentiment analysis of the uploaded audio. The response from the LLM should be presented as an audio response by sending it to the TTS API
- The option for text input is no longer required and should be removed.

Provide a report of your application, architecture, code and design decisions, with a focus on what you learned.

Course Project

Photo Storage Application

Using Google Cloud's Dialogflow, build an agent that can answer questions and can carry a conversation about a book.

Download a pdf book from the Project Gutenberg [website](#). I recommend something from the [top 100 from the last 30 days](#).

Provide a report of your application, architecture, code and design decisions, with a focus on what you learned. Include in your report, link to the chosen book, with sample questions and the generated answers based on the book you chose. Make sure to add screenshots of the configurations changed as well as an explanation for the reason for changing them to the values/settings chosen.