My Solution for Sainya Ranakshetram AI challenge

1. How to View my Solution

A. README.md

This is a README.md file for my solution for sainya-ranakshetram ai challenge. This README.md file is written in markdown format. You can read more about Markdown format here. There are Two ways in which you can read this README.md file

Option 1: Read this README.md file on using grip (GitHub markdown previewer)

To render this readme.md , open the terminal and cd into this directory and run the following command in a bash shell:

```
$ grip
```

which will give the following output:

```
* Serving Flask app 'grip.app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use
a production WSGI server instead.
* Running on http://localhost:6419
Press CTRL+C to quit
```

Click on the link http://localhost:6419 to view the rendered README.md file.

Incase you are running this solution on a remote server, you can forward the port 6419 to a remote tunnel using cloud-flared tunneling service. To do so, run the following command in a bash shell:

```
$ cloudflared tunnel --url http://localhost:6419
```

this will give the following output:

```
2022-12-22T11:28:34Z INF Thank you for trying Cloudflare Tunnel. Doing so, without a
Cloudflare account, is a quick way to experiment and try it out. However, be aware
that these account-less Tunnels have no uptime guarantee. If you intend to use
Tunnels in production you should use a pre-created named tunnel by following:
https://developers.cloudflare.com/cloudflare-one/connections/connect-apps
2022-12-22T11:28:34Z INF Requesting new quick Tunnel on trycloudflare.com...
2022-12-22T11:28:36Z INF +----------------------------------------------------------
----------------------------------+
2022-12-22T11:28:36Z INF |  Your quick Tunnel has been created! Visit it at (it may
take some time to be reachable):  |
2022-12-22T11:28:36Z INF |  Url Here (unique url will be created here every time)
|
2022-12-22T11:28:36Z INF +----------------------------------------------------------
----------------------------------+
2022-12-22T11:28:36Z INF Version 2022.12.1
2022-12-22T11:28:36Z INF GOOS: linux, GOVersion: go1.19.3, GoArch: amd64
2022-12-22T11:28:36Z INF cloudflared will not automatically update if installed by a
package manager.
2022-12-22T11:28:36Z INF Generated Connector ID: a3eea567-7fe4-4f24-bbab-
eaba6e003265
2022-12-22T11:28:36Z INF Initial protocol quic
2022-12-22T11:28:36Z INF ICMP proxy will use 10.42.32.18 as source for IPv4
2022-12-22T11:28:36Z INF ICMP proxy will use :: as source for IPv6
2022-12-22T11:28:36Z INF Starting metrics server on 127.0.0.1:42055/metrics
```

The Url Here area will have your unique url. Click on the link to view the rendered README.md file.

Option 2: Read README.pdf

You can open a rendered pdf of README.md By opening the file README.pdf in this directory.

B. Video

You can also view the video of my solution here ## ADD vidoe link here

2. How to Run my soltion

Step 1 : Make sure All requirements are installed

Docker

To check this run the following command in a bash shell

```
$ docker --version
```

If this command runs successfully then you have docker installed on your system. If not then install docker using the following command

```
$ bash install_docker.sh
```

Which will install docker on your system

Compute Requirements

This Repo requires an Nvidia GPU with a minimum of 10GB of memory to run to fit the transcription model.

Audio File Requirements

The Audio File passes should be in either of the following formats:

- .wav
- .mp3
- .m4a
- .flac
- .ogg
- .aac
- .avi

more might be supported but these are the ones that I have tested.

Step 2 : Clone the Docker container

To clone the docker container run the following command in a bash shell

```
$ sudo docker pull mithilaidocker/audiotranscribe:master
```

Step 3 : Run the Docker container

To run the docker container run the following command in a bash shell

```
sudo docker run --gpus all --ipc=host --ulimit memlock=-1 --net="host" --ulimit
stack=67108864 -it -v "/home/":/home \
--rm mithilaidocker/audiotranscribe:master
```

By running this command you will enter the docker container.

Step 4 : Run the Solution

Option 1 : Using the GUI in the form of a Flask Web Server

Step 1 : Run the Flask Web Server

To Run the flask app for the solution run the following command in a bash shell.(Make sure you are in the `/app` dir)

```
root@xx:/app#  python -m flask run --host= 0.0.0.0
```

This will run the flask app which contains the solution. The command will give the following output

```
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a
production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.42.32.18:5000
Press CTRL+C to quit
```

If you are running on the same machine as the server then you can access the solution at http://127.0.0.1:5000. In case you running this solution on a remote server you will need to forward the port 5000 to your local machine. To do this we can use cloudfared tunnel (*already installed on the* *docker image*) to forward the port 5000 to our local machine. To do this run the following command in a bash shell

```
$ cloudflared tunnel --url http://127.0.0.1:5000
```

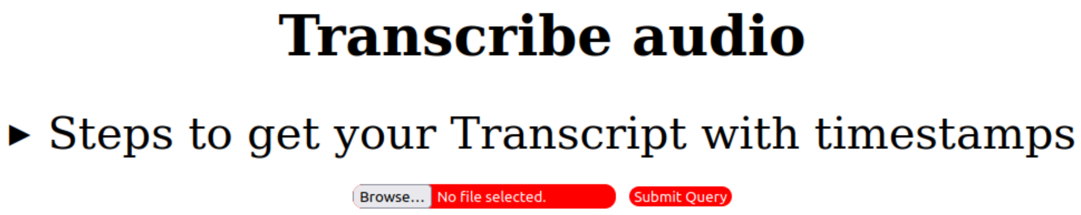
Which will Give the following output

```
2022-12-22T11:28:34Z INF Thank you for trying Cloudflare Tunnel. Doing so, without a
Cloudflare account, is a quick way to experiment and try it out. However, be aware
that these account-less Tunnels have no uptime guarantee. If you intend to use
Tunnels in production you should use a pre-created named tunnel by following:
https://developers.cloudflare.com/cloudflare-one/connections/connect-apps
2022-12-22T11:28:34Z INF Requesting new quick Tunnel on trycloudflare.com...
2022-12-22T11:28:36Z INF +----------------------------------------------------------
----------------------------------+
2022-12-22T11:28:36Z INF |  Your quick Tunnel has been created! Visit it at (it may
take some time to be reachable):  |
2022-12-22T11:28:36Z INF |  Url Here (unique url will be created here every time)
|
2022-12-22T11:28:36Z INF +----------------------------------------------------------
----------------------------------+
2022-12-22T11:28:36Z INF Version 2022.12.1
2022-12-22T11:28:36Z INF GOOS: linux, GOVersion: go1.19.3, GoArch: amd64
2022-12-22T11:28:36Z INF Settings: map[protocol:quic url:http://127.0.0.1:5000]
2022-12-22T11:28:36Z INF cloudflared will not automatically update if installed by a
package manager.
2022-12-22T11:28:36Z INF Generated Connector ID: a3eea567-7fe4-4f24-bbab-
eaba6e003265
2022-12-22T11:28:36Z INF Initial protocol quic
2022-12-22T11:28:36Z INF ICMP proxy will use 10.42.32.18 as source for IPv4
2022-12-22T11:28:36Z INF ICMP proxy will use :: as source for IPv6
2022-12-22T11:28:36Z INF Starting metrics server on 127.0.0.1:42055/metrics
```

Click on the link in the area of the output that says `Url Here` to view the solution. You will have your own unique url every time ***NOTE:*** The Cloudflare tunnel is only a quick way to access the solution. It is not a production ready solution. If you want to use this solution in production you should use a pre-created named tunnel by following: https://developers.cloudflare.com/cloudflare-one/connections/connect-apps

Step 2 : How to use the Flask App

Upon opening the url you will be greeted with the following page

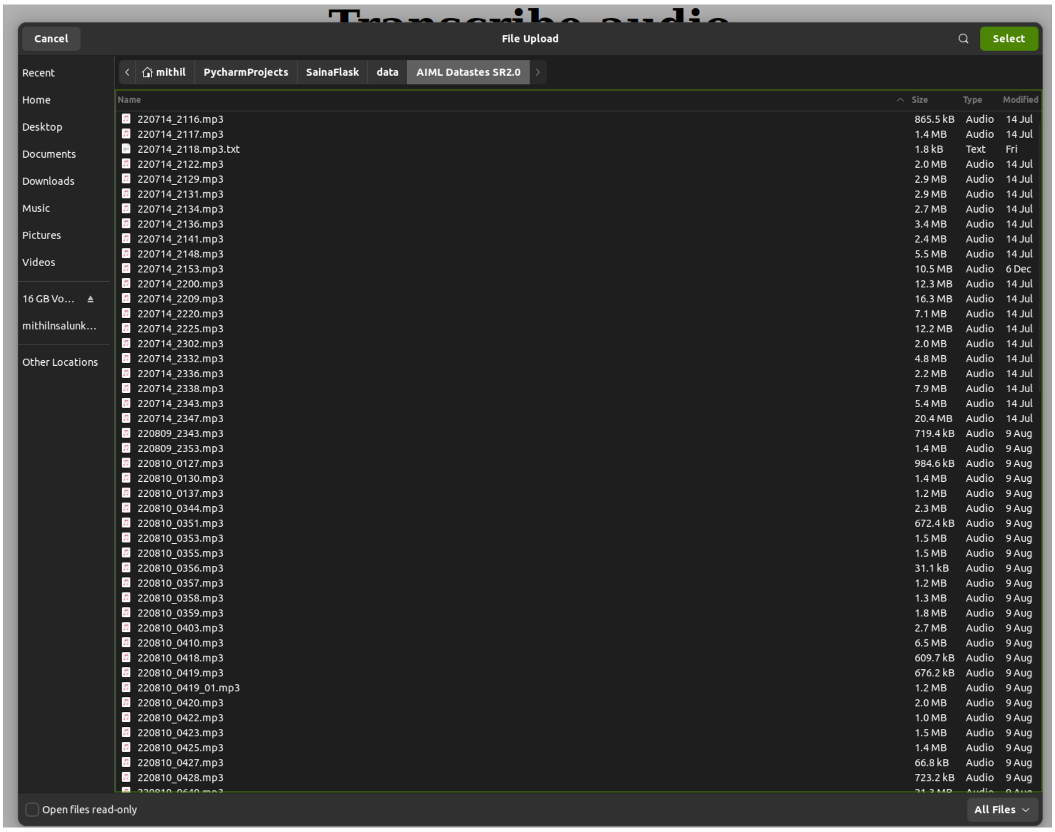


Transcribe audio

▶ Steps to get your Transcript with timestamps

Steps To Transcribe The Audio File are following from here

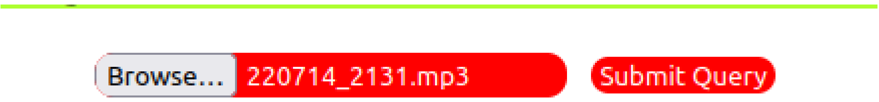
1. Click on the `Choose File` or the `Browse` button to select the audio file you want to transcribe. The following window will open on Linux



Name	Size	Type	Modified
220714_2116.mp3	865.5 kB	Audio	14 Jul
220714_2117.mp3	1.4 MB	Audio	14 Jul
220714_2118.mp3.txt	1.8 kB	Text	Fri
220714_2122.mp3	2.0 MB	Audio	14 Jul
220714_2129.mp3	2.9 MB	Audio	14 Jul
220714_2131.mp3	2.9 MB	Audio	14 Jul
220714_2134.mp3	2.7 MB	Audio	14 Jul
220714_2136.mp3	3.4 MB	Audio	14 Jul
220714_2141.mp3	2.4 MB	Audio	14 Jul
220714_2148.mp3	5.5 MB	Audio	14 Jul
220714_2153.mp3	10.5 MB	Audio	6 Dec
220714_2200.mp3	12.3 MB	Audio	14 Jul
220714_2209.mp3	16.3 MB	Audio	14 Jul
220714_2220.mp3	7.1 MB	Audio	14 Jul
220714_2225.mp3	12.2 MB	Audio	14 Jul
220714_2302.mp3	2.0 MB	Audio	14 Jul
220714_2332.mp3	4.8 MB	Audio	14 Jul
220714_2336.mp3	2.2 MB	Audio	14 Jul
220714_2338.mp3	7.9 MB	Audio	14 Jul
220714_2343.mp3	5.4 MB	Audio	14 Jul
220714_2347.mp3	20.4 MB	Audio	14 Jul
220809_2343.mp3	719.4 kB	Audio	9 Aug
220809_2353.mp3	1.4 MB	Audio	9 Aug
220810_0127.mp3	984.6 kB	Audio	9 Aug
220810_0130.mp3	1.4 MB	Audio	9 Aug
220810_0137.mp3	1.2 MB	Audio	9 Aug
220810_0344.mp3	2.3 MB	Audio	9 Aug
220810_0351.mp3	672.4 kB	Audio	9 Aug
220810_0353.mp3	1.5 MB	Audio	9 Aug
220810_0355.mp3	1.5 MB	Audio	9 Aug
220810_0356.mp3	31.1 kB	Audio	9 Aug
220810_0357.mp3	1.2 MB	Audio	9 Aug
220810_0358.mp3	1.3 MB	Audio	9 Aug
220810_0359.mp3	1.8 MB	Audio	9 Aug
220810_0403.mp3	2.7 MB	Audio	9 Aug
220810_0410.mp3	6.5 MB	Audio	9 Aug
220810_0418.mp3	609.7 kB	Audio	9 Aug
220810_0419.mp3	676.2 kB	Audio	9 Aug
220810_0419_01.mp3	1.2 MB	Audio	9 Aug
220810_0420.mp3	2.0 MB	Audio	9 Aug
220810_0422.mp3	1.0 MB	Audio	9 Aug
220810_0423.mp3	1.5 MB	Audio	9 Aug
220810_0425.mp3	1.4 MB	Audio	9 Aug
220810_0427.mp3	66.8 kB	Audio	9 Aug
220810_0428.mp3	723.2 kB	Audio	9 Aug

2. Select the audio file which are in the supported audio formats you want to transcribe and click on `Select` (or any other button you get depending upon your os).

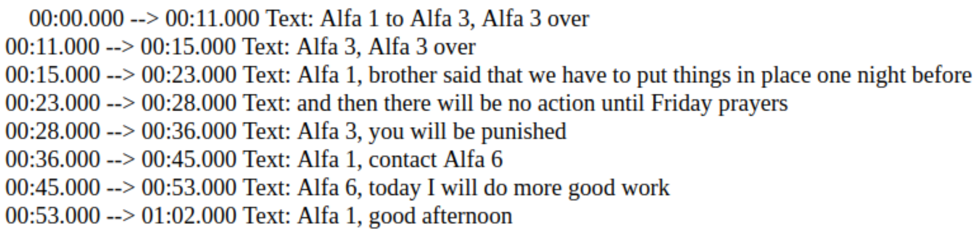
3. The file name will be displayed in the `File Name` text box Example -



4. Click on the `Submit Query` button to start the transcription process. ***NOTE:*** The Submit Query Button can have a different name depending upon the browser you are using. For example in Firefox it is `Submit Query` but in Chrome it is `Submit`

5. The transcription process will take some time depending upon the length of the audio file and the type of GPU you have. It is important not close the web page once clicking upon the `Submit Query` button. Once the process is complete you will be greeted with the result page. Let us take an example of the following audio file. 221001_0134.mp3

6. The result page will look like this for the following audio file is transcribed



00:00.000 --> 00:11.000 Text: Alfa 1 to Alfa 3, Alfa 3 over
00:11.000 --> 00:15.000 Text: Alfa 3, Alfa 3 over
00:15.000 --> 00:23.000 Text: Alfa 1, brother said that we have to put things in place one night before
00:23.000 --> 00:28.000 Text: and then there will be no action until Friday prayers
00:28.000 --> 00:36.000 Text: Alfa 3, you will be punished
00:36.000 --> 00:45.000 Text: Alfa 1, contact Alfa 6
00:45.000 --> 00:53.000 Text: Alfa 6, today I will do more good work
00:53.000 --> 01:02.000 Text: Alfa 1, good afternoon

7. The Transcript will be saved in the `/home/transcripts` directory in the docker container. The file name will be the same as the audio file name with the extension `.txt`. So for the above example the transcript will be saved in the `/home/transcripts/221001_0134.txt` file.

8. You can print the txt file following command in the docker container

```
root@xx:/app# cat /home/transcripts/YourAudioFile.txt
```

With the 221001_0134.txt file being the name of the audio file you want to transcribe.

Option 2 : Using the CLI

Using the CLI is much more straightforward. To run the CLI for the solution run the following command in a bash shell.(Make sure you are in the `/app` dir)

```
root@xx:/app#  python model.py --path your_audio_file
```

Let us the Same of the file we did above for the flask App 221001_0134.mp3 so here the command will be

```
root@xx:/app#  python model.py --path 221001_0134.mp3
```

which would give us the following output

```
Reducing Noise
/opt/conda/lib/python3.9/site-packages/librosa/util/decorators.py:88: UserWarning:
PySoundFile failed. Trying audioread instead.
  return f(*args, **kwargs)
Transcribing... 221001_0134.mp3
Detecting language using up to the first 30 seconds. Use `--language` to specify the
```

```
language
Detected language: Urdu
[00:00.000 --> 00:11.000]  Alfa 1 to Alfa 3, Alfa 3 over
[00:11.000 --> 00:15.000]  Alfa 3, Alfa 3 over
[00:15.000 --> 00:23.000]  Alfa 1, brother said that we have to put things in place
one night before
[00:23.000 --> 00:28.000]  and then there will be no action until Friday prayers
[00:28.000 --> 00:36.000]  Alfa 3, you will be punished
[00:36.000 --> 00:45.000]  Alfa 1, contact Alfa 6
[00:45.000 --> 00:53.000]  Alfa 6, today I will do more good work
[00:53.000 --> 01:02.000]  Alfa 1, good afternoon
Transcription complete. Saved it to /home/transcripts/221001_0134.txt
```

the transcript will be saved in the `/home/transcripts` directory in the docker container. The file name will be the same as the audio file name with the extension `.txt`. So for the above example the transcript will be saved in the `/home/transcripts/221001_0134.txt` file. You can print the txt file following command in the docker container

```
root@xx:/app# cat /home/transcripts/YourAudioFile.txt
```

so here the YourAudioFile.txt will be the name of the audio file we are transcribing so for the above example it will be 221001_0134.txt