

FIT3162

User Guides

Authors:

- 1. Toh Xi Heng (33200548)**
- 2. Yew Yee Perng (32205481)**
- 3. Yeoh Ming Wei (32205449)**

End User Guide.....	4
1.0 How to access the software.....	4
End User Prerequisites.....	4
Browser Compatibility.....	4
2.0 Limitations.....	5
3.0 Overview of Main Page.....	6
A) View of Main Page.....	6
b) Overview of Non-Interactive Sections in Home Page.....	7
c) Overview of Interactive Components/Features on Main page.....	8
4.0 How to Operate the Application.....	8
Technical Guide.....	12

Overview

This document will consist of two parts, the End User Guide and Technical Guide. They serve as a guide to instructions with steps on running and setting up the application as an End User.

End User Guide

End users of our application are content creators and influencers or even individuals. Our application aims to provide a platform for them to generate a singing video with their desired audio and image as inputs to the application. This application primarily takes in two inputs, an audio and an image as input uploaded by the End User and returns a video generated with the audio added to the image.

1.0 How to access the software

To access and use the software, refer to the Technical Guide below to setup the application before proceeding with this section's walkthrough to use the application as an End User. Upon launching the application, you will be brought to the Main Page. Do note that the software prerequisites stated below are to be followed to ensure a smooth experience with the application.

End User Prerequisites

To access the Main Page: the End User must meet the following prerequisites:

- Ensure all the command lines stated in Technical Guide are executed to ensure the application runs smoothly. This includes installation of packages and cloning of github projects.

Browser Compatibility

Upon launching the software, it will automatically open a new tab in your default web browser to access the application. The browsers that have been tested and are confirmed to be compatible include:

We recommend using Chrome Browser to ensure the best experience. This User Guide, including all instructions and screenshots, was created using Chrome to access the application.

2.0 Limitations

Our software has been designed with specific usage constraints to ensure optimal performance and user experience. Please note the following limitations:

- **Input Image Format:** This application only accepts only one image at a time, and the image must be in PNG format.
- **Input Audio Format:** This application only accepts only one audio at a time, and the audio must be in Wav or mp3 format.

It's important to adhere to these limitations to prevent any issues while using the software. If you encounter a situation where the input format does not meet these requirements, you may be directed and required to upload an acceptable input with the listed formats.

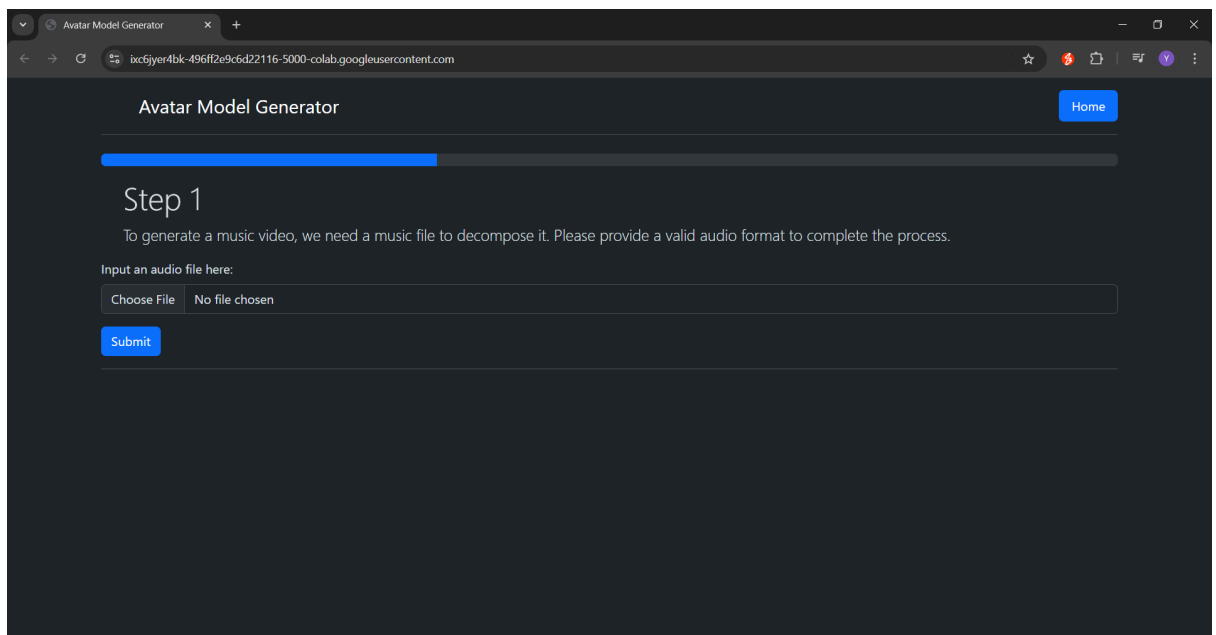
3.0 Overview of Main Page

The Main page serves as the primary interface of our software, divided into three sections to provide a basic understanding and navigation for the End Users.

- **A) View of Main Page:** This subsection offers a visual overview of the Main page as it appears right after the application is successfully launched.
- **B) Overview of Non-Interactive Sections on the Main Page:** This section describes the areas of the Main Page that remain static and do not allow or require user interaction. Familiarising yourself with these sections will help you better understand the information and results shown on the screen.
- **C) Overview of Interactive Sections on the Main Page:** This part covers all the interactive elements found on the Main Page. It explains the various actions you can perform, such as navigating menus, adjusting application settings for ease of use and accessibility, entering data, downloading, and managing results, ensuring you are informed about the available features and functions of the application.

Each section is created to provide a concise understanding of the application's interface to ensure a smooth and easy interaction.

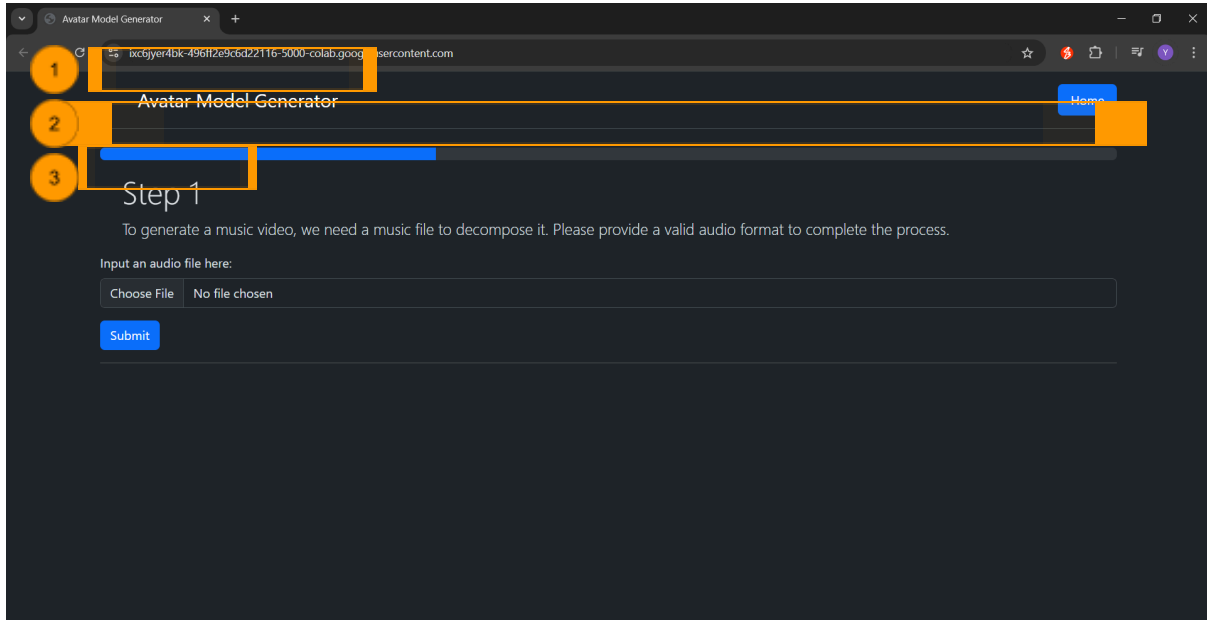
A) View of Main Page



Upon launching the application as by following the steps in the Technical Guide, the Main Page will appear in your default browser. Do take note that this guide is used in Chrome for demonstration purposes and might vary depending on your default browser.

b) Overview of Non-Interactive Sections in Home Page

This part of the guide focuses on the non-interactive components of the Main Page that provide essential information but do not require user interaction.

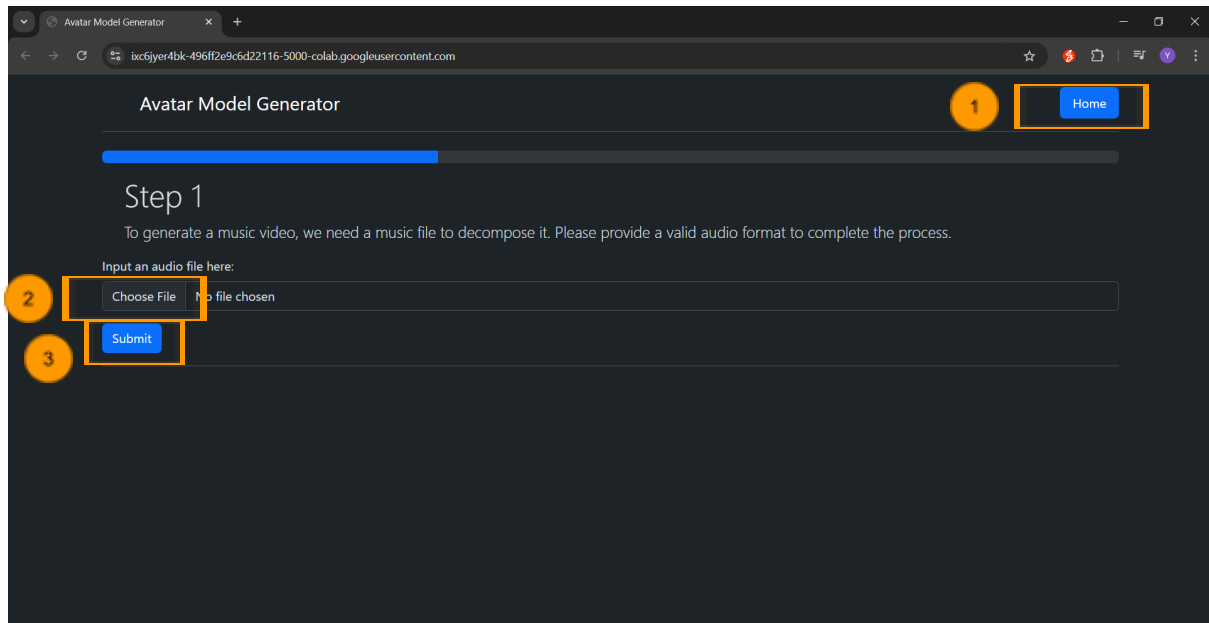


The Main Page of our application contains some static elements that do not require user interaction. These include:

1. A Header section that displays the name of the application.
2. A progress bar to show the approximate current status in process. This bar is static and updates automatically as you progress through the steps.
3. A step number and description below the progress bar to show the current step the End User is while using the application.

c) Overview of Interactive Components/Features on Main page

This section outlines the interactive features available on the Main Page, detailing the actions you can perform to fully engage with the application.



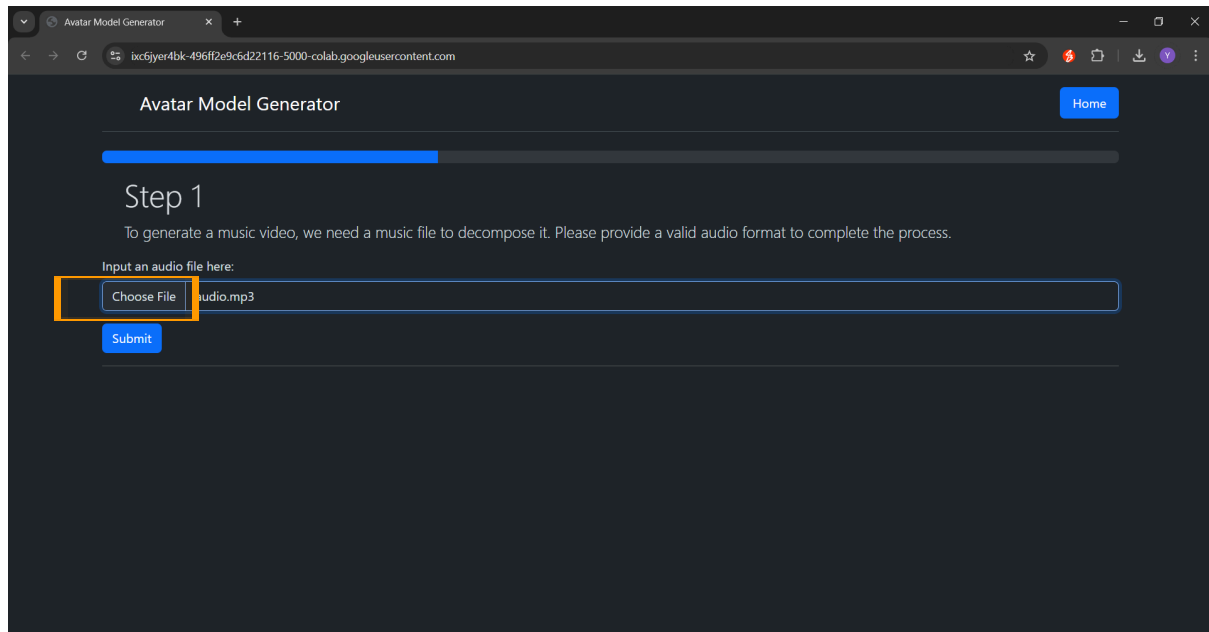
The interactive elements on the Main Page of our application includes:

1. A home button that navigates back to home when pressed.
2. A choose file button that allows the user to upload an audio file with the allowed format.
3. A submit button that enables the user to proceed when the audio file is uploaded.

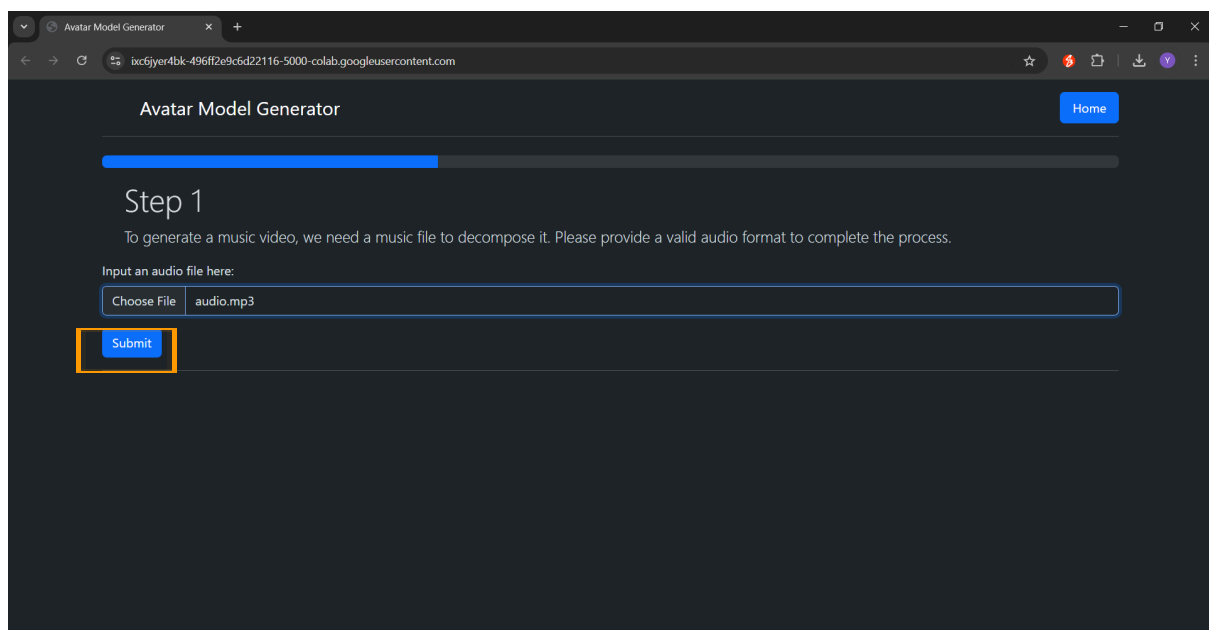
4.0 How to Operate the Application

This section is a guide with steps on how to use the application such as uploading an audio file or an image. Please do ensure that you have successfully launched the application and have arrived at the Main page.

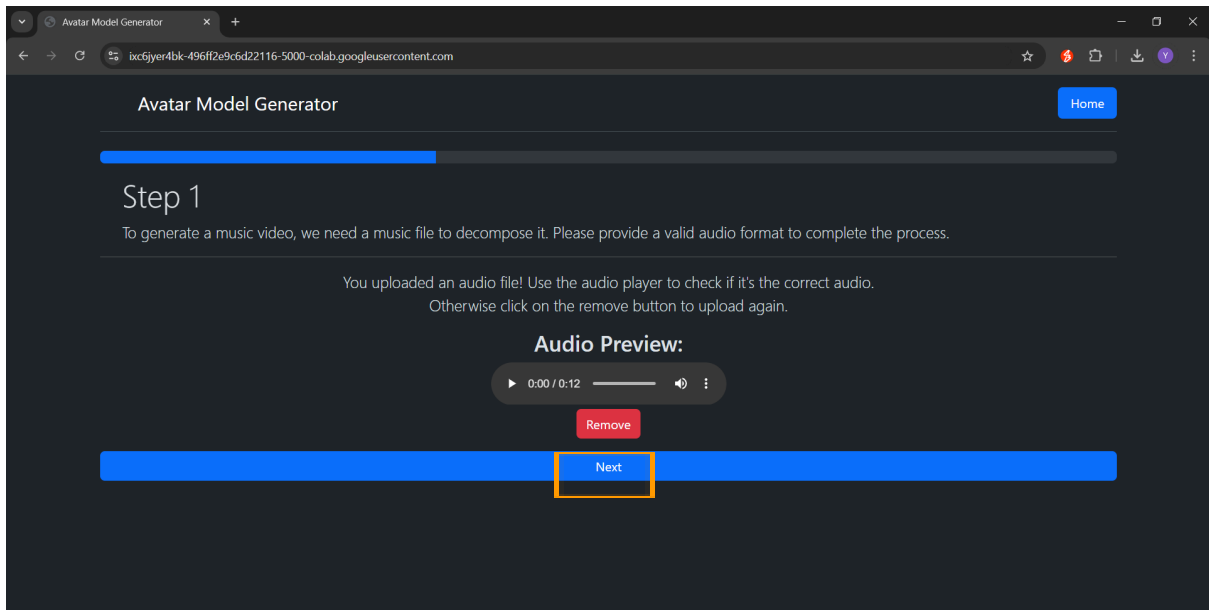
Step 1: Upload your audio file(wav/mp3 format). You are only required to upload an audio file in the format of .wav or .mp3.



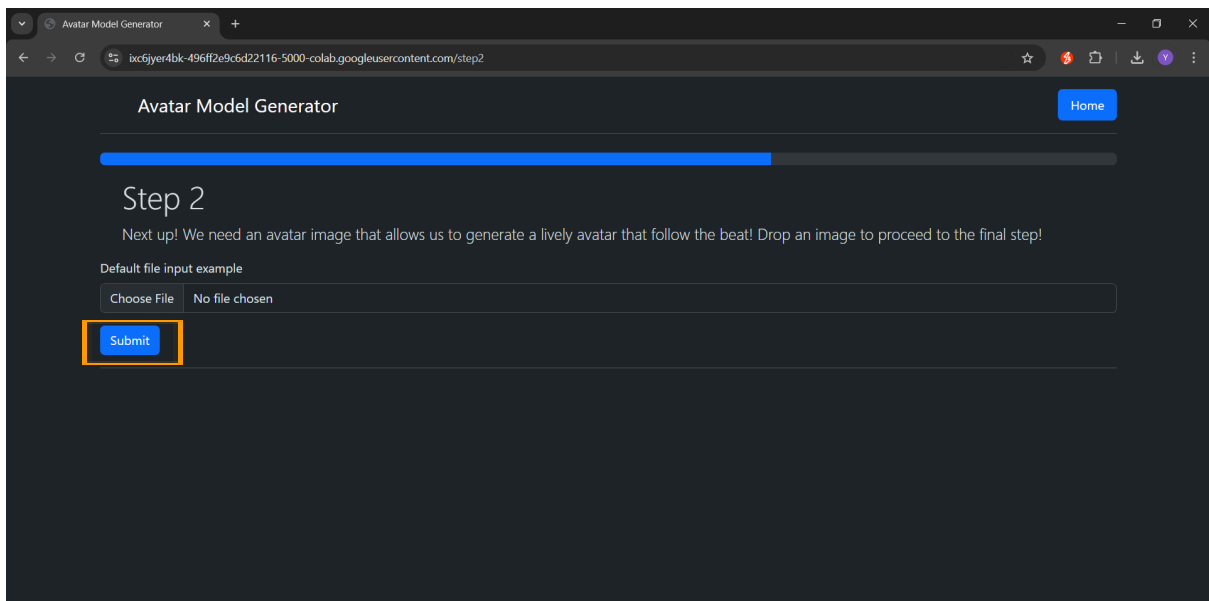
Step 2: Click on submit and you will be navigated to step 2.



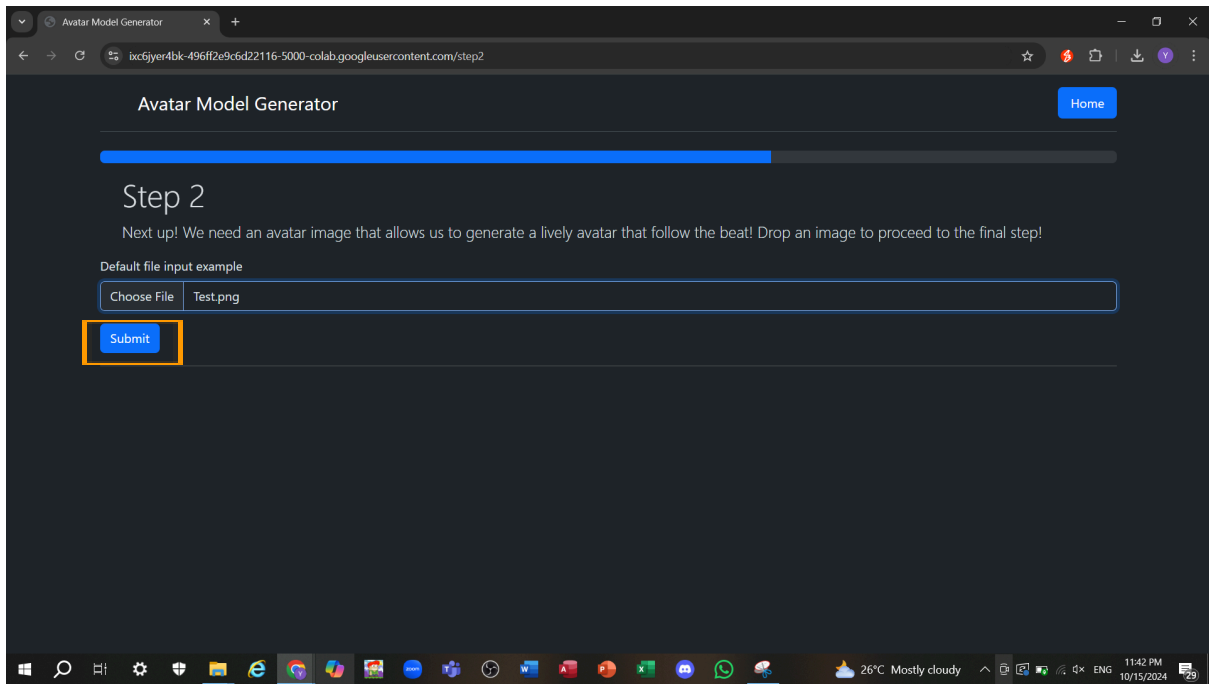
Step 3: An audio preview will be shown with the given audio. Press on next to proceed or remove to go back to audio selection.



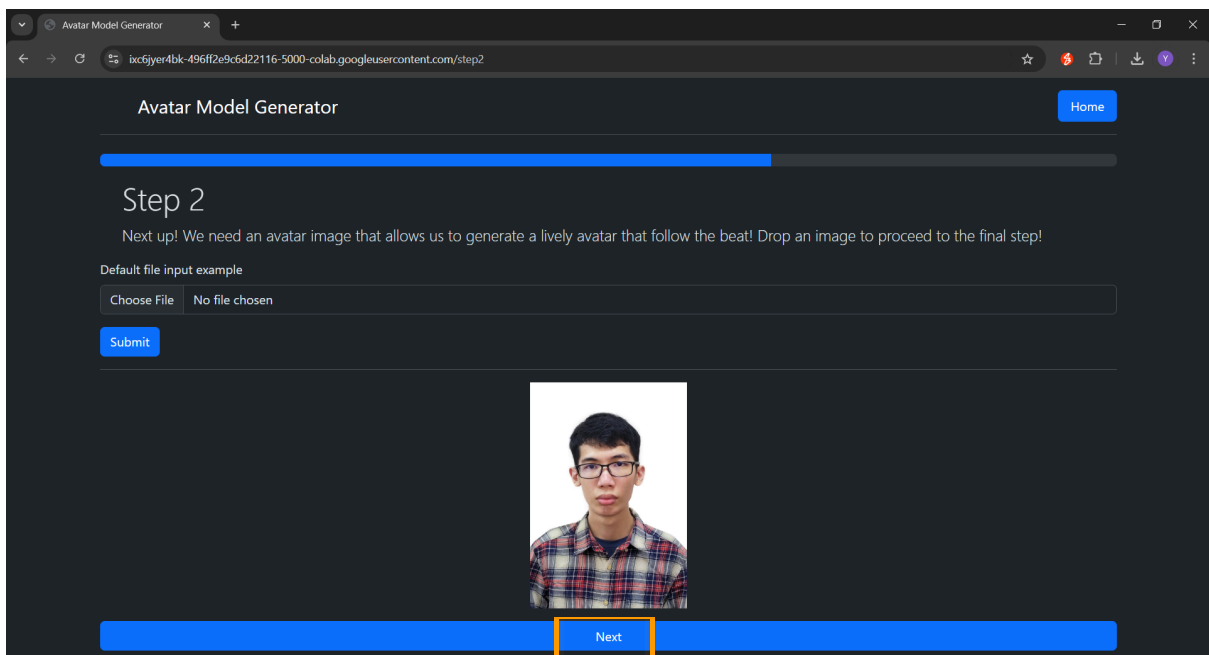
Step 4: We then advance to step 2. In this step, you will be required to upload an image in a .png format by clicking on choose file.



Step 5: After choosing the file, click on submit to proceed.



Step 6: The image will be uploaded and shown to allow the user to ensure they uploaded the correct desired image. After confirming, Click on next to proceed. This might take up to 5 minutes for the generation of the video.



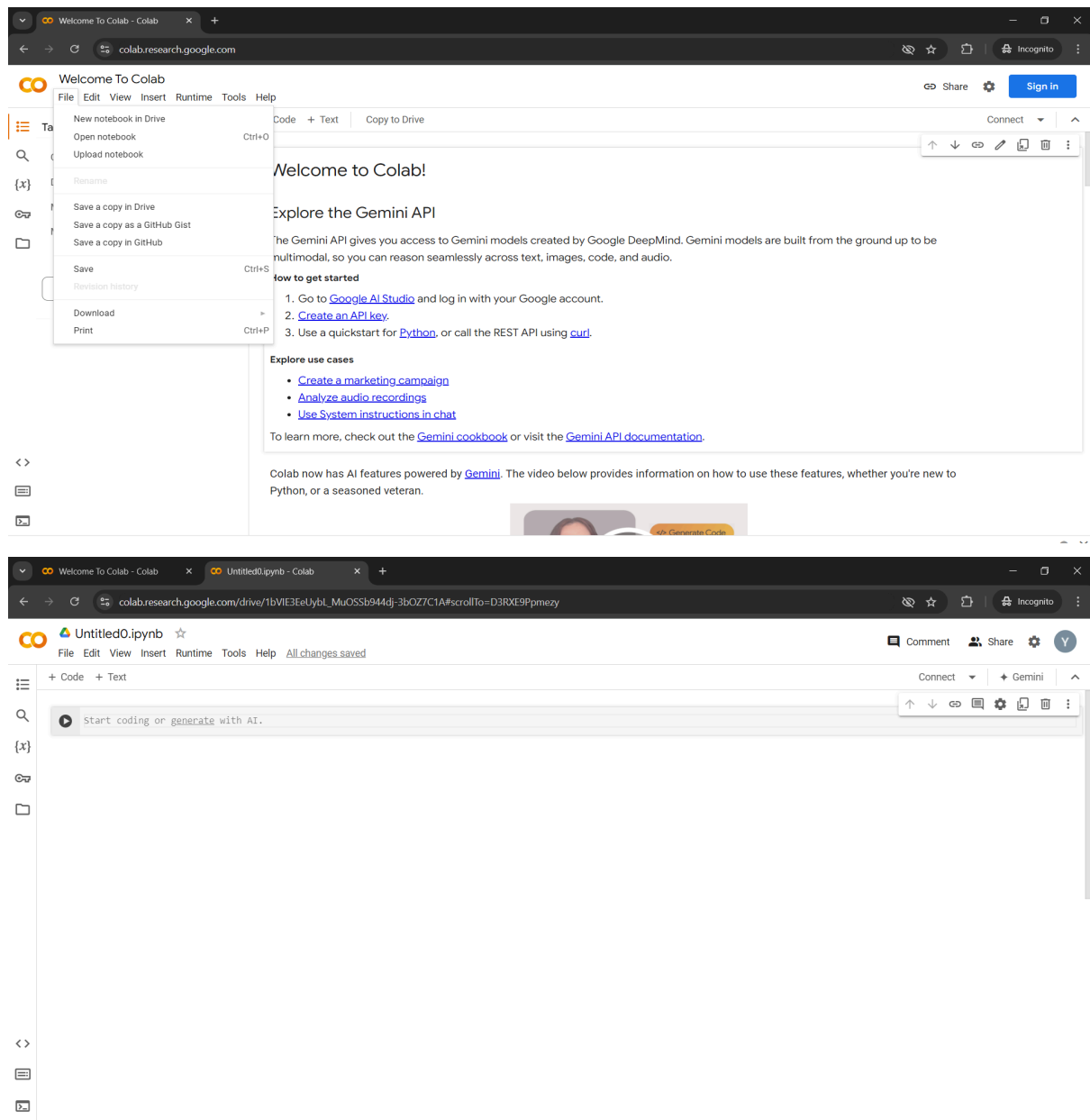
Step 7: At last, the video is generated and the user will be able to view their a generated video with their image and audio as input.

Technical Guide

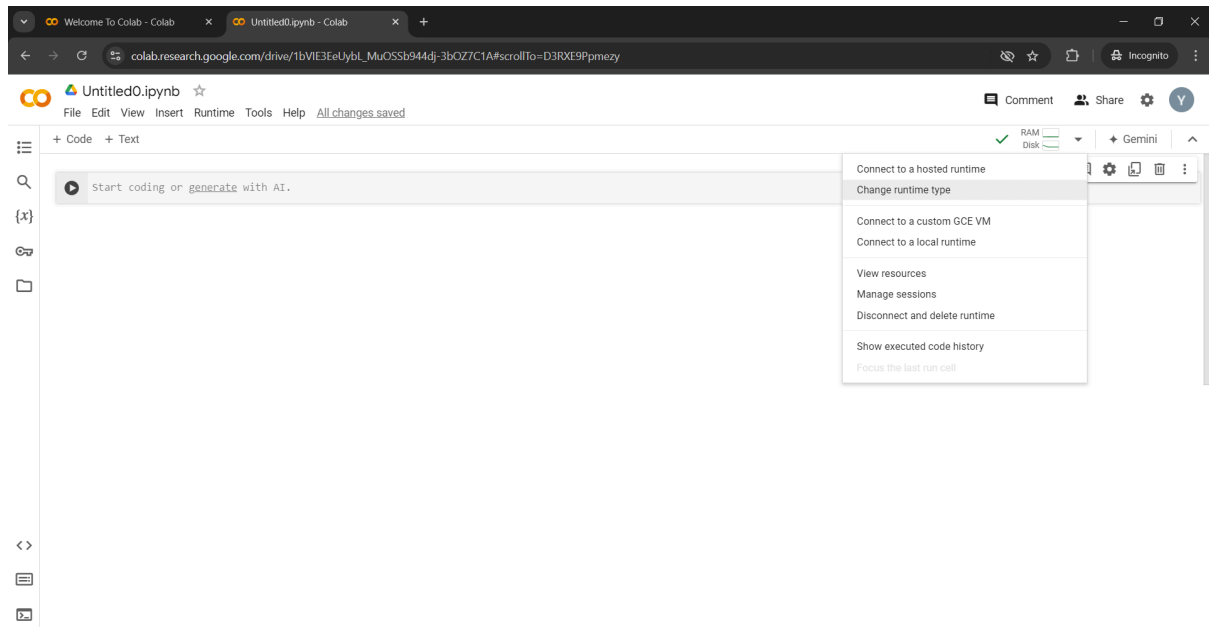
This section provides a step to step guide for the system administrator to set up the application for the End User's use.

1. Open up Google Colab by using this [link](#). You are required to sign in to a google account in order to use Google Colab.

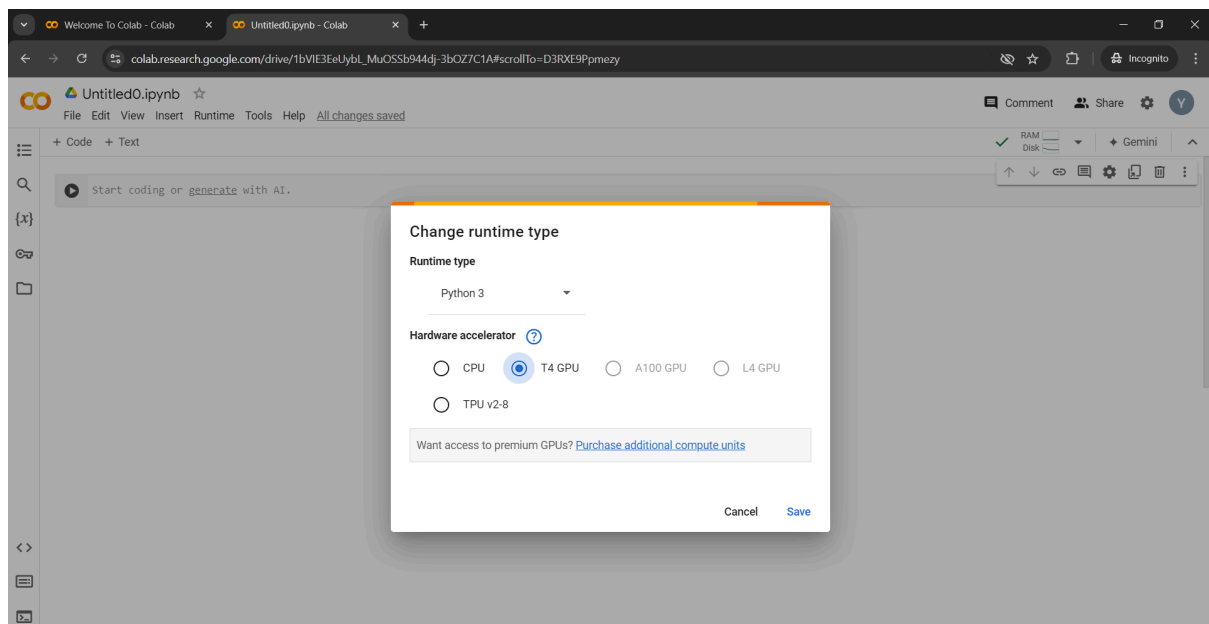
2. Then, direct to files and click on “New notebook in Drive” to open a new notebook as shown below.



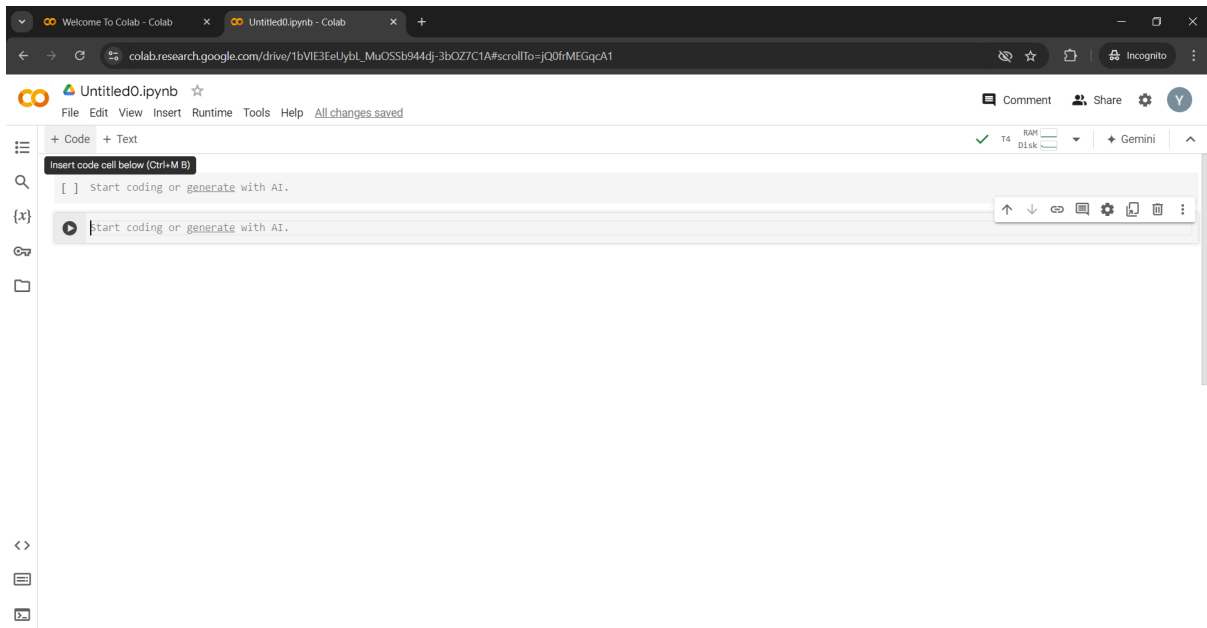
3. To begin with, click on the down-arrow button next to the connect button on the top right corner. Then, click on change runtime type.



Ensure that your runtime type is in Python 3 and Hardware accelerator is with T4 GPU. Click on save to save your selections.



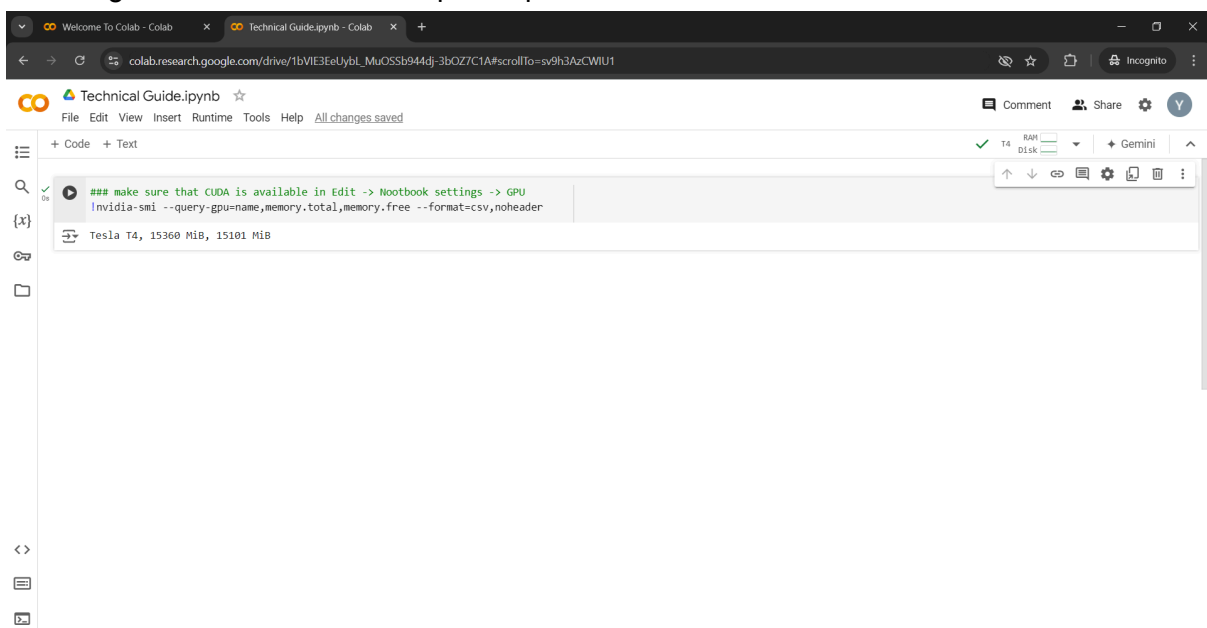
4. For every block of code we add, we are required to click on the + Code button on the top left to add a box to allow us to run every chunk of code we add in. This step will be required for the next steps when we are required to add in new chunks of installation code.



5. Copy and paste the code below into a box. This code checks the status and availability of the GPU. Run this code before advancing to the next step. To run the code, click on the right arrow button on the left side on each box.

```
### make sure that CUDA is available in Edit -> Notebook settings -> GPU
!nvidia-smi --query-gpu=name,memory.total,memory.free --format=csv,noheader
```

The image below shows an example output of the code.



6. Copy and paste the code below into a box. This piece of code is a setup script designed to configure the environment, install the required Python version (3.8), set up dependencies, and clone and prepare the model required. Run this code before advancing to the next step. This can take up to 5 minutes.

```
!update-alternatives --install /usr/local/bin/python3 python3
/usr/bin/python3.8 2
!update-alternatives --install /usr/local/bin/python3 python3
/usr/bin/python3.9 1
!sudo apt install python3.8

!sudo apt-get install python3.8-distutils

!python --version

!apt-get update

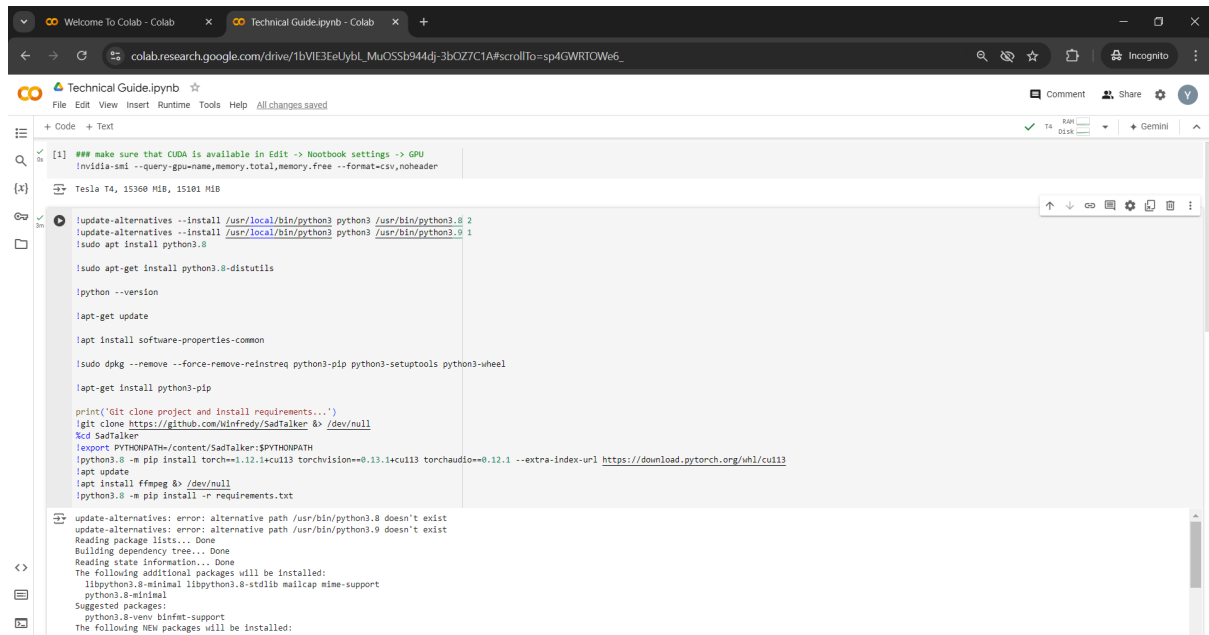
!apt install software-properties-common

!sudo dpkg --remove --force-remove-reinstreq python3-pip
python3-setuptools python3-wheel

!apt-get install python3-pip

print('Git clone project and install requirements...')
!git clone https://github.com/Winfredy/SadTalker &> /dev/null
%cd SadTalker
!export PYTHONPATH=/content/SadTalker:$PYTHONPATH
!python3.8 -m pip install torch==1.12.1+cu113 torchvision==0.13.1+cu113
torchaudio==0.12.1 --extra-index-url
https://download.pytorch.org/whl/cu113
!apt update
!apt install ffmpeg &> /dev/null
!python3.8 -m pip install -r requirements.txt
```

This image below shows an example of the code output.



```
## make sure that CUDA is available in Edit -> Notebook settings -> GPU
!nvidia-smi --query=gpu-name,memory.total,memory.free --format=csv,noheader

Tesla T4, 15360 MiB, 15101 MiB

!update-alternatives --install /usr/local/bin/python3 python3 /usr/bin/python3.8 2
!update-alternatives --install /usr/local/bin/python3 python3 /usr/bin/python3.9 1
!sudo apt install python3.8

!sudo apt-get install python3.8-distutils

!python --version

!apt-get update

!apt install software-properties-common

!sudo dpkg --remove --force-remove-reinstreq python3-pip python3-setuptools python3-wheel

!apt-get install python3-pip

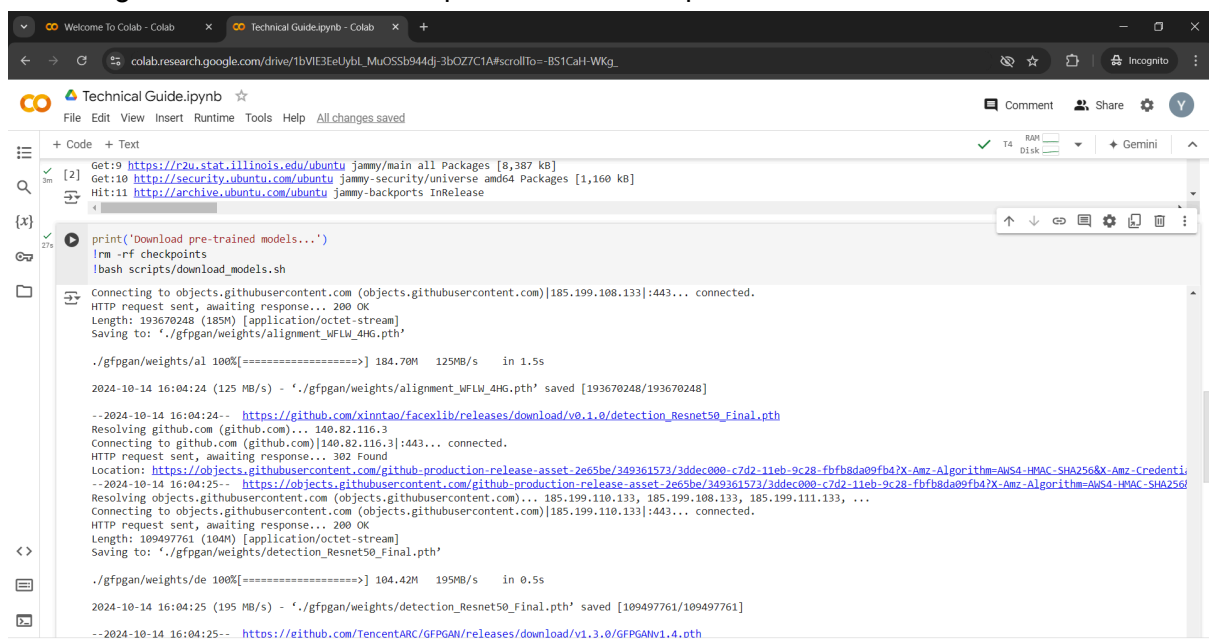
print('git clone project and install requirements...')
!git clone https://github.com/minfredy/SadTalker & /dev/null
!cd SadTalker
!export PYTHONPATH=/content/SadTalker:$PYTHONPATH
!python3.8 -m pip install torch==1.12.1+cu113 torchvision==0.13.1+cu113 torchaudio==0.12.1 --extra-index-url https://download.pytorch.org/whl/cu113
!apt update
!apt install ffmpeg & /dev/null
!python3.8 -m pip install -r requirements.txt

update-alternatives: error: alternative path /usr/bin/python3.8 doesn't exist
update-alternatives: error: alternative path /usr/bin/python3.9 doesn't exist
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libpython3.8-minimal libpython3.8-stdlib mailcap mime-support
python3.8-minimal
Suggested packages:
  python3.8-venv binfmt-support
The following NEW packages will be installed:
```

7. Copy and paste the code into a box. This piece of code is responsible for downloading the pre-trained models required for the model used. Run this code before advancing to the next step. This can take up to 1 minute.

```
print('Download pre-trained models...')
!rm -rf checkpoints
!bash scripts/download_models.sh
```

This image below shows an example of the code output.



```
Get:9 https://r2u.stat.illinois.edu/ubuntu jammy/main all Packages [8,387 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1,160 kB]
Hit:11 http://archive.ubuntu.com/ubuntu jammy-backports InRelease

Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 193670248 (185M) [application/octet-stream]
Saving to: './gfpgan/weights/alignment_WFLW_4HG.pth'

./gfpgan/weights/al 100%[=====] 184.70M 125MB/s in 1.5s

2024-10-14 16:04:24 (125 MB/s) - './gfpgan/weights/alignment_WFLW_4HG.pth' saved [193670248/193670248]

--2024-10-14 16:04:24-- https://github.com/xinntao/facefix/releases/download/v0.1.0/detection_Resnet50_Final.pth
Resolving github.com (github.com)... 140.82.116.3
Connecting to github.com (github.com)|140.82.116.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/349361573/3ddec000-c7d2-11eb-9c28-fbf8da09fb47X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=...
--2024-10-14 16:04:25-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/349361573/3ddec000-c7d2-11eb-9c28-fbf8da09fb47X-Amz-Algorithm=AWS4-HMAC-SHA256
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.108.133, 185.199.111.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 109497761 (104M) [application/octet-stream]
Saving to: './gfpgan/weights/detection_Resnet50_Final.pth'

./gfpgan/weights/de 100%[=====] 104.42M 195MB/s in 0.5s

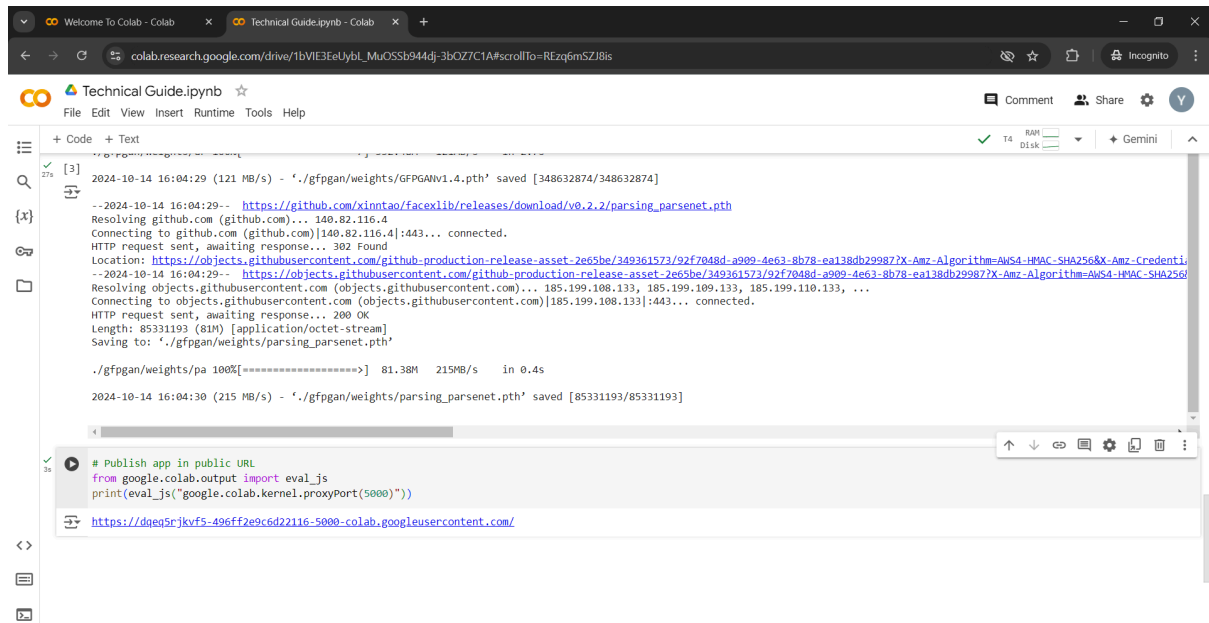
2024-10-14 16:04:25 (195 MB/s) - './gfpgan/weights/detection_Resnet50_Final.pth' saved [109497761/109497761]

--2024-10-14 16:04:25-- https://github.com/TencentARC/GFPGAN/releases/download/v1.3.0/GFPGANv1.4.pth
```


8. Copy and paste the code into a box. This piece of code is specific to Google Colab and is used to expose a locally running Flask (or any other web app) on a public URL within Colab's environment. Run this code before advancing to the next step.

```
# Publish app in public URL
from google.colab.output import eval_js
print(eval_js("google.colab.kernel.proxyPort(5000)"))
```

This image below shows an example of the code output.



```
2024-10-14 16:04:29 (121 MB/s) - './gfpgan/weights/GFPGANv1.4.pth' saved [348632874/348632874]
--2024-10-14 16:04:29-- https://github.com/xinntao/faceXlib/releases/download/v0.2.2/parsing_parsenet.pth
Resolving github.com (github.com)... 149.82.116.4
Connecting to github.com (github.com)[149.82.116.4]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/349361573/92f7048d-a909-4e63-8b78-ea138db29987?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=aws4_credentials&X-Amz-Date=20241014T160429Z&X-Amz-Security-Token=token&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3Bfilename%3Dparsing_parsenet.pth
--2024-10-14 16:04:29-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/349361573/92f7048d-a909-4e63-8b78-ea138db29987?X-Amz-Algorithm=AWS4-HMAC-SHA256
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)[185.199.108.133]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85331193 (81M) [application/octet-stream]
Saving to: './gfpgan/weights/parsing_parsenet.pth'

./gfpgan/weights/pa 100%[*****] 81.38M 215MB/s in 0.4s

2024-10-14 16:04:30 (215 MB/s) - './gfpgan/weights/parsing_parsenet.pth' saved [85331193/85331193]

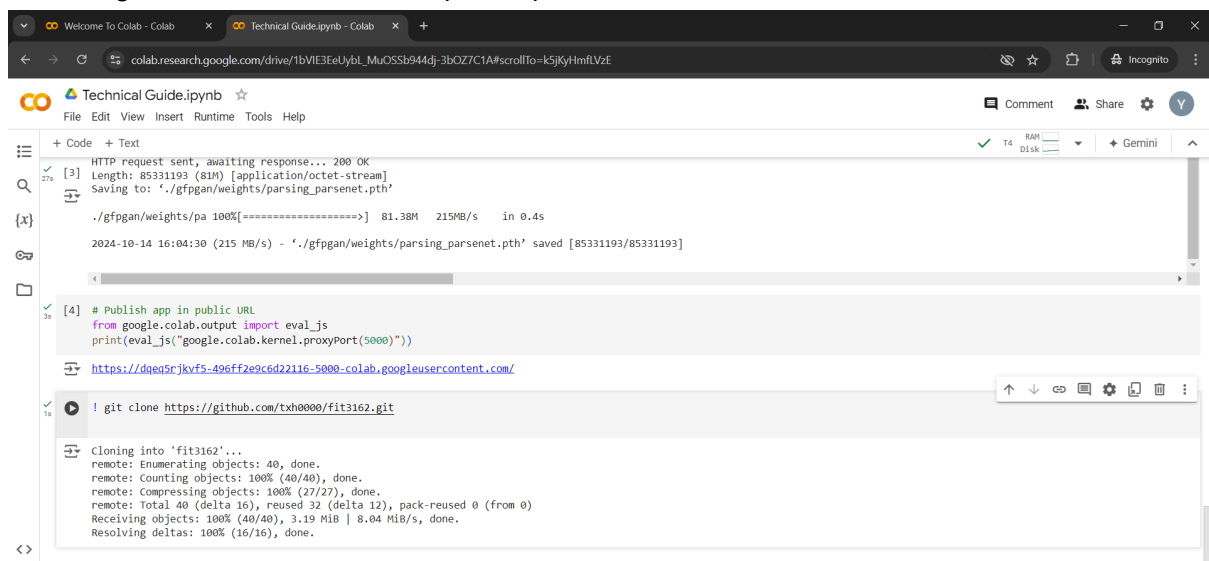
[3] # Publish app in public URL
from google.colab.output import eval_js
print(eval_js("google.colab.kernel.proxyPort(5000)"))

https://dqe95rjkvf5-496ff2e9c6d22116-5000-colab.googleusercontent.com/
```

9. Copy and paste the code into a box. This piece of code is using Git to clone a repository from GitHub into the current environment. Run this code before advancing to the next step.

```
! git clone https://github.com/txx0000/fit3162.git
```

This image below shows an example output of the code.



```
HTTP request sent, awaiting response... 200 OK
Length: 85331193 (81M) [application/octet-stream]
Saving to: './gfpgan/weights/parsing_parsenet.pth'

./gfpgan/weights/pa 100%[*****] 81.38M 215MB/s in 0.4s

2024-10-14 16:04:30 (215 MB/s) - './gfpgan/weights/parsing_parsenet.pth' saved [85331193/85331193]

[4] # Publish app in public URL
from google.colab.output import eval_js
print(eval_js("google.colab.kernel.proxyPort(5000)"))

https://dqe95rjkvf5-496ff2e9c6d22116-5000-colab.googleusercontent.com/

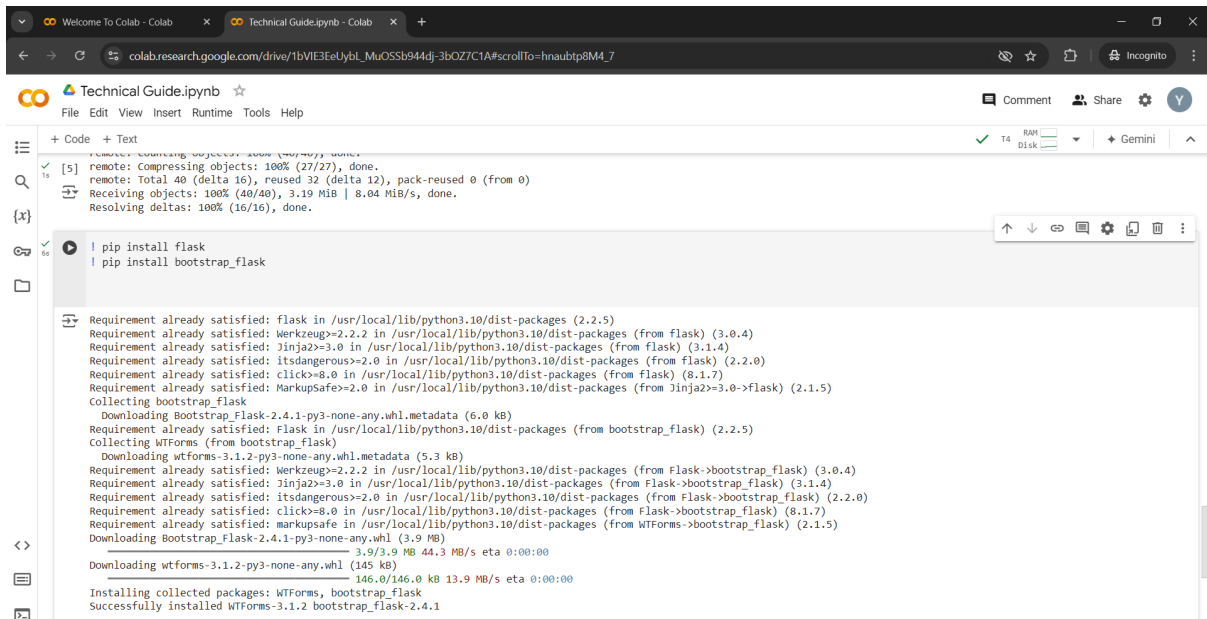
! git clone https://github.com/txx0000/fit3162.git

Cloning into 'fit3162'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 40 (delta 16), reused 32 (delta 12), pack-reused 0 (from 0)
Receiving objects: 100% (40/40), 3.19 MiB | 8.04 MiB/s, done.
Resolving deltas: 100% (16/16), done.
```

10. Copy and paste the code into a box. This code is used to install two Python libraries, Flask and Bootstrap-Flask, in the current environment. It uses the pip package manager to do so.

```
! pip install flask
! pip install bootstrap_flask
```

This image below shows an example output of the code.

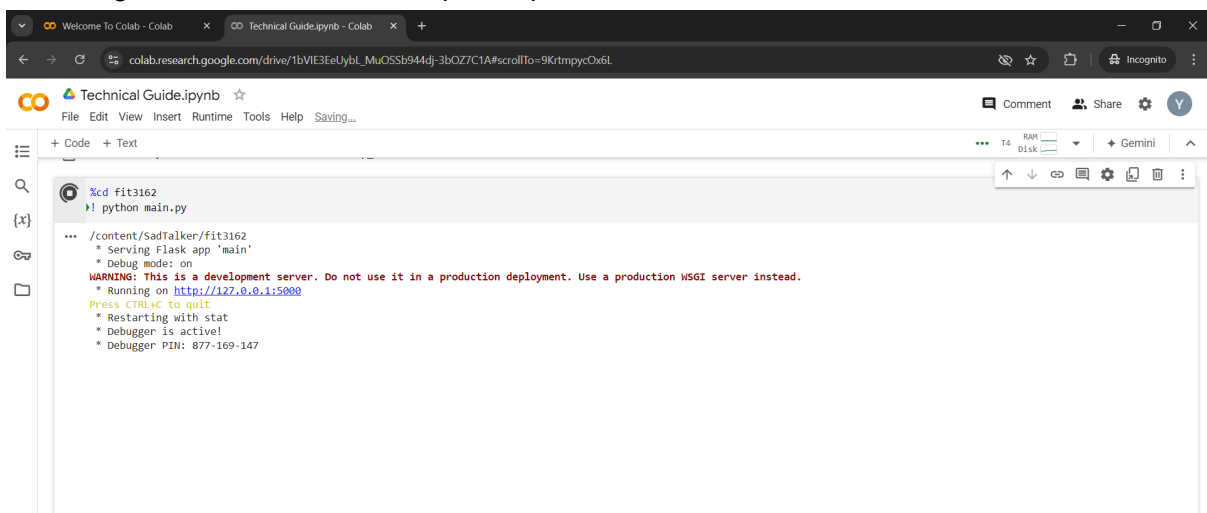


The screenshot shows a Jupyter Notebook interface with a code cell containing the commands `! pip install flask` and `! pip install bootstrap_flask`. The output of the first command shows that Flask and its dependencies (Werkzeug, Jinja2, itsdangerous, click, MarkupSafe) are already installed. The output of the second command shows the installation of Bootstrap-Flask, which depends on Flask and WTForms. The output also shows the download progress for WTForms and Bootstrap-Flask, and a final message indicating that the packages were successfully installed.

11. Copy and paste the code into a box. This piece of code is intended to navigate to a directory and run a Python script for the UI. Run this code before advancing to the next step.

```
%cd fit3162
! python main.py
```

This image below shows an example output of the code.



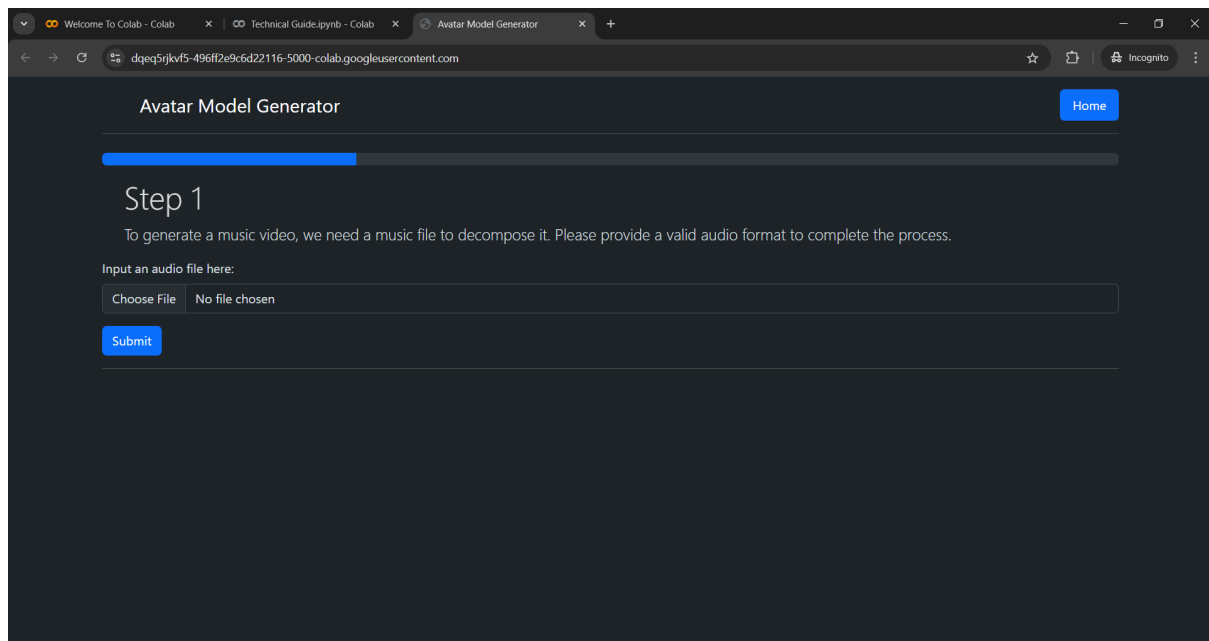
The screenshot shows a Jupyter Notebook interface with a code cell containing the commands `%cd fit3162` and `! python main.py`. The output of the first command shows the current directory is `/content/SadTalker/fit3162`. The output of the second command shows the output of running the `main.py` script, which includes a warning about using a development server, the URL `http://127.0.0.1:5000`, and information about the debugger.

12. After running all the code above, go back to the output we got in step 8. Click on the output link given as shown below to enter the UI.

```
✓ 35 # Publish app in public URL
from google.colab.output import eval_js
print(eval_js('google.colab.kernel.proxyPort(5000)'))

🔗 https://dqe95rjkvf5-496ff2e9c6d22116-5000-colab.googleusercontent.com/
```

After clicking on the link, a new tab will appear in your browser as shown below. This is the Main page and the application is ready to be used as per Part 1: End User Guide.



By doing so, all the steps are completed and the application is ready to use as outlined in Part 1 of the End User Guide.