Speech Dubbing Software

A
Major Project Report
for the award of

Bachelor of Technology

in
Computer Science and Engineering

Under the faculty of
Computer Science and Engineering

Submitted to

B. P. Mandal College of Engineering, Madhepura

Aryabhatta Knowledge University
Patna, (Bihar), India

by

Student's Name

Name	Reg. No.
1. Pushkar Kumar	17105128016
2. Kapil Kumar	17105128013
3. Pranav Ravi	17105128012
4. Sudama Manjhi	17105128003

Under The Supervision
of

Mr. Raj Kumar
Assistant Professor
Department of
Computer Science & Engineering
B.P. Mandal College Of Engineering
Madhepura (Bihar)

Year - 2021

Dedicated to

Our Parents, Teachers and Friends.

B. P. Mandal College of Engineering, Madhepura

(Aryabhatta Knowledge University, Patna)

DECLARATION

We declare that the project entitled **"Speech Dubbing Software"** is our work conducted at B. P. Mandal College of Engineering, Madhepura (Bihar), approved by the Committee. We have put in more than 90 days of attendance with the Supervisors at the B. P. Mandal College of Engineering, Madhepura.

We further declare that, to the best of our knowledge, the report does not contain the work which has been submitted for the award of the degree either in the University or in any other University/Deemed University without proper citations.

Dated: …….

Place: …….

Mr. Raj Kumar
Assistant Professor
Computer Science & Engineering
B.P. Mandal College of Engineering
Madhepura (Bihar)

Prof. Murlidhar Prasad Singh
Assistant Prof.& Head of Department
Computer Science & Engineering
B.P. Mandal College of Engineering
Madhepura, (Bihar)

Prof. Arbind Kumar Amar
Principal of
B.P. Mandal College of Engineering
Madhepura, (Bihar)

B. P. Mandal College of Engineering, Madhepura

(Aryabhatta Knowledge University, Patna)

CERTIFICATE

This is to certify that the major project report entitled **"Speech Dubbing Software"** project work done by **Pushkar Kumar, Kapil Kumar, Pranav Ravi, Sudama Manjhi** under our guidance and supervision for the degree of Bachelor of Engineering of B. P. Mandal College of Engineering, Madhepura, Bihar (India). The candidate has put in the required attendance of more than 90 days with me. To the best of my knowledge and brief the project:

I. Embodies the work of the candidate himself.

II. Has duly been completed.

III. Is up to the standard both in respect of contents and language for being referred to the examiner.

Dated: _____

Place: _____

Forwarded by

Mr. Raj Kumar
Assistant Professor
Computer Science & Engineering
B.P. Mandal College of Engineering
Madhepura (Bihar)

Prof. Murlidhar Prasad Singh
Assistant Prof.& Head of Department
Computer Science & Engineering
B.P. Mandal College of Engineering
Madhepura, (Bihar)

Prof. Arbind Kumar Amar
Principal
B.P. Mandal College of Engineering
Madhepura, (Bihar)

B. P. Mandal College of Engineering, Madhepura

(Aryabhatta Knowledge University, Patna)

ACKNOWLEDGEMENT

We express our sincere thanks to our respected project supervisor **Mr. Raj Kumar** (Assistant Professor), Department of Computer Science and Engineering, B.P. MANDAL COLLEGE OF ENGINEERING, MADHEPURA for his guidance, encouragement, and motivation towards real-time research work right from the beginning of the research up to the completion of this dissertation. It was a pleasure to work under his supervision. He was always available with new ideas and suggestions during the difficult phases of my dissertation. All the help is gratefully acknowledged.

We sincerely thank **Prof. Murlidhar Prasad Singh**, as the Head of the Department of Computer Science & Engineering for his kind support and permission to use the facilities available in the institute.

We sincerely thanks to our Principal **Prof. Arvind Kumar Amar** for his support, motivation and encouraging us.

My thanks go to all the staff of the Department of Computer Science and Engineering,
B. P. MANDAL COLLEGE OF ENGINEERING, MADHEPURA for their continuous support throughout the dissertation work.

Last, but not least, We also want to give warm thanks to all our classmates, and non-technical staff. Without their thankful hand, We could never complete our dissertation successfully.

Name	Reg. No.	Sign.
1. Pushkar Kumar	17105128016	……
2. Kapil Kumar	17105128013	……
3. Pranav Ravi	17105128012	……
4. Sudama Manjhi	17105128003	……

Date: _____

Place: _____

B. P. Mandal College of Engineering, Madhepura

(Aryabhatta Knowledge University, Patna)

CERTIFICATE OF APPROVAL

The project work entitled "Speech Dubbing Software" being submitted by Pushkar Kumar (17105128016), Kapil Kumar (17105128013), Pranav Ravi (17105128012) & Sudama Manjhi (17105128003) has been examined by us and is hereby approved for the award of degree "BACHELOR OF TECHNOLOGY" (Computer Science and Engineering), for which it has been submitted. It is understood that by this approval the undersigned does not necessarily endorse or approve any statement made, opinion expressed, or conclusion drawn therein, but approve the project work only for the purpose for which it has been submitted.

Internal Examiner **External Examiner**

Head of Department **Principal**

Date : _____

Place : _____

ABSTRACT

The Project “Speech Dubbing Software” aims to replace English speech with Hindi Speech on a .mp4/.mkv file. So that the resultant video speech is in Hindi. Speech Dubbing Software Features that it converts the English speech to Hindi Speech on a Video. When we watch NPTEL Swayam lecture, all lecture is in English. We can’t understand fully because we belong to a rural area. So, we will decide that if we convert that video English speech to Hindi Speech. It will beneficial for all students who are week in English.

For making this project, we used PyCharm, Python, Qt Designer. We used the Tkinter for taking the (.mp4/.mkv) file from File Explorer, then, we convert the .mp4 to .wav file after that, we conquer the .wav file then, we recognize the English text file from the .wav file. After that We translated the English text to Hindi text then We will generate the Hindi speech then, overlapped the input file with generated Hindi speech and We remove the unnecessary file which is not further used in the future.

In this project, after describing our dubbing methodology, we focus on synchronizing the translated Hindi speech with the original input file (.mp4/.mkv extension). We present empirical results for English-to-Hindi dubbing on an English Speech Video.

CONTENTS

LIST OF FIGURES

Speech Dubbing Software

1. Introduction

1.1 About Project

Speech Dubbing Software: It is a graphical user interface-based (GUI) software in which we convert the English video speech to Hindi video speech. It is Machine learning-based software, it is automatically converted the .mp4 video (English language) file to .mp4 video (Hindi language).

1.2 Objective

The main objective of this project (Speech Dubbing Software) is to build software that can automatically translate the video language from English to Hindi. With the help of machine learning and natural language processing.

1.3 Background

Dubbing, mixing or re-recording, is a post-production process used in filmmaking and video production in which additional or supplementary recordings are lip-synced and "mixed" with original production sound to create the finished soundtrack.

The process usually takes place on a dub stage. After sound editors edit and prepare all the necessary tracks – dialogue, automated dialogue replacement (ADR), effects, Foley, music – the dubbing mixers proceed to balance all of the elements and record the finished soundtrack.

Dubbing is sometimes confused with ADR, also known as "additional dialogue replacement", clarification needed"automated dialogue recording" and "looping", in which the original actors re-record and synchronize audio segments.

Outside the film industry, the term "dubbing" commonly refers to the replacement of the actor's voices with those of different performers speaking another language, which is called "revoicing" in the film industry.

further explanation needed The term "dubbing" is only used when talking about replacing a previous voice, usually in another language.

When a voice is created from scratch for animations, the term "original voice" is always used because, in some cases, this media is partially finished before the voice is implemented. The voice work would still be part of the creation process, thus being considered the official voice.

Automated dialogue replacement (ADR) is the process of re-recording dialogue by the original actor (or a replacement actor) after the filming process to improve audio quality or make changes to the originally scripted dialogue. In the early days of talkies, a loop of the film would be cut and spliced together for each of the scenes that needed to be rerecorded, then one-by-one the loops would be loaded onto a projector.

1.4 Motivation

To understand the English video Lecture in the Hindi language.

- Easy to understand the complex topic (when you are a Hindi medium student)
- There are lots of good video lecture available on NPTEL or other foreign countries university portal but few students able to see and understand the lecture topic. In this situation very helpful for students who do not understand the well English lecture videos.

2. Description of Used Tools & Technology

2.1 Python

- Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs, as well as its object-oriented approach, aim to help programmers write clear, logical code for small and large-scale projects.

Figure No 2.1

- Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.
- Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting.
- Python 3.0 was released in 2008 and was a major revision of the language that is not completely backwards-compatible. Python 2 was discontinued version 2.7.18 in 2020.

2.1.1 These are some of the reasons why use Python?

- **Getting Started**

  Python can be easy to pick up whether you're a first-time programmer or you're experienced with other languages.

- **Friendly & Easy to Learn**

  The community hosts conferences and meetups collaborate on code, and much more.

- **Applications**

  The Python Package Index (PyPI) hosts thousands of third-party modules for Python. Both Python's standard library and the community-contributed modules allow for endless possibilities.

  - Web and Internet Development
  - Database Access
  - Desktop GUIs
  - Scientific & Numeric
  - Education
  - Network Programming
  - Software & Game Development

- **Open-source**

  Python is developed under an OSI-approved open source license, making it freely usable and distributable, even for commercial use. Python's license is administered by the Python Software Foundation.

2.1.2 Python 3 Installation on windows

a) Downloading

1. Go to Official Website: www.python.org

The following page will appear in your browser.

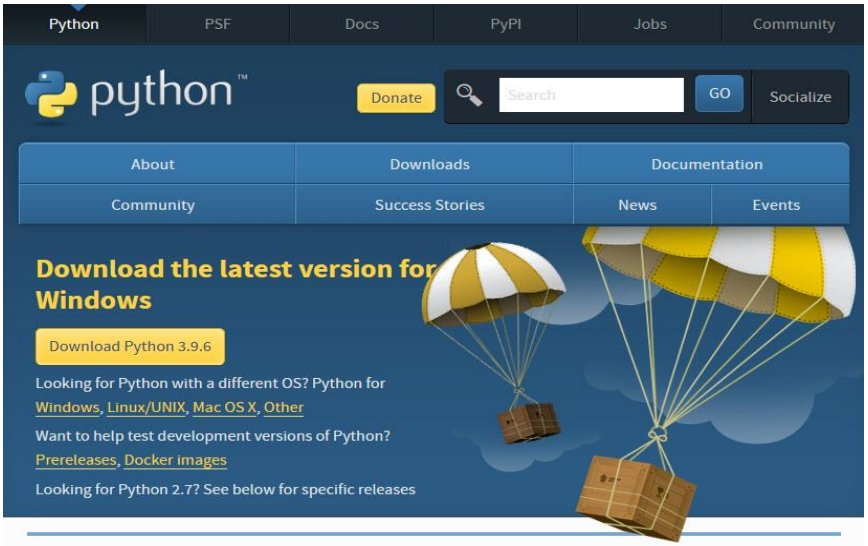


Figure No 2.2

2. Click the Download Python 3.9.6 button.

The following pop-up window titled Opening python-3.96-amd64.exe will appear.

Figure No 2.3

3. Click the Save File button.

The file named python-3.9.6-amd64.exe should start downloading into your standard download folder. This file is about 25 Mb so it might take a while to download fully if you are on a slow internet connection (it took me about 10 seconds over a cable modem).

The file should appear in your Downloads folder as

Figure No 2.4

Move this file to a more permanent location, so that you can install Python(and reinstall it easily later, if necessary).

b) Installation

1. **Double-click the icon labelling the file python-3.9.6-amd64.exe.**

A Python 3.9.6 (64-bit) Setup pop-up window will appear.

Figure No 2.5

Ensure that both the Install launcher for all users (recommended) and the Add Python 3.9 to PATH checkboxes at the bottom are checked: typically only first is checked by default.

If the Python Installer finds an earlier version of Python installed on your computer, the Install Now message may instead appear as Upgrade Now (and the checkboxes will not appear).

2. **Highlight the Install Now (or Upgrade Now) message, and then click it.**

When run, a User Account Control pop-up window may appear on your screen. I could not capture its image, but it asks, Do you want to allow this app to make changes to your device.

3. **Click the Yes button.**

A new Python 3.9.6 (64-bit) Setup pop-up window will appear with a Setup Progress message and a progress bar.

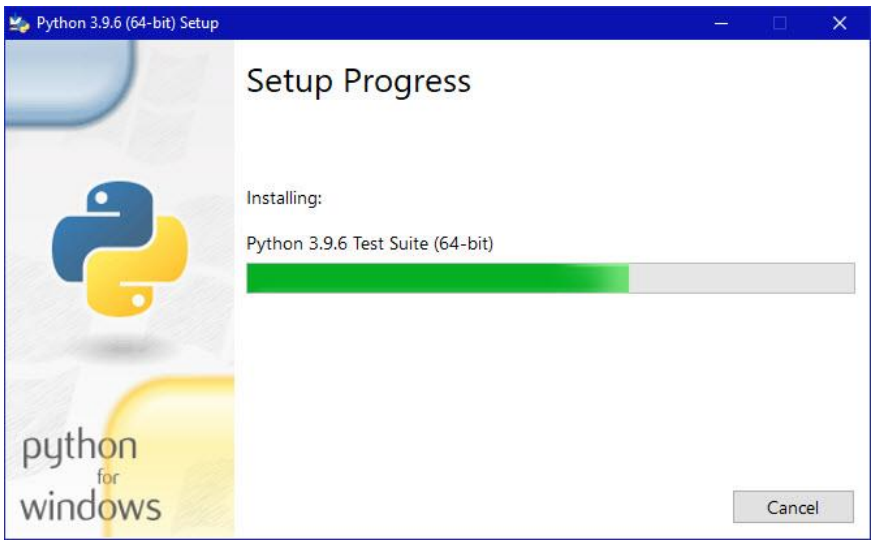


Figure No 2.6

During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new Python 3.9.6 (64-bit) Setup pop-up window will appear with a Setup was successfully message.

4. **Click the Close button.**

Python should now be installed.

c) Verifying

1. **Navigate to the**
directory C:\Users\Pattis\AppData\Local\Programs\Python\Python39

2. **Double-click the icon/file python.exe.**

3. The following pop-up window will appear.

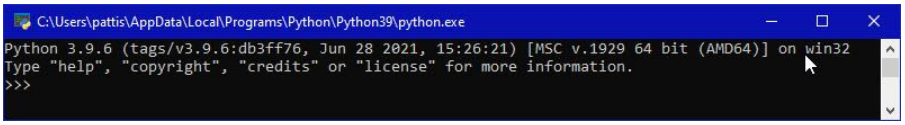


Figure No 2.7

A pop-up window with the title:

C:\Users\Pattis\AppData\Local\Programs\Python\Python39\python .exe

appears, and inside the window; on the first line is the text Python 3.9.6 ... (notice that it should also say 64 bit). Inside the window, at the bottom left, is the prompt >>>: type exit() to this prompt and press enter to terminate Python.

You should keep the file python-3.9.6.exe somewhere on your computer in case you need to reinstall Python (not likely necessary).

2.2 PyCharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as IntelliJ).

Figure No 2.8

It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.

2.2.1 Features

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes.
- Project and code navigation: specialized project views, file structure views, and quick jumping between files, classes, methods, and usages.
- Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down, and others.
- Support for web frameworks: Django, web2py, and Flask [professional edition only.
- Integrated Python debugger.
- Integrated unit testing, with line-by-line code coverage.
- Google App Engine Python development [professional edition only.
- Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce, and CVS with change lists and merge

2.2.2 Plugins

PyCharm provides an API so that developers can write their plugins to extend PyCharm features. Several plugins from other JetBrains IDE also work with PyCharm. There are more than 1000 plugins that are compatible with PyCharm.

2.2.3 History

The beta version was released in July 2010, with the 1.0 arriving 3 months later. Version 2.0 was released on 13 December 2011, version 3.0 on 24 September 2013, and version 4.0 on 19 November 2014.

PyCharm Community Edition, the open-source version of PyCharm, became available on 22 October 2013

2.2.4 Licensing

PyCharm Professional Edition has several license options that differ in their features, price, and terms of use.

- General and commercial use is paid.
- PyCharm Professional Edition is free for open-source projects and some educational uses.
- An Academic license is discounted or free.

PyCharm Community Edition is distributed under Apache 2 license, with full source code available on GitHub.

2.2.5 Prerequisites

- 64-bit versions of Microsoft Windows 8 or 10
- Minimum of 2 GB RAM, with 8 GB RAM recommended
- A minimum of 2.5 GB SSD hard drive space (SSD is recommended)
- A minimum screen resolution of 1024x768
- A version of Python 2.7+, or Python 3.5+

2.2.6 Installation

Step 1:

Download PyCharm from the JetBrains website. You will have the option of using the Professional version or the Community version. This guide installs the Community edition. Note: It is also a good idea to confirm that Python is installed.

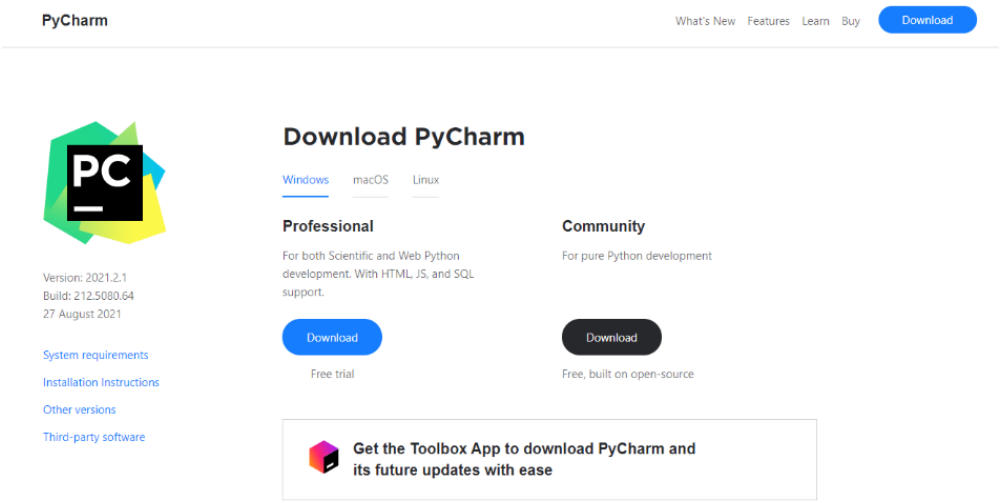


Figure No 2.9

Step 2:

After running the installer, select the installation location. The most common location is in our Program Files directory, but this can be changed to an alternative location if needed, but this is usually unnecessary.

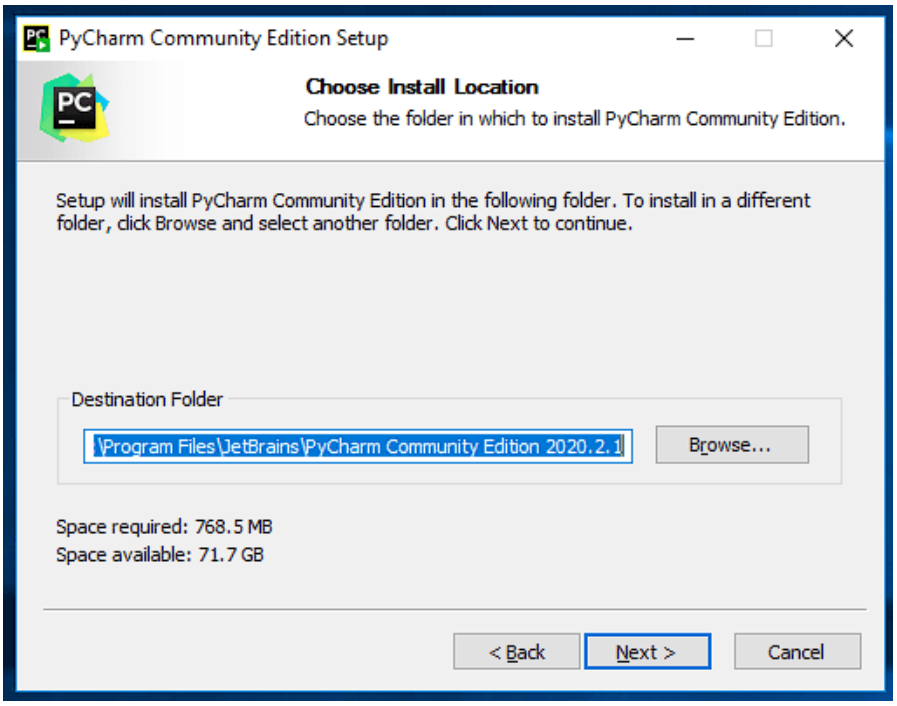


Figure No 2.10

Step 3:

Next, we will be prompted to select our installation options. These include the option to add a desktop shortcut, updating your path to include PyCharm launcher along with any associating file types, and a context menu. You can see below the choices I have selected for my installation. Feel free to choose what is best for your development style.

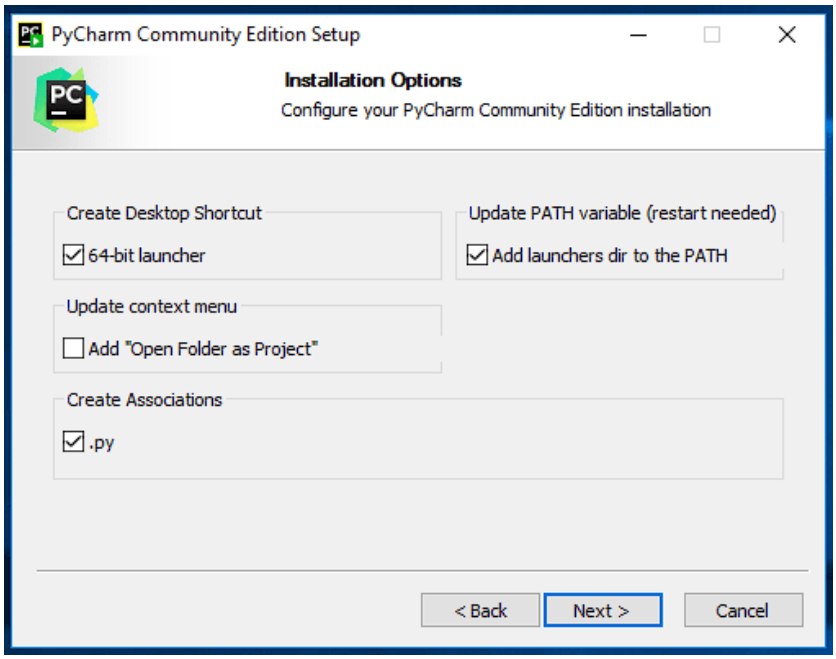


Figure No 2.11

Step 4:

The next step is to choose your start menu folder location, for a quick reference from your desktop start page.

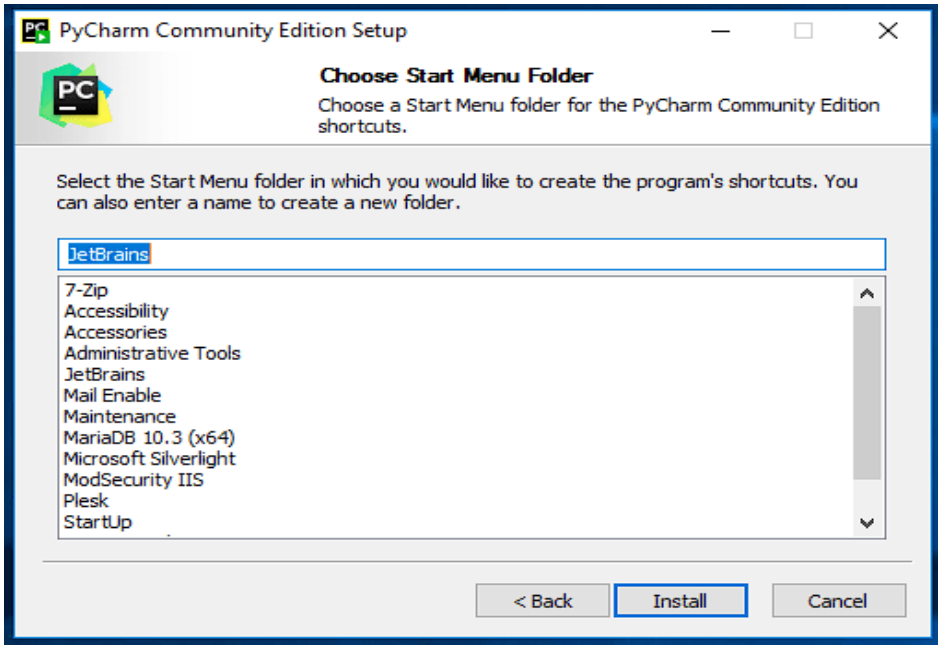


Figure No 2.12

Step 5:

Now, we wait for PyCharm to install. If you wish to see additional details of the installation, you can select show details.

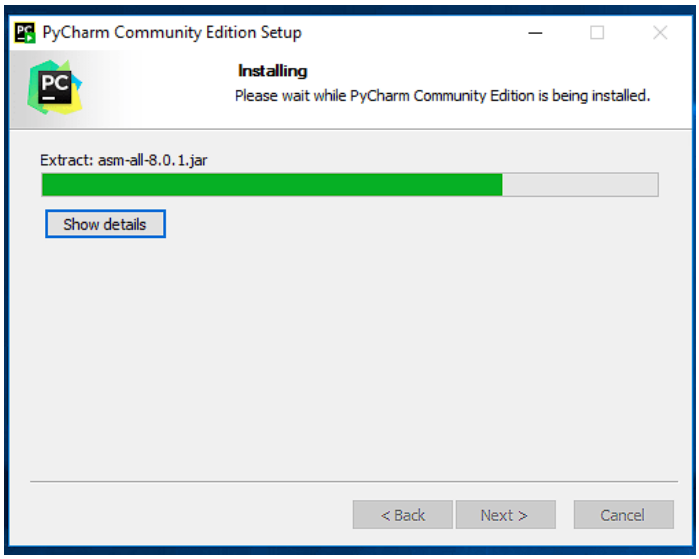


Figure No 2.13

Step 6:

Finally, reboot the computer at the end of the PyCharm installation process.

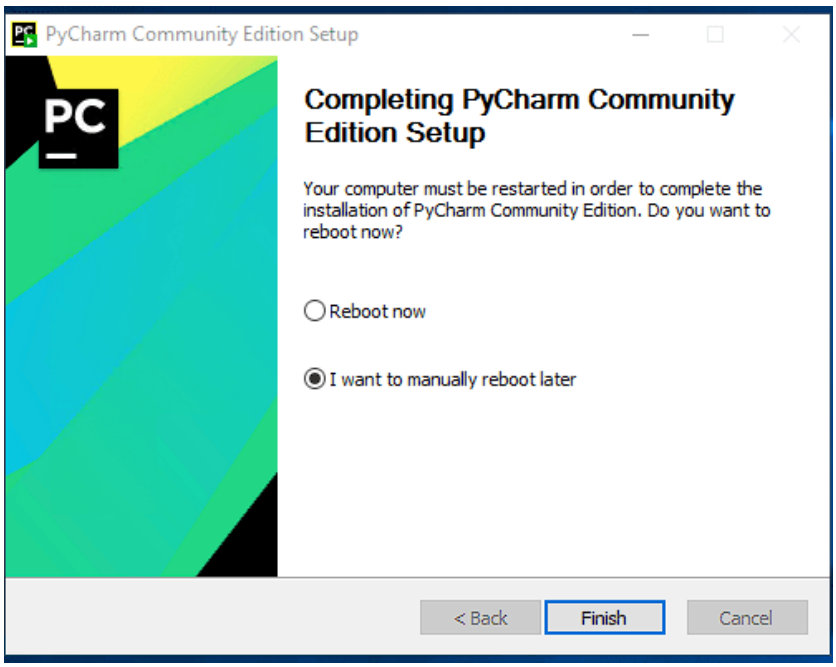


Figure No 2.14

2.2.7 How Do I Configure PyCharm?

Step 1:

When we open PyCharm the first time, we will initially be prompted with featured plugins we can download to emulate a Vim editor, R language support, or an AWS Toolkit.

A more detailed plugin list is available in settings. Feel free to install and enable the ones that best suit your needs. The image below and most likely, your screen will only show the featured plugins, but you can find more in your settings - External tools - packages.

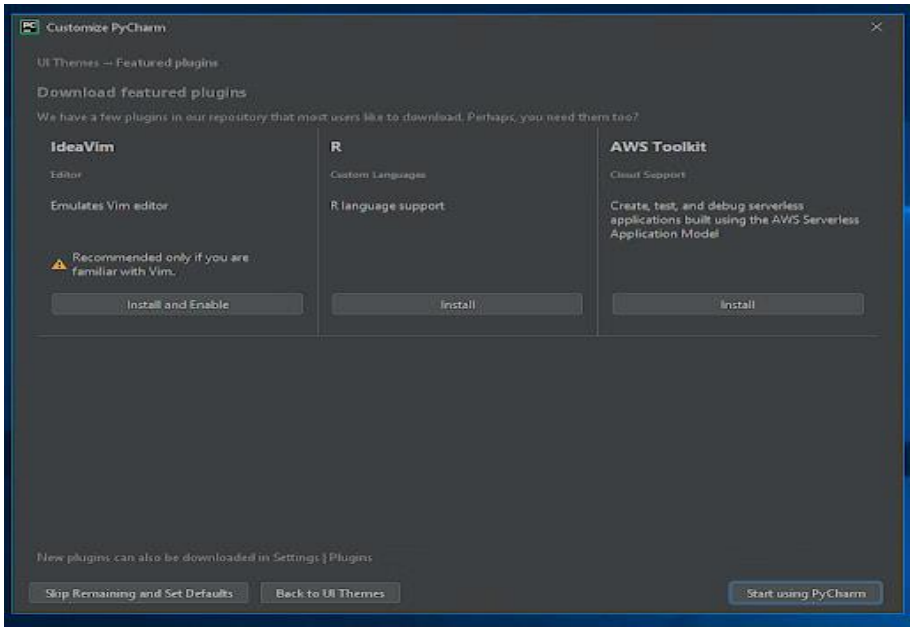


Figure No 2.15

Step 2:

From the main screen, we can create a new project, open an existing one, or select one from our versioning control system of choice, like GIT. We can also see the configuration options in the bottom right corner, and ask for help which can be useful in moving forward.

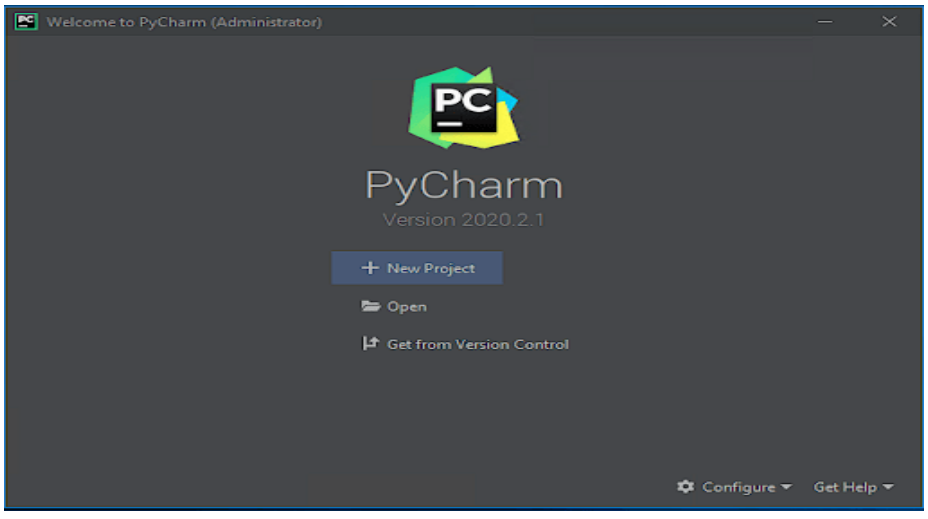


Figure No 2.16

Step 3:

When creating a new project, we will be asked a few questions like where would we like to place our project folder and any related files which need to be saved. Or, whether we would like a main.py created by default, and other Python interpreter options unique to our environment.

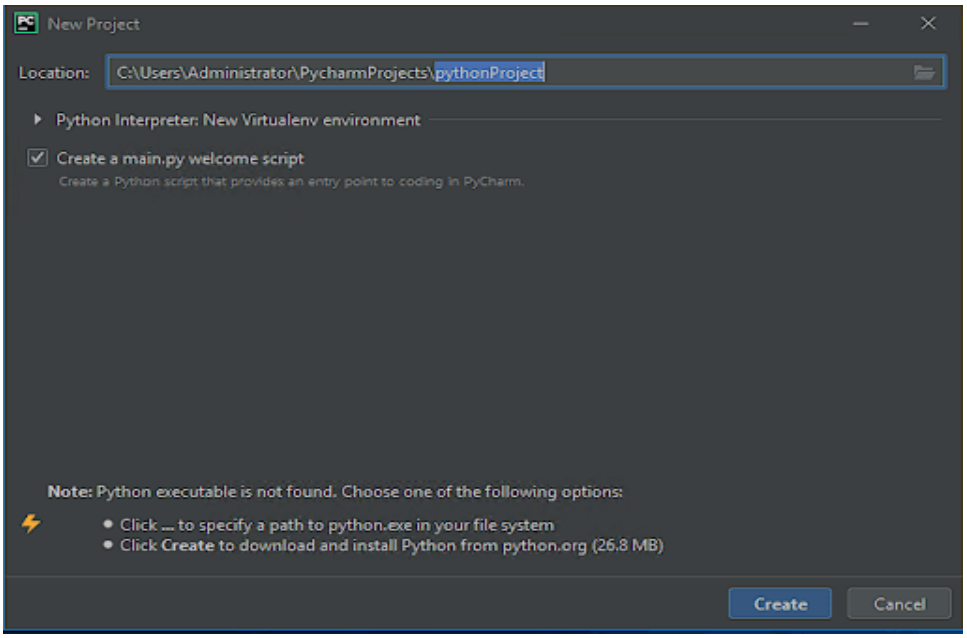


Figure No 2.17

Step 4:

Finely you will see the dashboard of your project

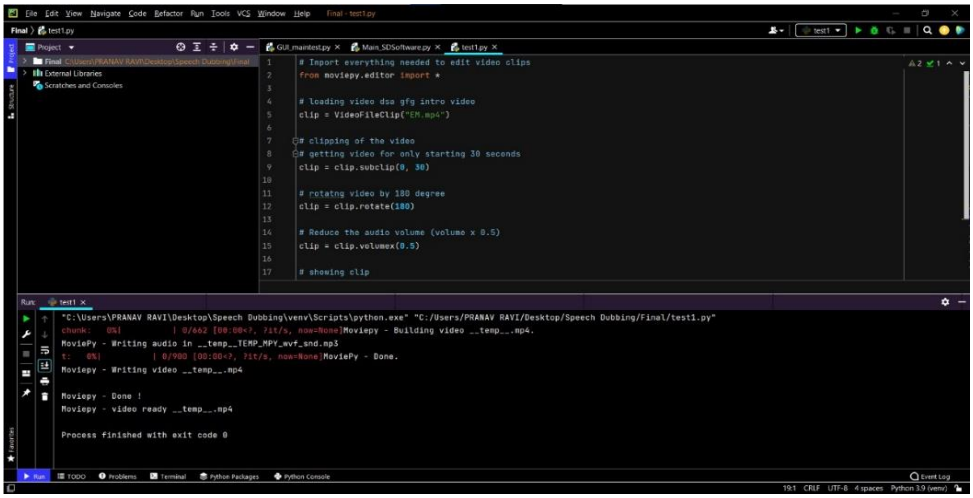


Figure No 2.18

2.3 Anaconda

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS.

It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012.

2.3.1 Benefits of Using Python Anaconda

- 2.3.1 It is free and open-source
- It has more than 1500 Python/R data science packages
- Anaconda simplifies package management and deployment
- It has tools to easily collect data from sources using machine learning and AI
- It creates an environment that is easily manageable for deploying any project
- Anaconda is the industry standard for developing, testing, and training on a single machine
- It has good community support- you can ask your questions there.

2.3.2 Installation of Anaconda

Step 1:

Go to the official website: www.anaconda.org

Step 2:

You can download the installer for Python 3.7 or for Python 2.7 (at the time of writing). And you can download it for a 32-bit or 64-bit machine.

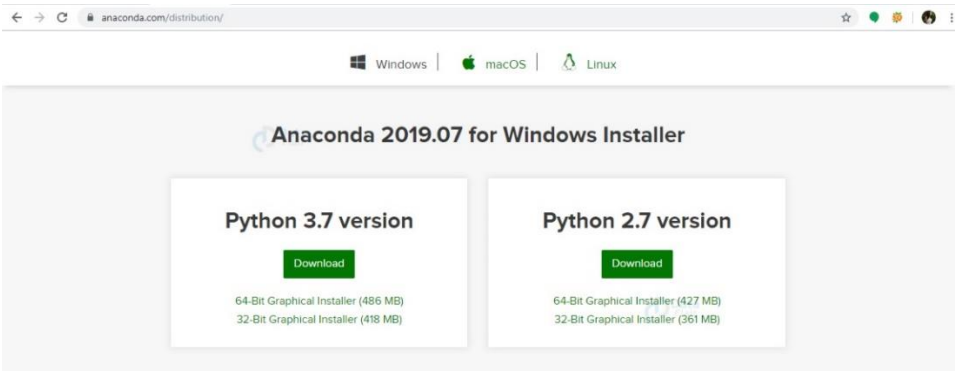


Figure No 2.19

Step 3:

Click on the downloaded .exe to open it. This is the Anaconda setup. Click next.

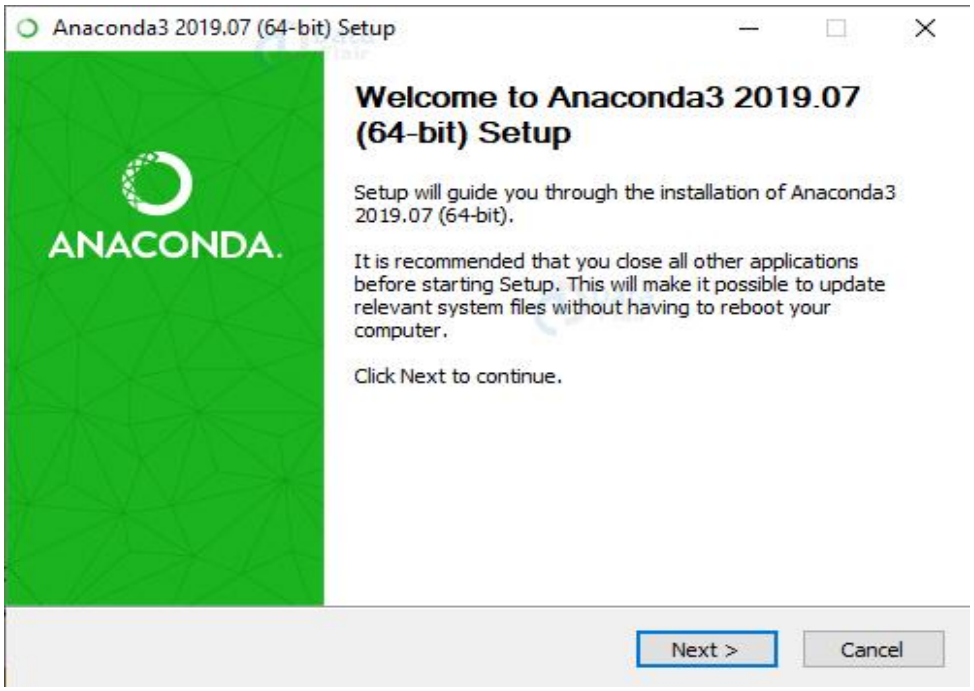


Figure No 2.20

Step 4:

Now, you'll see the license agreement. Click on I Agree

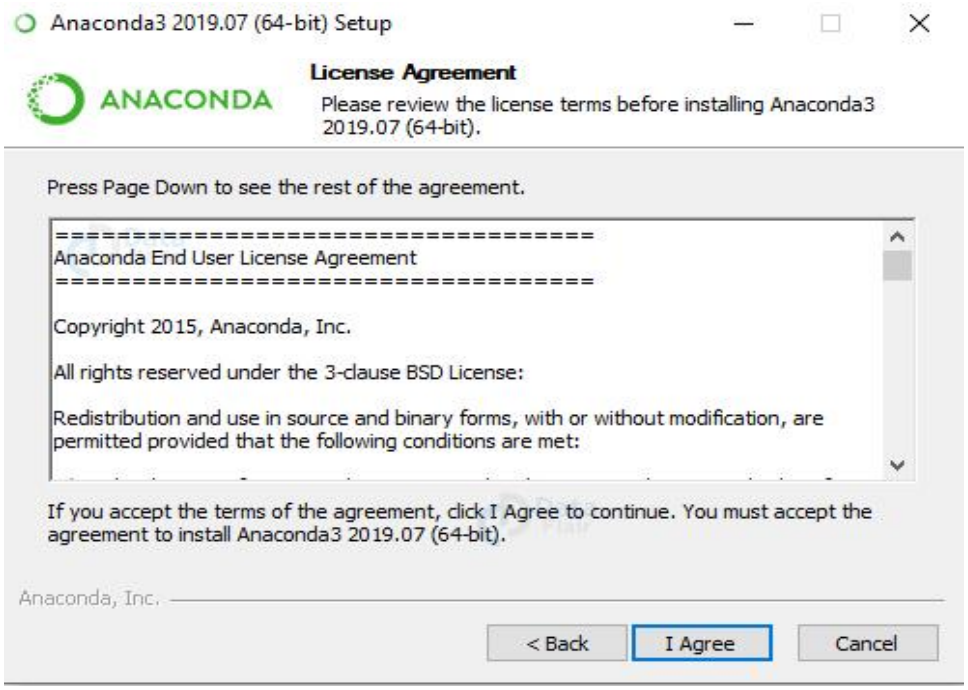


Figure No 2.21

Step 5:

Click finish

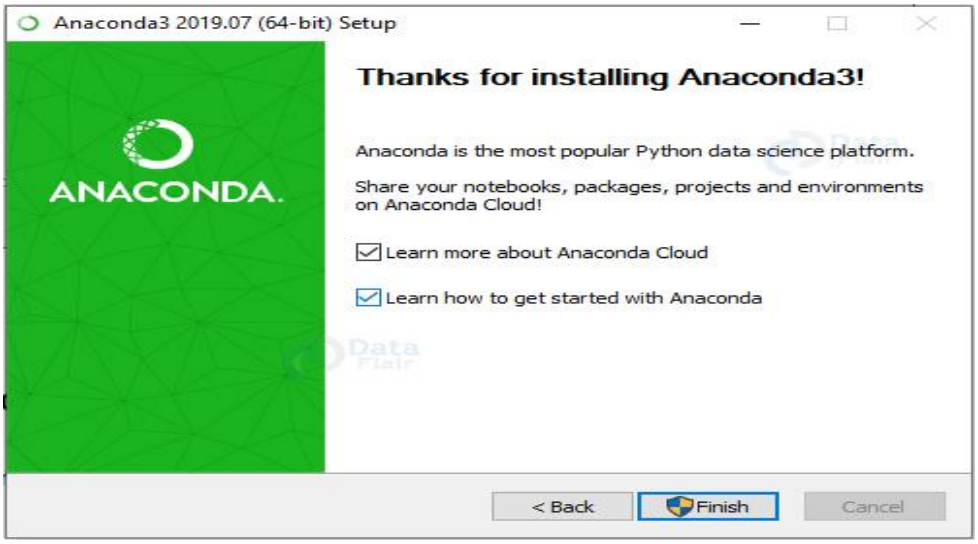


Figure No 2.22

2.4 Qt Creator

Qt Creator is a cross-platform C++, JavaScript, and QML integrated development environment that simplifies GUI application development. It is part of the SDK for the Qt GUI application development framework and uses the Qt API, which encapsulates host OS GUI function calls.

Figure No 2.23

It includes a visual debugger and an integrated WYSIWYG GUI layout and forms designer. The editor has features such as syntax highlighting and autocompletion. Qt Creator uses the C++ compiler from the GNU Compiler Collection on Linux. On Windows, it can use MinGW or MSVC with the default install and can also use Microsoft Console Debugger when compiled from source code. Clang is also supported.

2.4.1 Features of Creator

- Sophisticated Code Editor. Qt Creator's Advanced Code Editor Lets You Code In C++, Qml, JavaScript, Python, And Other Languages
- Version Control
- Integrated Ui Design
- Project & Build Management
- Multiple Targets
- Qt Device Emulator
- Qt Quick Compiler

2.4.2 Installation

Step 1:

Go to official website - www.qt.io

Step 2:

Download Installer

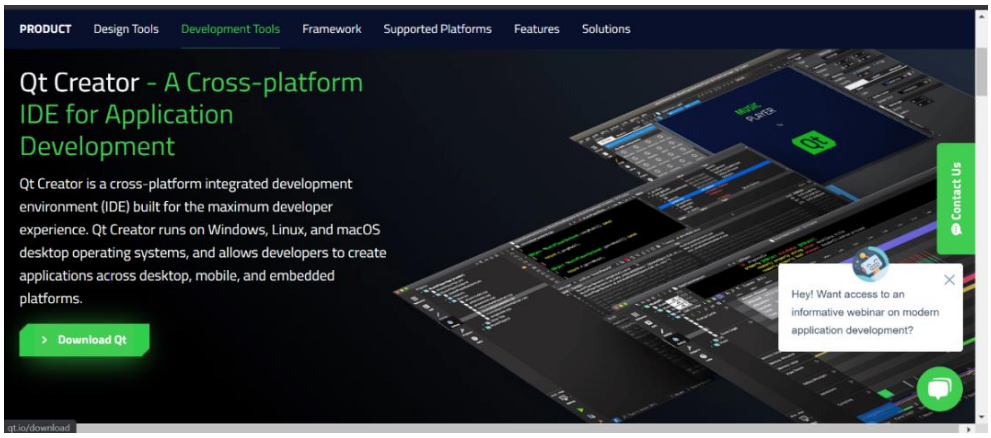


Figure No 2.24

Step 3: Run Installer

The downloaded installer is named something like qt-unified-windows version.exe. Double-click to run it.

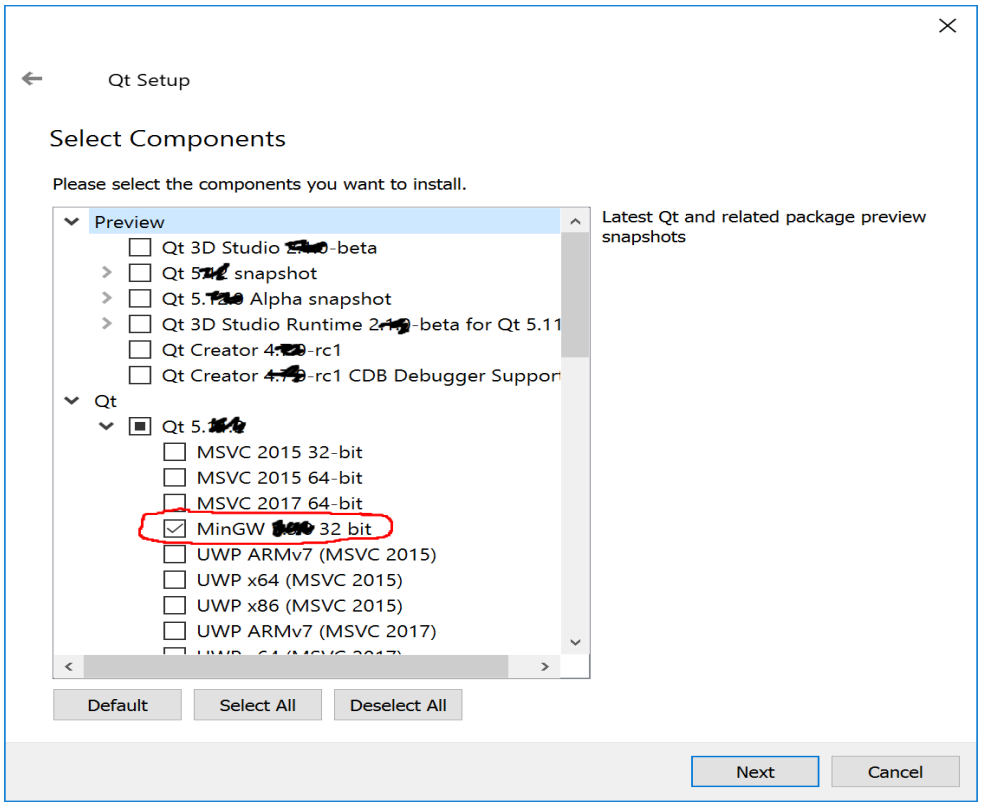


Figure No 2.25

Step 4: Configure Project

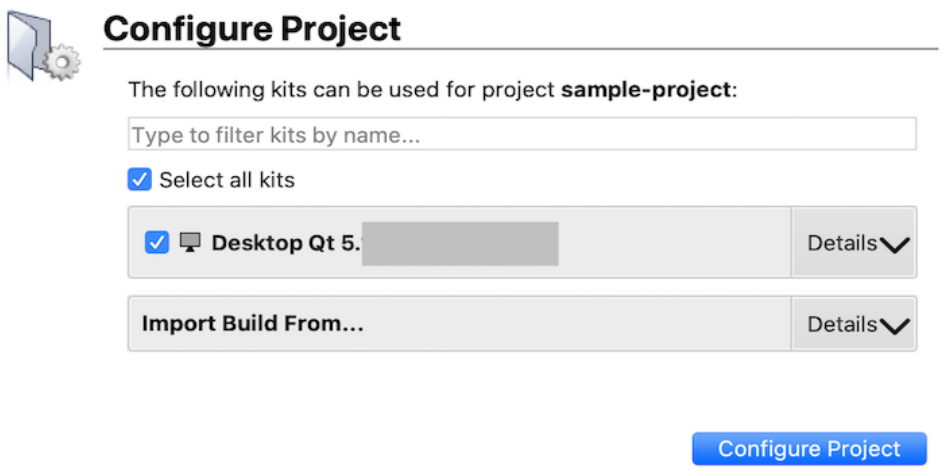


Figure No 2.26

2.4 Qt Designer

Qt Designer is a tool for designing and building graphical user interfaces (GUIs) from Qt widgets. It is possible to compose and customize the widgets or dialogs and test them using different styles and resolutions directly in the editor.

Widgets and forms created with Qt Designer are integrated with programmed code, using the Qt signals and slots mechanism.

Qt Quick Designer is a tool for developing animations by using the declarative programming language QML.

3. Description of Used Python's Library

3.1 PyQt5

PyQt5 is a comprehensive set of Python bindings for Qt v5. It is implemented as more than 35 extension modules and enables Python to be used as an alternative application development language to C++ on all supported platforms including iOS and Android.

PyQt5 may also be embedded in C++-based applications to allow users of those applications to configure or enhance the functionality of those applications.

i. Installation Command

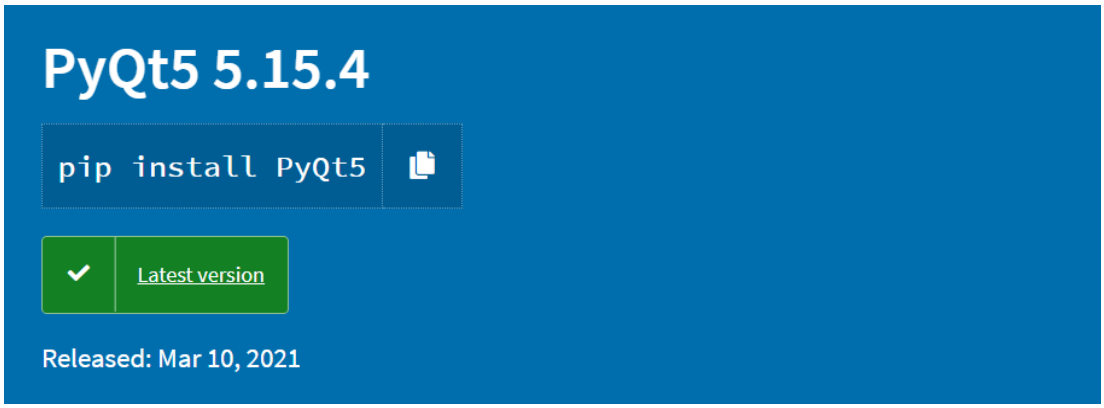


Figure No 3.1

3.2 QtWidgets

- Qt5 comes with a huge number of widgets built-in, from simple text boxes to digital displays, vector graphics canvas, and a full-blown web browser. While you can build perfectly functional applications with the built-in widgets, sometimes your applications will need more.
- The Qt Widgets module provides a set of UI elements to create classic desktop-style user interfaces

3.2.1 Widgets

Widgets are the primary elements for creating user interfaces in Qt. Widgets can display data and status information, receive user input, and provide a container for other widgets that should be grouped. A widget that is not embedded in a parent widget is called a window.

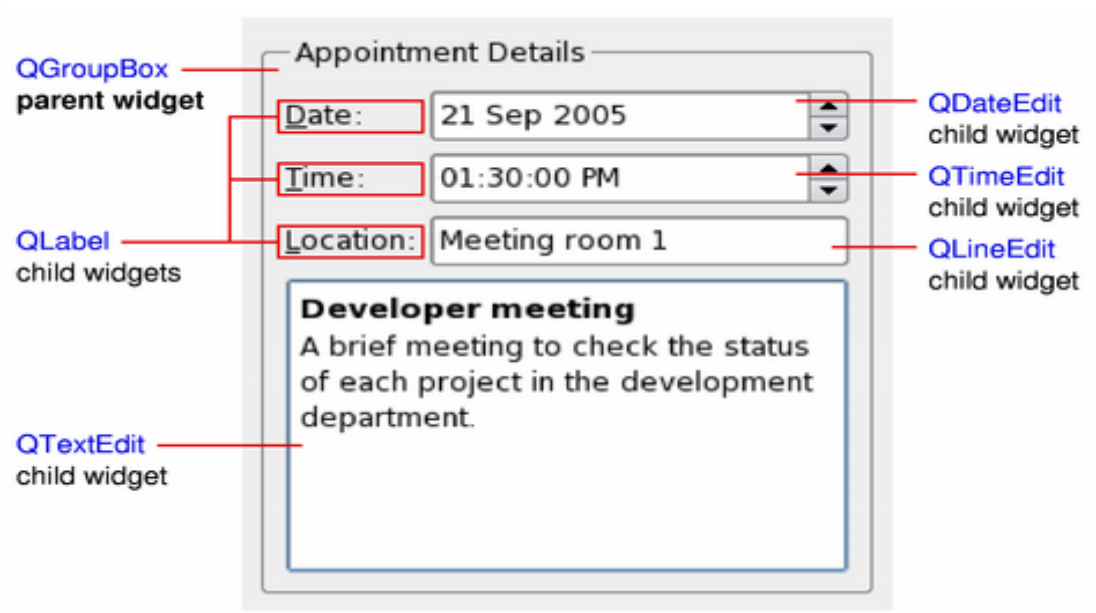


Figure No 3.2

The QWidget class provides the basic capability to render to the screen and to handle user input events. All UI elements that Qt provides are either subclasses of QWidget or are used in connection with a QWidget subclass. Creating custom widgets is done by subclassing QWidget or a suitable subclass and reimplementing the virtual event handlers.

- Window and Dialog Widgets
- Application Main Window
- Dialog Windows
- Keyboard Focus in Widgets

3.2.2 Styles

Styles draw on behalf of widgets and encapsulate the look and feel of a GUI. Qt's built-in widgets use the QStyle class to perform nearly all of their drawing, ensuring that they look exactly like the equivalent native widgets.

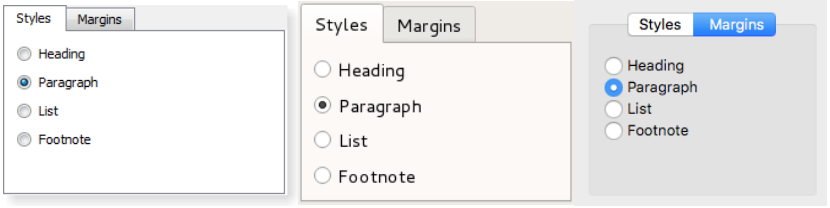


Figure No 3.3

Qt Style Sheets are a powerful mechanism that allows you to customize the appearance of widgets, in addition to what is already possible by subclassing QStyle.

3.2.3 Layouts

Layouts are an elegant and flexible way to automatically arrange child widgets within their container. Each widget reports its size requirements to the layout through the sizeHint and size policy properties, and the layout distributes the available space accordingly.

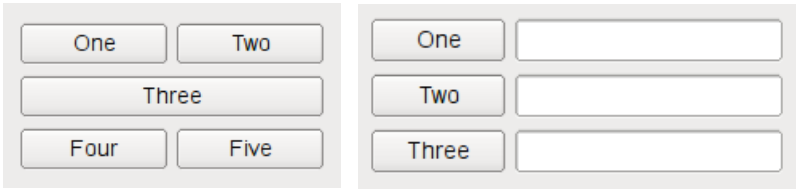


Figure No 3.4

3.2.4 Model/View Classes

The model/view architecture provides classes that manage the way data is presented to the user. Data-driven applications which use lists and tables are structured to separate the data and view using models, views, and delegates.

Figure No 3.5

3.2.5 Graphics View

The Graphics View Framework is for managing and interacting with a large number of custom-made 2D graphical items, and a view widget for visualizing the items, with support for zooming and rotation.

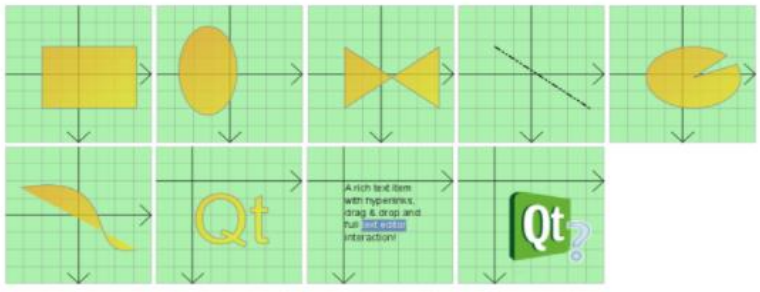


Figure No 3.6

3.2.6 Installation Command

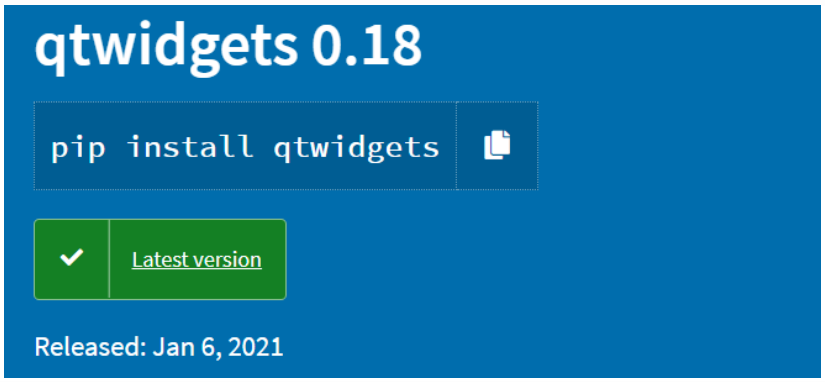


Figure No 3.7

3.3 QtCore

Qt provides thread support in the form of platform-independent threading classes, a thread-safe way of posting events, and signal-slot connections across threads. Multithreaded programming

3.3.1 Core Functionalities

- Qt Core adds these features to C++:
- a very powerful mechanism for seamless object communication called signals and slots
- queryable and designable object properties
- hierarchical and queryable object trees that organize
- object ownership in a natural way with guarded pointers (QPointer)
- a dynamic cast that works across library boundaries

3.3.2 Threading and Concurrent Programming

Qt provides thread support in the form of platform-independent threading classes, a thread-safe way of posting events, and signal-slot connections across threads. Multithreaded programming is also a useful paradigm for performing time-consuming operations without freezing the user interface of an application.

The Thread Support in Qt page contains information on implementing threads in applications. Additional concurrent classes are provided by the Qt Concurrent module.

3.3.3 Input/Output, Resources, and Containers

Qt provides a resource system for organizing application files and assets, a set of containers, and classes for receiving input and printing output.

- Container Classes
- Serializing Qt Data Types
- Implicit Sharing

In addition, Qt Core provides a platform-independent mechanism for storing binary files in the application's executable.

3.3.4 Additional Frameworks

Qt Core also provides some of Qt's key frameworks.

- The Animation Framework
- JSON Support in Qt
- The State Machine Framework
- How to Create Qt Plugins
- The Event System

3.4 Qt GUI

The Qt GUI module provides classes for windowing system integration, event handling, OpenGL and OpenGL ES integration, 2D graphics, basic imaging, fonts, and text.

3.5 PySide

PySide is the Python Qt bindings project, providing access to the complete Qt 4.8 framework as well as to generator tools for rapidly generating bindings for any C++ libraries.

3.5.1 Compatibility

PySide requires Python 2.6 or later and Qt 4.6 or better.

Note

Qt 5.x is currently not supported.

3.2.6 Installation Command

3.5.2 Installation Command

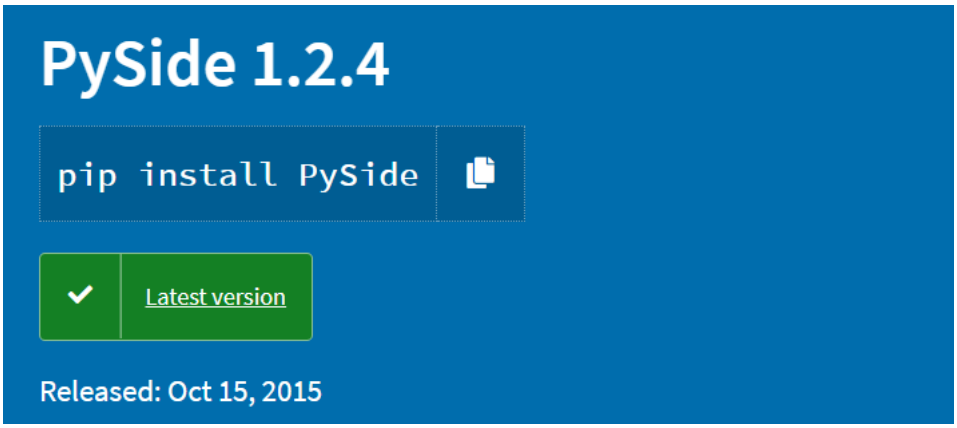


Figure No 3.8

3.6 Speech Recognition

Speech Recognition is an important feature in several applications used such as home automation, artificial intelligence, etc. This article aims to provide an introduction on how to make use of the SpeechRecognition library of Python. This is useful as it can be used on microcontrollers such as Raspberry Pis with the help of an external microphone.

3.6.1 Requirements

To use all of the functionality of the library, you should have:

- Python 2.6, 2.7, or 3.3+ (required)
- PyAudio 0.2.11+ (required only if you need to use microphone input, Microphone)
- PocketSphinx (required only if you need to use the Sphinx recognizer, recognizer_instance.recognize_sphinx)
- Google API Client Library for Python (required only if you need to use the Google Cloud Speech API, recognizer_instance.recognize_google_cloud)
- FLAC encoder (required only if the system is not x86-based Windows/Linux/OS X)

3.6.2 Installation Command

Figure No 3.9

3.7 Deep Translator

3.7.1 Motivation

I needed to translate a text using python. It was hard to find a simple way to do it. Other libraries can be used for this task, but most of them are buggy, not free, limited, not supported anymore, or complex to use.

Therefore, I decided to build this simple tool. It is 100% free, unlimited, easy to use, and provides support for all languages.

My goal was to integrate support for multiple famous translators in this tool.

3.7.2 When you should use it

- If you want to translate text using python
- If you want to translate from a file
- If you want to get translations from many sources and not only one
- If you want to automate translations
- If you want to compare different translations
- If you want to detect language automatically

3.7.3 Why you should use it

- It’s the only python tool that integrates many translators
- multi-language support
- supports batch translation
- High level of abstraction
- Automatic language detection
- Easy to use and extend
- Support for most famous universal translators
- Stable and maintained regularly
- The API is very easy to use
- Proxy integration is supported

3.7.4 Features

- Support for google translate
- Support for the Microsoft translator (version >= 1.3.5)
- Support for Pons translator
- Support for the Linguee translator
- Support for the Mymemory translator
- Support for the Yandex translator (version >= 1.2.1)
- Support for the QCRI translator (version >= 1.2.4)
- Support for the DeepL translator (version >= 1.2.5)
- Support for the Papago translator (version >= 1.4.4)
- Support for proxy usage
- Automatic single language detection
- Batch language detection

3.7.5 Installation Command

Figure No 3.10

3.8 Googletrans

Googletrans is a free and unlimited python library that implemented Google Translate API. This uses the Google Translate Ajax API to make calls to such methods as detecting and translate.

Compatible with Python 3.6+.

3.8.1 Features

- Fast and reliable - it uses the same servers that translate.google.com uses
- Auto language detection
- Bulk translations
- Customizable service URL
- HTTP/2 support

3.8.2 installation Command

Figure No 3.11

3.9 Tkinter

Tkinter is an open-source, portable graphical user interface (GUI) library designed for use in Python scripts.
Tkinter relies on the Tk library, the GUI library used by Tcl/Tk and Perl, which is in turn implemented in C. Therefore, Tkinter can be said to be implemented using multiple layers.

3.9.1 Advantages of Tkinter

i. Layered approach

The layered approach used in designing Tkinter gives Tkinter all of the advantages of the TK library. Therefore, at the time of creation, Tkinter inherited the benefits of a GUI toolkit that had been given time to mature. This makes early versions of Tkinter a lot more stable and reliable than if it had been rewritten from scratch. Moreover, the conversion from Tcl/Tk to Tkinter is trivial, so that Tk programmers can learn to use Tkinter very easily.

ii. Accessibility

Learning Tkinter is very intuitive, and therefore quick and painless. The Tkinter implementation hides the detailed and complicated calls in simple, intuitive methods. This is a continuation of the Python way of thinking since the language excels at quickly building prototypes. It is therefore expected that its preferred GUI library be implemented using the same approach. For example, here is the code for a typical “Hello world”-like application:

```
from Tkinter import *
root = Tk( )
root.title("A simple application")
root.mainloop( )
```

The first 2 lines allow creating a complete window. Compared to MFC programming, it makes no doubt that Tkinter is simple to use. The third line sets the caption of the window, and the fourth one makes it enter its event loop.

iii. Portability

Python scripts that use Tkinter do not require modifications to be ported from one platform to the other. Tkinter is available for any platform that Python is implemented for, namely Microsoft Windows, X Windows, and Macintosh. This gives it a great advantage over most competing libraries, which are often restricted to one or two platforms. Moreover, Tkinter will provide the native look and feel of the specific platform it runs on.

iv. Availability

Tkinter is now included in any Python distribution. Therefore, no supplementary modules are required to run scripts using Tkinter.

3.9.2 Drawbacks of Tkinter

The multi-layered approach taken in designing Tkinter can have some disadvantages as far as execution speed is concerned. While this could constitute a problem with older, slower machines, most modern computers are fast enough to cope with the extra processing in a reasonable time. When speed is critical, proper care must be taken to write code that is as efficient as possible.

3.10 GTTS

gTTS (Google Text-to-Speech), a Python library and CLI tool to interface with Google Translate's text-to-speech API. Write spoken mp3 data to a file, a file-like object (byte string) for further audio manipulation, or stdout. (Google Text-to-Speech)

3.9.1 Features

- Customizable speech-specific sentence tokenizer that allows for unlimited lengths of text to be read, all while keeping proper intonation, abbreviations, decimals, and more;
- Customizable text pre-processors can, for example, provide pronunciation corrections.

3.9.2 Installation Command

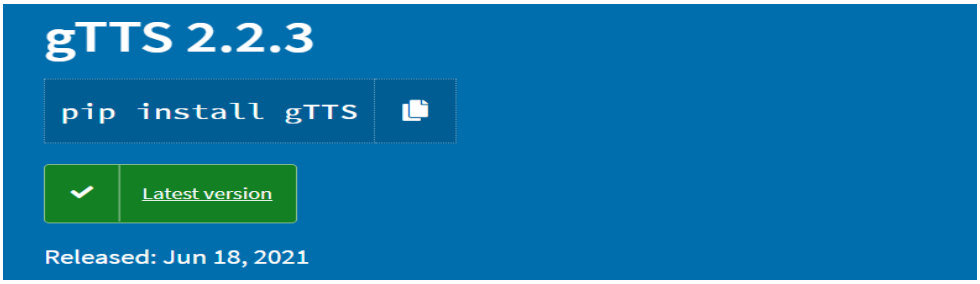


Figure No 3.12

3.11 Moviepy

MoviePy is a Python module for video editing, which can be used for basic operations (like cuts, concatenations, title insertions), video compositing (a.k.a. non-linear editing), video processing, or to create advanced effects. It can read and write the most common video formats, including GIF.

3.10.1 Advantages of MoviePy

- Simple: Basic operations can be done in one line, code is easy to learn and easy to understand for newcomers.
- Flexible: Users have total control over the frames of the video and audio, and creating their effects is easy as Py.
- Portable: The code uses very common software such as Numpy and FFMPEG. And can run on almost any machine with almost any version of Python.

3.10.2 Disadvantages of MoviePy :

- MoviePy cannot yet stream videos (read from a webcam or render a video live on a distant machine)
- It is not designed for video processing involving many successive frames of a movie (like video stabilization, you'll need another software for that)
- Memory problems can arise if the user use many video, audio, and image sources at the same time (>100)

3.10.3 Installation Command

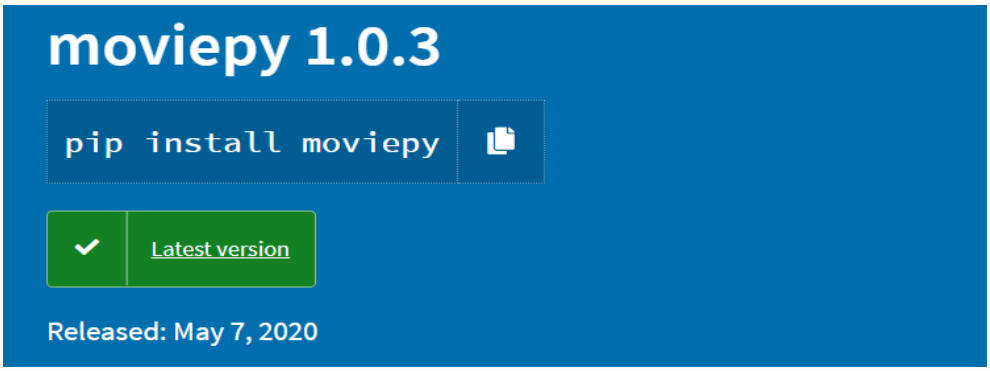


Figure No 3.13

4. Project Methodology

4.1 Block Diagram

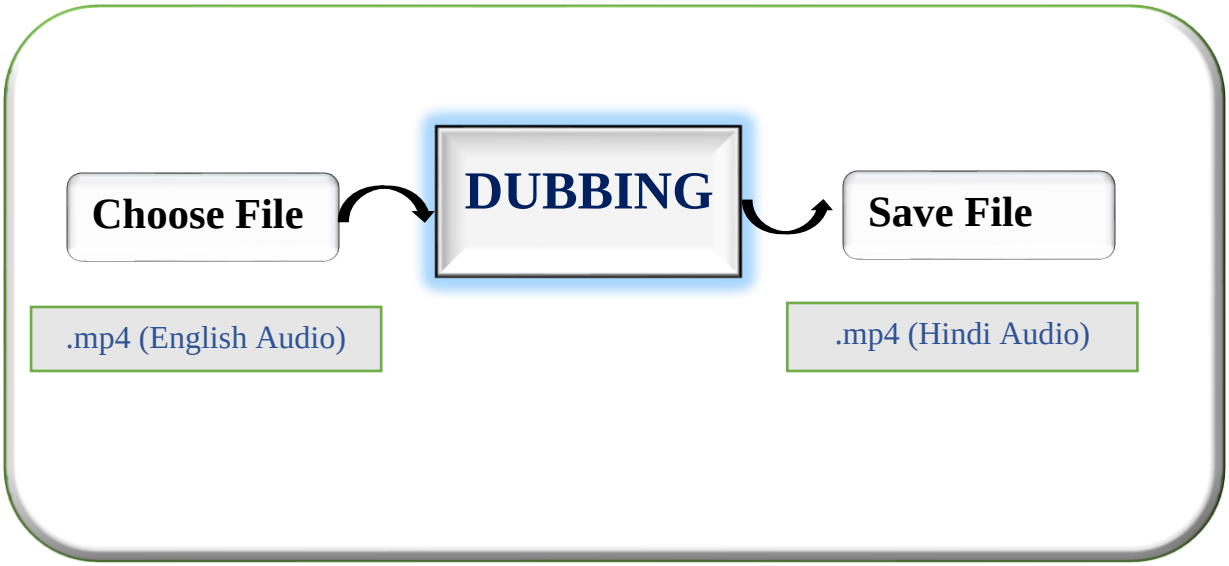
A **block diagram** is a diagram of a system in which the principal parts or functions are represented by **blocks** connected by lines that show the relationships of the blocks. They are heavily used in engineering in hardware design, electronic design, software design, and process flow diagrams.

Block diagrams are typically used for higher-level, less detailed descriptions that are intended to clarify overall concepts without concern for the details of implementation. Contrast this with the schematic diagrams and layout diagrams used in electrical engineering, which show the implementation details of electrical components and physical construction.

4.1.1 Block

Blocks or system building blocks are modular structures within **SysML (Systems Modeling Language)** that represent statistical concepts and objects in the system. In the world of software development, blocks describe data elements and operators or control flow elements.

4.1.2 Project Block Diagram



4.2 Project Algorithm

Step 1:

Starting the Execution of Program

Step 2:

Asking for input .mp4 English Audio file.

askopenfilename()

Step 3:

Store input file as a variable enteredVideoFile.

enteredVideoFile = askopenfilename()

Step 4:

Convert the entered .mp4 file to .mp3 and after that convert .mp3 to .wav file format and save .wav file as GeneratedSpeech.wav.

```
AudioClipFile = AudioFileClip(enteredVideoFile)
AudioClipFile.write_audiofile(TranscribedAudioFileName)
TranscribedAudioFileName = "GeneratedSpeech.wav"
```

Step 5:

Conquer the GeneratedSpeech.wav.

```
duration = frameNumber / float(frameRate)
total_duration = math.ceil(duration / 60)
```

Step 6:

Recognize audio of every conquer part of GeneratedSpeech.wav file and write it as text in the generatedTxtFile.txt file.

```
for i in range(0, total_duration):
    with sr.AudioFile(TranscribedAudioFileName) as source:
        audio = recognizeVoice.record(source, offset=i * 60, duration=60)
    Text_file = open("generatedTxtFile.txt", "a")
    Text_file.write(recognizeVoice.recognize_google(audio))
    Text_file.write(" ")
    Text_file.close()
```

Step 7:

Take input generatedTxtFile.txt file and Translate it into Hindi texts and store in translatedHindi as a variable.

```
translatedHindi = GoogleTranslator(source='english',
target='hindi').translate_file('generatedTxtFile.txt')
```

Step 8:

Generating Hindi Audio .mp3 file and save it as TranslatedHindiVoice.mp3

```
language = 'hi'
output = gTTS(text=translatedHindi, lang=language, slow=False)
output.save("TranslatedHindiVoice.mp3")
```

Step 9:

Mute audio of the input video file and overlap generated audio on the input video and save it as output.mp4.

```
combine_audio(enteredVideoFile, "TranslatedHindiVoice.mp3","output.mp4")
```

Step 10:

Remove all unnecessary generated files like GeneratedSpeech.wav, generatedTxtFile.txt, TranslatedHindiVoice.mp3

```
directory = os.getcwd()
test = os.listdir(directory)
for item in test:
    if item.endswith(".mp3"):
        os.remove(os.path.join(directory, item))
for item in test:
    if item.endswith(".wav"):
        os.remove(os.path.join(directory, item))
for item in test:
    if item.endswith(".txt"):
        os.remove(os.path.join(directory, item))
```

Step 11:

Ending the Execution of the Program.

4.3 Flow Chart

A **flowchart** is a graphical representation of **steps**. It was originated from computer science as a tool for representing **algorithms** and **programming** logic but had extended to use in all other kinds of processes.

Flowcharts play an extremely important role in displaying information and assisting reasoning. They help us visualize complex processes, or make explicit the structure of problems and tasks.

A flowchart can also be used to define a process or project to be implemented.

4.3.1 Flow Chart Symbols

Different flowchart shapes have different conventional meanings. The meanings of some of the more common shapes are as follows:

Terminator

The terminator symbol represents the starting or ending point of the system.

Process

A box indicates some particular operation.

Document

This represents a printout, such as a document or a report.

Decision

A diamond represents a decision or branching point. Lines coming out from the diamond indicates different possible situations, leading to different sub-processes.

Data

It represents information entering or leaving the system. An input might be an order from a customer. Output can be a product to be delivered.

On-Page Reference

This symbol would contain a letter inside. It indicates that the flow continues on a matching symbol containing the same letter somewhere else on the same page.

Off-Page Reference

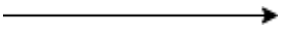
This symbol would contain a letter inside. It indicates that the flow continues on a matching symbol containing the same letter somewhere else on a different page.

Delay or Bottleneck

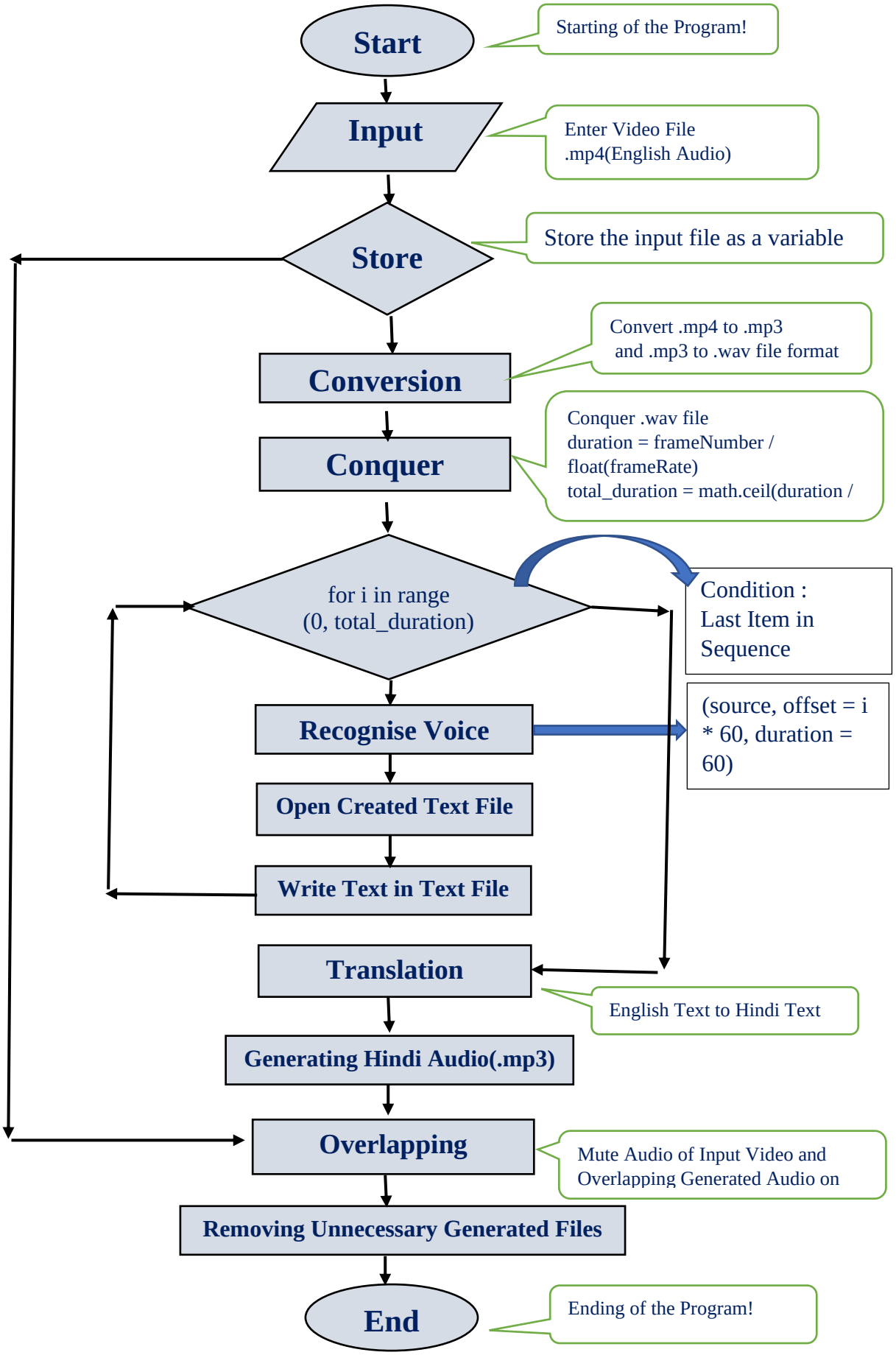
Identifies a delay or a bottleneck.

Flow

Lines represent the flow of the sequence and direction of a process.



4.3.2 Project Flow Chart Diagram



5. Project Implementation

5.1 Front End of Project

5.1.1 Create Window:

```
import sys

from PyQt5.QtCore import Qt
from PyQt5.QtWidgets import QApplication, QLabel,
QMainWindow

class Window(QMainWindow):
    """Main Window."""
    def __init__(self, parent=None):
        """Initializer."""
        super().__init__(parent)
        self.setWindowTitle("Speech Dubbing Software")
        self.resize(400, 200)
        self.centralWidget = QLabel("dashboard")
        self.centralWidget.setAlignment(Qt.AlignHCenter |
Qt.AlignVCenter)
        self.setCentralWidget(self.centralWidget)

if __name__ == "__main__":
    app = QApplication(sys.argv)
    win = Window()
    win.show()
    sys.exit(app.exec_())

        #Snip…
```

Main Window Screen :

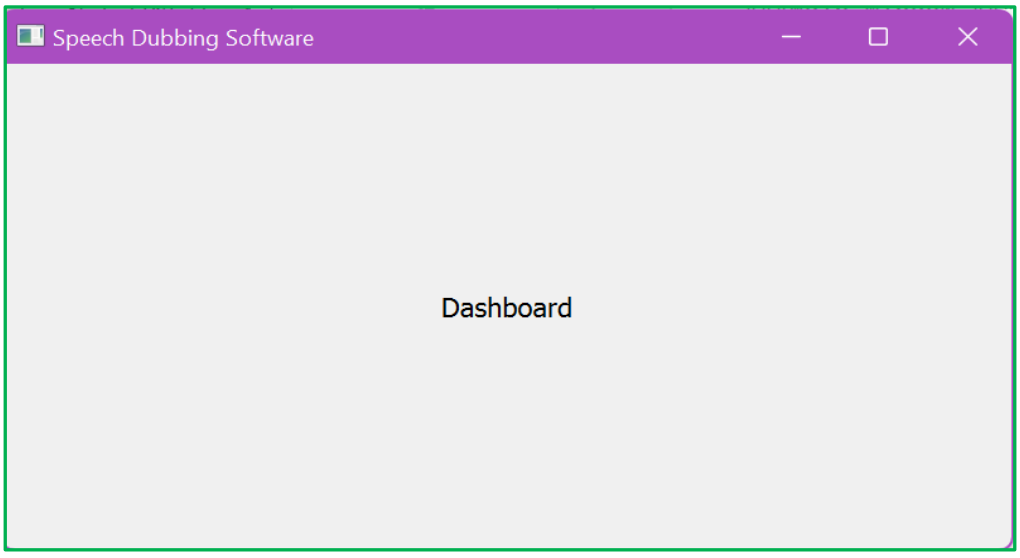


Figure No 5.1

5.1.2 Create Menu & Tool Bar

```
from PyQt5.QtWidgets import QMenuBar
from PyQt5.QtWidgets import QMenu

# Snip...
class Window(QMainWindow):
    # Snip...
    self._createMenuBar()
    self._createToolBars()

    def _createToolBars(self):
        # Using a title
        fileToolBar = self.addToolBar("File")
        # Using a QToolBar object
        editToolBar = QToolBar("Edit", self)
        self.addToolBar(editToolBar)
        # Using a QToolBar object and a toolbar area
        helpToolBar = QToolBar("Help", self)
        self.addToolBar(Qt.LeftToolBarArea, helpToolBar)

    def _createMenuBar(self):
        menuBar = self.menuBar()
        # Creating menus using a QMenu object
        fileMenu = QMenu("&File", self)
        menuBar.addMenu(fileMenu)
        # Creating menus using a title
        editMenu = menuBar.addMenu("&Edit")
        helpMenu = menuBar.addMenu("&Help")

        menuBar = QMenuBar(self)
        self.setMenuBar(menuBar)

                    #Snip…
```

Output:

- Main Window Screen with menu & Tool Bar:

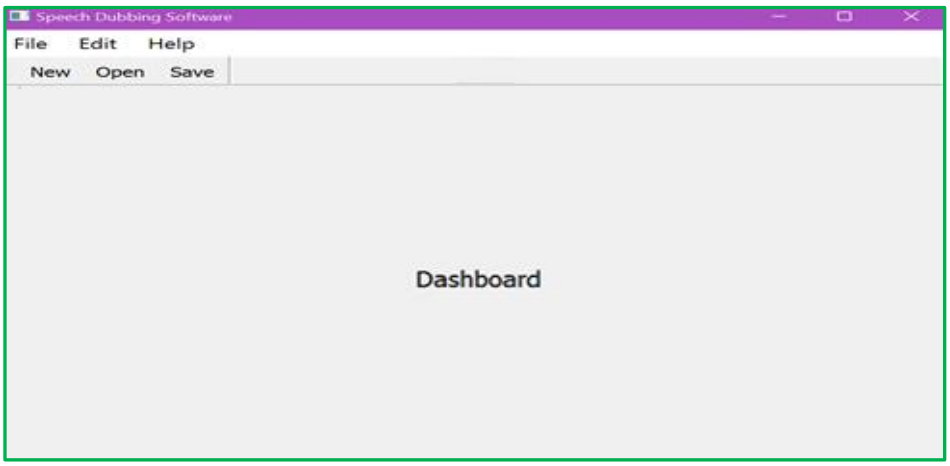


Figure No 5.2

5.1.3 Create Button with Dashboard

```
#Snip
chooseButton= QPushButton('Choose File', self)
chooseButton.resize(100, 32)
chooseButton.move(100, 100)
chooseButton.clicked.connect(self.chooseFile_)

#chooseButton.clicked.connect(self.chooseFile_)

saveButton = QPushButton('Save', self)
saveButton.resize(100, 32)
saveButton.move(315, 200)

quitButton = QPushButton('Quit', self)
quitButton.resize(60, 25)
quitButton.move(420, 280)

quitButton.clicked.connect(self.quiteApp)

  #Snip…
```

- Main Window Screen with a label, Choose Button, Save Button & Quit Button.

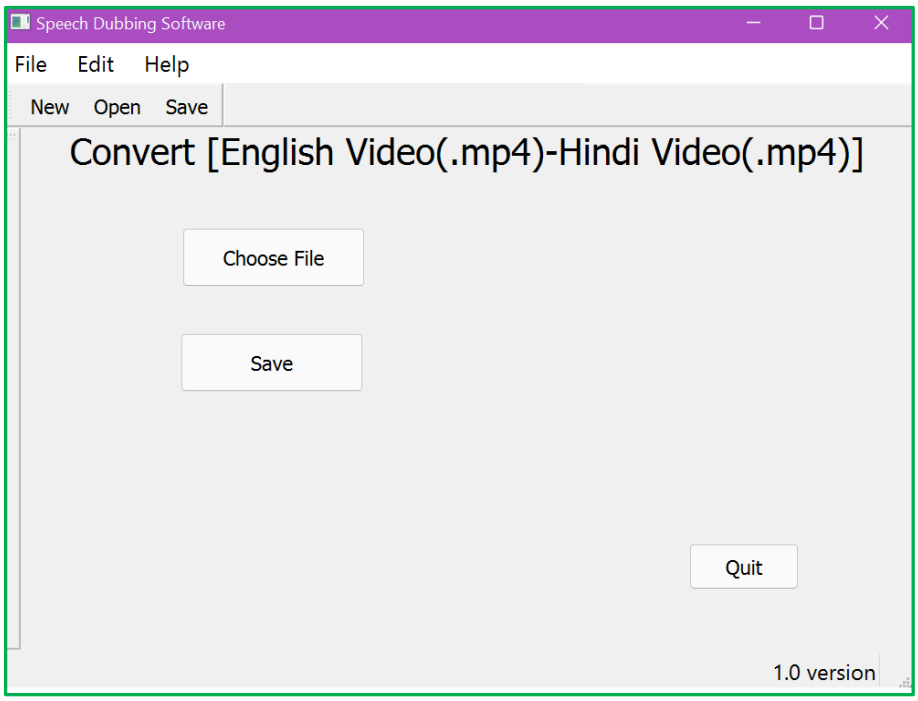


Figure No 5.3

5.2 Back-End of Project

5.2.1 Menu Bar

```
#Snip…

def _createActions(self):
    # Creating action using the first constructor
    self.newAction = QAction(self)
    self.newAction.setText("&New")
    self.newAction.setIcon(QIcon(":Framework/file-new_.svg"))
    self.openAction = QAction(QIcon(":file-open.svg"), "&Open...",
self)
    self.saveAction = QAction(QIcon(":file-save.svg"), "&Save",
self)

    # Creating actions using the second constructor
    self.openAction = QAction("&Open...", self)
    self.saveAction = QAction("&Save", self)
    self.exitAction = QAction("&Exit", self)

def _createContextMenu(self):
    # Setting contextMenuPolicy
    self.centralWidget.setContextMenuPolicy(Qt.ActionsContextMenu)
    # Populating the widget with actions
    self.centralWidget.addAction(self.newAction)
    self.centralWidget.addAction(self.openAction)
    self.centralWidget.addAction(self.saveAction)

def newFile(self):
 self.centralWidget.setText("<b>File > New</b> clicked")
def openFile(self):
    # Logic for opening an existing file goes here...
    self.centralWidget.setText("<b>File > Open...</b> clicked")
def saveFile(self):
    # Logic for saving a file goes here...
    self.centralWidget.setText("<b>File > Save</b> clicked")
def about(self):
    # Logic for showing an about dialog content goes here...
    self.centralWidget.setText("<b>Help > About...</b> In this
software"
                " we convert the .mp4Video_en to .mp4Video_hn")


    #connect the menu file to function
 def _connectActions(self):
    #Connect File actions
    self.newAction.triggered.connect(self.newFile)
    self.openAction.triggered.connect(self.openFile)
    self.saveAction.triggered.connect(self.saveFile)
      #Snip…
```

```
#Snip…
def getWordCount(self):
    # Logic for computing the word count goes here...
    return 1.0
#Create statusBar
def _createStatusBar(self):
    self.statusbar = self.statusBar()
    # Adding a temporary message
    self.statusbar.showMessage("Created By:Team Ciphers (BPMCE)",
90000)
    # Adding a permanent message
    self.wcLabel = QLabel(f"{self.getWordCount()} version")
    self.statusbar.addPermanentWidget(self.wcLabel)
#Snip…
```

5.2.2 Choose Button

```
 #Snip...
from tkinter.filedialog import *
import wave, math, contextlib
import speech_recognition as sr
from moviepy.editor import AudioFileClip
from deep_translator import GoogleTranslator
from gtts import gTTS
import moviepy.editor as mpe
import sys
import os

      #Snip...
      chooseButton = QPushButton('Choose File', self)
      chooseButton.resize(100, 32)
      chooseButton.move(100, 100)
      chooseButton.clicked.connect(self.chooseFile_)

def chooseFile_(self):
    mainSDS()  # calling mainSDS file for converting
def mainSDS():
    # from desktop for file access use askopenfilename()
    enteredVideoFile = askopenfilename()
   # enterText = input("Enter File Name:") + ".mp4"
    # to extract the audio from videoFile using AudioFileClip()
    AudioClipFile = AudioFileClip(enteredVideoFile)
    # declaration of TranscribedAudioFileName
    TranscribedAudioFileName = "GeneratedSpeech.wav"
    AudioClipFile.write_audiofile(TranscribedAudioFileName)
'''
    "LOGIC PART"
    duration = frameNumber / float(frameRate)
    total_duration = math.ceil(duration / 60)
'''
```

```
#Snip…
with contextlib.closing(wave.open(TranscribedAudioFileName, 'r')) as
file:
    # using getnframes(), for find the no of frame
    frameNumber = file.getnframes()
    # using getframerate(), for find the rate of frame
    frameRate = file.getframerate()
    duration = frameNumber / float(frameRate)
    total_duration = math.ceil(duration / 60)
    recognizeVoice = sr.Recognizer()

'''
    "LOGIC PART"
    audio = r.record(source, offset=i*60, duration=60
 '''
for i in range(0, total_duration):
    with sr.AudioFile(TranscribedAudioFileName) as source:
        audio = recognizeVoice.record(source, offset=i * 60,
duration=60)
    Text_file = open("generatedTxtFile.txt", "a")
    Text_file.write(recognizeVoice.recognize_google(audio))
    Text_file.write(" ")
    Text_file.close()
print(".mp4 to .txt okay...")

# translation english to hindi
translatedHindi = GoogleTranslator(source='english',
target='hindi').translate_file('generatedTxtFile.txt')
# print(translatedHindi)
print("translation okay...")

# Hindi text to hindi mp3
language = 'hi'
output = gTTS(text=translatedHindi, lang=language, slow=False)
output.save("TranslatedHindiVoice.mp3")
# os.system("start TranslatedHindiVoice.mp3")
print("Hindi text to mp3 okay...")
# Overlap the video and audio
print("Video & Audio Overlap Started... ")

def combine_audio(vidname, audname, outname, fps=60):
    my_clip = mpe.VideoFileClip(vidname)
    audio_background = mpe.AudioFileClip(audname)
    final_clip = my_clip.set_audio(audio_background)
    final_clip.write_videofile(outname, fps=fps)

combine_audio(enteredVideoFile, "TranslatedHindiVoice.mp3","output.mp4")
print("Video & Audio Overlap Successfully okay...")

# remove .txt, .wav, .mp3 file
import os

directory = os.getcwd()
test = os.listdir(directory)
#Snip…
```

```
#Snip…
for item in test:
    if item.endswith(".mp3"):
        os.remove(os.path.join(directory, item))

for item in test:
    if item.endswith(".wav"):
        os.remove(os.path.join(directory, item))
for item in test:
    if item.endswith(".txt"):
        os.remove(os.path.join(directory, item))
      #Snip…
```

5.2.3 Save Button

```
#Snip…
def saveFile_(self):
        userInfo = QMessageBox.question(self,
"Conformation", "Your file Converted.mp4 is Save at
DEFAULT LOCATION(D:\Speech Dubbing)!",
             QMessageBox.Yes | QMessageBox.No)
        if userInfo == QMessageBox.Yes:
            original = r'F:\work\Major
Project\Converted.mp4'
            target = 'D:\Speech Dubbing'
            shutil.move(original, target)
            print("Saving file at the desire location is
SUCCESSFULLY done.")
        elif userInfo == QMessageBox.No:
            pass
#Snip…
```

5.2.4 Quit Button

```
#Snip…
quitButton = QPushButton('Quit', self)
quitButton.resize(60, 25)
quitButton.move(420, 280)
quitButton.clicked.connect(self.quiteApp)
def quiteApp(self):
    userInfo = QMessageBox.question(self, "Conformation",
"Do you want the Quit the Application",
QMessageBox.Yes | QMessageBox.No)

    if userInfo == QMessageBox.Yes:
        app.quit()
    elif userInfo == QMessageBox.No:
        pass
```

6. How To Use

Step 1:

Download the executable file:

Click on the below link for the Speech Dubbing Software Executable file that is,

Speech Dubbing Software.exe.

https://drive.google.com/file/d/1pFVDR_Rq0M1kpqdkQHqZiV47p1GQzn09/view?usp=sharing

Step 2:

Installation/Configuration

After getting the .exe file you can create a desktop shortcut of that .exe file as following–

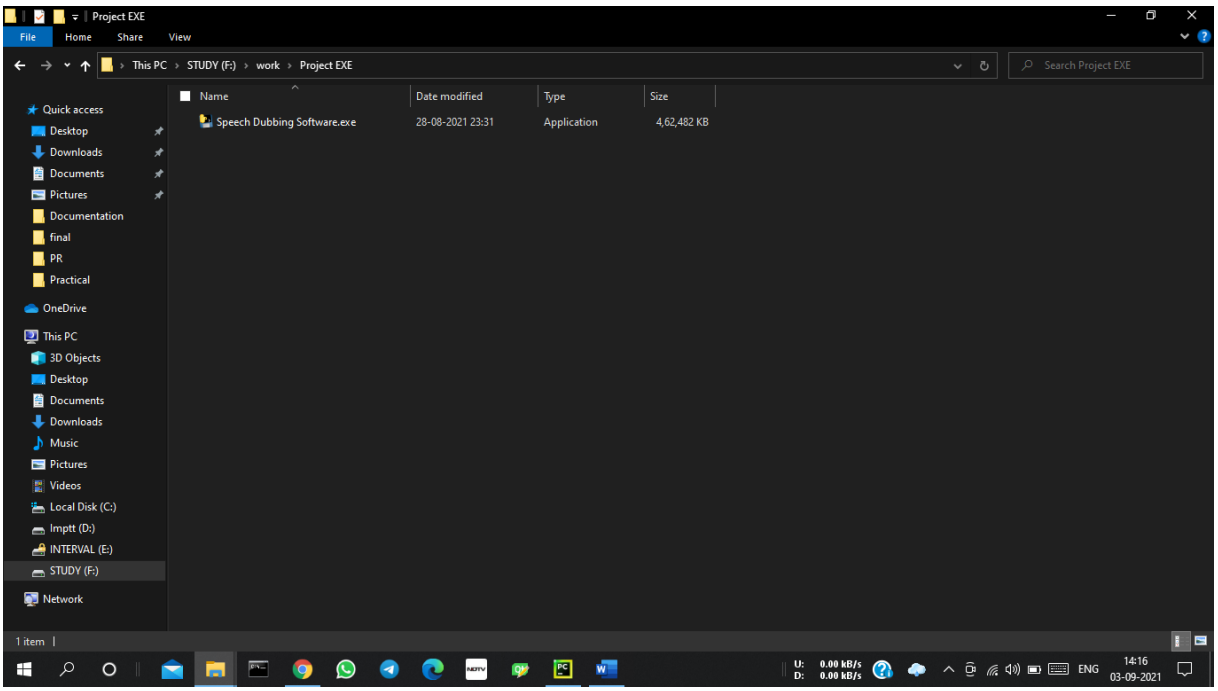


Figure No 6.1

1. Right-click on the .exe file. Now a popup window will be open.
2. Click on Create shortcut.

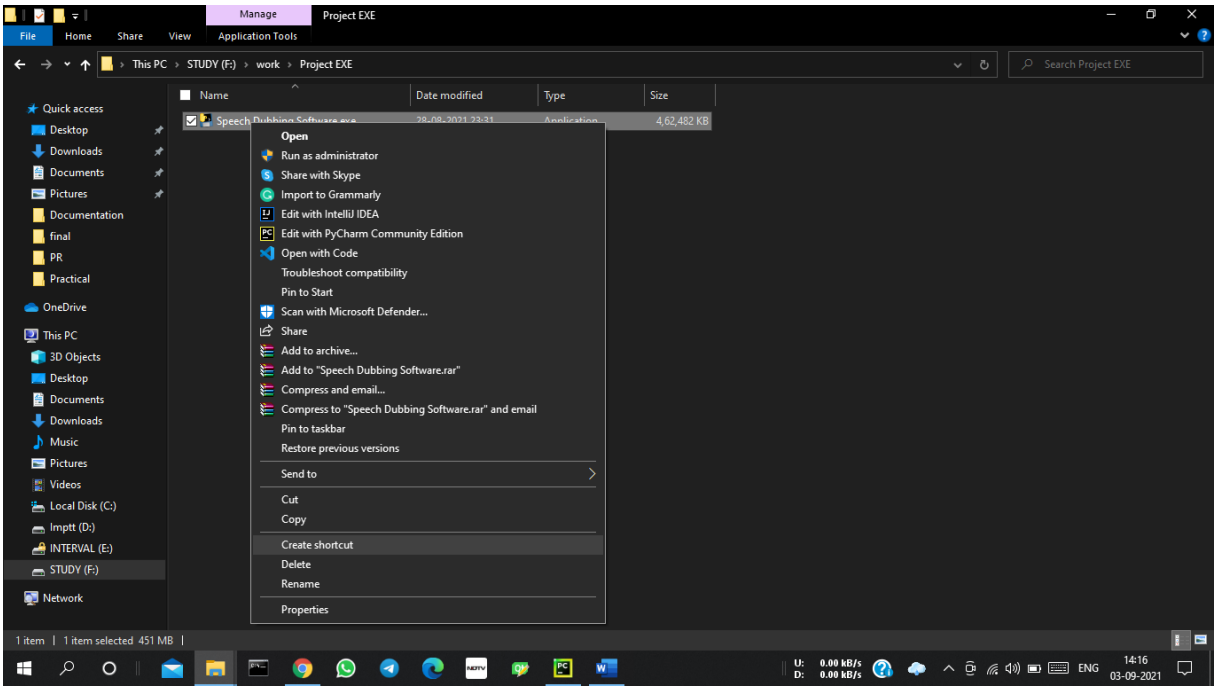


Figure No 6.2

3. Now, the shortcut of that file will be created.

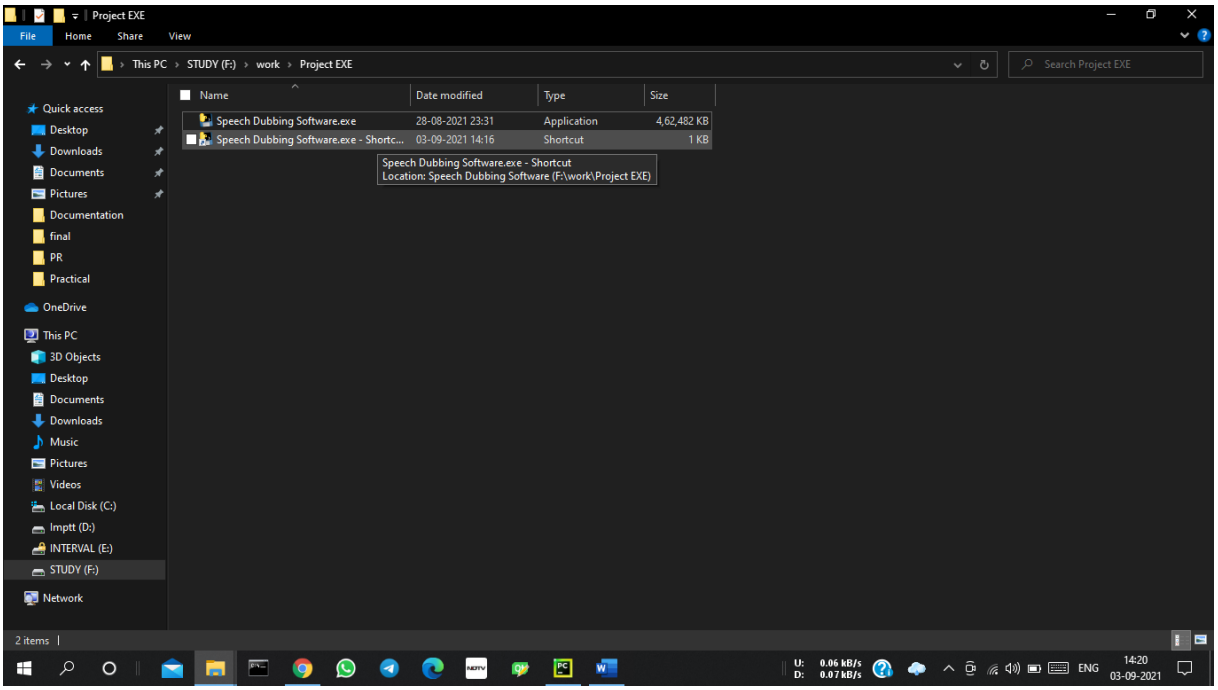


Figure No 6.3

4. You can drag your shortcut on the desktop.

Figure No 6.4

Step 3:

Running the Specch Dubbing Software

1) Double click on the desktop icon or the .exe file. By clicking the command prompt open.

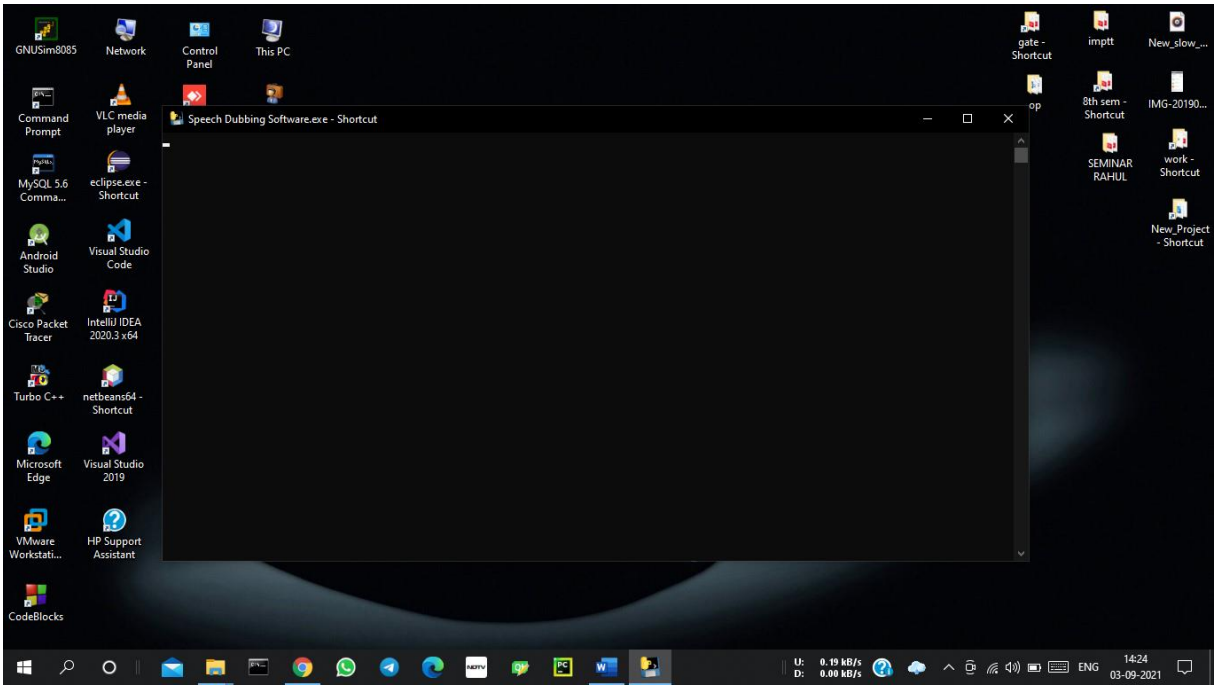


Figure No 6.5

2) You can maximize the appeared window for a better experience.

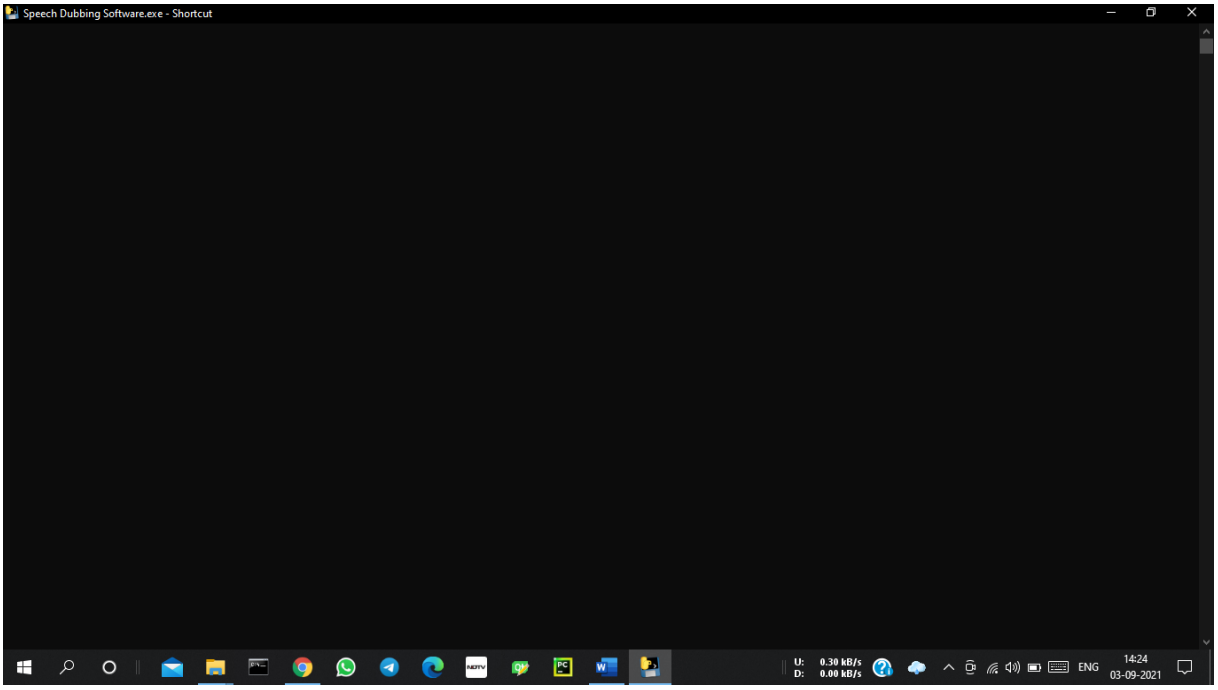


Figure No 6.6

3) After some seconds user interface of Speech Dubbing Software will appear.

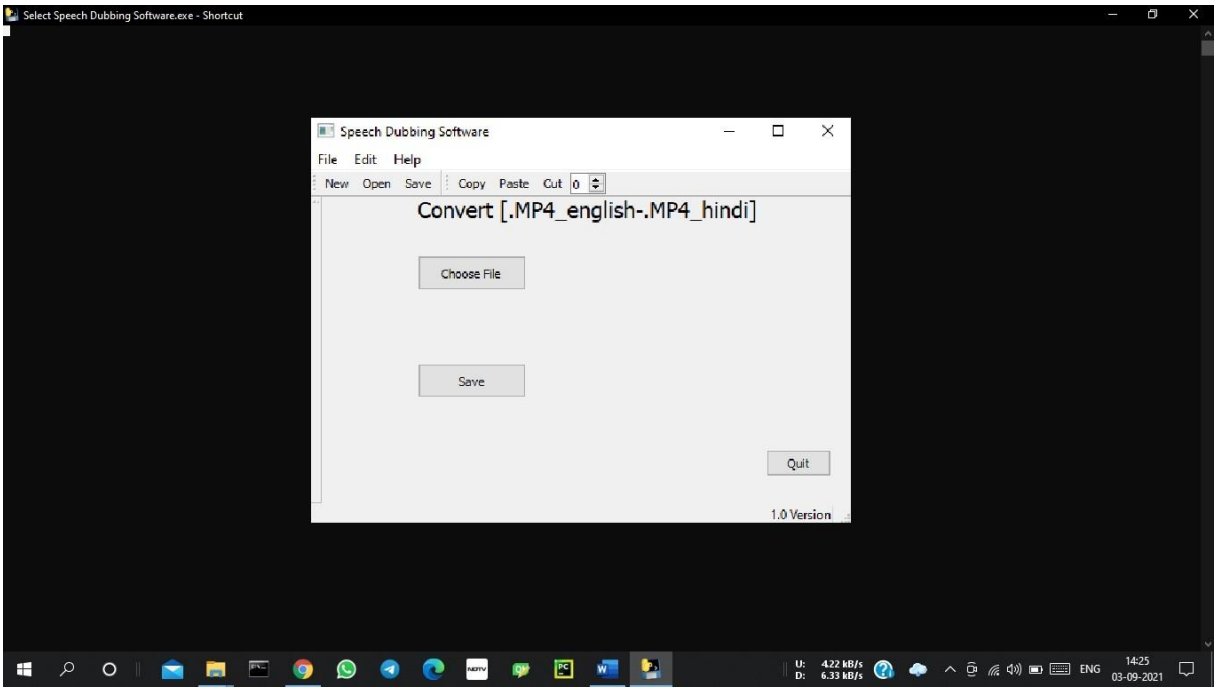


Figure No 6.7

Step 3:

Dubbing

1) Click on choose file for dubbing an English Audio Speech .mp4 file into Hindi Audio Speech .mp4 file.

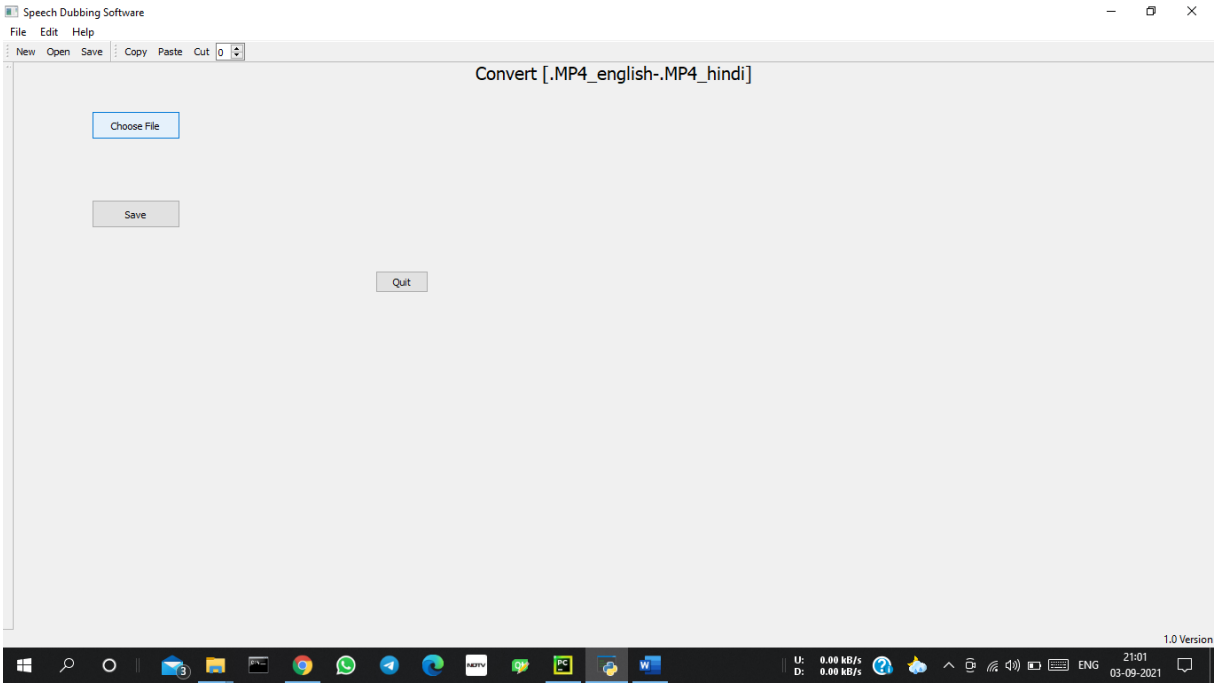


Figure No 6.8

2) Select your .mp4 file that you have to dub.

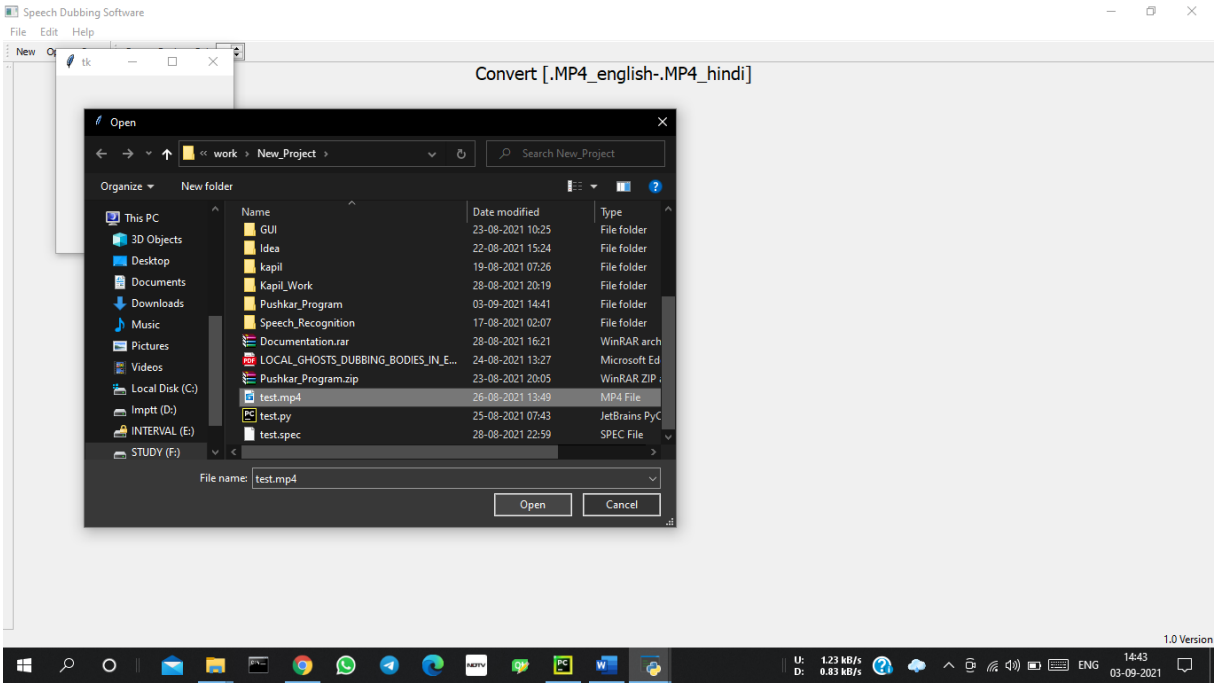


Figure No 6.9

3) After selection click on Open. Dubbing takes some more time, after some time a popup window appears that informs your file will be dubbed.

Step 4:

Saving

1) For saving dubbed files click on Save.

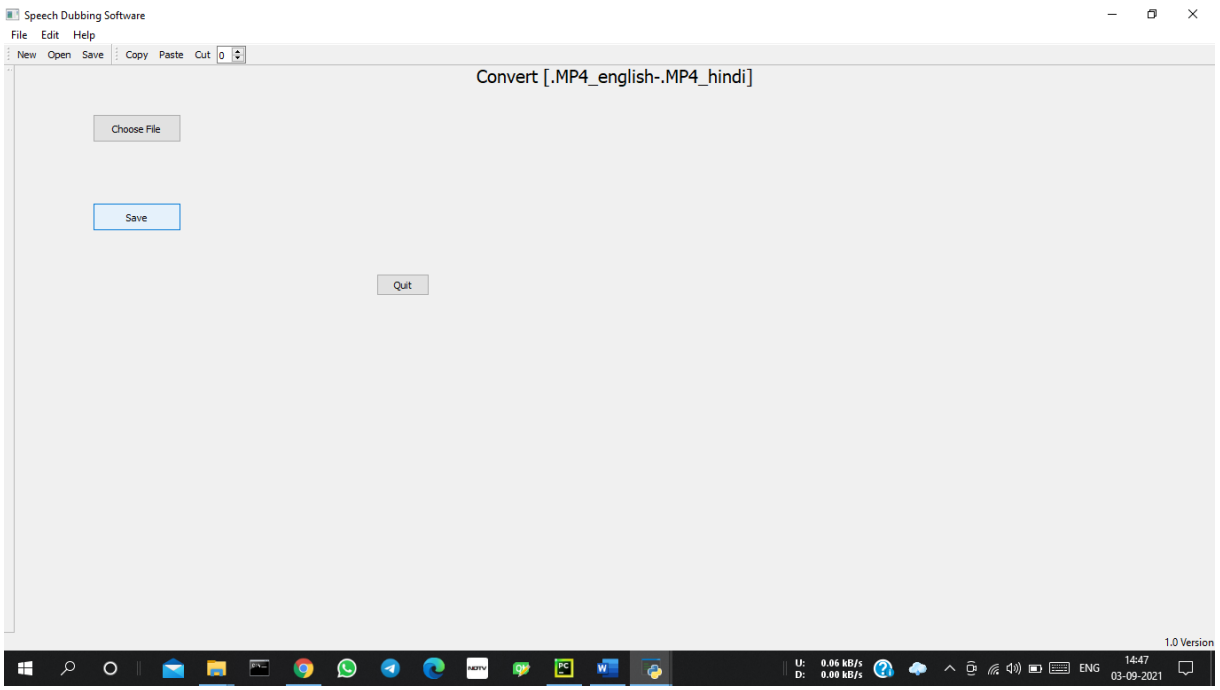


Figure No 6.10

2) Confirmation popup appears on window click on yes.
3) The file will be saved successfully.

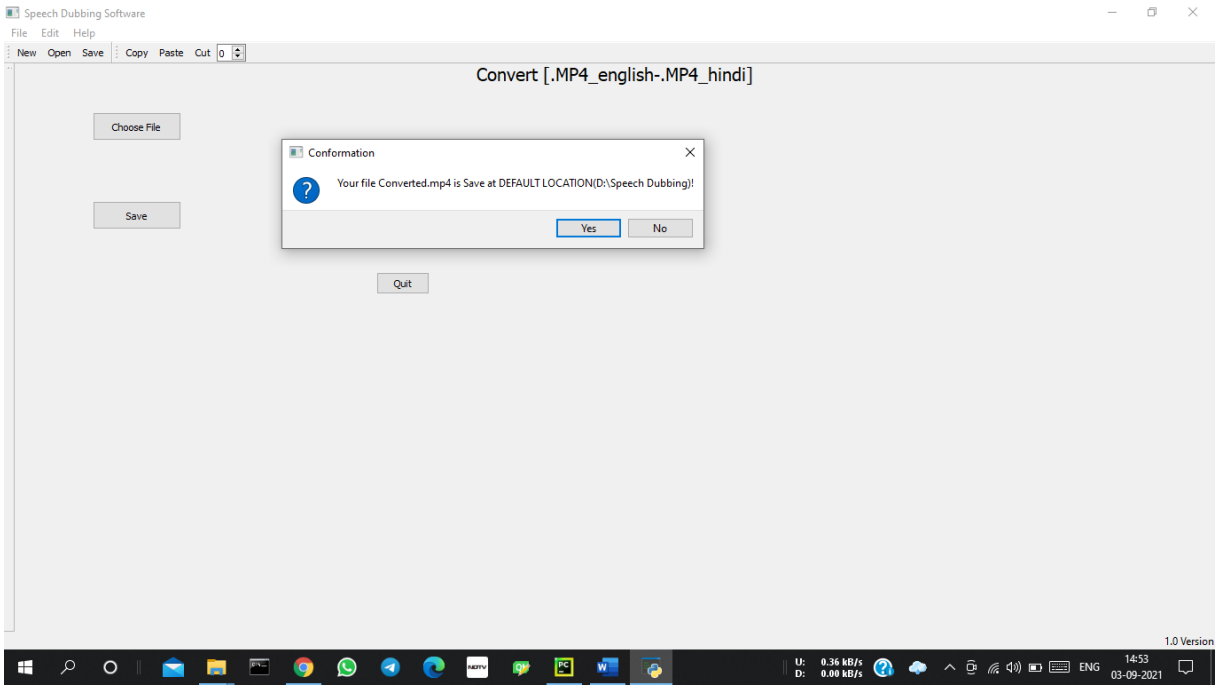


Figure No 6.11

Step 5:

Exit

1) To exit the application click on Quit.

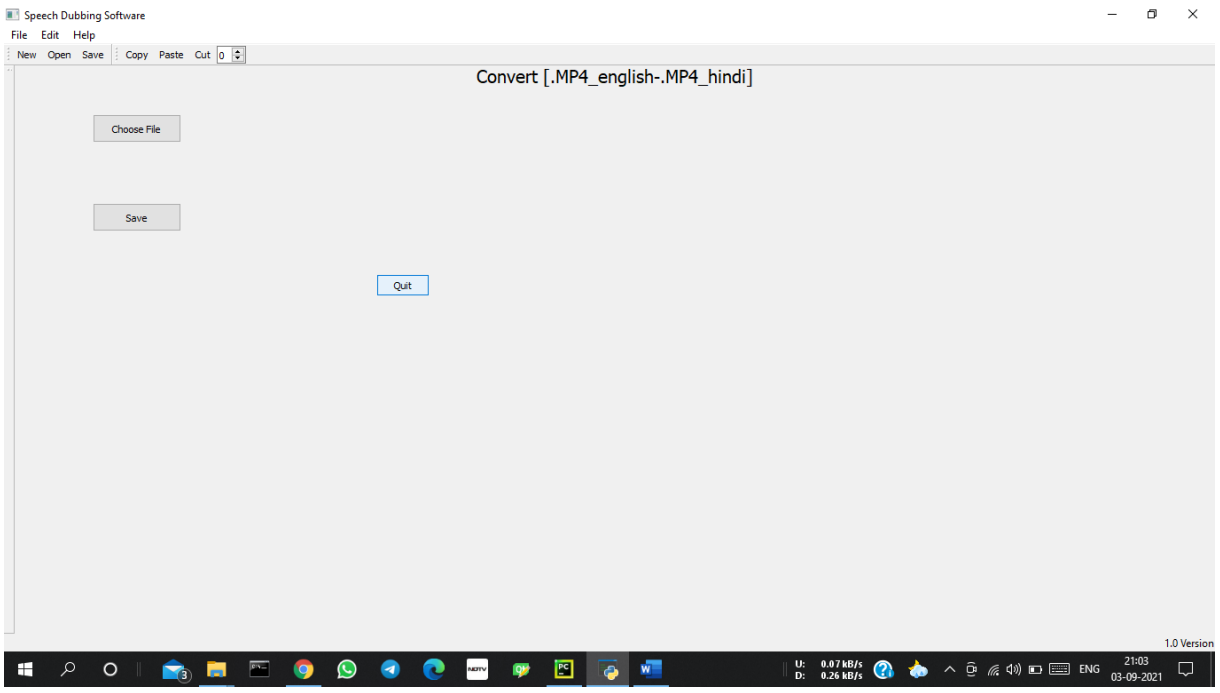


Figure No 6.12

2) A confirmation popup appeared.
3) Click on yes to exit the Application.

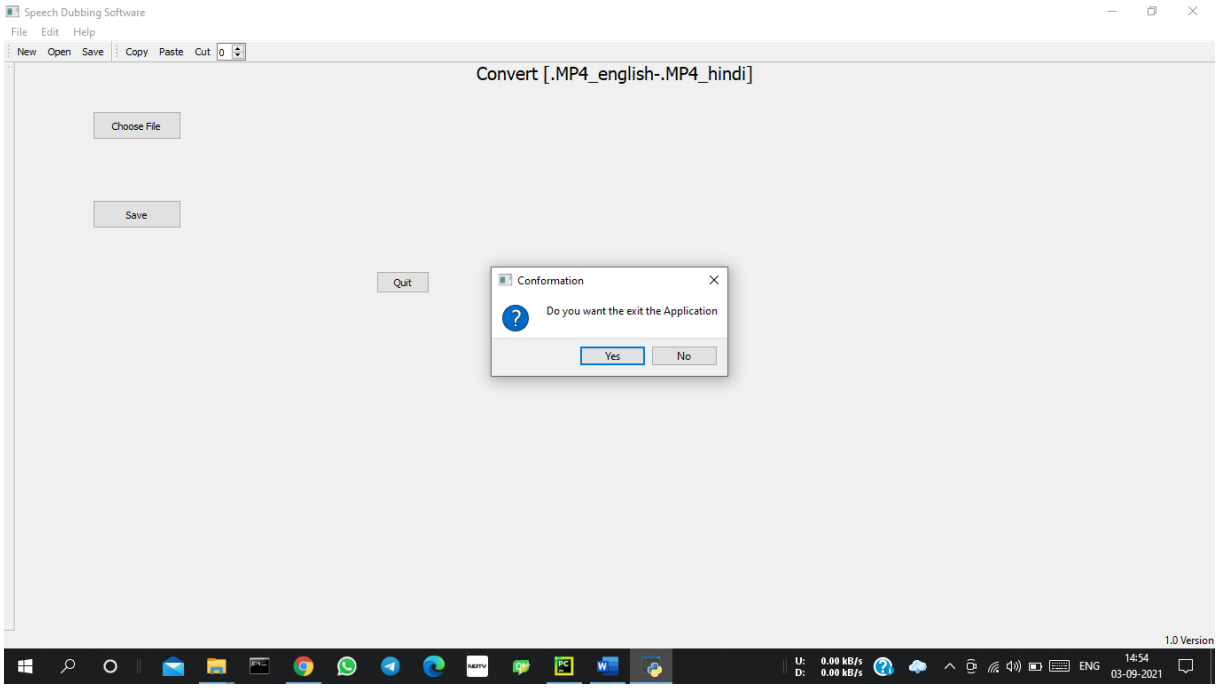


Figure No 6.13

Step 6:

Playing Converted .mp4 Hindi Audio File

1) Go to the directory **D:\Speech Dubbing.**

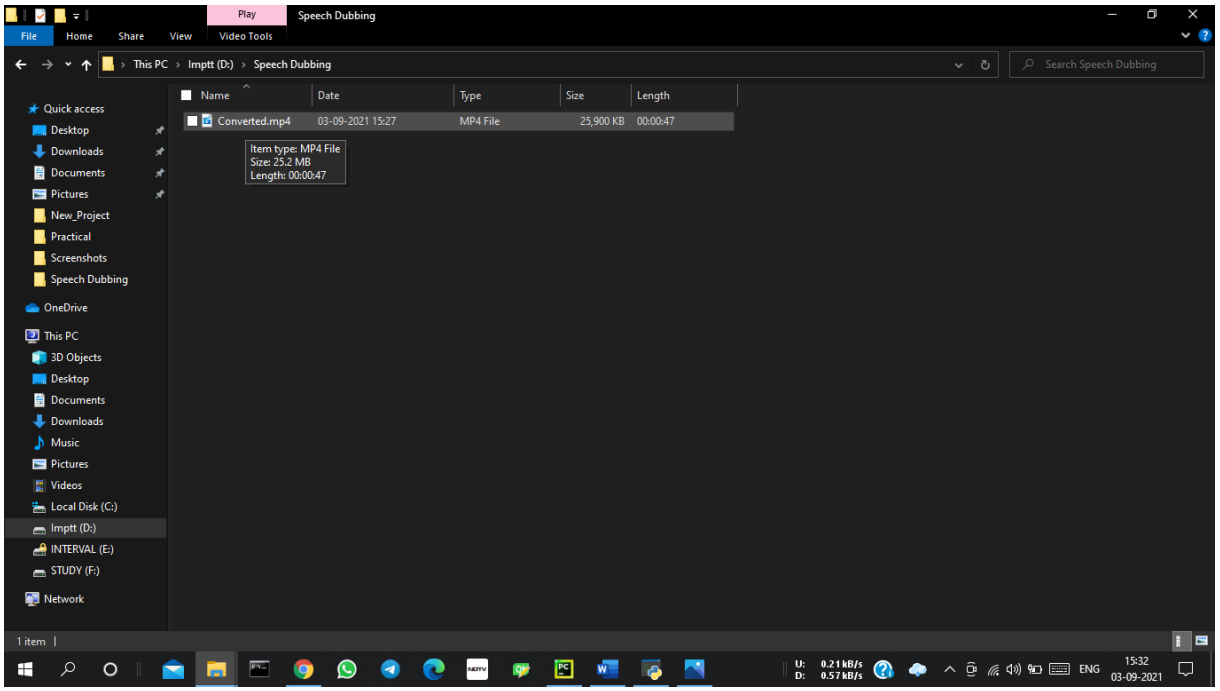


Figure No 6.14

2) Double click on the Converted.mp4 file.
3) Now enjoy your desire file.

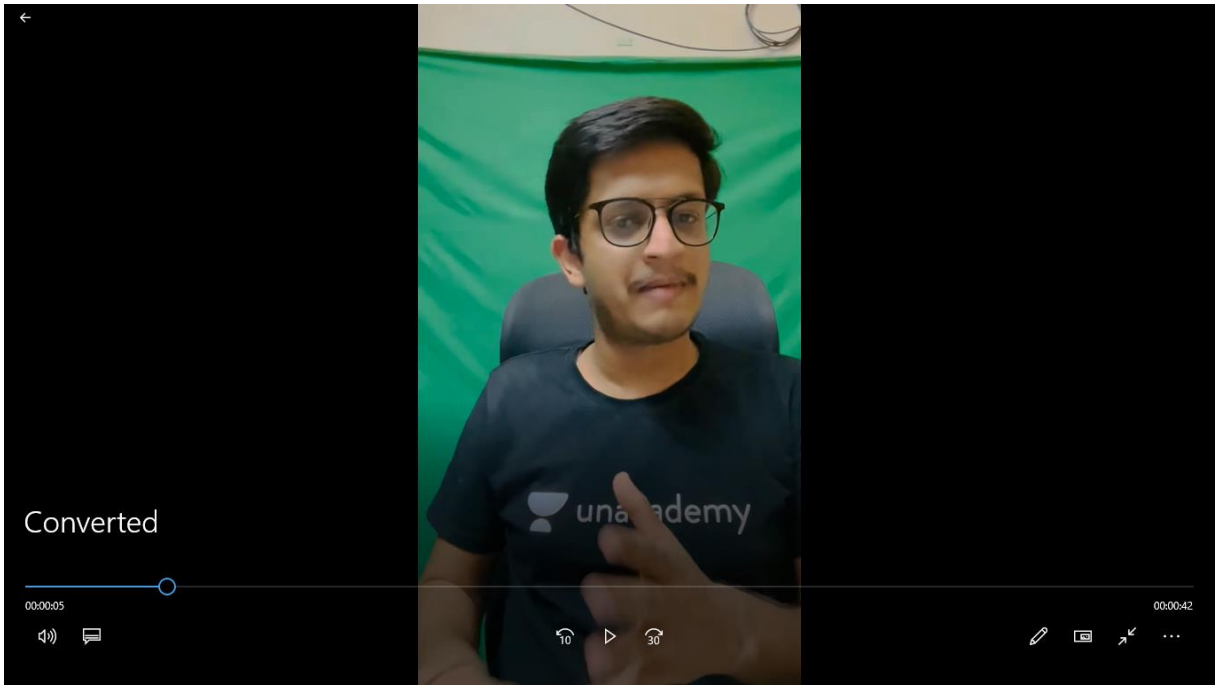


Figure No 6.15

7. Future Scope

- In this project when we tested the video file we achieve 80% accuracy and in the future that can be achieved by more than 80% by improving the generated audio synchronization & audio recognition.

- Future work will be devoted to better adapt machine translation to the style used in dubbing and to improve the quality of prosodic alignment, by generating more accurate sentence segmentation and by introducing more flexible synchronization.

- In the upcoming days, we will make a device which help for those people who are a tourist and that tourist are foreigner which native language is something else and tourist place language is different and that device will live to translate automatically the visiting place to tourist native language.

8. Conclusion

- In the project, we implemented the Speech Dubbing Software whereas the main concept was based on two things first Natural Language Processing and second Machine learning. Both are highly researchable are in Computer Science. We use the Natural language processing technique for recognition of speech & audio Generation of the video file & the Machine Learning technique for generated Audio synchronization.

- We have perceptually evaluated the naturalness of automatic speech dubbing after enhancing a baseline speech-to-speech translation system with the possibility to control the verbosity of the translation output, to segment and synchronize the target words with the speech-pause structure of the source utterances, and to enrich TTS speech with ambient noise and reverberation extracted from the original audio.

-
- We tested our project and it works for Indian & Non-Indian videos lecture/Speech.
- In this project, we learn lots of things regarding machine learning like math library in which we perform any mathematical operation on the file for desire and we learn lots of python modules whereas more powerful and more dynamic Compare to the other language

- Future work will be devoted to better adapt machine translation to the style used in dubbing and to improve the quality of prosodic alignment, by generating more accurate sentence segmentation and by introducing more flexible synchronization.

9. References

https://www.python.org/

https://en.wikipedia.org/wiki/Python_(programming_language)

https://zetcode.com/gui/pysidetutorial/firstprograms/

https://gtts.readthedocs.io/en/latest/

https://www.anaconda.com/

https://www.jetbrains.com/pycharm/

https://doc.qt.io/qt-5/qtdesigner

https://www.qt.io/download

https://www.qt.io/product/development-tools

https://en.wikipedia.org/wiki/Block_diagram

https://www.microtool.de/en/knowledge-base/what-is-a-block-diagram/

https://www.visual-paradigm.com/tutorials/flowchart-tutorial/

https://pypi.org/project/PyQt5/

https://doc.qt.io/qt-5/qtwidgets-index.html

https://www.cppdepend.com/qtcore

https://doc.qt.io/archives/qt-4.8/qtgui-module.html

https://en.wikipedia.org/wiki/Speech_recognition

https://pypi.org/project/SpeechRecognition/

https://www.deepl.com/en/translator

https://pypi.org/project/deep-translator/

https://pypi.org/project/googletrans/

https://docs.python.org/3/library/tkinter.html

https://gtts.readthedocs.io/en/latest/

https://pypi.org/project/gTTS/

https://towardsdatascience.com/transcribing-interview-data-from-video-to-text-with-python-5cdb6689eea1

https://realpython.com/python-pyqt-gui-calculator/

10. Project Source Code

To download the source file of this project goes to the below URL…

- **GitHub**

  https://github.com/RamanPushkar/Speech-Dubbing-Software-Video-English-Hindi-/blob/2be197713a87e456aaf20542f738acb7ca249405/README.md

- **Short-Url**
  https://bit.ly/3gZX2dl

- **Download the executable file:**

  https://drive.google.com/file/d/1pFVDR_Rg0M1kpgdkQHgZiV47p1GQzn09/view?usp=sharing