**PROJECT REPORT**

**ON**

**Text to speech converter**



REPORT SUBMITTED

TO

**VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE**

**FOR THE PBL OF PYTHON FOR ENGINEERS**

In

**DEPARTMENT OF ENGINEERING AND APPLIED SCIENCES**

BY

**Name of the Students -**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **ROLLNO.** | **DIVISION** | **BATCH** | **GRN NO.** |
| **DEEP JADHAV** | **7050** | **G** | **G3** | **22010818** |
| **ONKAR JAGTAP** | **7055** | **G** | **G3** | **22010898** |
| **DINESH RAUT** | **7058** | **G** | **G3** | **22010946** |
| **VISHAL NAGRE** | **7060** | **G** | **G3** | **22010978** |
| **AKSHAY MISAL** | **7064** | **G** | **G3** | **22011042** |

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Contents** | **Page No.** |
| **1** | **ABSTRACT** | 5 |
| **2** | **INTRODUCTION AND THEORY** | 6-7 |
| **3** | **FLOW OF PROGRAM AND FLOW OF CODE** | 8-9 |
| **4** | **SOURCE CODE OF THE PROGRAM** | 10-14 |
| **5** | **OUTPUT** | 15 |
| **6** | **LIST OF THE TOPICS/ CONCEPT WHICH ARE**  **COVERED FROM THE SYLLABUS** | 16 |
| **7** | **LIST OF THE TOPICS/ CONCEPT WHICH ARE**  **OUT OF SYLLABUS** | 16 |
| **8** | **CONCLUSION** | 17 |
| **9** | **REFERENCES** | 17 |

**Batch Teacher**

**Prof.Aparna Barbadekar**

**ABSTRACT**

The topic of our python PBL is “TEXT TO SPEECH CONVERTER’’. In this project we have utilized all resources available on the web to make this project better. Moreover we have also covered a wide range of topics that have been taught to by the college teachers and have implemented in this project ,some of the topics are like usage of modules(random module), conditionals (if and else statements), functions(global ,lambda and user defined), dictionary and strings. To make this project to look more attractive and representable we have also used GUI(Graphical user interface ) in it . As this topic has not been covered yet it was really difficult for us to deal with it but at the end we succeeded. Our project is just a simple application that everyone can use for the ease of converting the text into the speech.

**INTRODUCTION AND THEORY**

The “Speech to text converter” is a very interesting application. It can be used by a person who can use computer. In this application we have to choose the function according to our need. These are some rules for speech to text converter :

1. If you want to convert a basic line into speech, just type it in the text box hence displayed
2. Once the text is typed into the text box press the text to speech button on the screen. then press ok
3. If you want to convert pdf to speech , mention the name of pdf in the text box with “.pdf ” extension and then click pdf to speech button followed by ok.
4. Same procedure for scan to text , enter the name of image file with “ . jpeg” extension.
5. For a audio file, for converting thee audio file into the text the audio file must be in ‘WAV’ format , similarly enter file name with “.wav” extension.
6. Most important part of the project :- whatever it may be the pdf file , image, audio file which user want to execute or scan that should be only in the folder where the file of source code is pasted.

After understanding these rules now the app is ready to use for anyone.

In this project we have written a python code to run this app. Text To Speech is an amazing and easily expandable project. We got to know another application of PyQt5.QtWidgets and various features of python and how it is practically used . We came across different module especially the “ PYTTSX3 / PyAudio ”. This module converts scripts into audio. We could also change the rate of  speech  , volume, and also the voice.

We used some of python concepts in the code of this program.

1. **PyQt5 module :** 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.
2. **PYttsx3 module** :- Pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.The main core of this project which is Text to Speech conversion (TTS) is done using this module
3. **GTTS MODULE** :- Google text to speech converter is a language translator module in this program.
4. **Dictionary :** Python dictionary is an unordered collection of items. Each item of a dictionary has a key/value pair. Dictionaries are optimized to retrieve values when the key is known.
5. **Function And Creating a function :** A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result in Python a function is defined using the def keyword
6. Also we used many GUI tools or attributes like label , button , text , title , grid and some parameters of grid like padx , pady , row , column etc.

**FLOW OF PROGRAM**

1] Importing all modules and libraries

Firstly we worked on few imports like importing the random module and also imported all the stuff from **Pyqt5** so as to easily create a GUI.Pyqt5 is the standard GUI library for Python. Pyqt5 provides a fast and easy way to create GUI applications.

2]Creating the Masterscreen

After importing the modules and libraries we created the Masterscreen and have assigned it a suitable tittle here the tittle is “Text to speech forum”.

3] Making Buttons and Dummy labels

We have now created buttons that will appear on our Masterscreen. The buttons that we have included are “PDF TO SPEECH”, “TRANSLATION OF SPEECH”, “TEXT TO SPEECH ”, “TRANSLATION OF TEXT”, “SCAN TO TEXT” , “TRANSLATED SCAN TO TEXT” , “AUDIO TO TEXT” and the name of our project “Text to Speech forum”. We also gave the button attributes like text ,font etc. so that they look more presentable. Each button has been assigned a particular width so that they have proper spacing between them. We made sure that there is some space left between the bottom of screen and the buttons

4]Defining functions and Variables

Here comes the most important part of defining function here we have defined the outcome handler function it is very useful every time when the user is going to click a button for making a choice the outcome handler function is going to run and it is going to pass a value that is assigned to that variable which we have stored in a dictionary

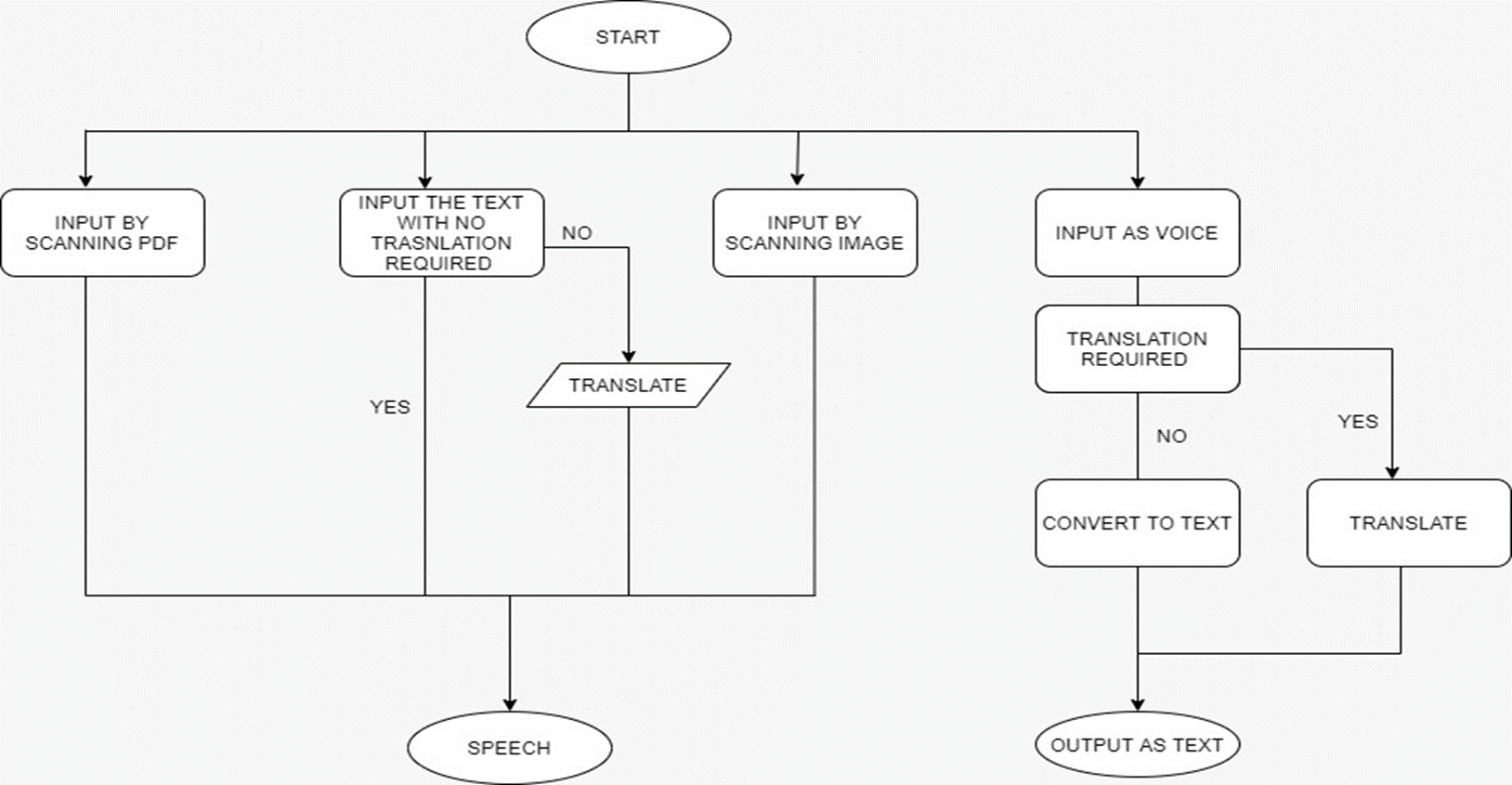
5] Executing the code

Whenever the outcome variable is accessed it goes to the code and executes the key and runs the code.

7] Displaying the Screen

Now at last we have to display our Masterscreen that we made for this program. The master screen have all the options to execute to program.

**FLOW OF CODE :-**



**SOURCE CODE OF THE PROGRAM :-**

import sys

from PyQt5.QtWidgets import QApplication, QWidget, QLabel

from PyQt5.QtGui import QIcon, QFont

from PyQt5.QtWidgets import QMainWindow, QApplication, QWidget, QPushButton, QAction, QLineEdit, QMessageBox

from PyQt5.QtGui import QIcon

from PyQt5.QtCore import pyqtSlot

import PyPDF2

import pyttsx3

from gtts import gTTS

import translators as ts

import os

import speech\_recognition as sr

class App(QMainWindow):

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

        self.title = 'TEXT TO SPEECH FORUM (G-2 GROUP)'

        self.left = 500

        self.top = 150

        self.width = 800

        self.height = 680

        self.label\_1 = QLabel("  Vishwakarma Institute Of Information Technology ", self)

        self.label\_1.setStyleSheet("border :5px solid black;")

        self.label\_1.setFont(QFont('One Day', 20))

        self.label\_1.setGeometry(2, 15, 796, 100)

        self.label\_1.setStyleSheet("background-color: #b2ffff;")

        self.label\_2 = QLabel("  TEXT TO SPEECH FORUM  ", self)

        self.label\_2.setFont(QFont('One Day', 30))

        self.label\_2.setStyleSheet("border :3px solid black;")

        self.label\_2.setGeometry(80, 135, 660, 90)

        self.label\_2.setStyleSheet("background-color: #f4a460;")

        self.setGeometry(self.left, self.top, self.width, self.height)

        self.setStyleSheet("background-color: #e3f988;")

        self.initUI()

    def initUI(self):

        self.setWindowTitle(self.title)

        self.setGeometry(self.left, self.top, self.width, self.height)

        # Create textbox

        self.textbox = QLineEdit(self)

        self.textbox.setGeometry(105, 270, 600, 150)

        self.setGeometry(self.left, self.top, self.width, self.height)

        self.textbox.setStyleSheet("background-color: white;")

        # Create a button 1 in the window

        self.button\_1 = QPushButton('PDF TO SPEECH ', self)

        self.button\_1.setGeometry(120, 470, 200, 40)

        self.button\_1.setStyleSheet("background-color: white;")

        # Create a button 2 in the window

        self.button\_2 = QPushButton(' TRANSLATION OF PDF  ', self)

        self.button\_2.setGeometry(500, 470, 200, 40)

        self.button\_2.setStyleSheet("background-color: white;")

        # Create a button 3 in the window

        self.button\_3 = QPushButton('TEXT TO SPEECH ', self)

        self.button\_3.setGeometry(120, 525, 200, 40)  # (left,top)

        self.button\_3.setStyleSheet("background-color: white;")

        # Create a button 4 in the window

        self.button\_4 = QPushButton('TRANSLATION OF TEXT ', self)

        self.button\_4.setGeometry(500, 525, 200, 40)

        self.button\_4.setStyleSheet("background-color: white;")

        # Create a button 5 in the window

        self.button\_5 = QPushButton('SCAN TO TEXT ', self)

        self.button\_5.setGeometry(120, 585, 200, 40)

        self.button\_5.setStyleSheet("background-color: white;")

        # Create a button 6 in the window

        self.button\_6 = QPushButton('TRANSLATED SCAN TO TEXT ', self)

        self.button\_6.setGeometry(500, 585, 200, 40)

        self.button\_6.setStyleSheet("background-color: white;")

        # Create a button 7 in the window

        self.button\_7 = QPushButton('AUDIO TO TEXT ', self)

        self.button\_7.setGeometry(300, 630, 200, 40)

        self.button\_7.setStyleSheet("background-color: white;")

        # connect button to function on\_click 1

        self.button\_1.clicked.connect(self.on\_click\_1)

        self.show()

        # connect button to function on\_click 2

        self.button\_2.clicked.connect(self.on\_click\_2)

        self.show()

        # connect button to function on\_click 3

        self.button\_3.clicked.connect(self.on\_click\_3)

        self.show()

        # connect button to function on\_click 4

        self.button\_4.clicked.connect(self.on\_click\_4)

        self.show()

        # connect button to function on\_click 5

        self.button\_5.clicked.connect(self.on\_click\_5)

        self.show()

        # connect button to function on\_click 6

        self.button\_6.clicked.connect(self.on\_click\_6)

        self.show()

        # connect button to function on\_click 7

        self.button\_7.clicked.connect(self.on\_click\_7)

        self.show()

    @pyqtSlot()

    def on\_click\_1(self):

        textboxValue = self.textbox.text()

        book = open(textboxValue, 'rb')

        pdfReader = PyPDF2.PdfFileReader(book)

        pages = pdfReader.numPages

        speaker = pyttsx3.init()

        page = pdfReader.getPage(0)

        text\_1 = page.extractText()

        '''speaker.say(text\_1)

        speaker.runAndWait()'''

        language = 'en'

        output = gTTS(text=text\_1, lang=language, slow=False)

        output.save("output(1).mp3")

        os.system("start output(1).mp3")

    def on\_click\_2(self):

        textboxValue = self.textbox.text()

        book = open(textboxValue, 'rb')

        pdfReader = PyPDF2.PdfFileReader(book)

        pages = pdfReader.numPages

        speaker = pyttsx3.init()

        page = pdfReader.getPage(0)

        text\_2 = page.extractText()

        myText = text\_2

        textis = (ts.google(myText, from\_language='en', to\_language='mr'))

        language = 'mr'

        output = gTTS(text=textis, lang=language, slow=False)

        #output.save("output(2).mp3")

        os.system("start output(2).mp3")

    def on\_click\_3(self):

        textboxValue = self.textbox.text()

        speaker = pyttsx3.init()

        speaker.say(textboxValue)

        speaker.runAndWait()

    def on\_click\_4(self):

        textboxValue = self.textbox.text()

        myText = textboxValue

        textis = (ts.google(myText, from\_language='en', to\_language='mr'))

        language = 'mr'

        output = gTTS(text=textis, lang=language, slow=False)

        output.save("output(3).mp3")

        os.system("start output(3).mp3")

    def on\_click\_5(self):

        textboxValue = self.textbox.text()

        import pytesseract as tess

        tess.pytesseract.tesseract\_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

        from PIL import Image

        import pyttsx3

        img = Image.open(textboxValue)

        text\_3 = tess.image\_to\_string(img)

        print(text\_3)

        speaker = pyttsx3.init()

        speaker.say(text\_3)

        speaker.runAndWait()

    def on\_click\_6(self):

        textboxValue = self.textbox.text()

        import pytesseract as tess

        tess.pytesseract.tesseract\_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

        from PIL import Image

        import pyttsx3

        img = Image.open(textboxValue)

        text\_4 = tess.image\_to\_string(img)

        print(text\_4)

        textis = (ts.google(text\_4, from\_language='en', to\_language='mr'))

        language = 'mr'

        output = gTTS(text=textis, lang=language, slow=False)

        output.save("output(4).mp3")

        os.system("start output(4).mp3")

    def on\_click\_7(self):

        textboxValue = self.textbox.text()

        r = sr.Recognizer()

        audio\_file = sr.AudioFile(textboxValue)

        with audio\_file as source:

            audio\_text = r.record(source)

        print((r.recognize\_google(audio\_text)))

        QMessageBox.question(self, 'Message - audio to text', "Text: " + (r.recognize\_google(audio\_text)),

                             QMessageBox.Ok, QMessageBox.Ok)

        self.textbox.setText("")

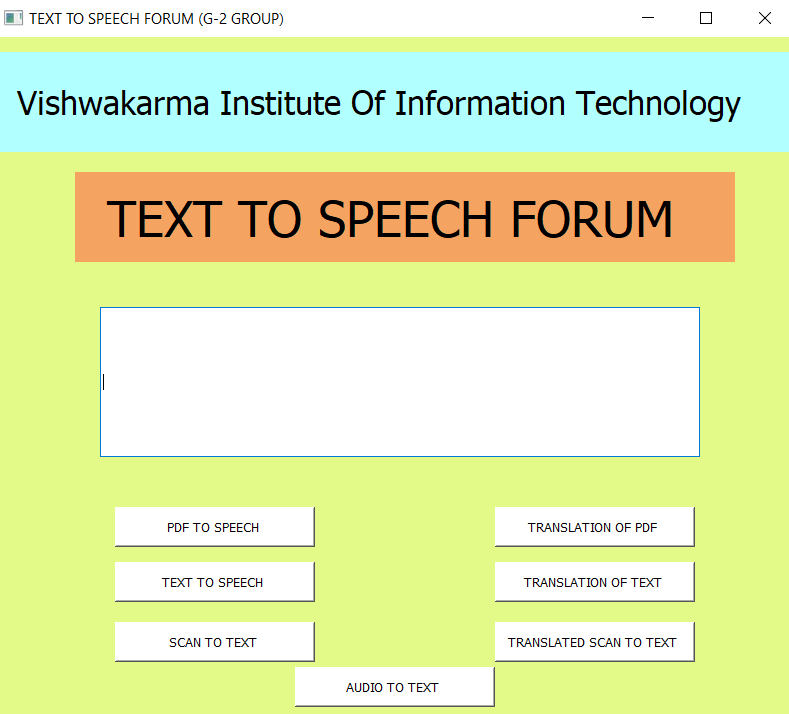
if \_\_name\_\_ == '\_\_main\_\_':

    app = QApplication(sys.argv)

    ex = App()

    sys.exit(app.exec\_())

**OUTPUT :-**

****

**List of topics/concepts covered from the syllabus**

1. Strings
2. Dictionary(nested dictionary)
3. Conditionals
4. Modules
5. Functions
6. Library

**List of Topics /concepts out of syllabus**

1. GUI [Graphical user interface]
2. Pyqt5 GUI toolkit

**Conclusion**

We have successfully made the “Text to speech converter” using GUI(Graphical User Interface).

**References**

1] https://geeksforgeeks.org/

2] <https://stackoverflow.com/>

3] https://www.python.org/