```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QLabel
from PyQt5.QtGui import QIcon, QFont
from PyQt5.QtWidgets import QMainWindow, QApplication, QWidget, QPushButton, Q
Action, 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 Technolo
gy ", 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.re
cognize_google(audio_text)),
                             QMessageBox.Ok, QMessageBox.Ok)
        self.textbox.setText("")


if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = App()
    sys.exit(app.exec_())
```