

Joy of Engineering

Project Report
(PSP1004)

VOICE ASSISTANT

GROUP MEMBERS

AYUSH SAINI

210C2030086

D.V.HARSHA VARDHAN REDDY

210C2030061

B.SAI CHAITANYA

210C2030075

DEEPAK

210C2050004



**BML MUNJAL
UNIVERSITY™**

FROM HERE TO THE WORLD

Department of Computer Science and Engineering
School of Engineering and Technology
January 2022

Declaration

We, Ayush, Harsha, Sai Chaitanya and Deepak, hereby declare that the project entitled "Voice Assistant" in fulfillment of completion of the 1st semester course – Joy of Engineering as part of Bachelor of Technology (B.Tech) program at the School of Engineering and Technology, BML Munjal University is an authentic record of our experimental work.

Due acknowledgements have been made in the text of the project to all other materials used.

This project was done in the full compliance with the requirements and constraints of the prescribed curriculum.

Ayush Saini	210C2030086
D.V.Harsha Vardhan Reddy	210C2030061
B.Sai Chaitanya	210C2030075
Deepak	210C2050004

Date: 20.01.22



**BML MUNJAL
UNIVERSITY™**

FROM HERE TO THE WORLD

Contents

01

Problem Definition
Project Objectives and Deliverables
Hardware and Software Components

02

Plan of Action
Assistant Code

07

Ui Code

09

Run Code

10

Challenges
Conclusion
References

Problem Definition

What is a voice assistant?

A voice assistant is a computer program which can respond on our voice assistant for making our life easy. Den (our assistant name) can help you to deal with some normal daily tasks with just a voice command.

Project Objectives and Deliverables

Solving daily problems is the main task of making this assistant. taking to us replying to our voice commands and opening our apps when needed.

Hardware and Software Components

As our project is software based there in no use of any external hardware.

Software Used: Vscode, Jupyter Notebook and QTDesigner

Language used: Python



Plan of Action

Assistant Code

Main assistant file was most important of our project. This consists of the voice detection , modulation , recognition and interpretation.

Libraries Used

- Speech Recognition
- pyttsx3
- Wikipedia
- Pyaudio
- Pyjokes
- Pywhatkit
- PyQt5
- Pyautogui
- Web browser
- Subprocess

Engine For Audio

```
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)

def speak(audio):
    engine.say(audio)
    engine.runAndWait()
```

Library Used For Speech Recognition

```
def takeCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Catching your words....")
        r.pause_threshold = 0.5
        audio = r.listen(source)

    try:
        print("Got it...")
        query = r.recognize_google(audio, language='en-in')
        print(f"User said: {query}\n")

    except Exception as e:
        print("sorry can you repeat...")
        return "None"
    return query
```

Assistant Full Code

```
1 import pytsxs3
2 import speech_recognition as sr
3 import datetime
4 from datetime import date
5 import calendar
6 import time
7 import math
8 import wikipedia
9 import webbrowser
10 import os
11 import smtplib
12 import winsound
13 import pyautogui
14 from tkinter import *
15 from playsound import playsound
16 import subprocess
17 import pywhatkit
18
19 engine = pytsxs3.init('sapi5')
20 voices = engine.getProperty('voices')
21 engine.setProperty('voice', voices[0].id)
22
23
24 def speak(audio):
25     engine.say(audio)
26     engine.runAndWait()
27
28
29 def takeCommand():
30     r = sr.Recognizer()
31     with sr.Microphone() as source:
32         print("Catching your words....")
33         r.pause_threshold = 0.5
34         audio = r.listen(source)
35
36     try:
37         print("Got it...")
38         query = r.recognize_google(audio, language='en-in')
39         print(f"User said: {query}\n")
40
41     except Exception as e:
42         print("sorry can you repeat...")
43         return "None"
44     return query
45
46
47 def wishMe():
48     hour = int(datetime.datetime.now().hour)
49     if hour >= 0 and hour < 12:
50         speak("Good Morning!")
51
52     elif hour >= 12 and hour < 18:
53         speak("Good Afternoon!")
54
55     else:
56         speak("Good Evening!")
57
58     speak("Hello den this side, how can i help you")
59
60
```

```
47 def wishMe():
48     hour = int(datetime.datetime.now().hour)
49     if hour >= 0 and hour < 12:
50         speak("Good Morning!")
51
52     elif hour >= 12 and hour < 18:
53         speak("Good Afternoon!")
54
55     else:
56         speak("Good Evening!")
57
58     speak("Hello den this side, how can i help you")
59
60
61 def brightness():
62     try:
63         query = takeCommand().lower()
64         if '25' in query:
65             pyautogui.moveTo(1880, 1050)
66             pyautogui.click()
67             time.sleep(1)
68             pyautogui.moveTo(1610, 960)
69             pyautogui.click()
70             pyautogui.moveTo(1880, 1050)
71             pyautogui.click()
72             speak('If you again want to change brihtness, say, change brightness')
73         elif '50' in query:
74             pyautogui.moveTo(1880, 1050)
75             pyautogui.click()
76             time.sleep(1)
77             pyautogui.moveTo(1684, 960)
78             pyautogui.click()
79             pyautogui.moveTo(1880, 1050)
80             pyautogui.click()
81             speak('If you again want to change brihtness, say, change brightness')
82         elif '75' in query:
83             pyautogui.moveTo(1880, 1050)
84             pyautogui.click()
85             time.sleep(1)
86             pyautogui.moveTo(1758, 960)
87             pyautogui.click()
88             pyautogui.moveTo(1880, 1050)
89             pyautogui.click()
90             speak('If you again want to change brihtness, say, change brightness')
91         elif '100' in query or 'full' in query:
92             pyautogui.moveTo(1880, 1050)
93             pyautogui.click()
94             time.sleep(1)
95             pyautogui.moveTo(1835, 960)
96             pyautogui.click()
97             pyautogui.moveTo(1880, 1050)
98             pyautogui.click()
99             speak('If you again want to change brihtness, say, change brightness')
100        else:
101            speak('Please select 25, 50, 75 or 100..... Say again.')
102            brightness()
103    except:
104        speak('Something went wrong')
```

```
107 def assisrun():
108
109     wishMe()
110
111     while True:
112         query = takeCommand().lower()
113
114         if 'who is' in query:
115             speak('Searching Wikipedia...')
116             query = query.replace("wikipedia", "")
117             results = wikipedia.summary(query, sentences=2)
118             speak("According to Wikipedia")
119             print(results)
120             speak(results)
121
122         elif 'open youtube' in query:
123             webbrowser.open("https://youtube.com")
124             speak("ok opening youtube")
125             break
126
127         elif 'open Amazon Prime' in query or 'open prime' in query:
128             webbrowser.open("https://primevideo.com")
129             speak('ok opening prime video')
130             break
131
132         elif 'open hotstar' in query:
133             webbrowser.open("https://hotstar.com")
134             speak('ok opening hotstar')
135             break
136
137         elif 'open google' in query:
138             webbrowser.open("https://google.com")
139             break
140
141         elif 'open netflix' in query:
142             webbrowser.open('https://www.netflix.com/in/')
143             speak('ok opening netflix')
144             break
145
146         elif 'the time' in query:
147             strTime = datetime.datetime.now().strftime("%H:%M:%S")
148             speak(f"Sir, the time is {strTime}")
149
150         elif 'whats the time' in query:
151             strTime = datetime.datetime.now().strftime("%H:%M:%S")
152             speak(f"Sir, the time is {strTime}")
153
154         elif "den" in query or "hi there" in query:
155
156             wishMe()
157             speak("den in your service")
158
159         elif 'how are you' in query:
160             speak("I am fine, Thank you")
161             speak("How are you, Sir")
162
163         elif 'fine' in query or 'good' in query:
164             speak("It's good to know that your fine")
165
166         elif "who made you" in query or "who created you" in query:
167             speak("I have been created by Universe.")
168
169         elif 'play' in query:
170             song = query.replace('play', '')
171             speak('playing' + song)
172             pywhatkit.playonyt(song)
173             break
174
175         elif 'stop' in query:
176             speak("thank you for using den")
177             break
178
179         elif 'ok stop' in query:
180             speak("thank you for using den")
181             break
182
183         elif 'terminate' in query:
184             speak("thank you for using den")
185             break
186
187         elif 'bye' in query:
188             speak("Bye sir")
189             break
```

```

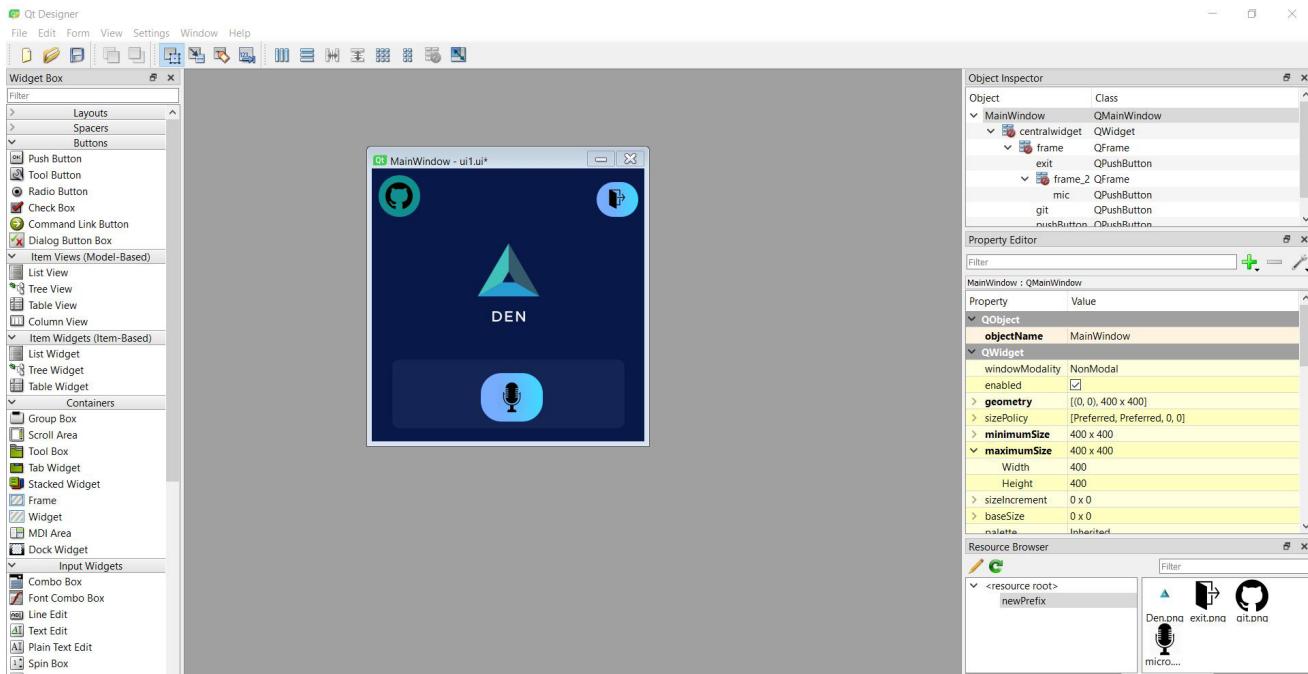
177 elif 'calculat' in query:
178     os.system('calc.exe')
179     break
180 elif ('increase' in query or 'decrease' in query or 'change' in query or 'minimize' in query or 'maximize' in query) and 'brightness' in query:
181     speak('At what percent should I kept the brightness, 25, 50, 75 or 100?')
182     brightness()
183 elif 'open map' in query or ('let' in query and 'map' in query and 'open' in query):
184     speak('ok opening maps.')
185     webbrowser.open('https://www.google.com/maps')
186     break
187 elif 'google maps' in query or ('let' in query and 'map' in query and 'open' in query):
188     speak('ok opening maps.')
189     webbrowser.open('https://www.google.com/maps')
190     break
191 elif 'news' in query:
192     webbrowser.open('https://www.bbc.com/news/world/asia/india')
193     break
194 elif 'online shop' in query or (('can' in query or 'want' in query or 'do' in query or 'could' in query) and 'shop' in query) or ('let' in query and 'shop' in query):
195     speak(
196         'From which online shopping website, you want to shop? Amazon or flipkart?')
197     query = takeCommand().lower()
198     if 'amazon' in query:
199         speak('ok opening amazon')
200         webbrowser.open('https://www.amazon.in')
201         break
202     elif 'flip' in query:
203         webbrowser.open('https://www.flipkart.com')
204         break
205     else:
206         speak(
207             'Sorry sir, the website are you talking about is new to me.')
208         break
209 elif 'dictionary' in query:
210     webbrowser.open('https://www.dictionary.com')
211     time.sleep(3)
212     speak('Enter the word, in the search bar of the dictionary, whose defination or synonyms you want to know')
213     break
214 elif 'cabs' in query or (('can' in query or 'want' in query or 'do' in query or 'could' in query) and 'taxi' in query):
215     speak(
216         'From which service do you like to get a cab.')
217     query = takeCommand().lower()
218     if 'ola' in query:
219         speak('ok opening ola')
220         webbrowser.open('https://www.olacabs.com')
221         break
222     elif 'uber' in query:
223         speak('ok opening uber')
224         webbrowser.open('https://www.uber.com')
225         break
226     else:
227         speak(
228             'Sorry sir, the website are you talking about is new to me.')
229         break
230
231 elif 'food' in query or (('can' in query or 'want' in query or 'do' in query or 'could' in query) and 'food' in query) or ('hungry' in query):
232     speak(
233         'from which platform do you like to get some food.')
234     query = takeCommand().lower()
235     if 'swiggy' in query:
236         speak('ok opening swiggy')
237         webbrowser.open('https://www.swiggy.com')
238         break
239     elif 'zomato' in query:
240         speak('ok opening zomato')
241         webbrowser.open('https://www.zomato.com')
242         break
243     else:
244         speak(
245             'Sorry sir, the website are you talking about is new to me.')
246         break
247 elif 'open facebook' in query:
248     webbrowser.open("https://www.facebook.com")
249     speak('ok opening facebook')
250     break
251 elif 'open instagram' in query:
252     webbrowser.open("https://www.instagram.com")
253     speak('ok opening instagram')
254     break
255 elif 'open twitter' in query:
256     webbrowser.open("https://www.twitter.com")
257     speak('ok opening twitter')
258     break
259

```

Ui Code

This code is used to create a user interactive program that can be used for running our assistant code.

With the use of QT designer we design our interface that can further be used with our run file to take commands.



This ui file can further be changed into a python file...

```

1 import resource
2 from PyQt5 import QtCore, QtGui, QtWidgets
3
4
5
6 class Ui_MainWindow(object):
7     def setupUi(self, MainWindow):
8         MainWindow.setObjectName("MainWindow")
9         MainWindow.resize(400, 400)
10        MainWindow.setMinimumSize(QtCore.QSize(400, 400))
11        MainWindow.setMaximumSize(QtCore.QSize(400, 400))
12        self.centralwidget = QtWidgets.QWidget(MainWindow)
13        self.centralwidget.setObjectName("centralwidget")
14        self.frame = QtWidgets.QFrame(self.centralwidget)
15        self.frame.setGeometry(QtCore.QRect(0, 0, 400, 400))
16        self.frame.setMinimumSize(QtCore.QSize(400, 400))
17        self.frame.setMaximumSize(QtCore.QSize(400, 400))
18        self.frame.setStyleSheet("QFrame {background-color: rgb(7,24,73);}\n"
19                               "\n"
20                               "")
21        self.frame.setFrameShape(QtWidgets.QFrame.StyledPanel)
22        self.frame.setFrameShadow(QtWidgets.QFrame.Raised)
23        self.frame.setObjectName("frame")
24        self.git = QtWidgets.QPushButton(self.frame)
25        self.git.setGeometry(QtCore.QRect(10, 10, 61, 61))
26        self.git.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
27        self.git.setStyleSheet("QPushButton {background-color: rgb(18,144,144);}\n"
28                               "border-radius:30px;}\n"
29                               "")
```

```

21     self.frame.setFrameShape(QtWidgets.QFrame.StyledPanel)
22     self.frame.setFrameShadow(QtWidgets.QFrame.Raised)
23     self.frame.setObjectName("frame")
24     self.git = QtWidgets.QPushButton(self.frame)
25     self.git.setGeometry(QtCore.QRect(10, 10, 61, 61))
26     self.git.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
27     self.git.setStyleSheet("QPushButton {background-color: rgb(18,144,144);\\n"
28                           "border-radius:30px;}\n"
29                           "")
30     self.git.setText("")
31     icon = QtGui.QIcon()
32     icon.addPixmap(QtGui.QPixmap(":/newPrefix/git.png"),
33                   QtGui.QIcon.Normal, QtGui.QIcon.Off)
34     self.git.setIcon(icon)
35     self.git.setIconSize(QtCore.QSize(40, 40))
36     self.git.setObjectName("git")
37     self.frame_2 = QtWidgets.QFrame(self.frame)
38     self.frame_2.setGeometry(QtCore.QRect(30, 280, 341, 101))
39     self.frame_2.setStyleSheet("QFrame {background-color: rgb(19, 36, 85);\\n"
40                           "border-radius:10px;}")
41     self.frame_2.setFrameShape(QtWidgets.QFrame.StyledPanel)
42     self.frame_2.setFrameShadow(QtWidgets.QFrame.Raised)
43     self.frame_2.setObjectName("frame_2")
44     self.mic = QtWidgets.QPushButton(self.frame_2)
45     self.mic.setGeometry(QtCore.QRect(130, 20, 91, 71))
46     self.mic.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))

47     self.mic.setStyleSheet("QPushButton\\n"
48                           "{background-color: qlineargradient(spread:pad, x1:0.979211, y1:0.977, x2:0.00526316, y2:0.994, stop:0 rgba(66, 216, 255, 255), stop:0.6894\\n"
49                           "border-radius:30px;}\n"
50                           "QPushButton:hover{ background-color: qlineargradient(spread:pad, x1:0, y1:1, x2:1, y2:1, stop:0 rgba(147, 231, 101, 255), stop:1 rgba(51, 255, 255, 255);}\\n"
51                           " color: rgb(255, 255, 255);}\\n"
52                           "")
53     self.mic.setText("")
54     icon1 = QtGui.QIcon()
55     icon1.addPixmap(QtGui.QPixmap(":/newPrefix/micro.png"),
56                   QtGui.QIcon.Normal, QtGui.QIcon.Off)
57     self.mic.setIcon(icon1)
58     self.mic.setIconSize(QtCore.QSize(45, 45))
59     self.mic.setObjectName("mic")
60     self.exit = QtWidgets.QPushButton(self.frame)
61     self.exit.setGeometry(QtCore.QRect(330, 20, 61, 51))
62     self.exit.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
63     self.exit.setStyleSheet("QPushButton\\n"
64                           "{background-color: qlineargradient(spread:pad, x1:0.979211, y1:0.977, x2:0.00526316, y2:0.994, stop:0 rgba(66, 216, 255, 255), stop:0.689\\n"
65                           "border-radius:25px;}\n"
66                           "QPushButton:hover{ background-color: qlineargradient(spread:pad, x1:0, y1:1, x2:1, y2:1, stop:0 rgba(147, 231, 101, 255), stop:1 rgba(51, 255, 255, 255);}\\n"
67                           " color: rgb(255, 255, 255);}\\n"
68                           "")
69     self.exit.setText("")
70     icon2 = QtGui.QIcon()
71     icon2.addPixmap(QtGui.QPixmap(":/newPrefix/exit.png"),
72                   QtGui.QIcon.Normal, QtGui.QIcon.Off)
73     self.exit.setIcon(icon2)
74     self.exit.setIconSize(QtCore.QSize(32, 32))
75     self.exit.setObjectName("exit")
76     self.pushButton = QtWidgets.QPushButton(self.frame)
77     self.pushButton.setGeometry(QtCore.QRect(110, 100, 181, 141))
78     self.pushButton.setStyleSheet("QPushButton{background-color: rgb(7,24,73);\\n"
79                               "border-radius:25px;\\n"
80                               "}")
81     self.pushButton.setText("")

82     icon3 = QtGui.QIcon()
83     icon3.addPixmap(QtGui.QPixmap(":/newPrefix/Den.png"),
84                   QtGui.QIcon.Normal, QtGui.QIcon.Off)
85     self.pushButton.setIcon(icon3)
86     self.pushButton.setIconSize(QtCore.QSize(300, 300))
87     self.pushButton.setObjectName("pushButton")
88     self.frame_2.raise_()
89     self.git.raise_()
90     self.exit.raise_()
91     self.pushButton.raise_()
92     MainWindow.setCentralWidget(self.centralwidget)

93     self.retranslateUi(MainWindow)
94     QtCore.QMetaObject.connectSlotsByName(MainWindow)

95     def retranslateUi(self, MainWindow):
96         _translate = QtCore.QCoreApplication.translate
97         MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))

98     if __name__ == "__main__":
99         import sys
100        app = QtWidgets.QApplication(sys.argv)
101        MainWindow = QtWidgets.QMainWindow()
102        ui = Ui_MainWindow()
103        ui.setupUi(MainWindow)
104        MainWindow.show()
105        sys.exit(app.exec_())
106
107
108
109
110

```

Run Code

Run file integrates our assistant file and our ui file for functioning in proper manner. This file gives functionality to our buttons we designed in our QT Designer software.

```
● 1 import assistant
  2 from ui import Ui_MainWindow
  3 from PyQt5 import QtCore, QtGui, QtWidgets
  4 from PyQt5.QtCore import *
  5 from PyQt5.QtGui import *
  6 from PyQt5.QtWidgets import *
  7 from PyQt5.QtCore import Qt, QTime, QTimer, QDate
  8 from PyQt5.uic import loadUiType
  9 import os
10 import sys
11 import webbrowser as web
12
13
14 class MainThread(QThread):
15
16     def __init__(self):
17         super(MainThread, self).__init__()
18
19     def run(self):
20         assistant.assisrun()
21
22
23 startExe = MainThread()
24
25
26 class Gui_start(QMainWindow):
27
28     def __init__(self):
29         super().__init__()
30         self.gui = Ui_MainWindow()
31         self.gui.setupUi(self)
32         self.gui.mic.clicked.connect(self.start)
33         self.gui.exit.clicked.connect(self.close)
34         self.gui.git.clicked.connect(self.chrome_app)
35
36     def chrome_app(self):
37         web.open("https://www.github.com/ayushsaini12/DenTheAssistant.git")
38
39     def start(self):
40         startExe.start()
41
42
43 GuiApp = QApplication(sys.argv)
44 assistantui = Gui_start()
45 assistantui.show()
46 sys.exit(GuiApp.exec_())
47
```

For further code reference you can navigate to our github repository, which can be accessed using github logo on our ui or can navigate to the link given below:
<https://github.com/ayushsaini12/DenTheAssistant>

Challenges

The main challenges we faced are like gathering the main information on the assistant ,like related to the coading part. While adding the website links in code and finding the error, installing the different types of packages like eg:speech recognition,pyttsx3,pywhatkit,pyjokes.

After completing the coding part the main challange is making the user interface for the coding part. As its a new task for us we had the main problem in creating that UI on our own while gathering some information on it it has given some good output and we continues it and created that.

At last while we need to give a name to that and we came across that it should be some small and new .So that we named it as 'DEN'.

Conclusion

After the completion of the whole work we have seen the expected results what we want.

We made the the assistant for simple use and for the essential needs like for playing videos on YouTube, time, food, cabs, shopping, changing brightness, calculator and some more.

And also we had learnt the making of UI for program.

Reference

These are the some of the references of our project

- <https://youtu.be/AWvsXxDtEkU>
- <https://github.com/ProgrammingHero1/romantic-alexa>
- <https://youtu.be/Lp9Ftuq2sVI>
- <https://codewithharry.com/videos/python-tutorials-for-absolute-beginners-120>
- <https://www.geeksforgeeks.org/voice-assistant-using-python/>



DEN

Plagiarism Report

Ayush Saini User Info Messages Student English Help Logout

turnitin

Class Portfolio My Grades Discussion Calendar

NOW VIEWING: HOME > SOHRAB HOSSAIN

Welcome to your new class homepage! From the class homepage you can see all your assignments for your class, view additional assignment information, submit your work, and access feedback for your papers. Hover on any item in the class homepage for more information.

Class Homepage

This is your class homepage. To submit to an assignment click on the "Submit" button to the right of the assignment name. If the Submit button is grayed out, no submissions can be made to the assignment. If resubmissions are allowed the submit button will read "Resubmit" after you make your first submission to the assignment. To view the paper you have submitted, click the "View" button. Once the assignment's post date has passed, you will also be able to view the feedback left on your paper by clicking the "View" button.

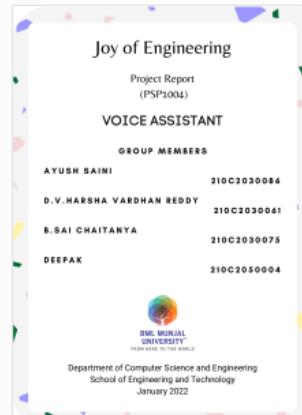
Assignment Inbox: Sohrab Hossain				
Assignment Title	Info	Dates	Similarity	Actions
JoE Course	①	Start 15-Jan-2022 11:22AM Due 22-Jan-2022 11:59PM Post 23-Jan-2022 12:00AM	7% 	Submit View 

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author: Ayush Saini
 Assignment title: JoE Course
 Submission title: Voice Assistant
 File name: Joy_of_Engineering.pdf
 File size: 7.87M
 Page count: 13
 Word count: 659
 Character count: 3,508
 Submission date: 22-Jan-2022 12:50AM (UTC+0530)
 Submission ID: 1745584356



Copyright 2022 Turnitin. All rights reserved.