iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** I **Month of publication:** January 2022

DOI: https://doi.org/10.22214/ijraset.2022.40083

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue I Jan 2022- Available at www.ijraset.com

Customized Automated Email Bot

Shruti Mane[1], Prof. Bhanuse Vijaykumar Rayappa[2]
[1]*Vishwakarma Institute of Technology*

Abstract: ***Email is one of the most reliable, authentic, genuine and frequently used modes of communication among the corporate sector. It is a common practice in corporate culture to send an automatic response for the new incoming messages but it is very generic and limited in extent of its functionality. A system can be designed to substitute the tedious work of replying to thousands of mails using human force. This work proposes a system which in overall automates the work of responding to a client/user. It aims to minimize the laborious task of going through email manually by customizing the response as per the query of the user and designing a reply as per the condition. The proposed system gives us the information regarding effectiveness of the approach in its implementation***

I. INTRODUCTION

One of the most ubiquitous and widely used modes of communication in professional as well as personal aspects is Email. It is always exciting when one seeks someone or some enterprise out through their email and they respond without wasting much of the time. Now-a-Days in society, especially as technology is growing, people are looking for immediate response and satisfaction. Not only does it provide with a boost in efficiency but also it may be a great and a very useful business tool if used in a proper manner. For example, if you cannot respond within 24-48 hours, automated emails are
great ways to inform the email recipients when you will have a delay in your response due to a project, vacation, and so on. Email is amongst one of the most robust and secures means of online communication and it has become communication tool of paramount importance for most enterprises as well as professionals.

II. METHODOLOGY

Imported the required libraries.
Gave access to email id.
Used assistant for the project in English language.
Added a contact list for sending emails.
Used if-while for adding clients to whom email is to sent, subject and content of the project.
And then the email is sent successfully.

III. LIMITATION

A. It does not operate unless it is given further commands
B. A bot is not a person.
C. They need time for implementation of thing to be done.
D. They don't understand the human context directly.

IV. CONCLUSION

The work proposed in this paper is a system which to be used for generating intelligent replies to certain class of emails.
The proposed system is intended to build a framework for efficient email management system, capable of classifying and along with that also automatically replying to a certain class of emails.

Check By: Dupli Checker

Code :-
```
import smtplib
from email.message import EmailMessage
import speech_recognition as sr
import pyttsx3
```

```
listener = sr.Recognizer()
engine = pyttsx3.init()
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)
server = smtplib.SMTP("smtp.gmail.com", 587)
server.starttls()
server_login_mail = "maneshrutii123@gmail.com"
server_login_password = "Satara@123"
server.login(server_login_mail, server_login_password)


def say(text):
    engine.say(text)
    engine.runAndWait()

say("hello, how can i help you? myself email assistant")

def assistant_listener():
    try:
        with sr.Microphone() as source:
            print("Listening...")
            voice = listener.listen(source)
            info = listener.recognize_google(voice, language="en-in").lower()
            return info

    except:
        return "no"

def send_email(rec, subject, message):
    email = EmailMessage()
    email['From'] = server_login_mail
    email['To'] = rec
    email['Subject'] = subject
    email.set_content(message)
    server.send_message(email)

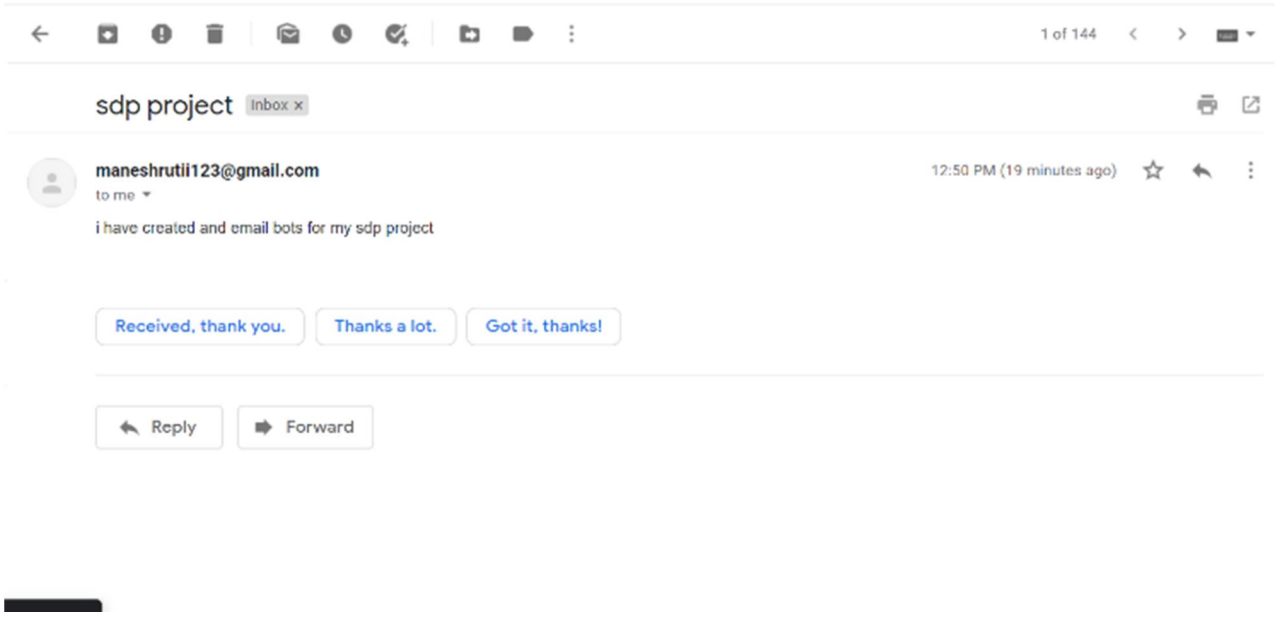
contact = {
    "vit": "shruti.mane20@vit.edu",
    "shruti": "shrutimane5122@gmail.com"
}

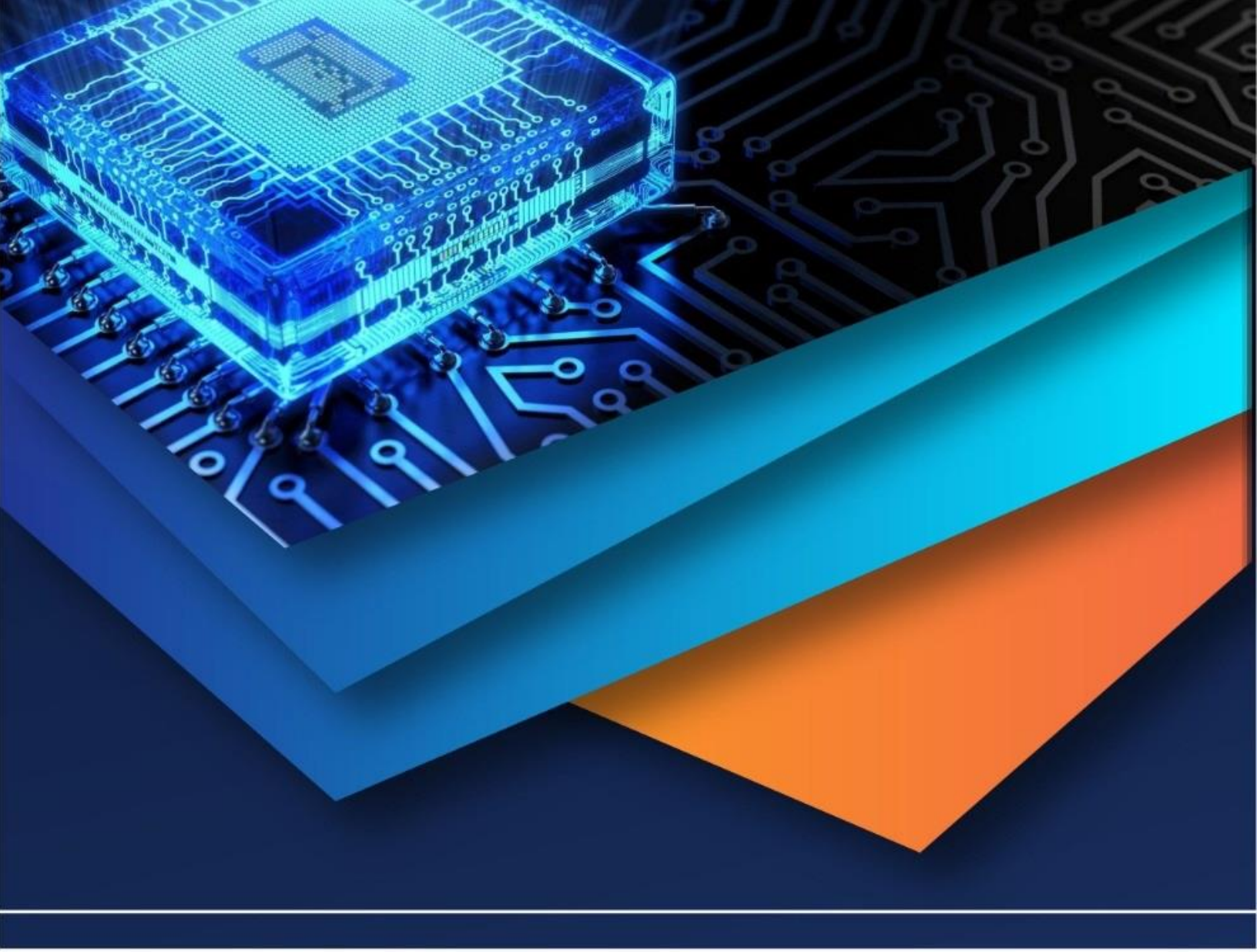
def whattodo():
    listen_me = assistant_listener()
    if "assistant" in listen_me:
        if "write mail" in listen_me:
            say("To whom you want to send mail?")
            try:
```

```
        user = assistant_listener()
        email = contact[user]
    except:
        say(user+" not found in your contacts!")
        return 0
    say("What you want to be subject?")
    subject = assistant_listener()
    say("what should be the message?")
    message = assistant_listener()
    send_email(email, subject, message)
    say("Email Send Successfully")

while True:
    whattodo()
```

V. RESULT





10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)