

A Novel Python-based Voice Assistance System for reducing the Hardware Dependency of Modern Age Physical Servers

Rajdip Paul¹, Nirmalya Mukhopadhyay²

¹Dept. of Computer Science and Engineering, , Assam down town University, Assam, India

²Assistant Professor, Dept. of Computer Science and Engineering, Assam down town University, Assam, India

Abstract - With technology getting better and advanced every single day through intensive inventions and discoveries carried out by experts all around the world, our lives are getting modified and bent more towards being dependent on technology and making the best out of it. With such a level of enhancements, it will be really illogical, if we still struggle to interact with our systems through various contemporary I/O devices (input/output). As, we need to interact with our smart systems almost every hour of the day and it is really necessary to use hands-free techniques other than the use of peripheral devices to communicate with these computer systems; techniques that are faster, easier and affordable. To cope up with these requirements, voice assistants are being made and developed during past few years. The main goal of a voice assistant is to reduce the use of I/O devices, to make interaction with our systems easy and possible for all. Also, it helps to save the space consumed by I/O devices and the cost we need to bear for buying and maintaining them properly. In this research paper, we have proposed a new python-based voice assistant that will be able perform almost every task that we use our devices for, just by listening to voice commands. It will be affordable for any class of the society and at some points better than the prevalent voice assistants available in the market. We have also provided a framework that shows the flow of control of the system. Finally, an extensive comparison with the existing systems proved the worthiness of our project.

Key Words: Voice Assistant; Python, I/O, Python Libraries, I/O dependencies, Voice-based commands, Peripheral devices

1.INTRODUCTION

There are many situations which we face in our daily life where we want things to get done by our computers using our verbal commands instead of interacting with the system using the keyboard and mouse. Verbal command control over systems helps especially people with disabilities a lot in their day-to-day work life and also enable them to maintain efficiency and compete. Along with this, as we live in an era of multitasking it becomes really necessary to complete a task quickly and move on to another and voice command-based systems helps us in a great way. Google CEO Sundar Pichai says 'We are now witnessing a new shift in computing: the move from a mobile-first to an AI-first world.' Also, in this new normal it is safer for everybody to avoid physical

interaction with office systems or systems that come in contact with numerous numbers of people. One solution to all such kind of problems or situations is using a voice assistant. Some of the basic tasks supported by most of the voice assistants are:

Checking weather updates- knowing the weather before starting a day or planning a trip is a very common thing that most people do and with your voice assistant you can easily get the weather update of any place just by asking it to get the weather information for you.

Sending and checking mails- In this world of internet sending and checking mails is the most common task that we all perform at least once a day and if you are an office goer then mailing activities are surely an integral part of your life where you send and receive multiple emails. Now imagine if you could automatically maintain your mail sphere and get away from doing redundant mailing activities and also instead of wasting time typing mails you could speak out the content of the mail and let your assistant type it for you. It will surely save time because as humans we can only type 40 words per minute while we can speak 150 words per minute.

Search Wikipedia- Searching Wikipedia is also an integral part of our lives and with voice assistants we can ease out this task.

Streaming music, opening applications are other fundamental activities that we can perform easily with our voice assistants.

The world is tilting towards using voice assistants. According to ComScore 50% of all searches will be voice searches by 2020. Voice search will be used by the masses in the not-too-distant future because the easiest way to find an answer is to verbalize and ask a question. According to Google between 20 to 25% of mobile queries are voice searches. 21% of mobile voice search users said they use voice search because they don't like typing on their mobile device (Source: Statista 2015). According to Kleiner Perkins Caufield and Byers (KPCB) Internet trends, as of May 2016, 1 in 5 searches on an Android app in the U.S. were through speech. Since 2008, Google voice search queries are up by over 35X. Voice search is quicker – Humans can speak 150 words per minute vs typing 40 words per minute. As a result, 43% of people say that using voice search is quicker than using a website or an app (Source: Statista) – in addition, 28% of people think that voice search is a more accurate way of searching.

"Personal assistants" was the top marketing search of 2016 in Bing – earlier this year, Microsoft's Bing Ads released their top marketing-related searches of 2016. With increased interest in chatbots and virtual assistants like Alexa, Cortana and Amazon Echo, personal assistants and AI saw the biggest gains in search activity.

25% of searches on Windows 10 taskbar are voice – this coming from desktop and not even a mobile device!

19% of people use Siri at least daily with 37% of people using Siri monthly, 23% using Microsoft Cortana monthly and 19% using Amazon's Alexa monthly – according to HubSpot (Source: <https://www.hubspot.com/marketing-statistics>)

60% of smartphone users who use voice search have begun using it in the past 12 to 18 months – Search Engine Land 2015

About 30% of searches will be done without a screen by 2020 – thanks to technology such as Google Assistant for home, Amazon Alexa for Echo and chat bots (Source: Mediapost).

40% of adults use voice search once per day – (Source: comScore)

According to the Global Web Index, nearly 10% of the online population or 325.8 million people used voice control in the past month. – Source: <https://www.globalwebindex.net/>

It is estimated that 500,000 Google Home units shipped in 2016 – Bloomberg

It is estimated that Amazon has sold 5.1 million Echo device units in the US since launched in 2014 – with 2 million units being sold in the first nine months of 2016 – Source: GeekWire

Nearly 50% of people are now using voice search when researching products – Social Media Today

In a study from 2014, Google found that 55% of teens and 41% of adults in the US are now using voice search more than once per day – Not surprising when you think about it as teens are always ahead of the curve when it comes to new technology. They tend to talk to their phones more than the average adult. Source: Google Mobile Voice Study

According to that same study from Google, 40% of adults use voice search to find directions while 43% of teens use voice search to call someone. Google Mobile Voice Search Survey Results

78% of teens said that they use voice search because it's safer – while 73% of teens says it makes them more efficient. Compared to adults where 76% said voice search is safer but only 62% said it made them more efficient. (Source: Google)

Consumer usage is on the rise – 51% of consumers use voice assistants in automobiles and 39% use voice assistants in

their homes indicating an increased comfort level with voice search technology. (Source: Wall Street Journal)

But most people in our economy cannot afford buying the commercial voice assistants that are available or lack the technical knowledge of how to use a voice assistant efficiently. Moreover, rise in data theft in recent times is also a main reason that people do not prefer using a voice assistant as it requires entire system control. To cope up with all such issues we came up with the idea of building our very own native voice assistant which will be efficient enough to perform our day-to-day system task, which can be customized according to one's needs, while maintaining data security. The voice assistant that we have developed is a desktop-based built using python modules and libraries. Along with performing all the tasks which the existing voice assistants can exhibit, our voice assistant "Penny" can also check date and time as it is the most fundamental activity that we all do every day, take note to release you from writing important pointers and also helps to note pointers fast, also show the notes, open YouTube for you, open Google, close YouTube and Google, locate a place, open and close system applications. This assistant is still in its basic level and we strongly believe that we can do a lot more with it and maybe we can surpass the existing voice assistants someday. It is still to be merged with Machine Learning and Internet of Things (IoT) for better enhancements. We are also trying to implement the concept of Natural Language Processing (NLP) in our voice assistant to make it more flexible in terms of understanding the commands from different users. Moreover, our program uses the least amount of system resources which minimizes the expensive system requirements which are needed by existing voice assistants. Also, it reduces threat to your system as our system directly does not interact with servers and does not use the host as for storage rather uses google servers and databases. Along with these our program will probably be the cheapest voice assistant in the market if someday it comes as a market product. Also, it will reduce the use of various input devices thus further reducing system cost and save the physical space used by such devices. In a nutshell we are very sure that it is very much feasible system.

Timeline of Mainstream Voice Assistants



Fig -1: Timeline of popular voice assistants^[12]

2. Related Work

The field of voice-based assistants has observed major advancements and innovations. The main reason behind such rapid growth in this field is its demand in devices like smartwatches or fitness bands, speakers, Bluetooth earphones, mobile phones, laptop or desktop, television, etc. Most of the smart devices that are being brought in the market today have built in voice assistants. The amount of data that is generated nowadays is huge and in order to make our assistant good enough to tackle these enormous amounts of data and give better results we should incorporate our assistants with machine learning and train our devices according to their uses. Along with machine learning other technologies which are equally important are IoT, NLP, Big data access management. The use of voice assistants can ease out a lot of tasks for us. Just give voice command input to the system and all tasks will be completed by the assistant starting from converting your speech command to text command then taking out the keywords from the command and execute queries based on those keywords. In the paper "Speech recognition using flat models" by Patrick Nguyen and all, a novel direct modelling approach for speech recognition is being brought forward which eases out the measure of consistency in the sentences spoken. They have termed this approach as Flat Direct Model (FDM). They did not follow the conventional Markov model and their model is not sequential. Using their approach, a key problem of defining features has been solved. Moreover, the template-based features improved the sentence error rate by 3% absolute over the baseline [2]. Again, in the paper "On the track of

Artificial Intelligence: Learning with Intelligent Personal Assistant" by Nil Goksel and all, the potential use of intelligent personal assistants (IPAs) which use advanced computing technologies and Natural Language Processing (NLP) for learning is being examined. Basically, they have reviewed the working system of IPAs within the scope of AI [4]. The application of voice assistants has been taken to some higher level in the paper "Smart Home Using Internet of Things" by Keerthana S and all where they have discussed how the application of smart assistants can lead to developing a smart home system using Wireless Fidelity (Wi-Fi) and Internet of Things. They have used CC3200MCU that has in-built Wi-Fi modules and temperature sensors. The temperature that is sensed by the temperature sensor is sent to the microcontroller unit (MCU) which is then posted to a server and using that data the status of electronic equipment like fan, light etc is monitored and controlled [5]. The application of voice assistants has been beautifully discussed in the paper "An Intelligent Voice Assistant Using Android Platform" by Sutar Shekhar and all where they have stressed on the fact that mobile users can perform their daily task using voice commands instead of typing things or using keys on mobiles. They have also used a prediction technology that will make recommendations based on the user activity [6]. The incorporation of natural language processing (NLP) in voice assistants is really necessary which will also lead to the creation of a trendsetting assistant. These factors have been the key focus of the paper "An Intelligent Chatbot using Natural Language Processing" by Rishabh Shah and all. They have discussed how NLP can help to make assistants smart enough to understand commands in any native language and thus does not prevent any part of the society from enjoying its perks [7]. We also studied the systems developed by Google Text To Speech – Electric Hook Up (GTTS-EHU) for Query-by-example Spoken Term Detection (QbE-STD) and Spoken Term Detection (STD) tasks of the Albayzin 2018 Search on Speech Evaluation. For representing audio documents and spoken queries Stacked bottleneck features (sBNF) are used as frame level acoustic representation. Spoken queries are synthesized, average of sBNF representations is taken and then the average query is used for Qbe-STD [8]. We have seen the integration of technologies like gTTS, AIML (Artificial Intelligence Mark-up Language) in the paper "JARVIS: An interpretation of AIML with integration of gTTS and Python" by Tanvee Gawand and all where they have adopted the dynamic base Python pyttss which is a text to speech conversion library in python and unlike alternative libraries, it works offline [9]. The main focus of voice assistants should be to reduce the use of input devices and this fact has been the key point of discussion in the paper

“VOICE ACTIVATED DESKTOP ASSISTANT USING PYTHON” by Dhiraj Pratap Singh. They have discussed that their machine will work in three modes- supervised, unsupervised or reinforcement learning which will vary according to needs of users [10].

Information & Communication Use Cases Outpace Entertainment for Voice Assistants on Smartphones

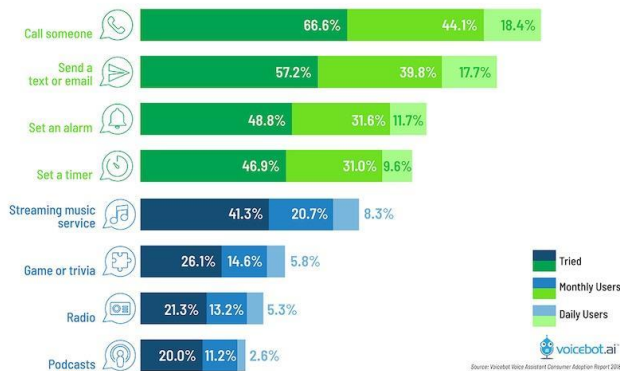


Fig -2: Usage of voice assistants^[13]

3. Theory

```

1 import pytsx3 #pip install pytsx3
2 import speech_recognition as sr #pip install speechRecognition
3 import datetime
4 import wikipedia #pip install wikipedia
5 import webbrowser
6 import os
7 import smtplib
8 import random
9 import sys
10 import subprocess
11 import selenium
12 from pygame import mixer
13 import requests #for weather
14 from urllib.request import urlopen

```

Fig -3: Imported modules

pytsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3. If you receive errors such as No module named win32com.client, No module named win32, or No module named win32api, you will need to additionally install pypiwin32.

The Speech Application Programming Interface or SAPI is an API developed by Microsoft to allow the use of speech recognition and speech synthesis within Windows applications.

speech_recognition is a library for performing speech recognition, with support for several engines and APIs, online and offline. Speech recognition, as the name suggests, refers to automatic recognition of human speech. Speech recognition is one of the most important tasks in the domain of human computer interaction. If you have ever interacted with Alexa or have ever ordered Siri to complete a task, you have already experienced the power of speech recognition.

Datetime module supplies classes for manipulating dates and times. While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation.

Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia. Search Wikipedia, get article summaries, get data like links and images from a page, and more. Wikipedia wraps the MediaWiki API so you can focus on using Wikipedia data, not getting it.

webbrowser module provides a high-level interface to allow displaying Web-based documents to users. Under most circumstances, simply calling the open() function from this module will do the right thing.

os module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see open(), if you want to manipulate paths, see the os.path module, and if you want to read all the lines in all the files on the command line see the fileinput module. For creating temporary files and directories see the tempfile module, and for high-level file and directory handling see the shutil module.

smtplib module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

random module implements pseudo-random number generators for various distributions. For integers, there is uniform selection from a range. For sequences, there is uniform selection of a random element, a function to generate a random permutation of a list in-place, and a function for random sampling without replacement.

sys module provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter. It is always available.

subprocess module allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes. This module intends to replace several older modules and functions

selenium package is used to automate web browser interaction from Python

Pygame is a cross-platform set of Python modules which is used to create video games. It consists of computer graphics and sound libraries designed to be used with the Python programming language. Pygame was officially written by Pete Shinners to replace PySDL. Pygame is suitable to create client-side applications that can be potentially wrapped in a standalone executable.

Requests allows you to send HTTP/1.1 requests extremely easily. The requests module allows you to send HTTP requests using Python. The HTTP request returns a

Response Object with all the response data (content, encoding, status, etc).

urllib.request is a Python module for fetching URLs (Uniform Resource Locators). It offers a very simple interface, in the form of the urlopen function. Our system is designed to function in the following ways:

- (a) It takes input from the user in the form of voice.
- (b) In order to make the assistant understand the input the speech input is converted to text input.
- (c) After generating the text input, it is processed in order to get the required result or output.
- (d) The input text contains the keywords which makes it possible for the assistant to understand which query it should execute. But in case the keywords do not match any of the queries in the code, then the assistant asks the user to speak the command again.
- (e) Once the exact query gets executed the user is prompted with the desired result.

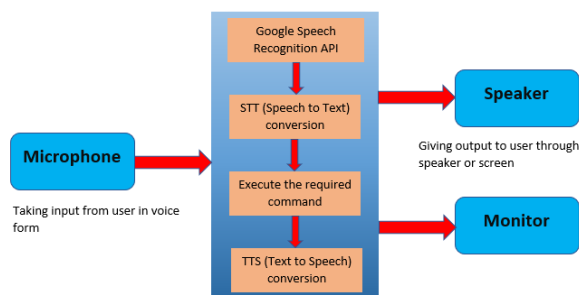


Fig -4: Processing block of STT to TTS

4. Methodology

At the outset we make our program capable of using system voice with the help of sapi5 and pyttsx3. 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 Speech Application Programming Interface or SAPI is an API developed by Microsoft to allow the use of speech recognition and speech synthesis within Windows applications. Then we define the speak function to enable the program to speak the outputs. After that we will define a function to take voice commands using the system microphone. The main function is then defined where all the capabilities of the program are defined.

The proposed system is supposed to have the following functionality:

- (a) The assistant asks the user for input and keeps listening for commands. The time for listening can be set according to user's requirement.
- (b) If the assistant fails to clearly grasp the command it will keep asking the user to repeat the command again and again.
- (c) This assistant can be customized to have either male or female voice according to user's requirement.

(d) The current version of the assistant supports features like Checking weather updates, Sending and checking mails, Search Wikipedia, Stream music, Open applications, Text messages, Check date and time, take note, show note, Open YouTube, Google, Close YouTube, Google, Open and close applications, Locate a place

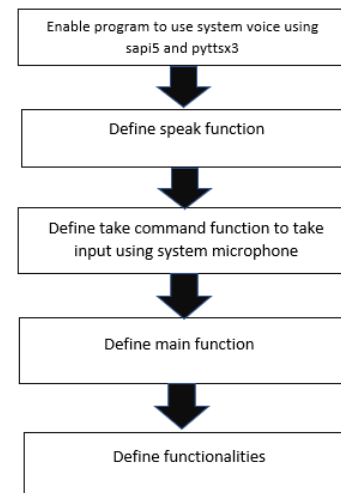


Fig -5: Methodology

5. Result and Discussion

We have tested our program with various inputs and go the results. Following are some of the screenshots of the results:

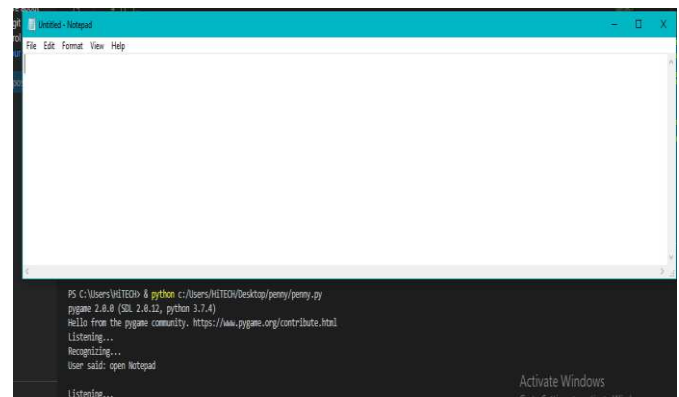


Fig -6: Opening Notepad using voice command

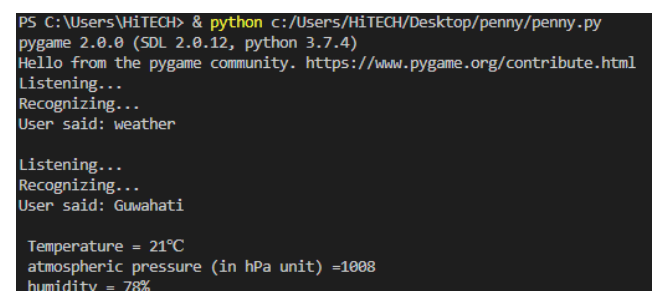


Fig -7: Viewing weather information

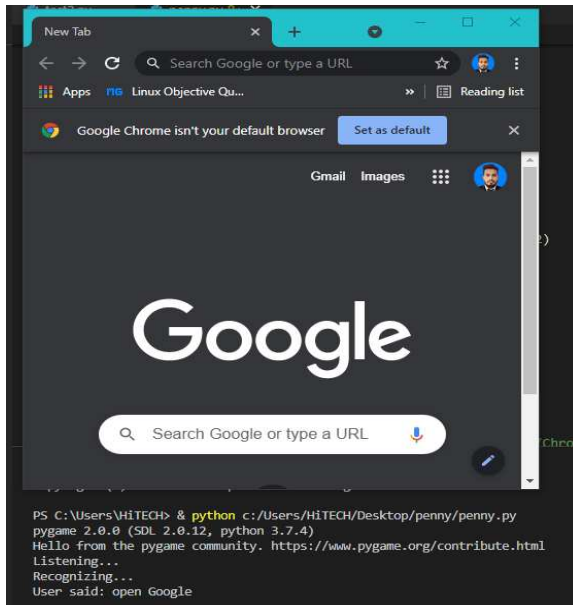


Fig -8: Opening Google with voice command

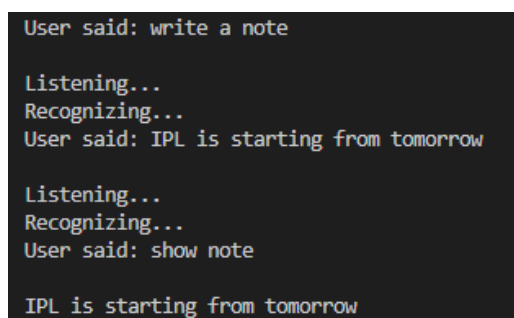


Fig -9: Taking notes

6. Conclusion and Future Scope

The current version of our assistant has some good features and is quite well responsive but there is lot that is yet to be improved. The understanding and reliability of our assistant can be improved more. The future development of the assistant includes merging NLP, Machine learning, IoT with it. By incorporating these technologies with our assistant, we will be able to achieve better results. What the virtual assistants can achieve is much beyond what we have achieved till now

ACKNOWLEDGEMENT

Throughout the writing of this dissertation, I have received a great deal of support and assistance. I would like to thank my mentor, Professor Nirmalya Mukhopadhyay, whose expertise was invaluable in formulating the research questions and methodology. His insightful feedback pushed me to sharpen my work to a higher level. I could not have

completed this dissertation without his support and the stimulating discussions as well as happy distractions to rest my mind outside of my research.

REFERENCES

- [1] D O'Shaughnessy, "Interacting With Computers by Voice: Automatic Speech Recognition and Synthesis", proceedings of THE IEEE, VOL. 91, NO. 9, 2003. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
- [2] Patrick Nguyen, Georg Heigold, Geoffrey Zweig, "Speech Recognition with Flat Direct Models", IEEE Journal of Selected Topics in Signal Processing, 2010.
- [3] Vinay Sagar, Kusuma SM, "Home Automation Using Internet of Things", IRJET, e-ISSN: 2395, 2015.
- [4] Nil Goksel, Canbek Mehmet, Emin Mutlu, "On the track of Artificial Intelligence: Learning with Intelligent Personal Assistant", proceedings of International Journal of Human Sciences, 2016.
- [5] Keerthana S, Meghana H, Priyanka K, Sahana V. Rao, Ashwini B "Smart Home Using Internet of Things", proceedings of Perspectives in Communication, Embedded-systems and signal processing, 2017
- [6] Sutar Shekhar, P. Sameer, Kamad Neha, Prof. Devkate Laxman, "An Intelligent Voice Assistant Using Android Platform", IJARCSMS, ISSN: 232-7782, 2017.
- [7] Rishabh Shah, Siddhant Lahoti, Prof. Lavanya. K, "An Intelligent Chatbot using Natural Language Processing", International Journal of Engineering Research, Vol.6, pp.281-286, 2017.
- [8] Luis Javier Rodríguez-Fuentes, Mikel Peñagarikano, Aparo Varona, Germán Bordel, "GTTS-EHU Systems for the Albayzin 2018 Search on Speech Evaluation", proceedings of IberSPEECH, Barcelona, Spain, 2018.
- [9] Ravivanshikumar, Sangpal, Tanvee, Gawand, Sahil Vaykar, "JARVIS: An interpretation of AIML with integration of gTTS and Python", proceedings of the 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT), Kanpur, 2019.
- [10] Dhiraj Pratap Singh, Deepika Sherawat, Sonia, "Voice activated desktop assistant using Python", proceedings of High Technology Letters, ISSN: 1006-6748, 2020.
- [11] Guido Van Rossum, Fred L. Drake, "An Introduction to Python", Network Theory Limited, UK, pp. 25-73, 2003.
- [12] Image source: Smartsheet.com.
- [13] Image source: voicebot.ai.

BIOGRAPHIES



Mr Rajdip Paul is pursuing Bachelor of Technology in CSE from Assam down town University, Guwahati, India since 2017. He will be graduating in July 2021. He is currently doing his research work in Python programming language and also setting up his own IT startup. He is a Google certified IT Support

expert, Google certified python automation expert, certified pentester from IIT Hyderabad, and certified Network administrator and analyst from IIHT. He is a member of ISTE & IMUN, a life member of the E-Cell of IITs since 2017 and also a distinguished ambassador of the technical and cultural boards for IITs in North East. He has published his last research paper in the journal of Assam down town University on the topic of wireless sensor nodes. His main research work focuses on Python language, Network security, wireless communication and innovating business processes through implementation of IT. He is a fresher with a zeal learning new technologies and implementing them to make life better.

teaching experience and 6 years of Research Experience.



Mr. Nirmalya Mukhopadhyay pursued Bachelor of Technology in CSE from MAKAUT, India in 2009 and Master of Technology in CSE from MAKAUT, India in 2012. He is currently doing his research work in Cloud domain and working as Assistant Professor in Department of Computer Science & Engineering, Assam down town University, Guwahati, India since 2019. He is a member of ACM, CSTA, IEEE, IAENG, CSI & AACSIT since 2013. He has published 14 research papers in reputed International Journals including UGC Care Listed Journals, Ebsco Indexed Journals and Copernicus Indexed Journals. He has been a Google Scholar since 2012. His main research work focuses on Virtualization, Cloud Resource Allocation, Cloud Scheduling, Cloud Load Balancers, Cloud Algorithms, Virtualization Security, Cloud Security and Privacy, High Performance Computing, Cloud Performance Analytics, Docker & Containerization, Edge Computing, Optimization Problems, Grid Computing, Genetic Algorithm, IoT and Computational Intelligence Elucidation. He has 10 years of