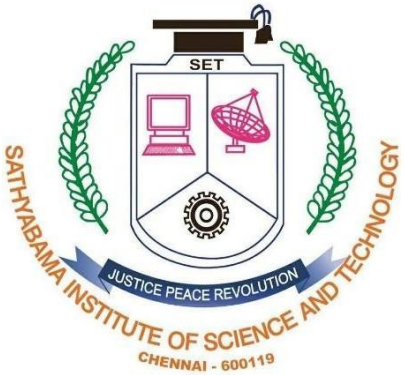
AI DESKTOP ASSISTANT

Submitted in partial fulfillment of the requirements
for the award of
Bachelor of Engineering degree in Computer Science and Engineering By

Sri Meenakshi Pandey (Reg.No - 39110967)
S Sindu (Reg.No – 39110948)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119

APRIL - 2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Sri Meenakshi Pandey(Reg.No - 39110967) and S Sindu(Reg.No - 39110948)** who carried out the Project Phase-2 entitled **"AI DESKTOP ASSISTANT"** under my supervision from January 2023 to April 2023.

Internal Guide
Ms. C A Daphine Desona Clemency, M.E.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on 20.4.2023

Internal Examiner **External Examiner**

DECLARATION

I, **Sri Meenakshi Pandey (Reg.No- 39110967),** hereby declare that the Project Phase-2 Report entitled "**AI DESKTOP ASSISTANT"** done by me under the guidance of **Ms. C A Daphine Desona Clemency, M.E.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:20.4.2023

Sri Meenakshi Pandey

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D**, **Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. C A Daphine Desona Clemency, M.E.** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

As we know Python is an emerging language, so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the process of converting speech into text. This is commonly used in voice assistants like Alexa, Siri, etc. In Python there is an API called Speech-Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. In the current scenario, advancement in technologies is such that they can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. The AI Desktop Assistant project aims to develop a cutting-edge software application that leverages the latest advancements in Artificial Intelligence to provide users with a personalized, intuitive, and efficient desktop assistant. The assistant will use natural language processing to interpret user commands, generate personalized responses, and perform various tasks, including but not limited to scheduling appointments, sending emails, managing files, and providing recommendations based on user preferences. The system will also learn from user interactions, continually improving its accuracy and functionality over time. The project will utilize machine learning techniques such as deep neural networks to develop robust models that can accurately understand and respond to user input. The end goal is to create a user-friendly, efficient, and reliable AI desktop assistant that can improve productivity, simplify tasks, and enhance the overall user experience.

TABLE OF CONTENTS

Chapter No	TITLE		Page No.
	ABSTRACT		v
	LIST OF FIGURES		viii
1	**INTRODUCTION**		1
2	**LITERATURE SURVEY**		4
	2.1 Inferences from Literature Survey		6
	2.2 Open problems in Existing System		7
3	**REQUIREMENTS ANALYSIS**		
	3.1	Feasibility Studies/Risk Analysis of the Project	11
	3.2	Software Requirements Specification Document	11
	3.3	System Use case	12
4	**DESCRIPTION OF PROPOSED SYSTEM**		17
	4.1	Selected Methodology or process model	18
	4.2	Architecture / Overall Design of Proposed System	19
	4.3	Description of Software for Implementation and Testing plan of the Proposed Model/System	19
	4.4	Project Management Plan	24
	4.5	Transition/ Software to Operations Plan	25
5	**IMPLEMENTATION DETAILS**		
	5.1	Development and Deployment Setup	27
	5.2	Algorithms	28
	5.3	Testing	32
6	**RESULTS AND DISCUSSION**		34
7	**CONCLUSION**		
	7.1	Conclusion	37
	7.2	Future work	38
	7.3	Research Issues	39
	7.4	Implementation Issues	39
	REFERENCES		41
	APPENDIX		

	A. SOURCE CODE	43
	B. SCREENSHOTS	48
	C. RESEARCH PAPER	53

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
3.1	*ER Diagram*	*13*
3.2	*Activity Diagram*	*14*
3.3	*Class Diagram*	*14*
3.4	*Use Case Diagram*	*15*
3.5	*Sequence Diagram*	*15*
3.6	*Component Diagram*	*16*
4.1	*A representation of an AI Assistant receiving commands from user*	*18*
4.2	*System Architecture for AI Desktop Assistant*	*19*
5.1	*K-Means Clustering*	*29*
5.2	*Difference between K-Means and MiniBatch K-Means*	*31*

CHAPTER 1

INTRODUCTION

An AI desktop virtual assistant is a software application that uses artificial intelligence and natural language processing technologies to interact with users in a conversational manner. It is designed to perform a variety of tasks, such as scheduling appointments, setting reminders, searching the web, sending emails, and controlling smart home devices, among others. The assistant can be activated through voice commands, typing or clicking on a button, and it is capable of learning from user interactions to improve its performance over time. AI desktop virtual assistants have become increasingly popular in recent years due to their ability to simplify daily tasks and enhance productivity, making them an essential tool for both personal and professional use.

Artificial Intelligence when used with machines, it shows us the capability of thinking like humans. In this, a computer system is designed in such a way that typically requires interaction from human. As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the Alexa, Siri, etc. In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. In the current scenario, advancement in technologies is such that they can perform any task with same effectiveness or can say more effectively than us.

By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. As the voice assistant is using Artificial Intelligence hence the result that it is providing are highly accurate and efficient. The assistant can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. The assistant is no less than a human assistant, but we can say that this is more effective and efficient to perform any task. The libraries and packages used to make this assistant focuses on the time complexities and reduces time. The functionalities include, it can send emails, it can read PDF, It can send text on WhatsApp, It can open command prompt, your favorite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast, It can give desktop reminders of your choice. It can have some basic conversation. Tools and technologies used are PyCharm IDE for making this project, and I created all py files in PyCharm. Along with this I used following modules and libraries in my project. pyttsx3, SpeechRecognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, pyQt etc. I have created a live GUI for interacting with the Assistant as it gives a design and interesting look while having the conversation.

An AI desktop virtual assistant is a revolutionary technology that has transformed the way we interact with our devices. The technology is designed to mimic human conversation, allowing users to communicate with their devices using natural

language. It has numerous applications and is being used in a variety of industries, from healthcare to finance and entertainment.

AI desktop virtual assistants are software programs that use artificial intelligence to understand and respond to user queries. They are designed to interact with users in a conversational manner, using natural language processing and machine learning algorithms to understand user input and provide relevant responses. They can be accessed through a range of devices, including desktops, laptops, and smartphones.

The use of AI desktop virtual assistants has become increasingly popular in recent years, as users seek to simplify their daily tasks and enhance their productivity. They can be used for a variety of purposes, such as scheduling appointments, setting reminders, searching the web, sending emails, and controlling smart home devices. This makes them an essential tool for both personal and professional use.

One of the key benefits of AI desktop virtual assistants is their ability to learn from user interactions. They use machine learning algorithms to analyze user data and improve their performance over time. This means that the more users interact with the assistant, the better it becomes at understanding their needs and providing relevant responses. As a result, the technology is becoming increasingly sophisticated, with new features and capabilities being added all the time.

Another benefit of AI desktop virtual assistants is their ability to integrate with other technologies. They can be used in conjunction with smart home devices, such as smart speakers and thermostats, to control the environment and perform other tasks. They can also be integrated with other software programs, such as productivity tools and customer relationship management (CRM) systems, to streamline workflows and improve efficiency.

In the healthcare industry, AI desktop virtual assistants are being used to improve patient care and reduce administrative burdens. They can be used to schedule appointments, send reminders, and provide patients with personalized healthcare information. They can also be used to analyze patient data and provide insights into patient health, allowing healthcare providers to make more informed decisions.

In the finance industry, AI desktop virtual assistants are being used to improve customer service and streamline workflows. They can be used to provide customers with personalized financial advice, analyze customer data to identify trends and opportunities, and perform routine tasks such as account balance inquiries and transaction histories.

In the entertainment industry, AI desktop virtual assistants are being used to enhance the user experience. They can be used to recommend movies and TV shows, provide personalized entertainment news and updates, and even create personalized playlists based on user preferences.

Despite the many benefits of AI desktop virtual assistants, there are also some concerns about their use. One of the key concerns is privacy and security. As these assistants are designed to collect and analyze user data, there are concerns about how this data is being used and who has access to it. There are also concerns about the accuracy and reliability of the technology, particularly in industries such as

healthcare where accuracy is critical.

To address these concerns, it is important to ensure that AI desktop virtual assistants are designed with privacy and security in mind. This means implementing robust security measures to protect user data, such as encryption and multi-factor authentication. It also means implementing transparency and accountability measures, such as clear privacy policies and user consent requirements.

Overall, AI desktop virtual assistants are a transformative technology that has the potential to revolutionize the way we interact with our devices. They offer numerous benefits, including improved productivity, personalized experiences, and enhanced customer service. As the technology continues to evolve, it is likely that we will see even more applications and benefits emerge, making them an essential tool for both personal and professional use.

CHAPTER 2

LITERATURE SURVEY

This area of digital assistants having speech reputation has visible some primary advancements or inventions. This is especially due to its call for in gadgets like smart watches or health bands, speakers, Bluetooth earphones, cellular telephones, computer or desktop, TV, and so forth. Almost all the digital gadgets which are coming nowadays with voice assistants, which assist to control the device with speech recognition. A new set of strategies is being evolved constantly to improve the performance of voice computerized seek. With using voice assistants, we are able to automate the project without difficulty, simply give the center to the machine within the speech shape and all the duties might be accomplished by means of it from changing your speech into textual content shape to putting off keywords from that text and execute the question to provide outcomes to the person. This has been one of the most beneficial improvements in era. Before AI we have been the ones who have been upgrading technology to do an assignment however now the gadget is itself able to counter new responsibilities and clear up it without need to involve the people to conform it.

A computer primarily based approach for performing a command via a voice consumer interface on a subset of objects. The subset is selected from a fixed of items, each having an object type at least one taggable field is associated with the object type and has a corresponding value. The set of objects is saved in the laptop memory. An utterance is acquired from the person and consists of a command, an object type choice, a tag-gable field selection, and a price for the taggable discipline. Responsive to the utterance, at least one item is retrieved from the set of gadgets, the item of the sort selected through the user and having a price within the taggable area selection that matches the taggable field fee obtained from the user the command is done on the item. The object includes textual content that's converted to voice output. They envisioned that someday computers will recognize natural language and count on what we need, whilst and where we need it, and proactively whole responsibilities on our behalf. However, speech recognition and machine getting to know have persevered to be refined, and based records served through packages and content providers have emerged. We agree with that as computer systems turn out to be smaller and greater ubiquitous [e.g., wearables and Internet of Things (IoT). The recognizer is designed to change a verbal articulation from an individual into an alternate method of data (e.g., text). A handheld individual colleague including a voice-recognizer and a characteristic dialect processor is disclosed. This snippet of data can be a plan for the day, data in the individual's logbook or data from the individual's address book, Such as a telephone number. The Most well-known utilization of iPhone is "SIRI" which causes the end client to impart end client versatile with voice and it additionally reacts to the voice charges of the client. It is named as Personal Assistant with Voice Recognition Intelligence, which takes the client contribution to type of voice or content and process it and returns the yield in different structures like activity to be performed or the item is directed to the end client. Furthermore, this proposed framework can change the method for communications between end client and the cell phones. Open Data is currently gathering consideration for imaginative administration creation, predominantly in the zone of government, bio science, and shrewd venture. Be that as it may, to advance its application more for purchaser administrations, a web crawler for Open Data to realize what sort of information is there would be of assistance. Virtual Personal Assistant (VPA) is the up-and-coming age of bearer administrations for

portable clients. VPA is accepted to be the smart advancement of administrations to take care of the regularly expanding demand by the portable experts for portability and network. The Virtual Personal Assistant (VPA) will empower the client to productively handle expanding interest of phone calls, messages, gatherings and different exercises. In any case, a great many people don't utilize them consistently. In particular, critical concerns rose around security, adaptation, information permanency and straight forwardness. Drawing on these discoveries we talk about key difficulties, including outlining for interrupt ability; reexamination of the human similitude; issues of trust and information proprietorship. As virtual assistants move toward becoming more intelligent and the IVA biological community of administrations and gadgets extends, there's a developing need to comprehend the security and protection dangers from this rising innovation. Better demonstrative testing can uncover such vulnerabilities and prompt more reliable frameworks. It enables the objective clients to connect with PCs and web-based administrations with a wide cluster of usefulness considering different web administrations and social media. There are four standard parts of the system: the voice recognition module, the natural language processing module, conversational agent and the content extraction module. The current screen per client writing computer programs is not fitting for getting to Internet in perspective of the base help they give for web content and the nonattendance of voice affirmation. The Virtual Right- hand programming open in the market are not especially given everything and unfit to utilize it similarly. Some may confront issue now too. This paper presents a usability of four Virtual assistant voice-based and contextual text (Google assistant, Coratan, Siri, Alexa). Cortana can likewise read your messages, track your area, watch your perusing history, check your contact list, watch out for your datebook, and set up this information together to propose valuable data, on the off chance that you enable it. You can likewise type your inquiries or solicitations, if you want to not stand up uproarious. It is only desktop based virtual assistant. Siri: Siri has been an integral part of iOS since the dispatch of iOS 5 of every 2011. It began with the nuts and bolts, for example, climate and informing, yet has extended significantly from that point forward to help all the more outsider mix with MacOS. While Siri's jokes are unbelievable, the virtual aide is getting more able consistently. Presently, you can request that it call individuals, send messages, plan gatherings, dispatch applications and recreations, and play music, answer questions, set updates, and give climate conjectures. Google Assistant: Google Assistant (which has consolidated capacities from the more seasoned Google now, as now is being eliminated) is unique in relation to Cortana and Siri. Survey on Virtual Assistant: Google Assistant, Siri, Cortana, Alexa 193 The significantly conversational VA is capable at interpreting essential vernaculars and understanding the importance behind unobtrusively complex request like, "What should we have for dinner?" It can in like manner see up to six unmistakable voices for couples and families, each voice settling to different logbook events and slants, great position amazing to Assistant and impeccable in a condition where everyone uses the voice helper on a singular gadget. Alexa: While sharing different features similarly as various VAs, Alexa is in its own one of a kind class. Amazon's voice partner isn't centered on portable or PC purposes, but instead for the independent Amazon Echo speaker and a set number of Amazon Fire gadgets, with a more prominent focus on entire house administration and administrations as opposed to PC situated errands. Each business visionary, side trickster and multitasking proficient out there would love to have a virtual assistant right hand to go up against a portion of the dull everyday errands that accompany existing in the advanced time. Similarly, as with any developing innovation, in any case, it can be hard to isolate the build up from the certainties. There are four noteworthy players seeking consideration: Amazon (Alexa),

Apple (Siri), (Google Assistant) and Microsoft (Cortana). I invested hours testing each of the four assistants by making inquiries and giving charges that numerous business clients would utilize. Amid the testing procedure, I noticed the accomplishment of the AI's reaction to me, and in addition different components a planned user may think about, for example, simplicity of setup, general capacity to perceive my voice and relevant comprehension. About each cell phone and PC available today has a brilliant right hand caught inside, like an accommodating phantom—however how might they stack up against each other? While it may seem like Siri, Cortana, and the mysterious Google Assistant are in general just assortments of the same virtual partners, they each have their own specific unconventionality's, imperfections, and characteristics. So, which one's best for clients? All things considered, that isn't a basic request to answer, as they're like the point that it's hard to take a gander at them without plunging significant into their capacities.

2.1 INFERENCES FROM LITREATURE SURVEY

Artificial Intelligence (AI) is any task performed by program machine, which otherwise human needs to apply intelligence to accomplish it. It is the science and engineering of making machines to demonstrate intelligence especially visual perception, speech recognition, decision-making, and translation between languages like human beings. AI is the simulation of human intelligence processes by machines, especially computer systems. Being a new technology, there is a huge shortage of working manpower having data analytics and data science skills; those in turn can be deputed to get maximum output from artificial intelligence. As the advancement of AI rising, businesses lack as killed professional who can match the requirement and work with this technology. Business owners need to train their professionals to be able to leverage the benefits of this technology. Artificial neural networks allow modeling of nonlinear processes and become a useful tool for solving many problems such as classification, clustering, dimension reduction, regression, structured prediction, machine translation, anomaly detection, pattern recognition, decision-making, computer vision, visualization, and others. This wide range of abilities makes it possible to use artificial neural networks in many areas. Recent developments in AI techniques complimented by the availability of high computational capacity at increasingly accessible costs, wide availability of labeled data, and improvement in learning techniques result in exploring the wide application domain for AI. AI improves lives of human beings by assisting in driving, taking personal care of aged
/handicap people, executing arduous and dangerous tasks, assisting in making informed decisions, rationally managing huge amounts of data that would otherwise be difficult to interpret, assisting in translating, and communicating multilingual while not knowing the language of our interlocutors and many more. Artificial intelligence is already everywhere and is widely used in ways that are obvious. The long-term economic effects of AI are uncertain. A survey of economists showed disagreement about whether the increasing use of robots and AI will cause a substantial increase in long-term unemployment, but they generally agree that it could be a net benefit, if productivity gains are redistributed. A 2017 study by PricewaterhouseCoopers sees the People's Republic of China gaining economically the most out of AI with 26,1% of GDP until 2030. A February 2020 European Union white paper on artificial intelligence advocated for artificial intelligence for economic benefits, including "improving healthcare (e.g. making

diagnosis more precise, enabling better prevention of diseases), increasing the efficiency of farming, contributing to climate change mitigation and adaptation, improving the efficiency of production systems through predictive maintenance", while acknowledging potential risks.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

Concern over risk from artificial intelligence has led to some high-profile donations and investments. A group of prominent tech titans including Peter Thiel, Amazon Web Services and Musk have committed $1 billion to OpenAI, a nonprofit company aimed at championing responsible AI development. The opinion of experts within the field of artificial intelligence is mixed, with sizable fractions both concerned and unconcerned by risk from eventual super humanly capable AI. Other technology industry leaders believe that artificial intelligence is helpful in its current form and will continue to assist humans. Oracle CEO Mark Hurd has stated that “AI will actually create more jobs, not less jobs" as humans will be needed to manage AI systems. Facebook CEO Mark Zuckerberg believes AI will "unlock a huge amount of positive things,” such as curing disease and increasing the safety of autonomous cars.

There are three identified challenges that vendors address in the voice-recognition domain: first improving speech recognition and command processing; second offering support for different languages, different accents, and bilingual users; and third understanding conversational contexts and establishing rapport.5 Vendors have recognized these challenges, and some proposals and efforts are promising. Deep learning algorithms, for instance, have enabled tremendous advances in speech recognition. These three challenges are well recognized and the community is working towards mitigating them.

2.2.1 Communication and conversation

Existing virtual assistants operate through predefined conditional—“if x, then y”— rules. For instance, Amazon’s Alexa uses a list of predefined skills that users can download and run on their devices.6 Existing systems usually cannot understand questions outside their knowledge base. Traditionally when a user poses a question that does not exist in the knowledge base of the agent, the agent answers with either a variant of “I don’t understand” or else pastes the user’s question directly into a web search. This limits the usability and reliability of these systems. However, there are promising algorithms and challenges that are in progress to enable dynamic, on the fly answer reconstruction. For example, CoQA (Reddy and Chen 1808) proposes a dataset to enable algorithms dynamically learning questions/answers, rather than extracting questions/answers from a static dataset. Sounding Board models, the user utterance by using multidimensional representation and content-oriented conversation segments. It creates a randomness in conversation and adjusts the answer to the user mental model. Another example is the conversational model of ERICA the robot, which analyzes the focused words of users’ utterances and constructs the response based on the focused words. As a result, it provides answers with improved accuracy than traditional methods. Research works are moving fast toward supporting more creative answering and information extraction. There might be two other approaches toward mitigating this challenge. (1) The agent can establish a “dialogue” with the user to collect more information, responding to unexpected questions by saying, for example:

“I don’t understand your question, can you provide more details? and “I will try to learn this topic from the web, thought it will take some times. Please ask me this question again later”. (2) The agent can search knowledge bases outside their own, such as analyzing information from the web. Note that this is different than simply reasserting the user’s question as a web search query. A stunning theorem was developed (Gödel’s incomplete-ness theorem), which provides a profound link between the concreteness the cybernetic systems described by Wiener, and the philosophical question of what machines can be proven to accomplish. In summary, Gödel’s Incompleteness Theorem provides that given a concrete mathematical system (with some easy to satisfy properties), there exist true statements that cannot be proven in finite time. In other words, while it may be true that artificial intelligence is capable of developing novel, creative material, it might not be possible to formally prove that they have such a capability. In interpreting Gödel’s incompleteness theorem, there are two schools of thought about Artificial Intelligence. One group including Lucas and Hofstadter (1980) describes that since machines are limited to a predefined formalization, decision making will be limited to the grammar of that formalization and machines cannot step outside of their formalization limits. On the other hand, there are scientists such as Norvig and Russell (2009) who do not agree with this argument and state a computer can invent a new formalization, and therefore it can implement creativity. Thus, the capability of creative communication by the AI and constructing answer without an accessing a knowledge base remains an unanswered question.

2.2.2 Context sensing and personalization

Existing systems collect limited contextual and sensor data, often neglecting most available sensors on the device. Many systems, such as Google Assistant or Siri, run on the smart-phone. Except calendar, location and email, they do not use other personal data or smartphone sensor data. At the time of writing this paper, there are no known virtual assistants that benefit from contextual data. Some social robots, such as Kuri,7 may designed to collect contextual data. Nevertheless, they are not produced in large scale due to their unsuccessful marketing campaign. Collecting these data might be associated with privacy risks as well, but contextual data collection will assist personalization of the services these systems provide. Social robots and also Internet of Things devices have two distinct differences from the traditional context-sensing systems found in smartphones and wearables. First, unlike smartphones and wearables, they are usually shared among household members. Therefore, they should identify a user from a small group of users (e.g. family or guests). This sort of identification allows more effective personalization. Second, unlike smartphones and wearables, social robots and smart speakers are not constantly attached to a user’s body. Accordingly, these devices can observe user activities and collect data from a third-person perspective. This enables more accurate activity recognition and mitigate data quality challenges existed in wearable devices Rawassizadeh etal. (2019). For example, a robot can be used to track weight-lifting and other physical activities, which include using weights. Currently, wearable and mobile devices per-form activity tracking, but this comes with several significant limitations. One is that these devices are incapable of collecting certain data about weight lifted by the user or details of their activities. Because they cannot accurately monitor users from a third person perspective. Google Fit Workout in Wear OS, for example, prompts users to enter workout and weight-data manually. A robot, on the other hand, could unobtrusively, but closely, follow a user to collect the same data and it obviates the need for manual data entry. For example, by using its camera and an image-recognition algorithm it can recognize the type of activity with high precision. Some may argue that because certain

AI technologies, like smart speakers, are non-mobile, they are incapable of collecting enough contextual information to be useful. How-ever, there are promising efforts to the contrary. Laput and Zhang (2017) proposed a static but powerful sensing device that can collect data from its context—for instance, data concerning activities a user performs in the kitchen. Even without mobility, these systems can collect useful data in their target environments. Furthermore, Cohen etal. (2006) acknowledge the benefit of personalizing of these systems and predict future assistant are aware about users' intentions, which require contextual information for such an awareness. As with any data collection, there are privacy concerns. To respect users' privacy, we recommend the device remains disconnected from the Internet and perform its data analysis locally. The lack of Internet connection may limit an agent's applications, but there will also be plenty of advantages. For instance, a device could undertake continuous health monitoring with no need for cloud storage.

2.2.3 User interface and embolization

There are longstanding studies for designing interfaces for intelligent agents (including virtual assistant and social robots). For instance, developers use animation toward embodification, or they add ambient displays toward conveying the emotions. Existing social robots present emotions either via shaking their heads or changing their facial expressions. Given the nascence of the technology, there is still much room for improvement. Another example is the advances in appearances of the robots' bodies, such as advantages of textile for embodification and touch based gestures, to improve usability of the robot. Recent version of smart speakers such as Amazon and Google use a textile instead of plastic on their smart speaker interface. There are studies that identify kids' expectations from robots, such as collaborative game play and peer- pressure of robots on decision making (Kline 2018). While these studies are mainly focused on kids, another interesting direction might be identifying other stakeholders such as patients with specific need and customizing robot interfaces based on their need, such as using robots to assists patients with cognitive impairment.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

Feasibility study can help you determine whether or not you should proceed with your project. It is essential to evaluate cost and benefit. It is essential to evaluate cost and benefit of the proposed system. Five types of feasibility study are taken into consideration.

1. Technical feasibility:
It includes finding out technologies for the project, both hardware and software. For virtual assistant, user must have microphone to convey their message and a speaker to listen when system speaks. These are very cheap now a days and everyone generally possess them. Besides, system needs internet connection. While using Assistant, make sure you have a steady internet connection. It is also not an issue in this era where almost every home or office has Wi-Fi.

2. Operational feasibility:
It is the ease and simplicity of operation of proposed system. System does not require any special skill set for users to operate it. In fact, it is designed to be used by almost everyone. Kids who still don't know to write can readout problems for system and get answers.

3. Economical feasibility:
Here, we find the total cost and benefit of the proposed system over current system. For this project, the main cost is documentation cost. User also would have to pay for microphone and speakers. Again, they are cheap and available. As far as maintenance is concerned, Assistant won't cost too much.

4. Organizational feasibility:
This shows the management and organizational structure of the project. This project is not built by a team. The management tasks are all to be carried out by a single person. That won't create any management issues and will increase the feasibility of the project.

5. Cultural feasibility:
It deals with compatibility of the project with cultural environment. Virtual assistant is built in accordance with the general culture. The project is named Assistant so as to represent Indian culture without undermining local beliefs. This project is technically feasible with no external hardware requirements. Also, it is simple in operation and does not cost training or repairs. Overall feasibility study of the project reveals that the goals of the proposed system are achievable. Decision is taken to proceed with the project.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

The IDE used in this project is PyCharm. All the python files were created in PyCharm and all the necessary packages were easily installable in this IDE. For this project following modules and libraries were used i.e. pyttsx3, Speech-Recognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, pyQt etc. I have created a live GUI for interacting with the Assistant as it gives a design and interesting look while having the conversation.

3.2.1. PYCHARM: It is an IDE i.e., Integrated Development Environment which has many features like it supports scientific tools (like matplotlib, numpy, scipy etc) web frameworks (example Django, web2py and Flask) refactoring in Python, integrated python debugger, code completion, code and project navigation etc. It also provides Data Science when used with Anaconda.

3.2.2. PYQT5 FOR LIVE GUI: PyQt5 is the most important python binding. It contains set of GUI widgets. PyQt5 has some important python modules like QTWidgets, QtCore, QtGui, and QtDesigner etc.

3.2.3. PYTHON LIBRARIES: In Assistant following python libraries were used:

pyttsx3: It is a python library which converts text to speech. **SpeechRecognition:** It is

a python module which converts speech to text.

pywhatkit: It is python library to send WhatsApp message at a particular time with some additional features.

Datetime: This library provides us the actual date and time. Wikipedia: It is a python module for searching anything on Wikipedia.

Smtplib: Simple mail transfer protocol that allows us to send mails and to route mails between mail servers.

pyPDF2: It is a python module which can read, split, merge any PDF.

Pyjokes: It is a python libraries which contains lots of interesting jokes in it.

Webbrowser: It provides interface for displaying web-based documents to users.

Pyautogui: It is a python library for graphical user interface.

os: It represents Operating System related functionality.

sys: It allows operating on the interpreter as it provides access to the variables and functions that usually interact strongly with the interpreter.

3.3 SYSTEM USE CASE

Use Case: Customer Support Chatbot
Actor: Customer

Basic Flow:

1. The customer visits the company's website and clicks on the chatbot icon.
2. The chatbot greets the customer and asks how it can assist them.
3. The customer provides a brief description of their issue or question.
4. The chatbot analyzes the customer's message and offers relevant responses or actions.
5. If the chatbot is unable to resolve the issue or if the customer requests additional assistance, the chatbot escalates the conversation to a human support agent.
6. The chatbot logs the conversation and any actions taken, then ends the conversation with the customer.

Alternative Flows:

- If the customer requests to speak with a human support agent at any point, the chatbot immediately escalates the conversation to an available agent.
- If the chatbot is unable to understand the customer's message, it asks the customer to clarify and repeats the step of analyzing the message.

Post-Conditions:

- The customer's issue has been resolved, or the customer has been escalated to a human support agent.
- The conversation has been logged and can be used for future reference or analysis.

3.3.1 ER DIAGRAM

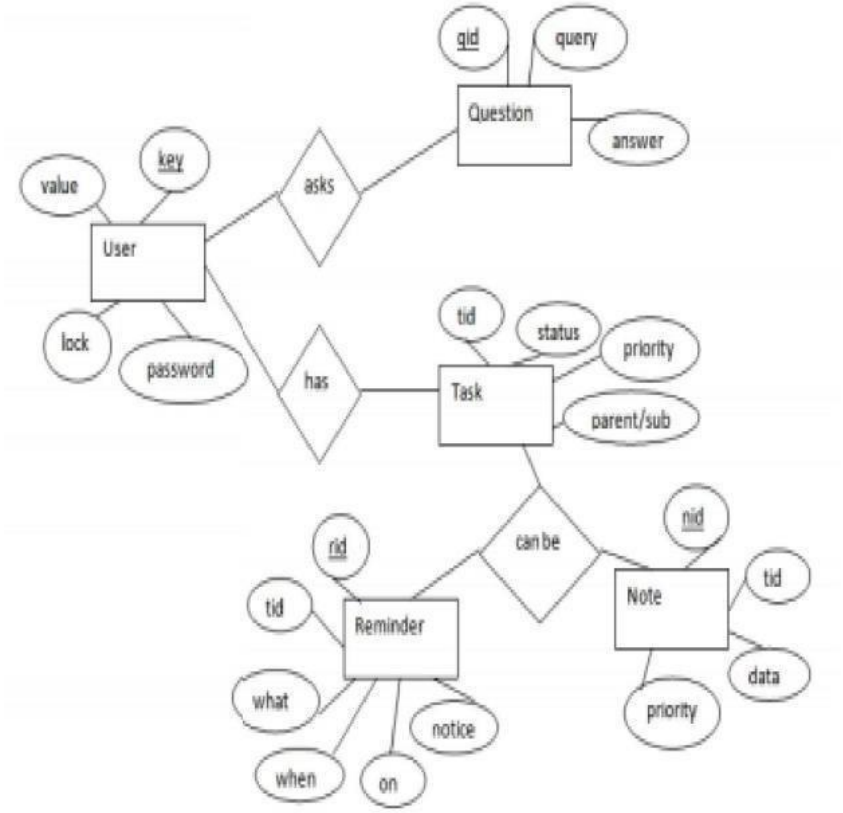


Fig. 3.1 ER Diagram

The above diagram shows entities and their relationship for a virtual assistant system.

We have a user of a system who can have their keys and values. It can be used to store any information about the user. Say, for key “name” value can be “Jim”. For some key’s user might like to keep secure. There he can enable lock and set a password (voice clip). Single user can ask multiple questions. Each question will be given ID to get recognized along with the query and its corresponding answer. User can also be having n number of tasks. These should have their own unique id and status i.e.
their current state. A task should also have a priority value and its category whether it is a parent task or child task of an older task.

3.3.2 ACTIVITY DIAGRAM

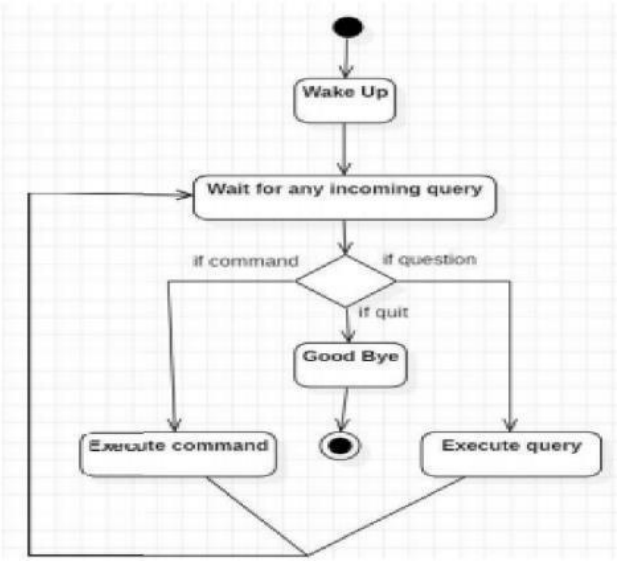


Fig. 3.2 Activity Diagram for the Use Case

Initially, the system is in idle mode. As it receives any wake up cal it begins execution. The received command is identified whether it is a questionnaire or a task to be performed. Specific action is taken accordingly. After the Question is being answered or the task is being performed, the system waits for another command. This loop continues unless it receives quit command. At that moment, it goes back to sleep.

3.3.3 CLASS DIAGRAM

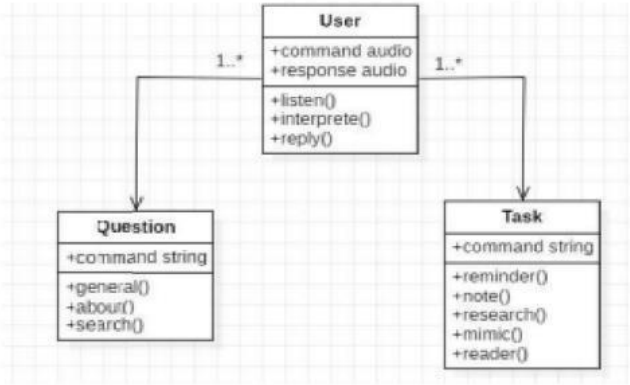


Fig. 3.3 Class Diagram for the Use Case

The class user has 2 attributes command that it sends in audio and the response it receives which is also audio. It performs function to listen the user command. Interpret

it and then reply or sends back response accordingly. Question class has the command in string form as it is interpreted by interpret class. It sends it to general or about or search function based on its identification. The task class also has interpreted command in string format. It has various functions like reminder, note, mimic, research and reader.

3.3.4 USE CASE DIAGRAM

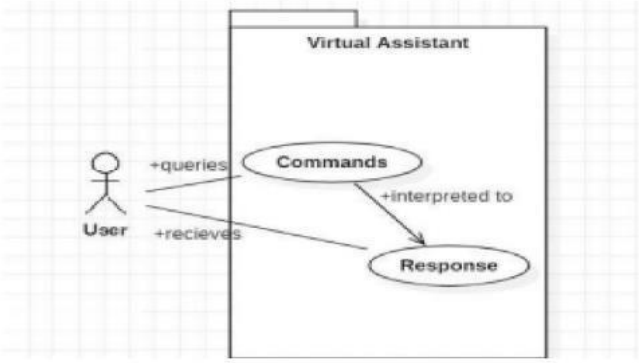


Fig. 3.4 Use Case Diagram

In this project there is only one user. The user queries command to the system. System then interprets it and fetches answer.The response is sent back to the user.

3.3.5 SEQUENCE DIAGRAM

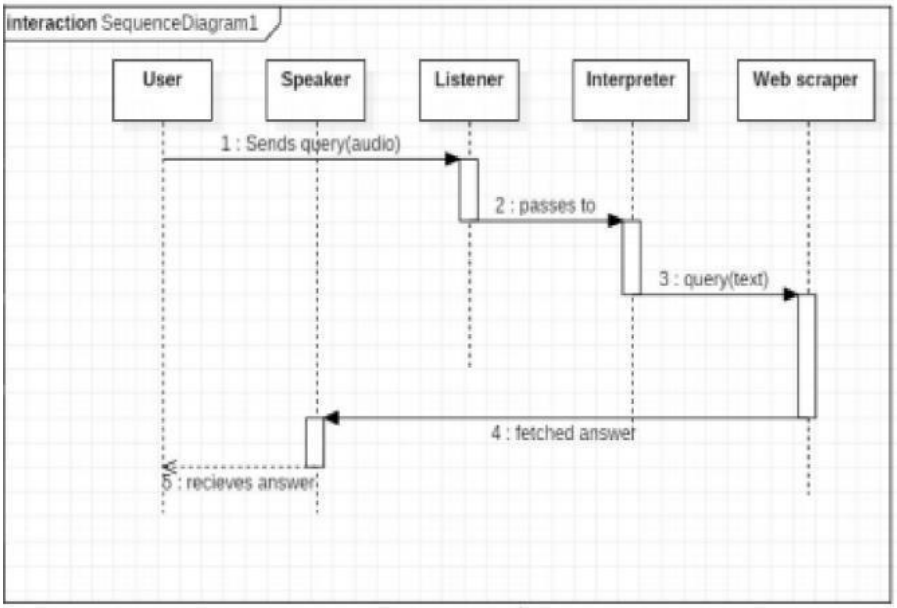


Fig. 3.5 Sequence Diagram for the use case

The above sequence diagram shows how an answer asked by the user is being fetched from internet. The audio query is interpreted and sent to Web scraper. The web scraper searches and finds the answer. It is then sent back to speaker, where it speaks the answer to user.

3.3.6 COMPONENT DIAGRAM

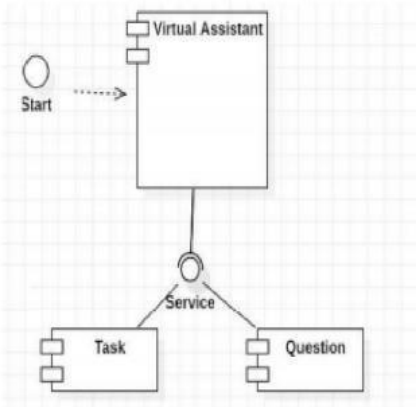


Fig. 3.6 Component Diagram for the use case

The main component here is the Virtual Assistant. It provides two specific service, executing Task or Answering your question.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

We are familiar with many existing voice assistants like Alexa, Siri, Google Assistant, Cortana which uses concept of language processing, and voice recognition. They listen the command given by the user as per their requirements and performs that specific function in a very efficient and effective manner. As these voice assistants are using Artificial Intelligence hence the result that they are providing are highly accurate and efficient. These assistants can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. These assistants are no less than a human assistant, but we can say that they are more effective and efficient to perform any task. The algorithm used to make these assistant focuses on the time complexities and reduces time. But for using these assistants one should have an account (like Google account for Google assistant, Microsoft account for Cortana) and can use it with internet connection only because these assistants are going to work with internet connectivity. They are integrated with many devices like, phones, laptops, and speakers etc.
It was an interesting task to make my own assistant. It became easier to send emails without typing any word, searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. Assistant is different from other traditional voice assistants in terms that it is specific to desktop and user does not need to make account to use this, it does not require any internet connection while getting the instructions to perform any specific task. The IDE used in this project is PyCharm. All the python files were created in PyCharm and all the necessary packages were easily installable in this IDE. For this project following modules and libraries were used i.e. pyttsx3, SpeechRecognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, pyQt etc. I have created a live GUI for interacting with the Assistant as it gives a design and interesting look while having the conversation. With the advancement Assistant can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. Functionalities of this project include, It can send emails, It can read PDF, It can send text on WhatsApp, It can open command prompt, your favorite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast, It can give desktop reminders of your choice. It can have some basic conversation.

The system is designed using the concept of Artificial Intelligence and with the help of necessary packages of Python. Python provides many libraries and packages to perform the tasks, for example pyPDF2 can be used to read PDF. The data in this project is nothing but user input, whatever the user says, the assistant performs the task accordingly. The user input is nothing specific but the list of tasks which a user wants to get performed in human language i.e., English.

Fig 4.1: A Representation of an AI Assistant receiving commands from User

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

Artificial Intelligence when used with machines, it shows us the capability of thinking like humans. In this, a computer system is designed in such a way that typically requires interaction from human. As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the Alexa, Siri, etc. In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favourite IDE with the help of a single voice command. In the current scenario, advancement in technologies is such that they can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. As the voice assistant is using Artificial Intelligence hence the result that it is providing are highly accurate and efficient. The assistant can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. The assistant is no less than a human assistant but we can say that this is more effective and efficient to perform any task. The libraries and packages used to make this assistant focuses on the time complexities and reduces time. The functionalities include, it can send emails, It can read PDF, It can send text on WhatsApp, It can open command prompt, your favourite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast, It can give desktop reminders of your choice. It can have some basic conversation. Tools and technologies used are PyCharm IDE for making this project, and I created all py files in PyCharm. Along with this I used following modules and libraries in my project. pyttsx3, SpeechRecognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, pyQt etc. I have created a live GUI for interacting with the Assistant as it gives a design and interesting look while having the conversation.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

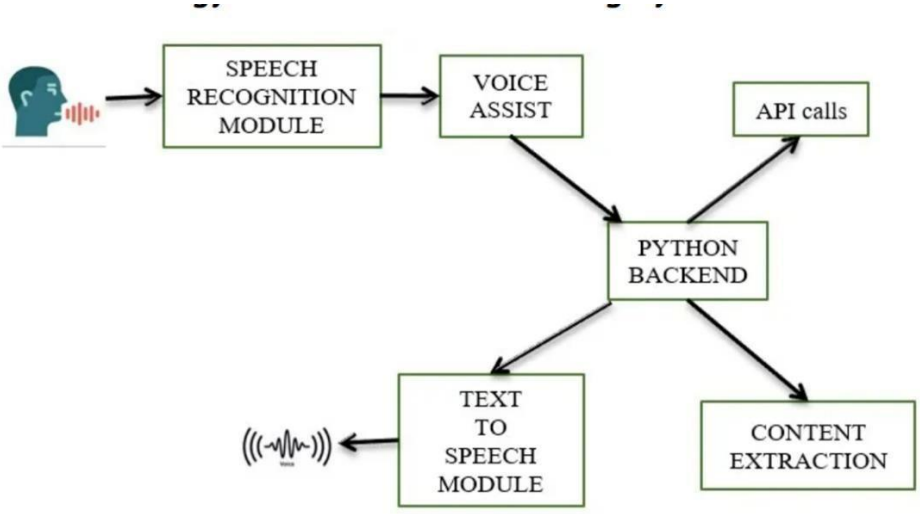


Fig 4.2: System Architecture for AI Desktop Assistant

The system architecture diagram of the proposed system has been shown in the above figures. The system is designed using the concept of Artificial Intelligence and with the help of necessary packages of Python. Python provides many libraries and packages to perform the tasks, for example pyPDF2 can be used to read PDF. The data in this project is nothing but user input, whatever the user says, the assistant performs the task accordingly. The user input is nothing specific but the list of tasks which a user wants to get performed in human language i.e., English.

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

Assistant, a desktop assistant is a voice assistant that can perform many daily tasks of desktop like playing music, opening your favorite IDE with the help of a single voice command. Assistant is different from other traditional voice assistants in terms that it is specific to desktop and user does not need to make account to use this, it does not require any internet connection while getting the instructions to perform any specific task.
As the first step, install all the necessary packages and libraries. The command used to install the libraries is “pip install” and then import it. The necessary packages included are as follows:

4.3.1. LIBRARIES AND PACKAGES

4.3.1.1. pyttsx3: It is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech. The pyttsx3 module supports two voices first is female and the second is male which is provided by “sapi5” for windows. It supports three TTS engines :

- *sapi5* – SAPI5 on Windows

- *nsss* – NSSpeechSynthesizer on Mac OS X
- *espeak* – eSpeak on every other platform

4.3.1.2. Speech-Recognition: It is a python module which converts speech to text. Speech recognition is a machine's ability to listen to spoken words and identify them. You can then use speech recognition in Python to convert the spoken words into text, make a query or give a reply. You can even program some devices to respond to these spoken words. You can do speech recognition in python with the help of computer programs that take in input from the microphone, process it, and convert it into a suitable form. Speech recognition seems highly futuristic, but it is present all around you. Automated phone calls allow you to speak out your query or the query you wish to be assisted on; your virtual assistants like Siri or Alexa also use speech recognition to talk to you seamlessly.

4.3.1.3. pywhatkit: It is python library to send WhatsApp message at a particular time with some additional features. Python offers numerous inbuilt libraries to ease our work. Among them pywhatkit is a Python library for sending WhatsApp messages at a certain time, it has several other features too. Following are some features of pywhatkit module:

- Send WhatsApp messages.
- Play a YouTube video.
- Perform a Google Search.
- Get information on a particular topic.

The pywhatkit module can also be used for converting text into handwritten text images.

4.3.1.4. Datetime: This library provides us the actual date and time. Python Datetime module comes built into Python, so there is no need to install it externally.
Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

4.3.1.5. Wikipedia: It is a python module for searching anything on Wikipedia. Python provides the Wikipedia module (or API) to scrap the data from the Wikipedia pages. This module allows us to get and parse the information from Wikipedia. In simple words, we can say that it is worked as a little scrapper and can scrap only a limited amount of data. Before we start working with it, we need to install this module on our local machine.

4.3.1.6. Smtplib: Simple Mail Transfer Protocol (SMTP) is used as a protocol to handle the email transfer using Python. It is used to route emails between email servers. It is an application layer protocol which allows to users to send mail to another. The receiver retrieves email using the protocols POP(Post Office Protocol) and IMAP(Internet Message Access Protocol). When the server listens for

the TCP connection from a client, it initiates a connection on port 587. Python provides a smtplib module, which defines an the SMTP client session object used to send emails to an internet machine. For this purpose, we have to import the smtplib module using the import statement.

4.3.1.7. pyPDF2: PyPDF2 is a free and open source pure-python PDF library capable of splitting, merging, cropping, and transforming the pages of PDF files. It can also add custom data, viewing options, and passwords to PDF files. PyPDF2 can retrieve text and metadata from PDFs as well. The low-level API (based on Pygments) allows writing programs that generate or efficiently manipulate documents. The high-level API (based on ReportLab) enables the creation of complex documents such as forms, books, or magazines with just a few lines of code. PyPDF2 supports:

- Converting PDF files into images (png or jpeg) or text files;
- Converting PDF to text, image to text
- Creating new PDF documents from scratch;
- Editing existing PDFs by adding, removing, replacing, or modifying pages;
- Modifying existing PDFs by rotating pages, adding watermarks, changing fonts, etc.;
- Signing documents with digital signatures (certificates must be present); PyPDF2 has

been designed with performance in mind. It uses native C code to handle the most time-consuming tasks (such as parsing) but never sacrifices the simplicity of its interface. The library is also thread-safe, and its memory footprint is not much larger than the one required by Python (around 1MB).

4.3.1.8. Pyjokes: Python supports creation of random jokes using one of its libraries. Let us explore it a little more, Pyjokes is a python library that is used to create one-line jokes for programmers. Informally, it can also be referred as a fun python library which is pretty simple to use.

4.3.1.9. Webbrowser: In Python, webbrowser module is a convenient web browser controller. It provides a high-level interface that allows displaying Web-based documents to users. webbrowser can also be used as a CLI tool. It accepts a URL as the argument with the following optional parameters: -n opens the URL in a new browser window, if possible, and -t opens the URL in a new browser tab.

4.3.1.10. Pyautogui: Python pyautogui library is an automation library that allows mouse and keyboard control. Or we can say that it facilitates us to automate the movement of the mouse and keyboard to establish the interaction with the other application using the Python script. It provides many features, and a few are given below.

- We can move the mouse and click in the other applications' window.
- We can send the keystrokes to the other applications. For example - filling out the form, typing the search query to browser, etc.
- We can also take snapshots and give an image.
- It allows us to locate a window of the application, and move, maximize,

minimize, resizes, or close it.
- Display alert and message boxes.

4.3.1.11. os: Python OS module provides the facility to establish the interaction between the user and the operating system. It offers many useful OS functions that are used to perform OS-based tasks and get related information about operating system. The OS comes under Python's standard utility modules. This module offers a portable way of using operating system dependent functionality.

4.3.1.12. sys: The python sys module provides functions and variables which are used to manipulate different parts of the Python Runtime Environment. It lets us access system-specific parameters and functions. The sys module comes packaged with Python, which means you do not need to download and install it separately using the PIP package manager. In order to start using the sys module and its various functions, you need to import it.

4.3.1.13. subprocess: The subprocess module present in Python(both 2.x and 3.x) is used to run new applications or programs through Python code by creating new processes. It also helps to obtain the input/output/error pipes as well as the exit codes of various commands.

4.3.1.14. pygame: Game programming is very rewarding nowadays and it can also be used in advertising and as a teaching tool too. Game development includes mathematics, logic, physics, AI, and much more and it can be amazingly fun. In python, game programming is done in pygame and it is one of the best modules for doing so.

4.3.1.14. pygame: Python provides a library named keyboard which is used to get full control of the keyboard. It's a small Python library which can hook global events, register hotkeys, simulate key presses and much more.
- It helps to enter keys, record the keyboard activities and block the keys until a specified key is entered and simulate the keys.
- It captures all keys, even onscreen keyboard events are also captured.
- Keyboard module supports complex hotkeys.
- Using this module we can listen and send keyboard events.
- It works on both windows and linux operating system.

4.3.2. FUNCTIONS

4.3.2.1. takeCommand(): The function is used to take the command as input through microphone of user and returns the output as string.

4.3.2.2. wishMe(): This function greets the user according to the time like Good Morning, Good Afternoon and Good Evening.

4.3.2.3. taskExecution(): This is the function which contains all the necessary task execution definition like sendEmail(), pdf_reader(), news() and many conditions in if condition like “open google”, “open notepad”, “search on Wikipedia” ,”play music” and “open command prompt” etc.

The system testing is done on fully integrated system to check whether the requirements are matching or not. The system testing for desktop assistant focuses on the following four parameters:

4.3.3. FUNCTIONALITY

In this we check the functionality of the system whether the system performs the task which it was intended to do. To check the functionality each function was checked and run, if it is able to execute the required task correctly then the system passes in that particular functionality test. For example, to check whether Assistant can search on Google or not, user said “Open Google”, then Assistant asked, “What should I search on Google?” then user said, “What is Python”, Assistant open Google and searched for the required input.

4.3.4. USABILITY

Usability of a system is checked by measuring the easiness of the software and how user friendly it is for the user to use, how it responses to each query that is being asked by the user. It makes it easier to complete any task as it automatically do it by using the essential module or libraries of Python, in a conversational interaction way. Hence any user when instruct any task to it, they feel like giving task to a human assistant because of the conversational interaction for giving input and getting the desired output in the form of task done. The desktop assistant is reactive which means it know human language very well and understand the context that is provided by the user and gives response in the same way, i.e. human understandable language, English. So user finds its reaction in an informed and smart way. The main application of it can be its multitasking ability. It can ask for continuous instruction one after other until the user “QUIT” it. It asks for the instruction and listen the response that is given by user without needing any trigger phase and then only executes the task.

4.3.5. SECURITY

The security testing mainly focuses on vulnerabilities and risks. As Assistant is a local desktop application, hence there is no risk of data breaching through remote access. The software is dedicated to a specific system so when the user logs in, it will be activated.

4.3.6. STABILITY

Stability of a system depends upon the output of the system, if the output is bounded and specific to the bounded input then the system is said to be stable. If the system works on all the poles of functionality, then it is stable.

4.4 PROJECT MANAGEMENT PLAN

Project management plan for an AI desktop virtual assistant would include various stages and activities that are critical to ensure the success of the project. These stages can be broadly classified into planning, execution, and monitoring and control.

The planning phase involves defining the scope of the project, setting objectives, and identifying the key stakeholders. In the case of an AI desktop virtual assistant, the project team would need to determine the specific features and functionalities that the assistant would need to perform and identify the user groups and target audience for the assistant. This would involve conducting a thorough analysis of the market and the needs of potential users.

The project team would also need to define the project schedule, including key milestones and deadlines, and identify the resources required for the project. This would include determining the necessary hardware and software infrastructure, as well as the personnel needed to develop and maintain the AI assistant.

The execution phase involves the actual development and implementation of the AI desktop virtual assistant. This would involve a range of activities, such as designing the user interface and programming the underlying AI algorithms. The project team would also need to conduct extensive testing and quality assurance to ensure that the assistant is functioning correctly and providing accurate and useful responses to user queries.

In addition, the project team would need to establish a process for collecting and analyzing user feedback. This would involve developing a system for tracking user interactions with the assistant, and using this data to continually improve the performance and functionality of the assistant.

The monitoring and control phase involves ongoing monitoring of the project progress and making adjustments as necessary to ensure that the project is meeting its objectives. This would involve regular project status updates, as well as ongoing performance monitoring of the AI assistant to identify and address any issues or areas for improvement.

Overall, a project management plan for an AI desktop virtual assistant would need to be comprehensive and flexible, considering the unique challenges and complexities of developing and implementing this type of technology. It would need to be focused on delivering a high-quality product that meets the needs of users, while also ensuring that the project stays on schedule and within budget. With careful planning and execution, an AI desktop virtual assistant can provide significant benefits to users across a range of industries and revolutionize the way we interact with our devices.

The project titled “A.I. DESKTOP ASSISTANT” was designed by our team. From installing of all the packages, importing, creating all the necessary functions, designing GUI in PyQT and connecting that live GUI with the backend, was all done by us. We have done all the research before making this project, designed the requirement documents for the requirements and functionalities, wrote synopsis and all the documentation, code and made the project in such a way that it is deliverable at each stage. We have created the front end (.ui file) of the project using PyQt designer, the

front end comprises of a live GUI and is connected with the .py file which contains all the classes and packages of the .ui file. The live GUI consists of moving GIFs which makes the front end attractive and user friendly. We have written the complete code in Python language and in PyCharm IDE from where it was very easy to install the packages and libraries, We have created the functions like takeCommand(), wishMe() and taskExecution() which has the following functionalities, like takeCommand() which is used to take the command as input through microphone of user and returns the output as string, wishMe() that greets the user according to the time like Good Morning, Good Afternoon and Good Evening and taskExecution()which contains all the necessary task execution definition like sendEmail(), pdf_reader(), news() and many conditions in if condition like “open Google”, “open notepad”, “search on Wikipedia”
,”play music” and “open command prompt” etc. While making this project we realized that with the advancement Assistant can perform any task with same effectiveness or can say more effectively than us. By making this project, we realized that the concept of AI in every field is decreasing human effort and saving time. Functionalities of this project include, It can send emails, It can read PDF, It can send text on WhatsApp, It can open command prompt, your favorite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast, It can give desktop reminders of your choice. It can have some basic conversation. At last, we have updated my report and completed it by attaching all the necessary screen captures of inputs and outputs, mentioning the limitations and scope in future of this project.

4.5 Transition/ Software to Operations Plan

A Transition/Software to Operations (T/S2O) plan is a detailed roadmap that outlines the steps required to transition an AI desktop assistant model from development to deployment and operations. Having a transition or software to operations plan is crucial for the successful implementation of an AI desktop assistant. This plan outlines the steps necessary to take the AI assistant from development to full deployment and use by end-users. It typically includes the following key components:

Deployment strategy: This outlines the approach that will be used to deploy the AI desktop assistant model, including the infrastructure required, software dependencies, and any third-party services that may be needed.

Testing and validation: This is a critical component of the T/S2O plan, as it ensures that the AI desktop assistant model is functioning as expected and meets the performance and accuracy requirements.

Monitoring and maintenance: Once the AI desktop assistant model is deployed, it needs to be monitored and maintained to ensure that it continues to perform optimally. This includes regular updates, bug fixes, and enhancements.

Security and compliance: Security and compliance are also important considerations in the T/S2O plan, as the AI desktop assistant model may handle sensitive data and needs to be compliant with relevant regulations and standards.

User support and training: Finally, the T/S2O plan should include provisions for user support and training, as the AI desktop assistant model may be used by individuals

with varying levels of technical expertise.

Integration with Existing Systems: The AI desktop assistant needs to be integrated with other systems and applications already in use by the organization to ensure that it functions effectively within the organization's technology infrastructure.

Training and Support: End-users need to be trained on how to use the AI desktop assistant effectively, and ongoing support should be provided to help users address any issues that may arise during use.

Maintenance and Upgrades: The AI assistant requires maintenance to ensure that it continues to function effectively, and updates and upgrades may be necessary to improve its performance or add new features.

By having a transition plan in place, organizations can ensure that the AI desktop assistant is implemented effectively and efficiently, and that end-users are adequately supported throughout the process. Overall, a T/S2O plan is an essential component of any AI desktop assistant project, as it helps to ensure that the model is deployed and operated effectively and efficiently and provides a framework for ongoing improvement and optimization.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

The code provides a Python implementation of an AI Desktop Assistant project that uses speech recognition to take user commands, perform various tasks, and provide verbal responses. The program imports several necessary packages for speech recognition, text-to-speech conversion, music playback, opening files and web pages, and accessing information from Wikipedia. The AI assistant can perform tasks like opening word, powerpoint, excel, zoom, notepad, and chrome applications, searching Wikipedia, telling jokes, opening web pages such as YouTube, Google, and Stack Overflow, and playing music. Additionally, the assistant can write notes, speak the current date, and greet the user based on the current time. Overall, the project demonstrates the potential of AI-powered desktop assistants in automating routine tasks and enhancing user productivity.

The development setup for a Python-based virtual assistant would typically involve using an integrated development environment (IDE) such as PyCharm, VSCode or Spyder. These IDEs provide tools for coding, debugging, and testing Python programs. The virtual assistant would also require the use of various Python libraries and packages, such as speech recognition, pyttsx3, wikipedia, webbrowser, and pygame, to name a few.

The deployment setup would depend on how the virtual assistant is intended to be used. If it is a personal project meant to be run on a local machine, the deployment process would involve installing any necessary packages on the target machine and running the program on that machine. However, if the virtual assistant is intended to be used by others or integrated into a larger application, more involved deployment processes would be necessary.

For deployment of virtual assistant, the following steps can be taken:

1. Create an executable file for the virtual assistant using tools such as Pyinstaller, cx_Freeze, or Py2Exe, which can package the Python program and its dependencies into a single executable file. This makes it easier to distribute the program and ensures that users don't need to install any additional packages.

2. Use cloud platforms like AWS, Azure or Google Cloud for deployment if the virtual assistant is intended to be used by multiple users over the internet. In this case, the program would be hosted on a cloud server and accessed through a web interface or API.

3. Use containers like Docker or Kubernetes for deployment if the virtual assistant is meant to be deployed on multiple machines. This makes it easier to manage and scale the deployment.

4. Set up a continuous integration/continuous deployment (CI/CD) pipeline to automate the deployment process. This ensures that any changes made to the code are automatically tested, built, and deployed to the target environment.

Overall, developing and deploying a virtual assistant requires careful planning and consideration of the intended use case and target audience.

5.2 ALGORITHMS

A virtual assistant, also called an AI assistant or digital assistant, is an application program that understands natural language voice commands and completes tasks for the user. Brief overview of some common machine learning algorithms that are used in the development and deployment of AI application are:

Regression: Regression is a statistical method used to estimate the relationship between a dependent variable and one or more independent variables. Linear regression is a common type of regression algorithm used to predict continuous values.

Classification: Classification is a supervised learning technique used to predict the class or category of a given data point. Some popular classification algorithms include logistic regression, decision trees, and support vector machines (SVM).

Clustering: Clustering is an unsupervised learning technique used to group similar data points together based on their characteristics. Some popular clustering algorithms include K-means, hierarchical clustering, and DBSCAN.

Neural Networks: Neural networks are a type of machine learning algorithm inspired by the structure of the human brain. They are composed of multiple layers of interconnected nodes and are used for a variety of tasks such as image and speech recognition, natural language processing, and predictive modeling.

Reinforcement Learning: Reinforcement learning is a type of machine learning technique used to train an agent to make decisions in an environment. It involves the agent taking actions in the environment and receiving feedback in the form of rewards or punishments.

These are just a few examples of the many algorithms used in the development and deployment of AI applications. The choice of algorithm will depend on the specific task at hand, the size and complexity of the dataset, and other factors such as computational resources and time constraints.

5.2.1 Specific algorithms used in the code

5.2.1.1 K-Means Clustering: K-means is a popular clustering algorithm used to group data points into k clusters. In the code provided, K-means is used to cluster the RGB values of each pixel in the image into a specified number of clusters (n_colors). Here

K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

Determines the best value for K center points or centroids by an iterative process. Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

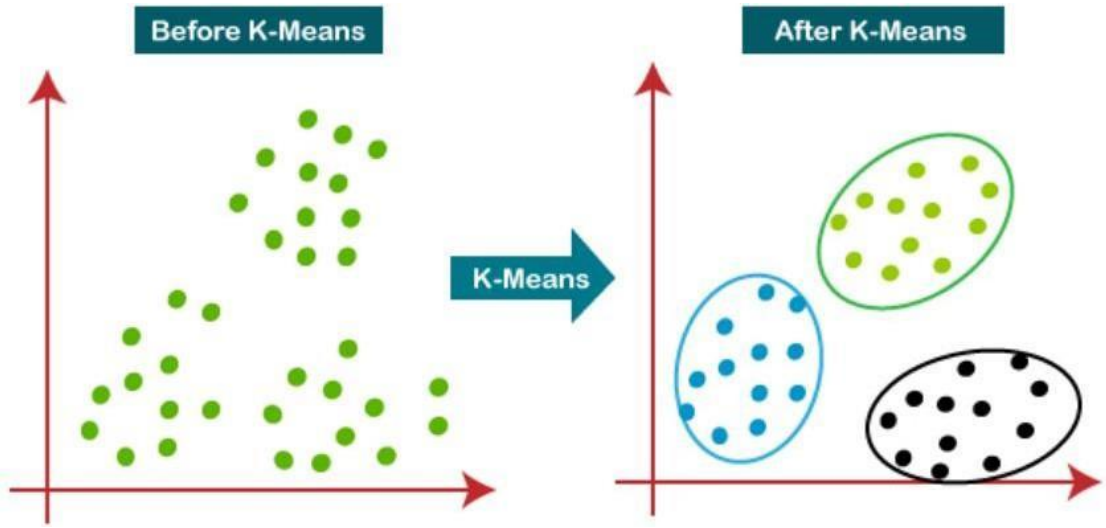


Fig. 5.1 K-Means Clustering

5.2.1.2 Principal Component Analysis (PCA): PCA is a technique used to reduce the dimensionality of a dataset by transforming it into a lower-dimensional space that still contains most of the information in the original dataset. Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the Principal Components. It is one of the popular tools that is used for exploratory data analysis and predictive modeling. It is a technique to draw strong patterns from the given dataset by reducing the variances.

PCA generally tries to find the lower-dimensional surface to project the high- dimensional data.
PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality.

Some real-world applications of PCA are image processing, movie recommendation system, optimizing the power allocation in various communication channels. It is a feature extraction technique, so it contains the important variables and drops the least important variable. In the code provided, PCA is used to further reduce the dimensionality of the clustered RGB values.

The PCA algorithm is based on some mathematical concepts such as:

- Variance and Covariance
- Eigenvalues and Eigen factors

Some common terms used in PCA algorithm:

- Dimensionality: It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
- Correlation: It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
- Orthogonal: It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
- Eigenvectors: If there is a square matrix M, and a non-zero vector v is given. Then v will be eigenvector if Av is the scalar multiple of v.
- Covariance Matrix: A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

5.2.1.3 MiniBatch K-Means: MiniBatch K-means is a variant of the K-means algorithm that is faster and more memory-efficient. Mini Batch K-means algorithm‘s main idea is to use small random batches of data of a fixed size, so they can be stored in memory. Each iteration a new random sample from the dataset is obtained and used to update the clusters and this is repeated until convergence. Each mini batch updates the clusters using a convex combination of the values of the prototypes and the data, applying a learning rate that decreases with the number of iterations. This learning rate is the inverse of the number of data assigned to a cluster during the process. As the number of iterations increases, the effect of new data is reduced, so convergence can be detected when no changes in the clusters occur in several consecutive iterations. The empirical results suggest that it can obtain a substantial saving of computational time at the expense of some loss of cluster quality, but not extensive study of the algorithm has been done to measure how the characteristics of the datasets, such as the number of clusters or its size, affect the partition quality.

Mini-batch K-means is a variation of the traditional K-means clustering algorithm that is designed to handle large datasets. In traditional K-means, the algorithm processes the entire dataset in each iteration, which can be computationally expensive for large datasets.

Mini-batch K-means addresses this issue by processing only a small subset of the data, called a mini-batch, in each iteration. The mini-batch is randomly sampled from the dataset, and the algorithm updates the cluster centroids based on the data in the mini-batch. This allows the algorithm to converge faster and use less memory than traditional K-means. In the code provided, MiniBatch K-means is used to cluster the

reduced data generated by PCA.

The mini batch K-means is faster but gives slightly different results than the normal batch K-means. Here we cluster a set of data, first with K-means and then with mini batch K-means, and plot the results. We will also plot the points that are labeled differently between the two algorithms.

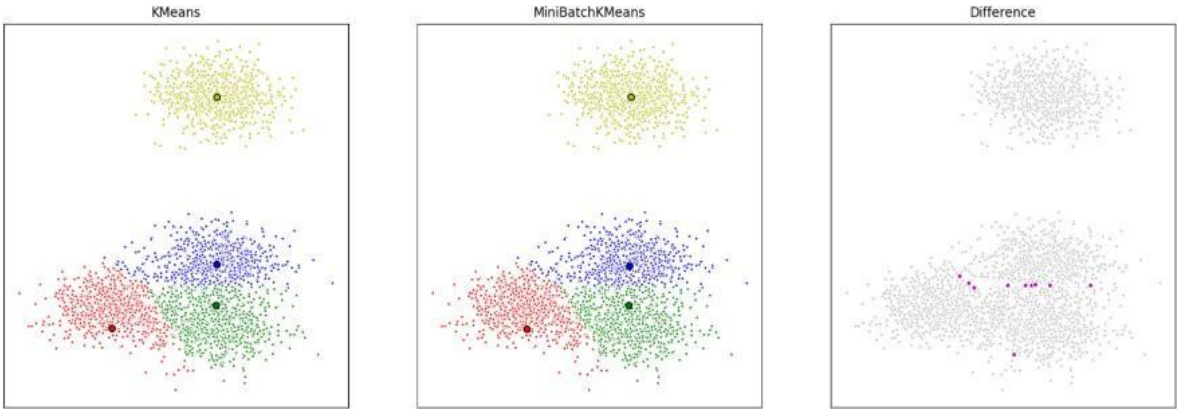


Fig. 5.2 Difference Between K-means and MiniBatch K-Means

5.2.1.4 Nearest Neighbors: Nearest Neighbors is a machine learning algorithm used for classification and regression. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data. These are the steps of the algorithm:

i. Initialize all vertices as unvisited.
ii. Select an arbitrary vertex, set it as the current vertex u. Mark u as visited.
iii. Find out the shortest edge connecting the current vertex u and an unvisited vertex v.
iv. Set v as the current vertex u. Mark v as visited.
v. If all the vertices in the domain are visited, then terminate. Else, go to step 3. The sequence of the visited vertices is the output of the algorithm.

The nearest neighbor algorithm is easy to implement and executes quickly, but it can sometimes miss shorter routes which are easily noticed with human insight, due to its "greedy" nature. As a general guide, if the last few stages of the tour are comparable in length to the first stages, then the tour is reasonable; if they are much greater, then it is likely that much better tours exist. Another check is to use an algorithm such as the lower bound algorithm to estimate if this tour is good enough.

In the worst case, the algorithm results in a tour that is much longer than the optimal tour. To be precise, for every constant r there is an instance of the traveling salesman problem such that the length of the tour computed by the nearest neighbor algorithm is greater than r times the length of the optimal tour. Moreover, for each number of cities there is an assignment of distances between the cities for which the nearest

neighbor heuristic produces the unique worst possible tour. (If the algorithm is applied on every vertex as the starting vertex, the best path found will be better than at least N/2-1 other tours, where N is the number of vertices.)

The nearest neighbor algorithm may not find a feasible tour at all, even when one exists.
In the code provided, the Nearest Neighbors algorithm is used to find the nearest cluster centers to each of the reduced data points.

5.3 TESTING

This code is a Python script that creates a virtual assistant which listens to voice commands and performs various tasks based on the commands received.
The virtual assistant can perform tasks such as opening files, searching on Wikipedia, opening web pages, telling jokes, playing music, and taking notes.
The code uses various Python libraries such as SpeechRecognition, pyttsx3, datetime, os, wikipedia, webbrowser, pygame, subprocess, and pyjokes.
The speak() function is used to convert the text to speech using the pyttsx3 library, and the takeCommand() function is used to listen to the user's voice commands using the SpeechRecognition library.
The wishMe() function greets the user based on the current time, and the play_music() function plays a song using the pygame library. The note() function takes a text input from the user and creates a note file using the subprocess library.
The code uses various conditional statements to execute different tasks based on the user's voice commands. The code is executed in a while loop that continuously listens to the user's voice commands until the user terminates the program.
This code is a Python script for a virtual assistant that can take voice commands from the user and perform various tasks, such as searching Wikipedia, opening websites, playing music, and more. Here is a brief summary of what the code does:

i. Imports necessary packages such as speech_recognition, pyttsx3, datetime, os, wikipedia, webbrowser, pygame, time, subprocess, and pyjokes.
ii. Initializes the speech engine and sets the voice property.
iii. Defines the function to speak out the text using pyttsx3.
iv. Defines the function to wish the user according to the time.
v. Defines the function to take command from the user using the microphone.
vi. Defines the function to play music using pygame.
vii. Defines the function to create a note with the user's input and open it in Notepad using subprocess.
viii. Defines the function to tell the user today's date.
ix. The main function calls the wishMe() function to greet the user and listens to the user's commands continuously.
x. The main function performs various tasks based on the user's commands such as searching Wikipedia, opening websites, playing music, and more.

This code is a simple implementation of a voice-controlled virtual assistant in Python using the speech recognition and pyttsx3 libraries. The virtual assistant can perform various tasks such as opening applications, searching for information on Wikipedia, telling jokes, playing music, and more.

Let's take a closer look at the code:

1. Importing necessary packages: The code starts by importing the necessary libraries for the project, including the speech_recognition, pyttsx3, datetime, os, wikipedia, webbrowser, pygame, time, subprocess, and pyjokes libraries.

2. Initializing the speech engine: The code initializes the pyttsx3 speech engine and sets the voice property to the first voice in the list of available voices.

3. Function to speak out the text: The speak() function is defined to speak out the text passed to it using the pyttsx3 speech engine.

4. Function to wish according to the time: The wishMe() function is defined to wish the user according to the time of day. The function uses the datetime library to get the current hour and then speaks a greeting based on that hour.

5. Function to take command from the user: The takeCommand() function is defined to take user input using the speech_recognition library. The function listens to the user's voice input and returns the recognized text.

6. Function to play music: The play_music() function is defined to play music using the pygame library. The function loads an MP3 file and plays it using the mixer module of the pygame library.

7. Function to create a note: The note() function is defined to create a note by opening the notepad.exe application and saving the text entered by the user in a text file.

8. Function to get the current date: The date() function is defined to get the current date and speak it out using the pyttsx3 speech engine. The function uses the datetime library to get the current month and day and then speaks out the month name and day number.

9. The main function: The main function contains the logic to execute various tasks based on user input. The function uses an infinite loop to keep listening to the user's voice input and then performs the appropriate task based on the input.

For example, if the user says "open youtube", the function will open the YouTube website in the default web browser. Similarly, if the user says "play music", the function will call the play_music() function to play music using the pygame library.

Overall, the code is a good starting point for creating a voice-controlled virtual assistant in Python. It can be expanded upon by adding more features and improving the accuracy of the speech recognition.

CHAPTER 6

RESULTS AND DISCUSSION

In this proposed concept effective way of implementing a Personal voice assistant, Speech Recognition library has many in-built functions, that will let the assistant understand the command given by user and the response will be sent back to user in voice, with Text to Speech functions. When assistant captures the voice command given by user, the under lying algorithms will convert the voice into text. And according to the keywords present in the text (command given by user), respective action will be performed by the assistant. This is made possible with the functions present in different libraries. Also, the assistant was able to achieve all the functionalities with help of some API's. We had used these APIs for functionalities like performing calculations, extracting news from web sources, and for some other things. We will be sending a request, and through the API, we're getting the respective output. API's like WOLFRAMALPHA, are very helpful in performing things like calculations, making small web searches. And for getting the data from web, not every API will have the capability to convert the raw JSON data into text. So, we used a library called JSON, and it will help in parsing the JSON Data coming form websites, to string format. In this way, we are able to extract news from the web sources, and send them as input to a function for further purposes. Also, we have libraries like Random and many other libraries, each corresponding to a different technology. We used the library OS to implement Operating System related functionalities like Shutting down a system, or restarting a system. pyautogui is a library that is implemented for functionalities like, capturing a screenshot. psutil is a library, and is used for functionalities like checking battery status.

The programming language used in this project is Python, which is known for its versatility, and availability of wide range of libraries. For programming the Virtual Assistant, we used Microsoft Visual Studio Code (IDE) which supports Python programming language. Speech Recognition library is present in Python, and is having some in-built functions. Initially, we will define a function for converting the text to speech. For that, we use pyttsx3 library. We will initialize the library instance to a variable. We use say() method and pass the text as an argument to that, for which the output will be a voice reply. For recognizing the voice command given by user, another function has been defined. In that function, define microscope source and within its scope, we use respective functions and store the output in a variable. For the whole process, we have many services to use, like Google Speech Recognition engine, Microsoft Bing Voice Recognition engine, and products of many other big companies like IBM, Houndify etc., For this project, we choose Google's Speech Recognition Engine, that will convert the respective analog voice command into a digital text format. We pass that text as an input to the Assistant, and it will search for the keyword. If the input command has a word that matches with the respective word, the respective function will be called, and it will perform the action accordingly, like telling time, or date, or telling battery status, taking a screenshot, saving a short note, and many more.For this Personal Virtual Assistant, the main advantage is that it saves a lot of time, and it can even handle queries from people, of different voices. There is no rule that one has to give any exact specified command to trigger a particular action. User has the flexibility to give command for user, in natural language. The programming

language used to design this Voice enabled Personal Assistant for PC is PYTHON 3.8.3. And the IDE (Integrated Development Environment) that we used is Microsoft Visual Code.

This Assistant consists of three modules. First is, assistant accepting voice input from user. Secondly, analysing the input given by the user, and mapping it to the respective intent and function. And the third is, the assistant giving user the result all along with voice.

Initially, the assistant will start accepting the user input. After receiving the input, the assistant will convert the analog voice input to the digital text. If assistant was not able to convert the voice into text, it will start asking user for the input again. If converted, it will start analyzing the input and will map the input with particular function. And later, the output will be given to user via the voice command.
The assistant, on starting, will initially wait for the input to be given from user. If the user gives input command, via voice, the assistant will capture it, and searches for the keyword present in the input command. If the assistant was able to find a key word, then it will perform the task accordingly, and returns the output back to user, in voice. If not, the assistant will again start waiting for the user to give input.Each of these functionalities are having their own importance in the whole system working.

- User Input—The assistant will wait for the user to give voice command for further processing.
- Introducing to user—The user who is asking assistant to introduce itself, will display the following.
- Reading out news—If the user asks the assistant to read out some news, the assistant will display the new line by line and it will also read out the news.
- Taking a sample note—If the user has a small note to be taken, he can ask the assistant to do so, and the assistant will take the notes and save it in a notepad file.
- Showing Note—If the user asks the assistant to display the note, and to speak out the note, the assistant will do so.
- YouTube searches—If the user asks the assistant to do some YouTube searches, the assistant will do that. It will ask the user, what to search in YouTube. After receiving the input, it will open the YouTube page with that respective search.
- Web Searches—If the user asks the assistant to do some web searches, the assistant will also do that. It will ask the user to search for what, and it will open the google search in a new tab of browser.
- Opening Applications—If the user asks the assistant to open an application, like MS Word, or any other, the assistant will do so immediately. And also, it will speak that it opens the application.

An AI desktop assistant model is a program that uses artificial intelligence and natural language processing to perform tasks and answer questions for the user. The assistant can perform tasks such as opening applications, playing music, setting reminders, making calls, sending messages, and more.

The model uses speech recognition to understand the user's commands and convert them into text, which is then processed to perform the relevant task. It can also access

the internet and use web scraping to provide the user with relevant information from websites such as Wikipedia or news portals.

The AI desktop assistant model can be customized to suit the user's needs, preferences, and behavior. It can learn from the user's interactions and improve its responses and functionality over time. The model can also be integrated with other devices and platforms such as smartphones, smart home devices, and email.

Overall, an AI desktop assistant model can be a powerful tool for increasing productivity, managing tasks, and simplifying daily life. It can provide a hands-free and intuitive experience for the user, allowing them to perform tasks with ease and efficiency.

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

This paper presents a comprehensive overview of the design and development of a Voice enabled personal assistant for pc using Python programming language. This Voice enabled personal assistant, in today's lifestyle will be more effective in case of saving time, compared to that of previous days. This Personal Assistant has been designed with ease of use as the main feature. The Assistant works properly to perform some tasks given by user. Furthermore, there are many things that this assistant is capable of doing, like turning our PC off, or restarting it, or reciting some latest news, with just one voice command. In conclusion, an AI desktop assistant can be a very useful tool for streamlining daily tasks, improving productivity, and providing personalized assistance to users. With advancements in natural language processing, machine learning, and speech recognition technologies, these assistants are becoming increasingly sophisticated and capable of handling a wide range of tasks.

Here are some additional points about AI desktop assistants:

- **Personalization:** AI desktop assistants can be personalized to suit individual users' preferences, habits, and needs. They can learn from previous interactions with the user to better understand their preferences and provide more accurate and relevant responses.

- **Multitasking:** AI desktop assistants can perform multiple tasks simultaneously, making them more efficient than humans in some cases. They can, for example, send emails, schedule meetings, and perform web searches while the user is working on other tasks.

- **Accessibility:** AI desktop assistants can help users with disabilities or impairments by providing them with voice-activated or other accessible interfaces. This makes it easier for them to interact with their computers and perform various tasks.

- **Integration with other technologies:** AI desktop assistants can be integrated with other technologies such as smart home devices, wearables, and other software applications. This makes it possible for users to control their devices and software with voice commands or other inputs.

- **Security:** AI desktop assistants can help improve security by providing users with secure logins and authentication processes. They can also help prevent data breaches by identifying and flagging suspicious activity.

Overall, AI desktop assistants have the potential to transform the way we interact with our computers and technology. As the technology continues to evolve, we can expect more advanced and personalized features that will further enhance their utility and

usefulness.

7.2 FUTURE WORK

The future of AI desktop assistants is quite exciting, as their potential applications are limitless. The future work of AI desktop assistants will likely focus on improving their natural language processing capabilities and expanding their functionality to perform more complex tasks. Some possible future developments include:

1. **Enhanced natural language processing:** As natural language processing improves, AI assistants will become even better at understanding and interpreting human language, making them even more useful for a wide range of tasks.

2. **Deeper integration with other systems:** AI assistants will become more deeply integrated with other systems and devices, allowing them to control a wider range of tasks and functions.

3. **Increased automation:** As AI assistants become more sophisticated, they will be able to automate more tasks and processes, saving users time and effort.

4. **Advancements in virtual and augmented reality:** AI assistants will be able to work with virtual and augmented reality systems, providing users with a more immersive and interactive experience.

5. **More personalized interactions:** AI desktop assistants could become better at recognizing individual users and their preferences, allowing for more personalized and efficient interactions.

6. **Integration with more devices and platforms:** AI desktop assistants could be integrated with more devices and platforms, including smart homes, cars, and wearables, allowing users to control their environment through voice commands.

7. **Increased intelligence and learning capabilities:** AI desktop assistants could be designed to continuously learn and adapt to users' needs, becoming smarter over time and providing more useful recommendations and advice.

8. **Improved security:** As AI desktop assistants become more integrated into our daily lives, ensuring their security and protecting user data will be critical. Future development could focus on enhancing security measures to prevent hacking and protect user privacy.

Overall, the future of AI desktop assistants is likely to be exciting, with advancements in technology making them even more useful and efficient for users. The future of AI desktop assistants looks bright, with the potential for these systems to become even more useful and integrated into our daily lives.

7.3 RESEARCH ISSUES

There are several research issues that are currently being explored in the development of AI desktop assistants. Here are a few:

1. **Natural Language Processing (NLP):** One of the biggest challenges in creating AI desktop assistants is enabling them to understand and interpret natural language. This involves developing advanced NLP algorithms that can accurately parse human language and respond appropriately.

2. **Personalization:** Another important issue is personalization. To be truly effective, AI desktop assistants need to be able to adapt to each user's unique needs and preferences. This requires developing algorithms that can learn from user interactions and adjust their responses accordingly.

3. **Contextual Awareness:** AI desktop assistants must also be able to understand the context in which they are being used. For example, an assistant that is being used in a noisy environment should be able to adjust its responses accordingly, speaking louder or providing visual cues instead of relying solely on speech.

4. **Privacy and Security:** As with any AI technology that collects data, privacy and security are major concerns. Researchers must develop robust security protocols to protect user data, as well as strategies for addressing potential ethical issues that may arise as these systems become more advanced.

5. **Multimodal Interactions:** Another area of research is the development of AI desktop assistants that can interact with users using multiple modes, such as speech, touch, and gesture. This requires developing sophisticated algorithms that can interpret and respond to multiple types of input.

6. **Integration with Other Systems:** Finally, researchers are exploring ways to integrate AI desktop assistants with other systems, such as smart homes, cars, and other devices. This requires developing standardized protocols for communication between different systems and ensuring that the assistant can operate seamlessly in different environments.

7. **Ethics:** As AI desktop assistants become more advanced, they raise ethical issues around their use. For example, what happens when an AI desktop assistant makes a mistake that harms a user? Researchers are working on developing ethical frameworks that can guide the development and use of AI desktop assistants.

7.4 IMPLEMENTATION ISSUES

Implementing an AI desktop assistant can pose several challenges, including:

Data privacy and security: AI assistants require access to a user's personal data, such as contacts, calendar, and email. Ensuring the privacy and security of this data is a major concern and requires careful implementation and monitoring.

Compatibility with different operating systems and devices: AI assistants need to be designed to work seamlessly with different operating systems and devices, including desktop computers, laptops, tablets, and smartphones.

Natural Language Processing (NLP): One of the key features of an AI desktop assistant is its ability to understand natural language commands and questions. However, NLP is a complex and evolving field, and developing accurate and efficient NLP algorithms for an assistant can be challenging.

Continuous learning: To improve its performance and provide better assistance, an AI desktop assistant needs to continuously learn and adapt to the user's preferences and behavior. This requires sophisticated machine learning algorithms and data analysis techniques.

Integration with third-party applications and services: An AI desktop assistant should be able to integrate with third-party applications and services, such as email clients, productivity tools, and social media platforms. However, this can be challenging due to compatibility issues and API restrictions.

Integration with existing systems: One of the biggest challenges of developing an AI desktop assistant is integrating it with existing systems such as calendars, emails, and messaging platforms. It requires careful design and development to ensure that the assistant can communicate seamlessly with other applications.

Natural language processing (NLP) accuracy: The accuracy of NLP algorithms used in AI desktop assistants is crucial to their effectiveness. However, NLP algorithms can be complex and require a large amount of data to train the models. Developers must ensure that the algorithms are accurate enough to understand and interpret user requests and respond appropriately.

User training and education: AI desktop assistants can only be effective if users know how to use them. Developers must provide adequate training and education resources to help users understand how to interact with the assistant and how to use its various features.

Platform compatibility: AI desktop assistants must be designed to work on different operating systems and hardware configurations. This can be challenging since different systems may have different requirements and limitations. Developers must ensure that their assistants work seamlessly across different platforms and configurations.

These are just a few of the implementation issues that can arise while developing an AI desktop assistant. Addressing these challenges requires careful planning, design, and development to ensure that the assistant is effective, user-friendly, and secure. Overall, implementing an AI desktop assistant requires a multidisciplinary approach, including expertise in machine learning, NLP, data privacy and security, software engineering, and user experience design.

REFERENCES: -

1. Elizabeth Sucupira Furtado, Virgilio Almedia And Vasco Furtado, “Personal Digital Assistants: The Need Of Governance”

2. Zecheng Zhan, Virgilio Almedia, And Meina Song, “Table-To-Dialog: Building Dialog Assistants To Chat With People On Behalf Of You”

3. Yusuf Ugurlu, Murat Karabulut, Islam Mayda “A Smart Virtual Assistant Answering Questions About COVID-19” Mathangi Sri “NLP In Virtual Assistants”

4. Anxo Pérez, Paula Lopez-Otero, Javier Parapar. “Designing An Open-Source Virtual Assistant”

5. C K Gomathy And V Geetha. Article: A Real Time Analysis Of Service Based Using Mobile Phone Controlled Vehicle Using DTMF For Accident Prevention. International Journal Of Computer Applications 138(2):11-13, March 2016. Published By Foundation Of Computer Science (FCS), NY, USA,ISSN No: 0975-8887

6. Dr.C.K.Gomathy , K. Bindhu Sravya , P. Swetha , S.Chandrika Article: A Location Based Value Prediction For Quality Of Web Service, Published By International Journal Of Advanced Engineering Research And Science (IJAERS), Vol-3, Issue-4 , April- 2016] ISSN: 2349-6495

7. C.K.Gomathy And Dr.S.Rajalakshmi.(2014),"A Business Intelligence Network Design For Service Oriented Architecture", International Journal Of Engineering Trends And Technology (IJETT) ,Volume IX, Issue III, March 2014, P.No:151-154, ISSN:2231-5381.

8. “VIRTUAL PERSONAL ASSISTANT (VPA) FOR MOBILE USERS”

9. D. SOMESHWAR, DHARMIK BHANUSHALI, SWATI NADKARNI, “IMPLEMENTATION OF VIRTUAL ASSISTANT WITH SIGN LANGUAGE USING DEEP LEARNING AND TENSORFLOW”

10. C.K.Gomathy.(2010),"Cloud Computing: Business Management For Effective Service Oriented Architecture" International Journal Of Power Control Signal And Computation (IJPCSC), Volume 1, Issue IV, Oct - Dec 2010, P.No:22-27, ISSN: 0976- 268X.

11. Dr.C K Gomathy, Article: A Semantic Quality Of Web Service Information Retrieval Techniques Using Bin Rank, International Journal Of Scientific Research In Computer Science Engineering And Information Technology (IJSRCSEIT) Volume 3 | Issue 1 | ISSN : 2456-3307, P.No:1563-1578, February-2018

12. Dr.C K Gomathy, Article: A Scheme Of ADHOC Communication Using Mobile Device Networks, International Journal Of Emerging Technologies And Innovative Research (JETIR) Volume 5 | Issue 11 | ISSN : 2349-5162, P.No:320-326, Nov-2018.

13. Dr.C K Gomathy, Article: Supply Chain-Impact Of Importance And Technology In Software Release Management, International Journal Of Scientific Research In Computer Science Engineering And Information Technology (IJSRCSEIT) Volume 3 | Issue 6 | ISSN : 2456-3307, P.No:1-4, July-2018

14. Hemalatha. C.Kand N. Ahmed Nisar (2011)., Explored Teachers’ Commitment In Self Financing Engineering Colleges, International Journal Of Enterprise Innovation Management Studies (IJEIMS), Vol2. No2. July-Dec 2011 ISSN: 0976-2698 Retrieved From Www.Ijcns.Com

15. Dr.C K Gomathy, Article: The Efficient Automatic Water Control Level Management Using Ultrasonic Sensor, International Journal Of Computer Applications (0975 – 8887) Volume 176 – No. 39, July 2020.

16. C K Gomathy And V Geetha. Article: A Real Time Analysis Of Service Based Using Mobile Phone Controlled Vehicle Using DTMF For Accident Prevention. International

Journal Of Computer Applications 138(2):11-13, March 2016. Published By Foundation Of Computer Science (FCS), NY, USA,ISSN No: 0975-8887
16. G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
17. W.-K. Chen, Linear Networks and Systems (Book styl\e).Belmont, CA: Wadsworth, 1993, pp. 123–135.
18. H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.
19. B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished. **20.** E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," IEEE Trans. Antennas Propagat., to be published.
21. Ardissono, L., Boella. And Lesmo, L. (2000) "A Plan-Based AgentArchitecture for Interpreting Natural Language Dialogue", International Journal of Human-Computer Studies.
22. Nguyen, A. and Wobcke, W. (2005), "An Agent-Based Approach to Dialogue Management in Personal Assistant", Proceedings of the 2005 International Conference on Intelligent User Interfaces.
23. Jurafsky & Martin. Speech and Language Processing – An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice- Hall Inc., New Jersey,2000.
24. Wobcke, W., Ho. V., Nguyen, A. and Krzywicki, A. (2005), " A BDI Agent Architecture for Dialogue Modeling and Coordination in a Smart Personal Assistant", Proceedings of the 2005 IEEE/WIC /ACM International Conference on Intelligent Agent Technology.
25. Knote, R., Janson, A., Eigenbrod, L. and Söllner, M., 2018. The What and How of Smart Personal Assistants: Principles and Application Domains for IS Research.
26. Feng, H., Fawaz, K. and Shin, K.G., 2017, October. Continuous authentication for voice assistants. In Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (pp. 343- 355). ACM.
27. Canbek, N.G. and Mutlu, M.E., 2016. On the track of artificial intelligence: Learning with intelligent personal assistants. Journal of Human Sciences, 13(1), pp.592-601.
28. Hwang, I., Jung, J., Kim, J., Shin, Y. and Seol, J.S., 2017, March. Architecture for Automatic Generation of User Interaction Guides with Intelligent Assistant. In Advanced Information Networking and Applications Workshops (WAINA), 2017 31st International Conference on (pp. 352-355). IEEE.
29. Buck, J.W., Perugini, S. and Nguyen, T.V., 2018, January. Natural Language, Mixed-initiative Personal Assistant Agents. In Proceedings of the 12th International Conference on Ubiquitous.

APPENDIX

A. Source Code:-

```
#importing necessary packages
import speech_recognition as sr
import pyttsx3
import datetime
import os
import wikipedia
import webbrowser
import pygame
import time
import subprocess                    #To open files
#from tkinter import *               #For the graphics
import pyjokes                      #For some really bad jokes
#from playsound import playsound         #To playsound
#import keyboard                     #To get keyboard


#initializing the speech engine
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice',voices[0].id)

#function to speak out the text
def speak(text):
    engine.say(text)
    engine.runAndWait()

#function to wish according to the time
def wishMe():
    hour = int(datetime.datetime.now().hour) if
    hour>=0 and hour<12:
        speak("Good morning!How can I help you?") elif
    hour>=12 and hour<18:
        speak("Good afternoon!How can I help you?") else:
        speak("Good evening!How can I help you?")

#function to take command from user
def takeCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1
        audio = r.listen(source)
    try:
        print("Recognizing...")
        query = r.recognize_google(audio,language='en-in')
```

```
        print(f"User said: {query}\n")
    except Exception as e:
        #if user does not speak out anything
        print("Say that again please...") return
        "None"
    return query

def play_music():
    pygame.init()
    pygame.mixer.init() #
    Load the MP3 file
    pygame.mixer.music.load("C:/Users/srpandey/Music/Kesariya - Brahmastra.mp3") # Play
    the audio
    pygame.mixer.music.play()
    # Wait for the audio to finish playing
    while pygame.mixer.music.get_busy():
        pygame.time.wait(1)
    # Close the mixer and pygame
    pygame.mixer.music.stop()
    pygame.mixer.quit()
    pygame.quit()

def note(text):
    date = datetime.datetime.now()
    file_name = str(date).replace(":", "-") + "-note.txt"

    with open(file_name, "w") as f:
        f.write(text)

    subprocess.Popen(["notepad.exe", file_name])


def date():
    now = datetime.datetime.now()
    month_name = now.month
    day_name = now.day
    month_names = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September',
'October', 'November', 'December']
    ordinalnames = [ '1st', '2nd', '3rd', ' 4th', '5th', '6th', '7th', '8th', '9th', '10th', '11th', '12th',
'13th', '14th', '15th', '16th', '17th', '18th', '19th', '20th', '21st', '22nd', '23rd','24rd', '25th', '26th', '27th',
'28th', '29th', '30th', '31st']

    speak("Today is "+ month_names[month_name-1] +" " + ordinalnames[day_name- 1] + '.')

#calling the function
if __name___=="__main
    _________": wishMe()
    while True:
        app_string = ["open word", "open powerpoint", "open excel", "open zoom", "open
```

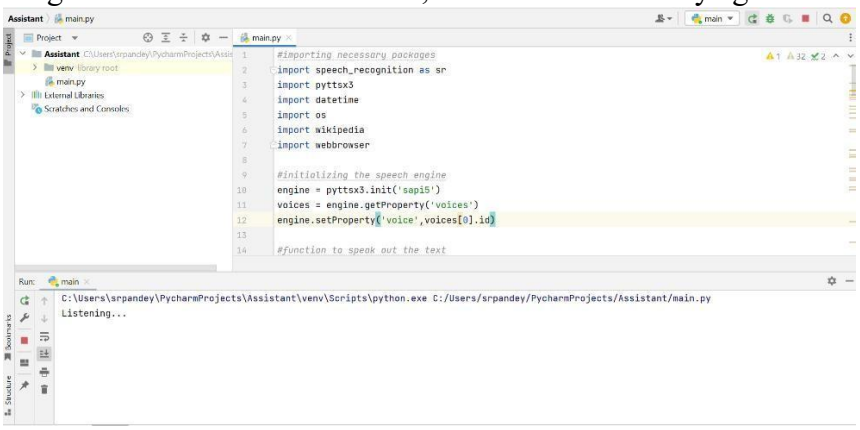
```
notepad", "open chrome"]
        app_link = [r'\Microsoft Office Word 2007.lnk', r'\Microsoft Office PowerPoint
2007.lnk',
                    r'\Microsoft Office Excel 2007.lnk', r'\Zoom.lnk', r'\Notepad.lnk', r'\Google
Chrome.lnk']
        app_dest      =      r'C:\Users\srpandey\Documents\Project\A-GUI-Virtual-Assistant-
with-python-main\A-GUI-Virtual-Assistant-with-python-main'

        query = takeCommand().lower() #converting the command to lower case #logic for
        executing tasks

        if "hello" in query or "hi" in query:
            wishMe()

        elif 'wikipedia' in query: speak('Searching
            Wikipedia...') query =
            query.replace("wikipedia","")
            results = wikipedia.summary(query,sentences=2)
            speak("According to Wikipedia")
            print(results)
            speak(results)

        elif 'joke' in query: speak(pyjokes.get_joke())

        elif 'open youtube' in query:
            webbrowser.open("https://www.youtube.com/")
            speak("Youtube open now")

        elif 'open google' in query:
            webbrowser.open("https://www.google.com/")
            speak("Google Chrome open now")

        elif 'open stack overflow' in query:
            webbrowser.open("https://stackoverflow.com/")
            speak("Stack Overflow open now")

        elif 'play music' in query:
            #music_dir = 'C:\\Users\\srpandey\\Music'
            #songs = os.listdir(music_dir) #print(songs)
            #os.startfile(os.path.join(music_dir,songs[0]))
            #song_path = "C:\\Users\\srpandey\\Music\\Kesariya - Brahmastra.mp3"
            play_music()

        elif 'open gmail' in query:
            webbrowser.open_new_tab("mail.google.com")
            speak("Google Mail open now")
            time.sleep(5)
```

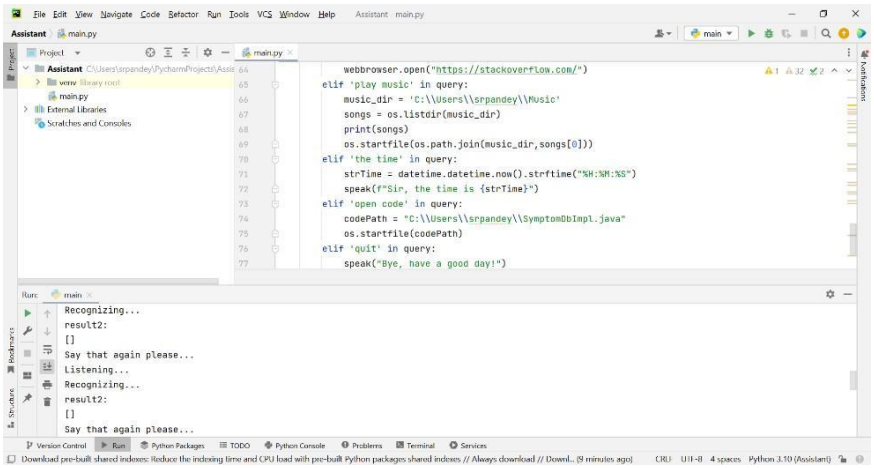
```
    elif 'open netflix' in query:
        webbrowser.open_new_tab("netflix.com/browse")
        speak("Netflix open now, Happy watching")

    elif 'open prime video' in query:
        webbrowser.open_new_tab("primevideo.com") speak("Amazon
        Prime Video open now, Happy watching") time.sleep(5)

    elif app_string[0] in query:
        os.startfile(app_dest + app_link[0])
        speak("Microsoft office Word is opening now")

    elif app_string[1] in query:
        os.startfile(app_dest + app_link[1])
        speak("Microsoft office PowerPoint is opening now")

    elif app_string[2] in query:
        os.startfile(app_dest + app_link[2])
        speak("Microsoft office Excel is opening now")

    elif app_string[3] in query:
        os.startfile(app_dest + app_link[3])
        speak("Zoom is opening now")

    elif app_string[4] in query:
        os.startfile(app_dest + app_link[4])
        speak("Notepad is opening now")

    elif 'the time' in query:
        strTime = datetime.datetime.now().strftime("%H:%M:%S")
        speak(f"Sir, the time is {strTime}")

    elif 'open code' in query:
        codePath = "C:\\Users\\srpandey\\SymptomDbImpl.java" os.startfile(codePath)

    elif 'news' in query:
        news=
webbrowser.open_new_tab("https://timesofindia.indiatimes.com/city/mangalore")
        speak('Here are some headlines from the Times of India, Happy reading')
        time.sleep(6)

    elif 'cricket' in query:
        news = webbrowser.open_new_tab("cricbuzz.com")
        speak('This is live news from cricbuzz') time.sleep(6)

    elif 'corona' in query:
        news=
```

```
webbrowser.open_new_tab("https://www.worldometers.info/coronavirus/") speak('Here
            are the latest covid-19 numbers')
            time.sleep(6)

        elif 'date' in query:
            date()

        elif 'who are you' in query or 'what can you do' in query:
            speak(
                'I am your personal assistant. I am programmed to minor tasks like opening youtube,
google chrome, gmail and search wikipedia etcetra')

        elif "who made you" in query or "who created you" in query or "who discovered you" in
query:
            speak("I was built by Sri Meenakshi Pandey & S Sindu")

        elif 'make a note' in query:
            query = query.replace("make a note", "")
            note(query)

        elif 'quit' in query:
            speak("Bye, have a good day!")
            exit()
```

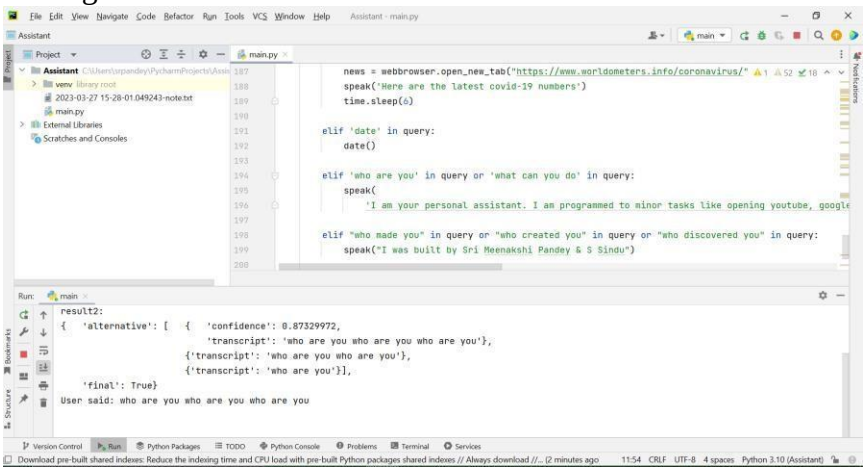
B. Screenshots:-

- User Input—The assistant will wait for the user to give voice command for further processing and if it doesn't understand, it asks the user to say again.

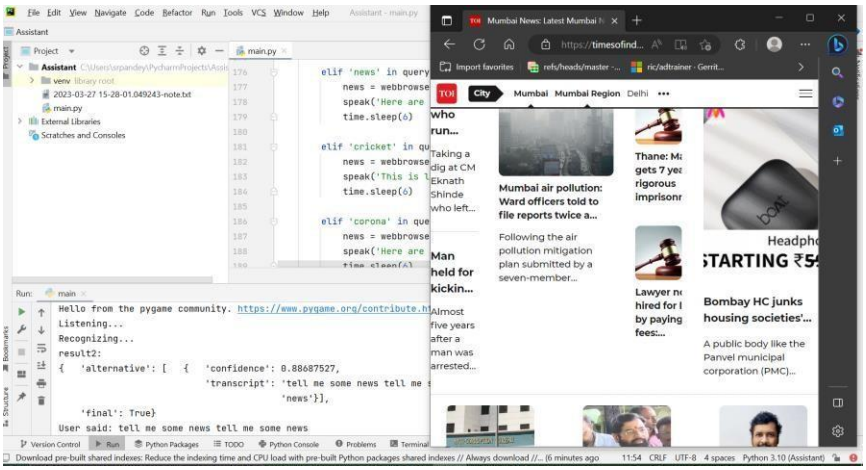




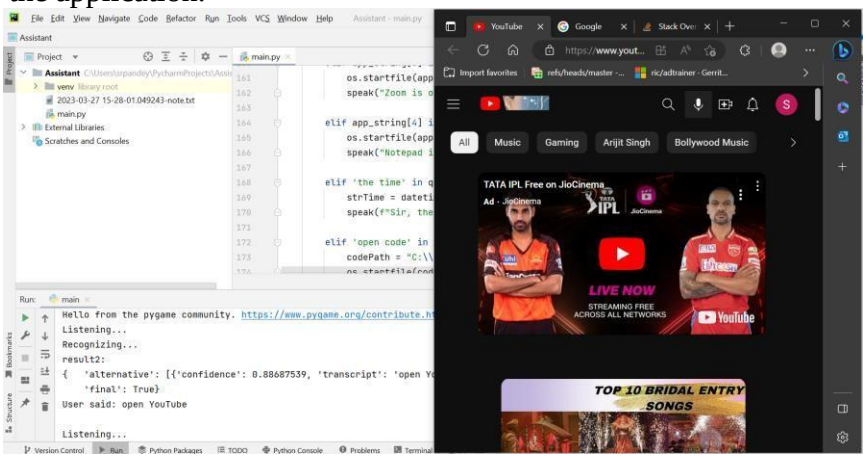
- Introducing to user—The user who is asking assistant to introduce itself, will display the following and introduce him.

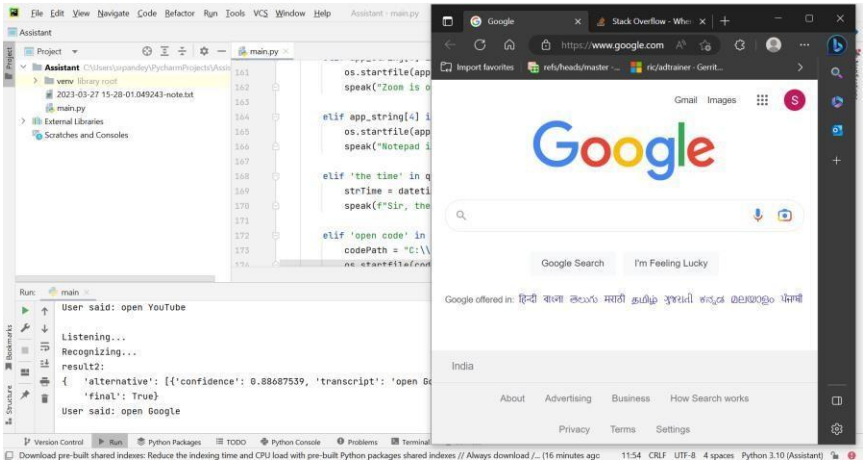


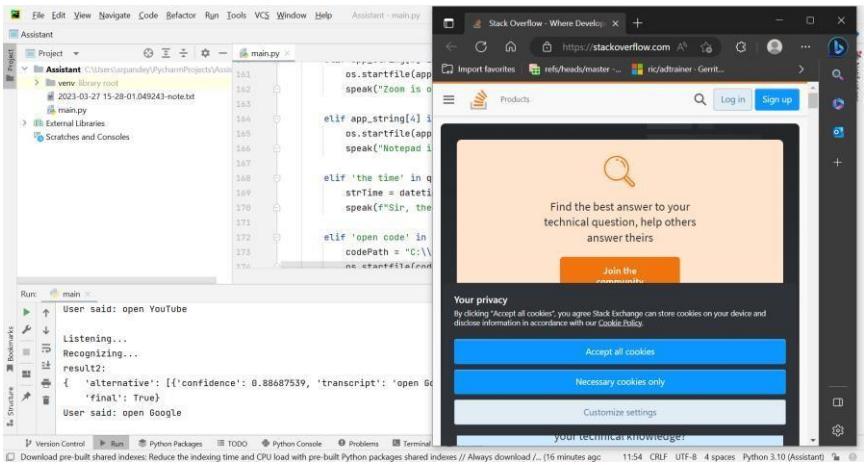
- Reading out news—If the user asks the assistant to read out some news, the assistant will display the new line by line and it will also read out the news.

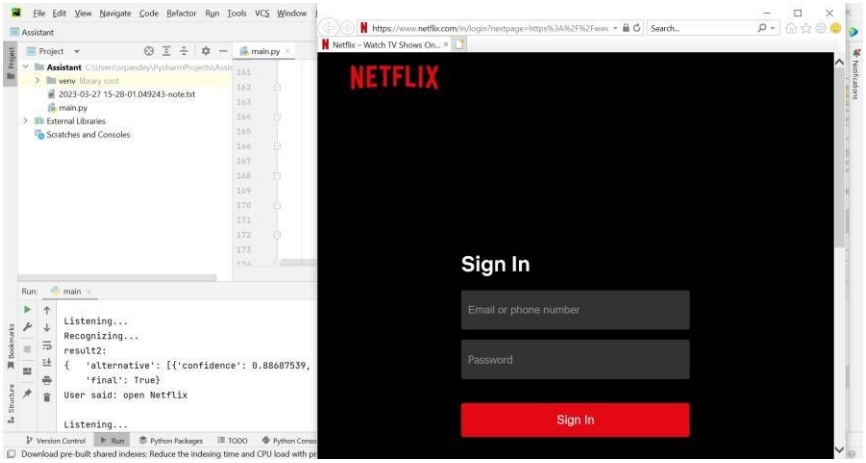


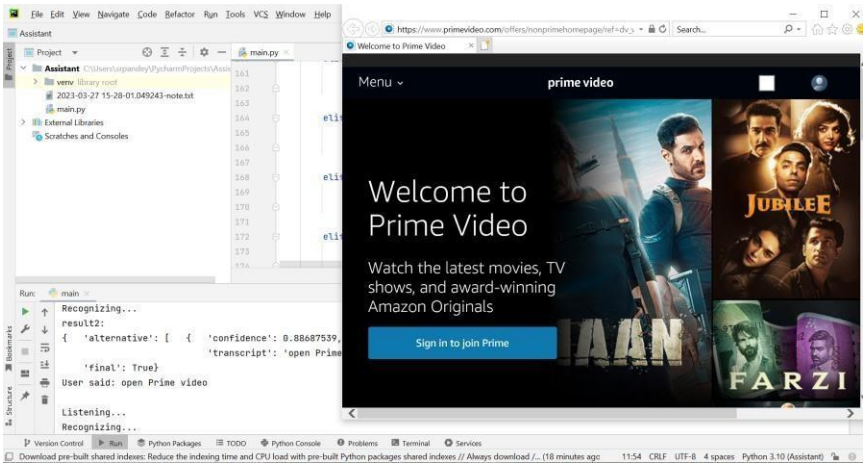
- Opening Applications—If the user asks the assistant to open an application, like MS Word, or any other, the assistant will do so immediately. And also, it will speak that it opens the application.



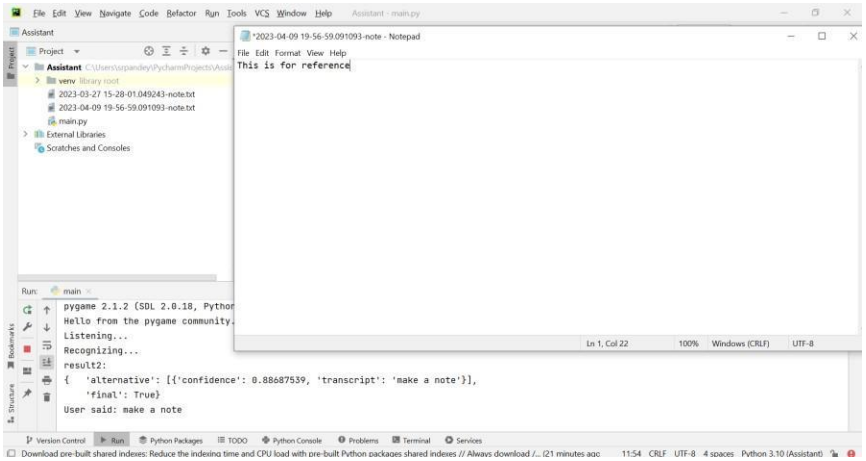




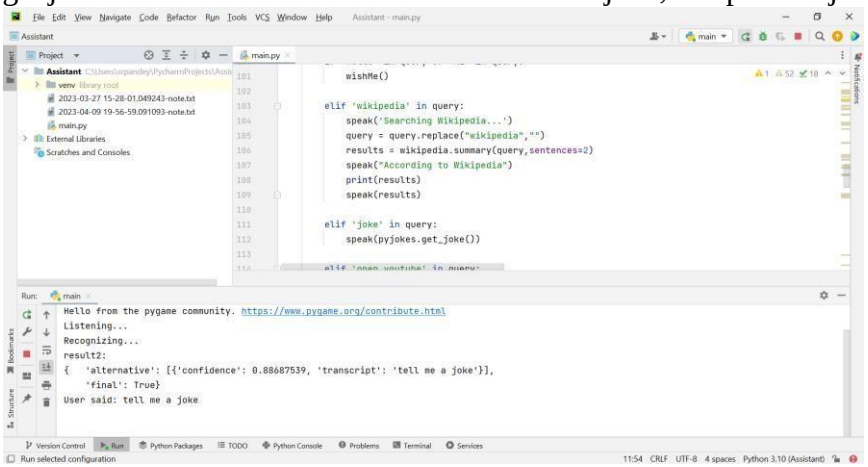




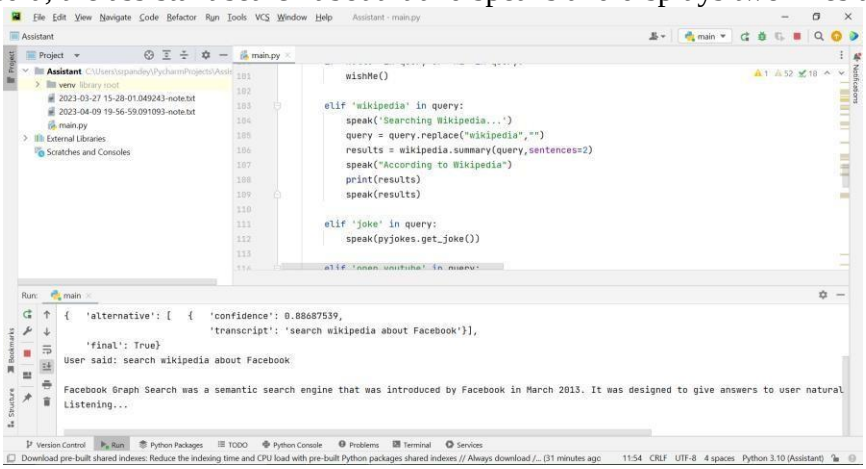
- Taking a sample note—If the user has a small note to be taken, he can ask the assistant to do so, and the assistant will take the notes and save it in a notepad file.



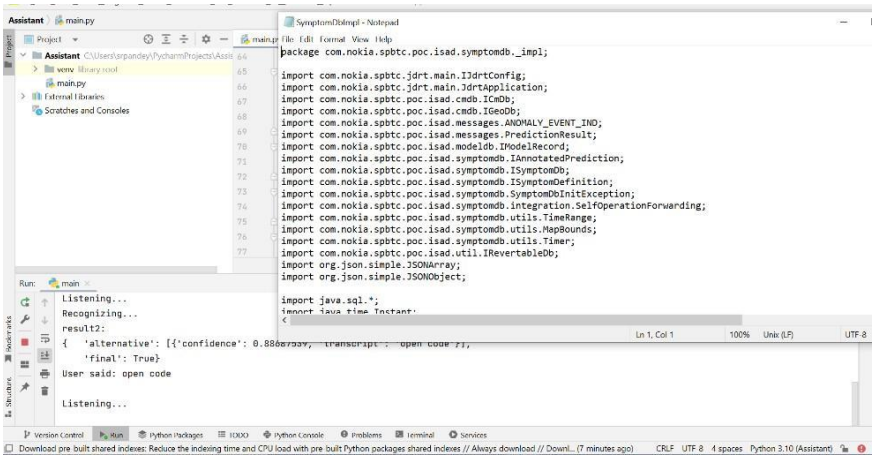
- Telling a joke—If the user asks the assistant to tell a joke, he speaks a joke randomly.



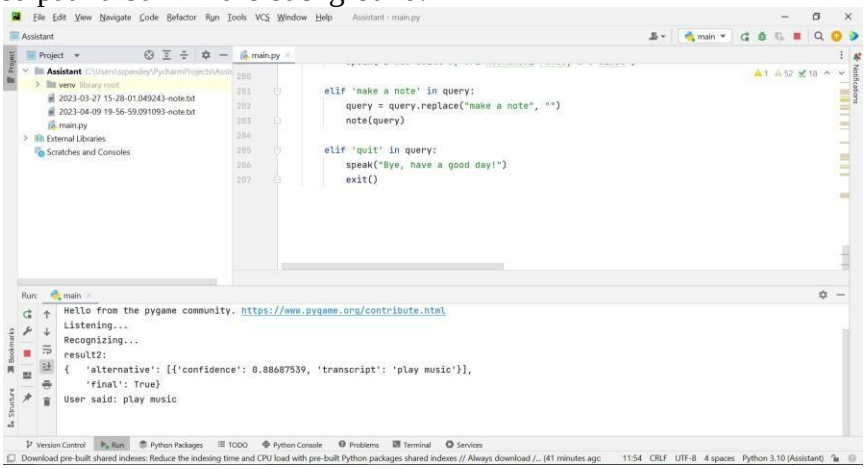
- Searching in Wikipedia—If the user asks the assistant to search about a certain topic in Wikipedia, the assistant search about it and speaks and displays two lines about it.



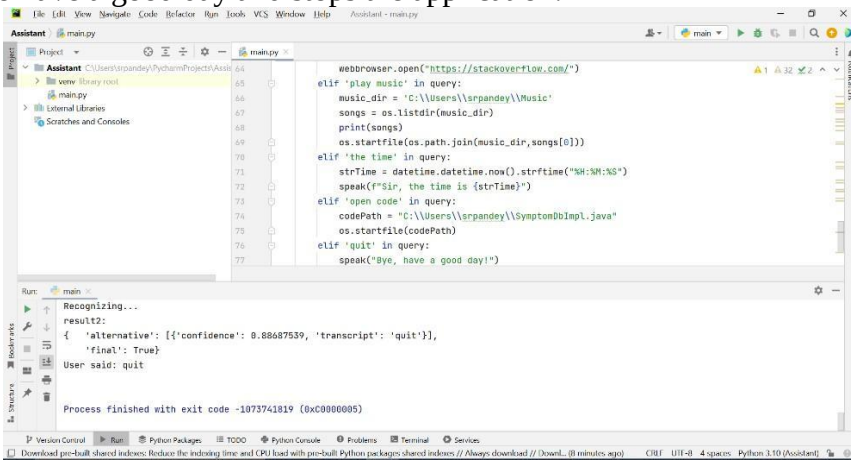
- Opening a file—If the user asks the assistant to open a certain file, the assistant opens and displays it.



- Playing music—If the user asks the assistant to play music, the assistant plays the directed path album in the background.



- Quitting the application—If the user asks the assistant to quit, the assistant greets the user to have a good day and stops the application.



C. Research Paper

AI BASED VIRTUAL ASSISTANT

Sri Meenakshi Pandey [a,*], S Sindu[a], C A Daphine Desona Clemency [a,b]
[a] Sathyabama Institute Of Science And Technology, Department Of Computer Science And Engineering
[b] Semmancheri, Chennai, India, 600119

*** Corresponding author email address**: daphine.cse@sathyabama.ac.in

Abstract

Artificial Intelligence has been fast emerging as a noteworthy technology that has the capability to revolutionize the cognitive behaviour of humans by simulating their intelligence for the betterment of the mankind. AI consists of multi-functional technologies which plays a significant role in our everyday lives like home automation where controlling the computer and performing multiple tasks using voice commands to remote monitoring and control activities. This study is aimed at designing an AI based virtual assistant that acts as a human language interface through automation and voice recognition based interaction from human based on Python language . Instructions for voice assistant are implemented in accordance with user requirement .The most successful Speech recognition software's like Alexa, Siri, etc has been the brainchild of AI technology. Speech Recognition API in python converts speech into text thereby sending and receiving the emails without typing, searching the keywords in Google without even opening the browser along with carrying out several tasks like playing music, scheduling meetings, checking mails etc., has been made possible through the help of this AI based virtual Assistant software. In the present scenario, innovation in digital technologies has resulted in increased effectiveness and accurateness of several tasks that would have required large amount of human effort and resources. Through utilization of AI in every domain, remarkable transformations have resulted in reduced time and labour. Thus AI based voice assistant software offers highly accurate and efficient solution to minimise human effort and time while performing a task that imitates a human assistant to carry out any particular task. Muti-functional aspects like voice commands, sending emails, reading PDF, sending text on WhatsApp, opening a command prompt or IDE, playing music, performing keyword searches in Wikipedia , giving weather forecast, desktop reminders of your choice etc are some of the major operations that can be performed by the developed AI based virtual assistant which also possess certain basic conversational abilities. Multiple python libraries and speech recognition tools has been utilised for the project. A live GUI has been designed for interacting with the AI virtual Assistant as it presents an elegant design framework to carry out the necessary conversation.

Keywords: AI, Python, WhatsApp, Wikipedia, GUI, IDE, Speech Recognition

1. Introduction

Recent times has been witnessing multiple innovative digital technologies based gadgets like smart phone, wearable bands ,fitness gears, multimedia speakers, Bluetooth ear pods, computers , laptops, television, etc., which primarily comprise of voice assistants. Almost every digital gadgets arriving to the market are being equipped with virtual assistants that can control the device through speech recognition. With the modern advancement in the form of AI, the conventional speech recognition based computer systems has got transformed into even more sophisticated and efficient. In today's

world, these technologies have been playing significant roles in our day-to-day lives, right from utilising them for entertainment, education, or interaction with others. This incredible advancement has lead to design and development of Intelligent Personal Assistants (IPAs), whose main aim has been to work towards making lives easier for end-users by facilitating seamless right to use devices through voice commands. This innovation can be termed as revolutionary in accomplishing human- system interaction. To execute the tasks in a more efficient manner, voice based interaction, conversion of speech to text has been utilised to accomplish thorough understanding of input.

Most of the popular voice assistants used in modern times have the ability to answer user based queries through voice commands that are based NLP technology which can be considered as an exact replica of instantaneous conversations between individuals.NLP mainly focuses on designing methods and ML algorithms for better understanding and generation of languages where users are presented with opportunities to interact with intelligent devices that can exactly replicate another human. The active role played by these virtual assistants in understanding the user's requirements in the form of voice commands and assist them by answering their commands is utilised to design a framework where further advancement is in progress. Popular voice assistants like Google assistant, Alexa, Siri and Cortana are real-time instances of impressive innovation and versatility of AI. As majority of tech behemoths like Amazon and Google have their own respective voice assistants, and the probabilities are very high that these virtual assistants will become even more significant in our every day chores in the near future too.

The actual operation of voice assistant begins after getting activated by the user(Fig 1). Normally certain keywords are used as activation inputs to these voice assistants. For instance, users call out "OK Google" or "Alexa" or "Hey Siri" to activate their respective Google, Amazon or Apple voice assistants. As soon the user conveys their minds, AI assistant transforms the acquired input in the form of voice to corresponding data that can be further processed. During the initial stage, any speech obtained from the user is transformed into text followed by means of syntactic and semantic processing of converted text, which refers to the understanding of voice assistant in interpreting actual meaning of translated text, by observing the actual sentence format, its grammatical context, related information and appropriate meaning of word entities. As soon as the exact meaning of obtained information has been understood, the voice assistant explores query related analysis from web or any cloud platform or some application to respond the user related query in an appropriate manner, by offering an answer in the form of text to express its response to user's query. The final step thus involves conversion of text into speech in the form of an audio output to users.

ROLE OF AI IN THE DESIGN AND DEVELOPMENT OF VOICE ASSISTANTS

Advancements in voice assistants in the modern times are aimed at making the assistants efficient in solving the problems along with guiding the users in arriving at appropriate solutions. Thus making these voice assistants smart can be accomplished through repetitive fine-tuning and refining of parameters by utilising machine learning and deep learning techniques. These techniques that can be seen as a subset of AI as it facilitates the computer systems embedded with VA's that has the capacity to involuntarily learn and update themselves through experience and do so without being explicitly programmed by any human. In essence, voice assistants utilize speech recognition ability to transform consumers' speech into audio and vice-versa. Nevertheless, its through advancement of AI and specifically ML and DL tools that facilitates virtual voice assistants to develop them into much smarter, by operating accurately and efficiently and work towards potentially achieving consumer satisfaction by offering them an enhanced user experience.

Source: www.Slanglabs.com

Figure 1: Working of Voice assistants

HISTORY AND EVOLUTION OF INTELLIGENT VOICE ASSISTANTS

Voice control has been initially employed in public forum through HAL 9000, followed by sentient computers in 2001 for Space Odyssey, which was taken up by Starship Enterprise's and quite recently in Iron Man ventures. The evolution of smart VAs in fiction as well as entertainment

industries has been very quick. However, complete utilisation of voice assistants has been possible in real life only from the start of 2010's, when Apple launched its smart phone-based voice assistant, Siri. This segment will offer a systematic review of evolution of voice assistants so far designed and its developments.

SIRI(2010)

Siri has been acquired by Apple which is unequivocally considered to be the most well-known and extremely efficient voice assistant that can perform a wide range of operations like sending text messages, scheduling the meetings, dialling phone calls, ,activate battery power optimising mode, enable DND etc., Siri has the ability to respond user queries, transmit electronic communications through mails, activate alarms, can carry out reservation in restaurants, provides directions to places by its interpretation knowledge of natural languages. Inspite of all the benefits that Siri has been offering, it has its own drawbacks like it can operate only in Apple devices, requires an active internet connection to operate, Siri works well with English commands, but must be spoken clearly to understand, conversing in a rapid manner or using strong accents cannot be identified by siri as its learning abilities may get reduced and it may impact its understanding of user's query. Moreover constraints like inaudible noise, presence of background disturbances and poor quality acoustic from headsets can also limit the voice assistant functions. Siri thus requires seamless Wi-Fi connection for its effective operational abilities.

ALEXA(2014)

Amazon Echo was launched by Amazon which is a smart speaker that facilitates users with voice assistant called Alexa that has been designed as an internal strategy to help Amazon focus and enhance its customer base and further increase revenue through facilitating online shopping experience. The main benefits are it's easier to operate process, nonstop music, shopping, timers etc., but the limitations are its mishearing and slower response rates.

GOOGLE ASSISTANT (2016)

This facility is available on smart phones and other home automation devices which is not only available on its own products but also offers operation in multiple gadgets through joint ventures with other enterprises.

OTHER VOICE ASSISTANTS

Along with the most popular and widely sought after voice assistants discussed above, there are still more smart voice assistants like Facebook's M and Microsoft 's Cortana etc.,. As people are more integrated with modern technologies and rapid proliferation of IoT, AI technology based evolution in the form of voice assistants is unambiguously sure to get this innovative technology to subsequent levels.

Our AI based voice assistant has been designed with the following objectives in mind

- To design effective personal assistant software that uses semantic data sources available on the internet, user generated content and knowledge acquired from knowledge databases.
- To efficiently answer questions posed by users with respect to various domains like business environment, website details, together with an appropriate chat interface.
- To efficiently save the time and efforts by presenting a systematic understanding on several information through detailed research and then making the report terms of our understanding.
- By presenting a rapid voice search mechanism where more time can be saved.

The organization of research is as follows: initial section presents a detailed introduction on voice assistant technology along with recent techniques available in market; second section offers a systematic review on various AI based voice assistants and their benefits and limitations. The section 3 is dedicated to our proposed methodology followed by results discussion and analysis in section 4.The study concludes by presenting a summary of our research in

conclusion and future enhancements section.

2. Literature Survey

Technological companies like Microsoft, Google, Apple and Amazon have been making use of NLP to design and develop their AI voice assistants. The major techniques employed by these software companies involve several processes right from transforming their work flow and enhancing the performance of their Personal Assistants in order to be compatible with their device handling by taking into account its compatibility and complexity. While Google has worked towards improving its voice assistant's capabilities through making use of deep learning methodologies to focus more on dialogue systems .Microsoft employs ML tools and other NN based facilities to improve the Cortana's language processing abilities. Amazon undertakes Speech Recognition technology based functionality to convert speech to text, and certain positive reception of users like tongue appreciations to understand the different dialects, tones and several nuances of text, thereby letting the researchers and developers to design voice assistants to enhance customer's experiences and facilitate realistic dialogue exchanging capacities.

Majority of voice assistants possess feministic tone even though users can modify the voice in accordance to their preferences. As voice Assistants allows us to inquire about everything, be it, location or weather or entertainment options, it also lets us to access translated information in almost over 100 languages. This feature of Google Assistant helps in home automation to control the home from remote places, favorite playlist can be played, and all these functions can be carried out from our smart phones throughout making use of hands-free mode of speech identification process.

Cortana is perhaps considered to be the leading archetypal device that comprises of multiple sensors to sense its environmental surroundings. As a part of Windows Shell, the special abilities of cortana in scheduling and assigning meetings together with a Bot Framework to build skills needed to engage in conversation with other digital assistants. It also learns about our time to be as useful in offering suitable answers along with completing basic tasks.

Alexa is the voice service of all Amazon services like Echo dot and Show that enables customers to undergo personalized understanding by offering facilities to realize their skillfulness. Companies like Uber, Capital One, Starbucks etc., makes use of Alexa-enabled gadgets to enhance their businesses. Following are some of the common tasks performed by voice assistants like

- Setting Reminders and alarms
- Sending and receiving messages
- Creating calendar entries
- Email briefing
- Scheduling meetings
- Play music
- Entertainment
- Gaming
- Weather forecast
- Voice based home automation
- Multi-language answering abilities
- Location information
- Maps
- Cloud and other online services

The voice assistants discussed above have certain limitations like most of the time is consumed in entering the entries than actual work getting done and often they don't possess or manage a detailed knowledge database of their own and their perceptive comprehension mainly arises from data acquired from domain as well as data models.

3. Proposed System Architecture

Majority of the famous and most widely used existing voice assistants uses NLP, and speech recognition technologies to accomplish the task of accurate recognition functionalities. By listening to the directives issued by users, the requirements are understood and specific function in performed in an efficient manner. Artificial Intelligence has been used to generate accurate results and reduce the overall labour and time while carrying out the specific task. As far as conventional typing is concerned, it has been reduced completely and this assistant has been designed to imitate a human assistant in facilitating an effective operation in hand. The

algorithm used focuses more on the time complexities and reduces time. In order to use virtual voice assistants its mandatory to have accounts like Google for

Google assistant, Microsoft account for Cortana etc.,and can be used only with internet connection. Our software is versatile and can be integrated with several devices like, mobile phones, laptops, speakers etc.

Our proposed smart voice assistant can mail without typing a single syllable, can search internet without entering a keyword or opening a browser, and can perform several tasks like playing music, games etc., with the assistance of voice command. Moreover our model is different from other conventional voice assistants that are specific to desktop and requires maintaining a dedicated account to use this; our system does not require any internet connection while obtaining instructions to execute several jobs. IDE used is PyCharm along with several modules and libraries which eased our understanding and assisted in designing and development of our model. GUI has been developed for interacting with the Virtual Assistant whose design and look has been made to enhance the conversation. With further advancement our proposed Assistant can perform any task more effectively than humans. AI is rapidly emerging in every field and decreases human efforts and saves more time and resources. Functionalities of this project include the following

- Sending emails
- Reads PDF
- Sends text on WhatsApp
- Opens command prompt
- Opens our favourite IDE, notepad
- Plays music
- Does Wikipedia searches
- Opens websites in a web browser
- Gives weather reports
- Choice of setting up Desktop reminders.
- Performs basic conversation.

The system has been designed by making use of trending technologies like AI and Python whose tools and libraries has been employed to perform the necessary tasks, for example reading pdf documents using pyPDF. The dataset used in this project is user input (Fig 2), as per user instructions, the assistant performs the tasks accordingly. If user input consists of performing various tasks , for example in case of client requesting to carry out the conversation in particular language i.e., French.

Fig 2: Representation of AI Assistant receiving commands from User

AI when utilized with machines has the potential to mimic and carry out tasks by thinking in the manner humans do. In this research, a voice recognition system has been designed to interact with human using Python which is an emerging language that has been used for scripting the Voice Assistant. Instructions for voice assistant are carried out as per the user requirement In Python , API known as Speech Recognition is present which facilitates us to convert speech into text thus sending emails and searching on Google can be done without opening the browser, and daily tasks like listening to music, playing games can be done with activating a single voice command.

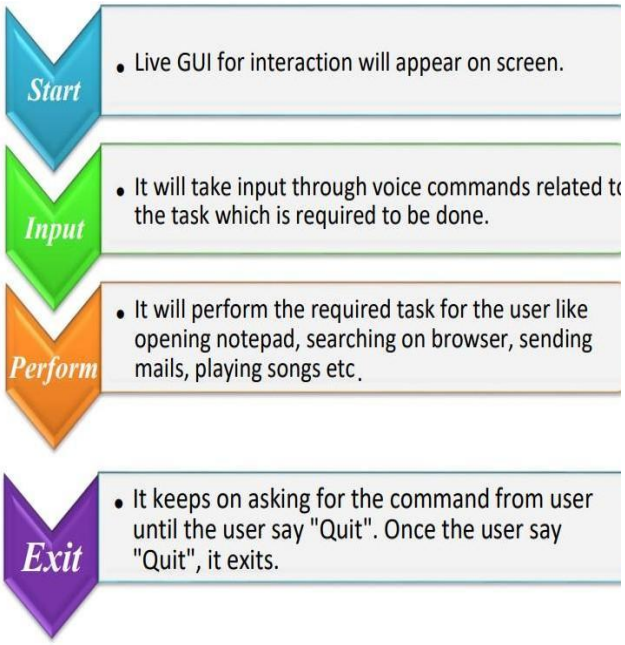


Fig 3: System Architecture for AI Desktop Assistant

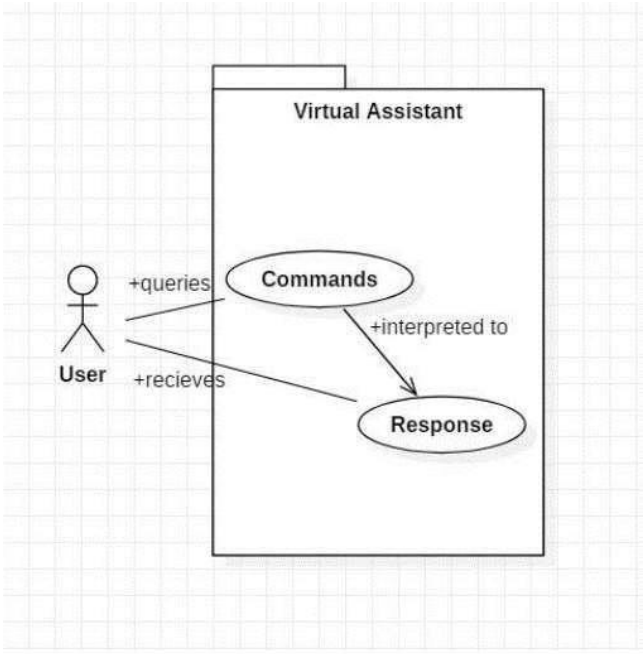


Fig 4:Block diagram of the proposed model

In the present circumstances, development in technologies like AI enables us to accomplish complex tasks in much accuracy and effectiveness than humans. Our proposed architecture and block diagrams are presented in figures 3 and 4.

4 Results Analysis and Discussion

Using NLP as a basic structural framework along with built-in synthetic speech recognition system we have made an effort to design a knowledgeable virtual digital assistant that can be of great help towards managing multiple applications, has the ability to acquire queries from customers, and perform necessary web based searches by conversing with the humans in recognisable tone. In addition, our program has been designated to have complete and seamless access to various programs in computer through knowledgeable interactions and management of associated devices, For example opening YouTube or Facebook, sending mails by logging into the mail account through voice commands, locking the PC, playing games, music, alarms, climate notifications, etc. Voice Assistant is relatively different in comparison to other conventional assistants in terms several requirements like desktop compatibility, requirement of separate user account, need for internet connection while attaining the instructions to perform particular task. To obtain accurate results and analysing them, the foremost step is to install all the essential python packages and libraries. The required command used to carry out this task is pip install and then perform importing them. The necessary packages included are as follows:

EMPLOYED LIBRARIES AND PACKAGES

pyttsx3: This library is responsible for converting text to speech.
Speech-Recognition Module: This component takes care of converting speech to text.
pywhatkit: In order to send WhatsApp messages by means of scheduling the time interval can be achieved through this.
Date and time: Exact date and time can be acquired.
Wikipedia: To facilitate searching for specific information.
Smtplib: This mail transfer protocol lets sending and receipt of mails along with appropriate routing function between servers.
pyPDF2: Used for reading, splitting, merging any

PDF document.

pyjokes: Used for jokes.
webbrowser: An API for specific demonstration of documents to required users.
pyautogui: To support graphical user based interface.
OS: Operating System support based operations.
sys: In order to act as an interpreter and allow access to various attributes this module is used that is responsible for necessary interaction .

FUNCTIONS USED IN PROPOSED MODEL

takeCommand(): To obtain command from users as input through device's microphone and it usually returns output in the form of a string.
wishMe(): This utility greets the user based on the time of day.
taskExecution(): Contains all the required commands to carry out execution in the form of reading mail, intercepting PDF, getting recent news and weather updates ,opening various websites, applications, keywords based search in web, playing music and games etc.,

To carry out complete testing of our proposed model in a fully integrated set up along with accurate testing and validation of system's performance and accuracy, our result analysis on developed virtual assistant focuses on the subsequent aspects namely:

Operational efficiency:

In order to effectively evaluate the required functionality of the system to ensure its efficient and reliable performance of any particular function, it becomes necessary to test the efficacy of each function and commands made use in our project. Testing every command and cross checking its results by repeated executions, helps in understanding the proper functioning of system .By making sure that our model passes the functionality test several commands has been directed by user and satisfactory outcomes has been received by receiving appropriate responses to queries.

Serviceability:

To measure the usefulness of our proposed system, it becomes mandatory to determine its flexibility and convenience while actual operation of developed model. To effectively perform this

testing, ability of the software in terms of its user friendly attribute, compatibility, accurate understanding of user queries and delivery of answers to each query needs to be tested. To accomplish this requirement, completion of designated task by picking up the appropriate python module needs to be done in normal human conversational manner. As it becomes essential for the user to experience as if instructions are given to another individual ad not to any robot. Thus uncomplicated conventional conversation based interactions to provide input and getting the desired output must be achieved. The developed desktop assistant must be highly intelligent to understand and intercept human languages in effective way and should possess the ability to comprehend/recognize the context supplied by the user and offer necessary response in the similar language in which question has been posed. The user must therefore be impressed by the output and assistant's multitasking ability. All instructions can be carried out and responses can be obtained until user prefers to quit. The major activator is the user instruction followed by necessary response of virtual assistant after listening and understanding the actual requirements.

Reliability:

To guarantee security and reliability of our proposed model, this testing mainly concentrates on potential vulnerabilities that may exist in our system. Since our model is local desktop based application, the probabilities of any risks and possible data theft can happen only through remote accessibility. To ensure further security, our software is dedicated for usability by specific system so that only authorized user can activate it by secured manner.

Durability:

The aim of any technology is to offer a stable and consistent performance which in turn depends on actual outputs of the system .Existence of dependencies between desired output and particular input can assure stability and durability in the system. Thus if a system manages to meet all desired functionalities in terms of validation and testing, then the system can be considered to be stable and reliable.

Research scope and management

The scope, objectives and detailed plan on execution of this proposed virtual assistant system based on AI has been performed by our team to effectively manage and develop a technique that can be of greater use to future innovations. Right from installation of all required software and tools, bringing in necessary packages and libraries, development of related functions and attributes, designing API/GUI to facilitate uninterrupted connection has been done meticulously by our entire team members. Relevant research studies has been taken up before actual implementation to ensure understanding the constraints and challenges involved in executing this project, all essential documents for user requirements and adequate functionalities, analysis, followed by supporting documentation, related programming code has been made available to ensure all necessary parameters in terms of functionality.

Front end GUI containing appropriate classes, libraries and functions has been loaded to guarantee convenience of use ,attractive interaction and user friendliness in real time implementation. Programming has been done in Python language, PyCharm IDE has been used to install all necessary packages and libraries, Several functions like takeCommand, taskExecution etc., have been used to acquire queries from users using microphone and performs necessary task executions like opening relevant application, searching for keywords, playing games or music. The main motivation of this project has been to integrate the latest technology like AI in creation of a system that has the ability to think and assist humans in a better way by minimising the time and effort. The basic functionality of our model includes

- Sending e-mails and reading
- Can read PDF documents
- Texting WhatsApp
- Opens command prompt,
- Opening IDE and applications like MS-Office, notepad etc.,
- Plays music
- Keyword based Wikipedia search
- Opening of websites (Google, Facebook, etc.,)
- Update on weather conditions
- anage schedules/set desktop reminders/alarms
- Engage in conversation

Necessary updates has been made in reports by appending relevant screen shots of inputs and obtained outputs, probable constraints and scope of this project in near future with possible enhancement.

5 Conclusion and Future enhancements

With NLP and speech recognition as the fundamental framework, our system has been built with necessary features to design an intelligent virtual desktop assistant that has the ability to manage various tasks ,applications, send appropriate replies to user queries, and carry out keyword searches and has the capacity to carry out human like conversation with users by interacting smartly and manages the inter-connected devices. The user friendly aspect along with provision of various benefits like updating the remainders, setting up schedules, acquiring weather updates are accomplished through assistance of our proposed model. After thorough analysis and systematic validation of multiple attributes and functionalities of our system in comparison with existing voice assistant systems, it can be assured that out proposed AI based model offers satisfactory performance and accuracy in terms of reliability and stability. Its environment friendly factor also adds to the much- organized requirement of any voice-controlled system. There are scopes of further enhancements specially to employ latest machine learning and deep learning methodologies to make our system more real-time and futuristic.

REFERENCES

[1] Gimpel, H. (2015). Interview with Thomas W. Malone on "Collective Intelligence, Climate Change, and the Future of Work." Business & Information Systems Engineering, 57(4), 275– 278. https://doi.org/10.1007/s12599-015-0382-4

[2]B. A. Shawar and E. Atwell, "Chatbots: are they really useful?", LDV Forum, vol. 22, no. 1, (2007)

[3] A. M. Turing, "Computing Machinery and Intelligence", Mind, (1950), pp. 433-460

[4] A. Khanna, “Pandorabots Chatbot Hosting Platform SARANG Bot”, (2015) April 19, Internet:http://pandorabots.com/pandora/talk?botid=9f0f09a71e34dcf8/
[5] A. I. Alice, “Foundation. Free A.L.I.C.E. AIML Set”, (2015) March 21, Internet: http://code.google.com/p/aiml-en-us-foundation- alice/
[6] B. Whitby, Author, “Artificial Intelligence”, The Rosen Publishing Group, (2009)
[7] Xin Lei, Andrew Senior, Alexander Gruenstein and Jeffrey Sorensen “Accurate and Compact Large Vocabulary Speech Recognition on Mobile Devices,” in INTERSPEECH. 2013, pp. 662–665, ISCA.
[8] Alex Graves, Santiago Fern´andez , Faustino Gomez , and J¨urgen Schmidhuber “Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks,” Proceedings of the 23 rd International Conference on Machine Learning, Pittsburgh, PA, 2006 .
[9] Brian Kingsbury, “Lattice-based Optimization of Sequence Classification Criteria for NeuralNetwork Acoustic Modeling”, 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan.
[10] Daniel L. Stufflebeam, and Anthony J. Shinkfield, “Evaluation Theory, Models, and Applications”, John Wiley & Sons, 2007.
[11] David Rybach, Michael Riley, and Chris Alberti, “Direct Construction of Compact Contextdependency Transducers from Data,” Proc. of INTERSPEECH, pp. 218–221, 2010.
[12] K. Beulen ; and H. Ney “Automatic Question Generation for Decision Tree Based State Tying,” Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98.
[13] Patrick Nguyen, Georg Heigold, Geoffrey Zweig, “Speech Recognition with Flat Direct Models,” IEEE Journal of Selected Topics in Signal Processing, Volume: 4 , Issue: 6 , Dec. 2010.
[14] Klaus Beulen, Hermann Ney, “Automatic Question Generation for Decision Tree Based State Tying,” Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98.