Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN: 2581 - 611X)

VOLUME:8, ISSUE:1, FEBRUARY 2025, PAPER NUMBER:54

Prometheus: Virtual Assistant for Desktop

Irfan Ramzan Parray
Department of CSE
Lovely Profressional University
Phagwara, Punjab
parrayirfan9@gmail.com

Himanshu Choudhary
Department of CSE
Lovely professional University
Phagwara,Punjab
himanshu07170@gmail.com

Gurpreet Singh
Department of CSE
Lovely professional University
Phagwara,Punjab
Gurpreetsinghf22@gmail.com

Ayush Gupta
Department of CSE
Lovely Professional University
Phagwara,Punjab
ag0308958@gmail.com

Abstract—**Virtual Assistants, in many forms, have impacted Human Society since their Innovation in Information Technology. Either it is a gadget or something that reduces human help in any task can be called as a virtual assistant. Majority of the IT Engineers are now spending most of their time in front of screens. So, the assistance for the same is an obvious requirement of all. With help of sufficient knowledge of Artificial Intelligence, Optical Character Recognition, Operating System, Data Structure & Algorithms and Functional Programming, it became possible to develop a Virtual Assistant like Prometheus which can provide tools and assistance regarding Desktop work. Anything like On-Screen Text Extraction, Data Processing, Language Translation, fixing Reminder, opening software or engaging in chat with Artificial Intelligence, Prometheus can be a good help. Though there are many options with Virtual Assistants but a need of exploration with newer ways derived this research.**

Keywords— **Virtual Assistant, Prometheus, Machine Learning.**

I. INTRODUCTION

Today, we are living in an era of competition that demands a quality time from any player of a field to surpass the improvements of competitors. Each individual may face a need of some assistance because of one's working limitations. Hence, most of the working-class professionals are in need of some sort of assistance but it can result in a real expense if it is provided by a human. Thus, a virtual assistant can be more cheap and efficient option. A virtual assistant is a software or an embedded system that, when allotted a particular task, works without any human interference. The assistance that it provides is measurable using certain aspects of the 5use. Basically, Virtual assistants are judged to be efficient in terms of user input accountability, integrated tools and services, multitasking and customizations. A revolutionizing today's era of artificial intelligence is making all these aspects even more required and emphasized. Virtual assistants can be of many types like robotic, software etc. Some can be designed for a particular task or some capable of providing multiple services. While all have their own features as per the needs, world of desktops is do full of such things. Bing Copilot, Cortana, Siri, Google Assistant are some of the well-known examples. When we talk about their features then each varies from each other on certain aspects while also maintain some common traits. Desktop Virtual Assistant literally means a software that assists you with activities going on the Desktop. The assistant can be either for basic Desktop Automations or Professional implementations. However, some assistants are made for particular tasks or software implementations. Regarding the project 'Prometheus', it was well clear to target the needs of a student with respect to the academics. When working, with academic work like an assignment, learning, research and evaluations, the student usually gets to search for tools at different places of the digital world. Data security issues on web tools, lack of integrity, internet requirement are some of the things that a student usually get worried of while academic tasks are to be done on desktop. Lower working dependency of the virtual assistant upon the device statistics, optimized operation, easy control and tool navigation are some other aspects emphasized while the development of the concerned virtual assistant. While keeping in mind the proper execution of each step of SDLC, project is much adaptable to modifications and maintenance. Because of the fact that it is made by following the traditional approach of functional programming, the modularity is insured by a complete and efficient testing of the project at all the three levels namely - unit testing, integration testing and system testing.

The project consists of various conceptual implementations like Machine Learning, Optical Character Recognition, Operating System, Graphical User Interfaces, Database Management System, Automation and many other. Here is an overview for each with respect to the project.

1.1 Machine Learning

Artificial Intelligence has many branches and machine learning is one of them. By its name, it is clear that a machine would be learning something. Machine Learning is a study of developing algorithms which are then feed with datasets to train for future tasks and decision makings when asked in terms of similar inputs. The algorithm, that is written, is called model. This model is always designed for a specific purpose and input. Similarly, Prometheus helps you, to choose movies out of a range of options, using such model called Movie Recommendation System. This model is trained with a dataset to provide approximate suggestions as per the user requirements and options. The suggestions may vary with respect

1.2 Optical Character Recognition

This concept allows to recognise, read and judge text in an image based on its language and characters. It is becoming very important for reading, sorting, collecting, searching and managing data. A very common example of it is google image search. But it is also emphasized by Bing, OpenAI and others. Out of create, read, update and delete operations on such image content, the project chose to work on efficient read of text form the image. The feature of instant launch of the functionality via keyboard shortcut along with snipping customization provides user the capacity to gather the image text into the clipboard for easier paste options.

1.3 Graphical User Interfaces

A necessity of desktop world without which those people can't imagine about working on computers who don't know to operate command lines. Graphical User Interfaces not only stand for ease of desktop users but also target certain other aspects like worldly-relation of desktop work, attractive designing and work picturization. Prometheus is assisted with Graphical User Interfaces to implement certain functionalities like voice reminder, YouTube audio-video downloader, text translator, logo animation, optical character recognition, movie recommendation system and tic-tac-toe.

1.4 Database Management System

Desktop world is completely dependent on data and data on Database Management System. So, Database Management System works as a fuel tanker of the Digital World. The concept requires certain important subareas like CRUD operations, ACID properties of databases and others. Tic Tac Toe game is integrated by DBMS to keep record of players' statistics. The system used is serverless called sqlite3. Hence, the game doesn't require internet connectivity to support any of its functionality.

1.5 Operating System

Work of a normal user starts after operating system in installed properly on one's machine. Certain principles without which the operating system can't operate are to be considered while software Development. Certain tools of operating system are used in the development namely multithreading and multiprocessing. In multithreading concepts like mutual exclusion, deadlock management, data updating and synchronization. It played main role in voice reminder, YouTube audio/video downloader and handling keyboard shortcut. Multiprocessing developed as a need when certain modules required separate space for their execution and perform parallelism.

1.6 Automation

Automation is need of the world where everyone is not free to do simple tasks which can be automated. Automation of software can be done by using Graphical User Interface elements. Keeping track of time and location of each element while maintaining a synchronized flow is to be cared thoroughly through the execution. The concept is implemented to send messages on WhatsApp by verbal command to the assistant.

II. Literature Review

P.Sabhita[1] designed a virtual assistant that intends to aid user by organizing meetings, managing work, interpreting languages, requesting chrome tabs and YouTube videos. It employs Python libraries such as pyttsx3 to provide voice collaborative outputs. Mehdi Mekni, Zakaria Baani, and Dalia Suliema[2] proposed a semi-voice enabled chatbot for the State University and College System that provides assistance and support to students. It implements NLP and NLU modules using platforms such as Diagflow, Witai, Watson Conversations, and Microsoft Bot. This paper recommended a single chatbot for all the state universities and colleges. While researching embodied intelligent virtual assistants, Kim et al. investigated the impact of IVA embodiment on collaborative decision-making processes. [3] It used an incremental method in three test cases: executing tasks without human intelligence, working with a disembodied voice assistance, and working with an embodied assistant. The research indicated Both assistance situations (disembodied and embodied) resulted in superior performance than executing the task alone. Interestingly, participants reported much reduced task loads when working with the embodied helper rather than the disembodied voice assistant. Vinayak Iyer et al created a virtual Assistant desktop for the sight impaired that is tailored to the specific demands of these users.[4] The suggested system takes a modular approach, incorporating multiple functionalities. Like, automates google websites with selenium and Beautiful Soup, Enabling Gmail administration via voice commands, using machine learning and NLP to summarize Wikipedia articles and deliver intelligent answers. This study emphasizes the need of developing inclusive technology solutions to improve the lives of persons with impairments. Mahesh T R et al developed a personal artificial intelligence desktop assistant based on Python. [5] Assistant aims to increase users' productivity and efficiency in their daily computer chores by utilizing NLP modules and speech recognition. This research effort bridges the gap between technology and everyday tasks by adding to the expanding field of AI-powered virtual assistants. analysis and comparison of the movie recommendation algorithms put forward in different research publications is done in this literature study. Numerous research papers that create movie

recommendation systems using various machine learning algorithms and data sources have been chosen by us. [6] Airen and Agrawal (2022) created a user-based, item-based, and hybrid approach to k-nearest neighbors recommendation system for movies. The results showed that compared to other options, the hybrid technique outperformed them in terms of accuracy. Chauhan et al. released a sentiment analysis movie recommendation system that uses user opinions to extract recommendations from reviews and make movie recommendations based on user sentiment. (2021) [7] The proposed solution performed better than more conventional collaborative filtering techniques, according to the results. A machine learning-based, unsupervised recommendation system for movies was created by Putri et al. (2020) [8] It groups movies based on their degree of similarity using clustering techniques like k-means and hierarchical clustering. The recommended strategy outperformed the conventional collaborative filtering methodology, according to the results. A movie recommendation system that suggests films to viewers by utilizing machine learning methods such as gradient boosting, decision trees, and random forests was presented by Furtado and Singh (2020) [9]. The findings demonstrated that the gradient boosting method performed more accurately than the other techniques. Goyani and Chaurasiya (2020) [10] studied the several methods—collaborative filtering, content-based filtering, and hybrid approaches—applied in a motion picture recommendation system review. The combined technology's results indicate that their functioning is superior to that of previous algorithms.

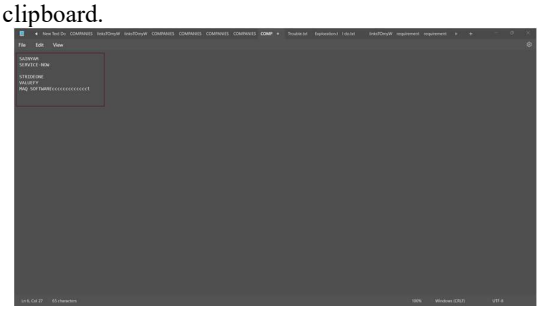
III. RESEARCH METHDOLOGIES

While working on the project, certain methods paved the way of research. The project in itself justifies a different methodology of developing a virtual assistant. The methodology consisted of certain aspects of development like programming approach, language, modularity and conceptual implementations.

3.1 Programming approach: The approach, our team considered efficient, is functional programming or development in terms of modularity. Each individual experimented in his own work space and performed technical emphasize towards the module one was working on. Because the virtual assistant requires to be implementing certain tools (modules), when wanted by user, to assist the work, it was well understood that the project is going to follow the traditional way of Software Development Life Cycle. Python appeared to be the most flexible, easier, versatile, featureful and advantageous language to use for the development. The provision of plenty of libraries stands the main reason to use and extend the capability and capacity of the virtual assistant, respectively.

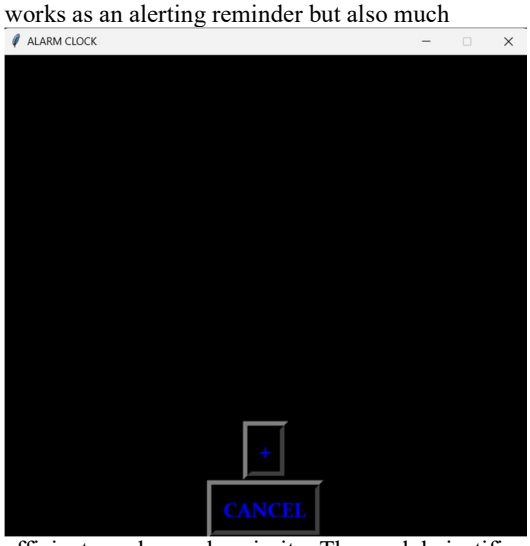
3.2 Modularity:
Modules, that together make up the software, have certain aspects that distinguish them from each other like working, libraries, modular independence and synchronization. All these parameters governed the analysis of modules.

3.2.1 OCR Scanner: A common need of today that takes into account the content written on images. Such implementation become even more sophisticated when it is easier to use, capable to be customized, works anywhere and adhere optimizations.
A user when use a key-shortcut, the feature comes into execution. While displaying an overlapping translucent (enough transparent for easier content visibility) window, it allows user to snip a rectangular portion of the screen and automatically pasting, the written content inside the portion, into clipboard.



Architectural framework of the module relies upon Tkinter library for its implementation of Graphical User Interface. Here it is acting as the translucent window and accounting the snip of the rectangular portion of the screen. In the module, Python Imaging Library works as an image handler because of the fact that the library is dedicatedly made for processing images. Pytesseract is the engine of the module. It provides the capability of optical character recognition of the image taken, as per the snip action of user recorded by Tkinter, by the PIL library.
The module can act as a standalone tool for the fact that its working doesn't require any pre-requisite information or any resulting output for further processing hence promoting modular independence. High cohesion is targeted by resolving user input irregularity or incorrectness to keep a synchronized single threaded flow of the execution.

3.2.2: Voice Reminder:
Verbal acknowledgement to a person busy at screen not only works as an alerting reminder but also much

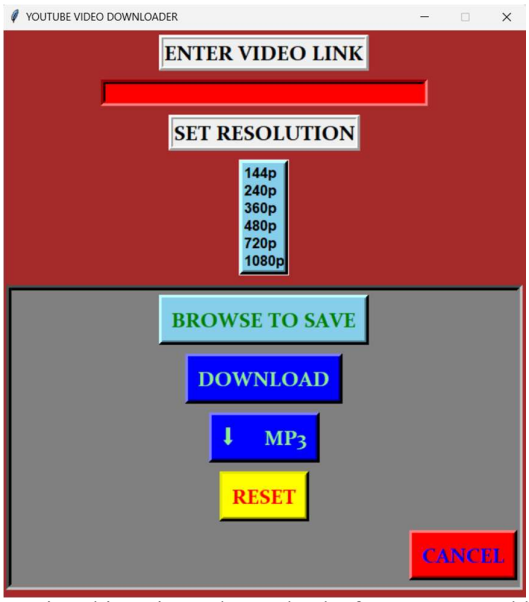


efficient as a humanly priority. The module justifies its name with its working for multiple alarms. The user tells a voice

command to Prometheus and then it will provide the interface to set timings for reminder.
The Graphical User Interface is made using Tkinter. The application is divided into two levels-parent window to add and show alarms while child window to set timings. So, when the alarm time comes, recorded voice command is reminded to the user by the Prometheus by speech. New alarm is set by a new thread by using Thread library for concurrency. This new thread then enters into the critical section of the module by adhering multithreading concepts like mutual exclusion, synchronization, functional consistency and old thread is killed by proper thread management.
As per the aspects of modular independence, the module is dependent on pre-requisite information (user voice command) to return a timed output as per the instructions. However, by eliminating voice recorded input and just using a default alerting sound can reduce the reminder to an alarm clock but making it an independent module or an application. Voice reminder is intensely cared at each stage of thread life cycle for a cohesive synchronization.

3.2.3: YouTube Audio/Video Downloader:

YouTube is now a center of entertainment in terms of video and audio content. A tool to let user download free portion out of that is must for a desktop virtual assistant.



Again, Tkinter is used to make the feature presentable to user and to download any audio or video as per need or interest. Thread Library provides another helping hand at operating system level to handle the interface while child thread is in task of downloading. Pytube is the library used for downloading audio/video as per the parameters selected by user through the Graphical User Interface.
The application is totally independent in terms of functionality. Synchronization, in this module, is related to keeping activities going on while a download is going on.

3.2.4: Text-Translator cum Translation reciter:

Language translation in form of text is a common need nowadays. This module delivers recitation of the translation in addition to that. It provides wide variety of languages. Module's GUI is again powered by Tkinter. One can easily select 'from' and 'to' language in two different scroll downs of languages. After writing the text for 'from' language, one can simply translate the language by clicking on 'translate' button. The translation is done using googletrans library. Another button for verbal clue of the output language initiates gtts library and it's working is handled by child thread using multithreading. The work of gtts is then continued by further processing of data into an mp3 file. This file is played using a library playsound and then deleted from the system upon recitation using os library.
The module is a tool in itself and do not demand interact with any other module except good internet connectivity. So, it can be used easily as a separate software because of its zero modular dependence. The translation in speech form demanded a little bit thread management, file handling, data processing and synchronization.

3.2.5: Voice-Translator:

Voice translation covers another aspect of communication need along with amazement of try and use. The voice input and output are practically much efficient in terms of learning and practice purposes.
The module requires language of input then required language and at last voice input to be translated. All inputs are to be provided verbally to the assistant. The inputs will then be converted into string form and then will be processed similar to the working of translation reciter explained before.

The module requires 3-time use of voice input i.e. dependency on another module. Hence the module can't be used as a stand-alone. In terms of synchronization, the module is very strict towards verbal input and works either in positive return or negative (user request unapplicable or denial) return.

3.2.6: WhatsApp Automation:

The systematic flow of world is now much dependent on automation. Such is the inspiration for this module that confessed about traditional approach towards sending messages on WhatsApp as a headache and boring thing. The module explores another way of sending WhatsApp messages using GUI Automation and verbal command directions from user.
User commands Prometheus to send a WhatsApp message and if WhatsApp is installed then it will proceed further else it will direct you to the website of WhatsApp. Because it is for installed one, it will check certain possibilities of obstacles like low power warning and then continue to ask for receiver's name. It will check for receiver's existence as per the name told by the user and then proceed to ask for the message if receiver exists. User will dictate message and it will send it using mapping of GUI elements of WhatsApp.

A combination of many libraries made it possible to automate the communication. Psutil library's function is used to check if WhatsApp is in running processes of CPU or not. Subprocess library is used for opening WhatsApp. Time library is used to synchronize executional flow of program by providing delays at obstacles and inputs' check. Pygetwindow library helped in handling WhatsApp window and expanding it to Fullscreen. Pyautogui library have played the main role in automating the WhatsApp like locating and identifying GUI elements of WhatsApp and performing Inputs as per user directions. Pyperclip library's functions are used to handle user verbal inputs.

3.2.7: Image to PDF converter:

As suggested from name this module creates pdf files from a series of images selected by the user using a Tkinter Gui. It allows user to select multiple image files using Tkinter file dialog. PIL (Python Imaging Library) will help user to select images and directory for saving the pdf file and also converts each image to RGB format (if it is not in that). Pathlib library is used for handling file paths and aids in creating a PDF file named 'myImages.pdf' in the specified directory. The first image is saved as the first page of the PDF, and subsequent images are appended to it. the function impdf() is somewhat modular as it encapsulates the entire process of image selection, processing, and PDF creation within a single function.

3.2.8: PDF Text Extractor:

Using a Tkinter GUI, the user can pick a PDF file and use the pdftext() function to extract text from it. The function imports pathlib's Path module, Tkinter for the GUI, and PyPDF2 for interacting with PDF files. It uses PdfFileReader from PyPDF2 to open the chosen PDF file.

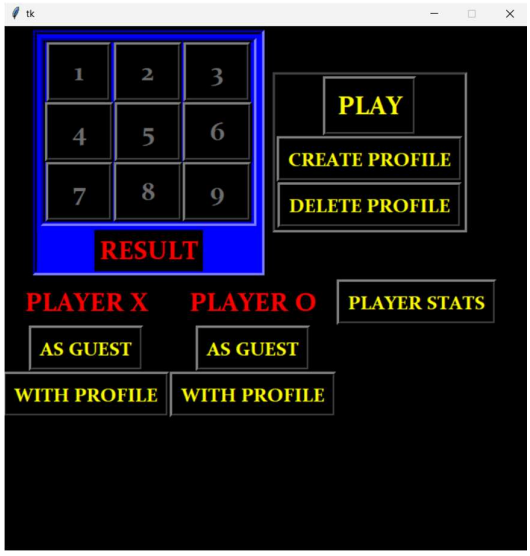
Using the getPage() and extractText() methods, iteratively goes over each PDF page, extracts the content, then concatenates the text into a single string. It uses UTF-8 encoding to copy the extracted text to a new text file called "new.txt.". This function contains all of the steps involved in extracting text from PDF files into a single function.

3.2.9: Tic Tac Toe:

Tic-tac-toe is a mind game that played on a 3x3 grid by two players who place the marks X and O in 1 of the 9 spaces in the grid.

The code creates a Tic Tac Toe interface using the Tkinter library. The game is a two-player game, so players are offered two options: to play as a guest or under a self-titled profile. The game statistics are recorded in terms of wins, losses and draws. Sound is implemented through the chime library to add sounds to game functionalities.

Game and player profile-related aspects are taken care of by functions like turny, disble, anable, work, mark, chng, credel, see, and stats. The code provided maintains synchronization between the frontend and backend parts of the game. Specifically, some of the synchronization work done by the code includes Gui elements such as buttons and labels are

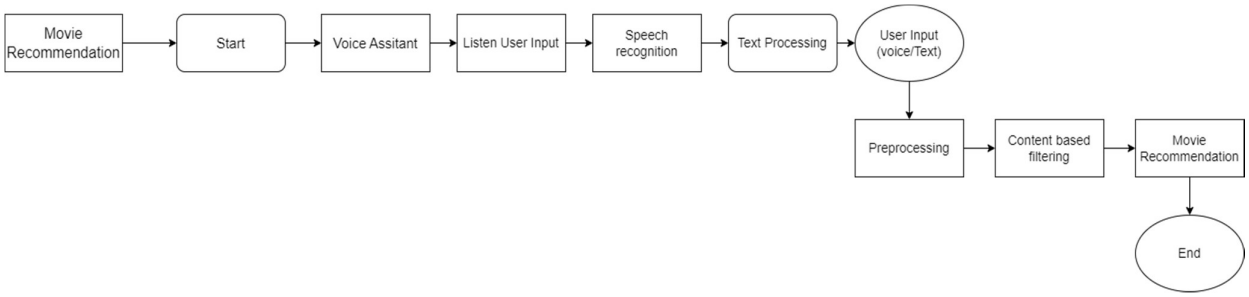


dynamically updated based on the state of the game and the action taken by a player. Database operations interact with the game logic to maintain an accurate record of player statistics Sound is played on cue whenever a player makes the necessary moves or passes milestones.

3.2.10 Movie Recommendation System

Dataset: movies, TV series, and other related media is the Movie Database (TMDb). Programmers commonly use it to construct applications that provide information about movies, including cast and crew, release dates, trailers, reviews, ratings, and other details. One of TMDb's most important features is its API, which One well-known website for information about allows programmers to access its massive data gathering and incorporate it into their applications. Obtaining an API key necessitates signing up for a free account on the TMDb website. This is necessary before using the TMDb API to retrieve data. Once we have the API key, we can use HTTP queries to the TMDb API endpoints to obtain data in JSON format.

Kaggle:



On Kaggle, data scientists and machine learning enthusiasts can connect virtually. When solving data science challenges, Kaggle users can collaborate, machine access and share datasets, use notebooks with GPU integration, and compete with other data scientists. In addition to a community of data scientists and learning practitioners who may cooperate and exchange expertise, Kaggle offers a variety of datasets and tasks for users to work on.

API:

These days, APIs are also gaining popularity. An application programming interface (API) is a collection of procedures, techniques, and tools that let various software applications communicate with one another. It is easier to conceptualize as a messenger that facilitates seamless data and service sharing between two distinct systems.

With APIs, developers may connect to a wide range of services and systems without having to grasp the underlying programming. They function by granting other apps online access to specific protocols, libraries, or services. Because of this, programmers can now utilize the characteristics of previously deployed systems or services to construct new software applications.

IV. RESULT AND DISCUSSION

Text to speech (pyttsx3) and speech to text (speech recognition library by Google) are built-in modules in Python that offer good accuracy as well as a simple and fast method of converting text. The speech-to-text module identified the words with 80 % accuracy using 3 distinct voice samples, each with 20 different inputs in a quiet to moderately-noise environment.

When Virtual Assistant gets the user input about movie recommendation. The movie recommendation system should process movie data. The essential content features of the movies to process were as follows: overview, genres, keywords, the cast details, and insights about cast. These are the parameters that processed to calculate the cosine similarity of movies. If given a movie title, it lists the similar movies based on the content. Moreover, the cosine similarity score provides the recommendations list of either short or long content since there are listeners that have been connected to how movie recommendations are preferential. The system is both quantitatively and qualitatively evaluated; hence the recommendations can be distilled, while the quantity is obtained from user statistics.

CONCLUSION

In this Paper, we have talked about using Python to create a personal virtual assistant for Windows. we addressed voice-activated virtual assistants, their popularity, and their future potential. Which may perform the duties specified by the user in audio format. It can search the internet, browse Wikipedia, download YouTube video, set an alarm, make reminders, automate WhatsApp messages, extract text from pdf, convert pdf to text, Optical Character Recognition, text translation cum reciter, verbal translation, and many more. This assistant makes human life easier. We can incorporate artificial intelligence and IoT into electronic gadgets to make them more advanced. Humans' lives are being made easier by virtual assistants. This Virtual personal assistant is a basic system with many features for handling onscreen tasks. The purpose of this paper is to describe how to develop and implement a desktop assistant that uses voice commands. We can also add some more functionality to the project if necessary.

The implementation of a movie recommendation system within the "Prometheus" project showcases the potential of machine learning algorithms in generating accurate and personalized recommendations. The consistent performance observed across different data subsets highlights the reliability and predictive capabilities of the system. With the advancements in AI and building of LLMs will make this recommendation system more powerful and the virtual assistant more helpful and engaging.

REFERENCES

[1] P. Sabitha, K Saikumar, Vadaboyina,

K Ravi Kiran, Appalaraju, V Rajesh. Design and Development of Intelligent Voice Personal Assistant using Python.In2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)

[2] Mehdi Mekni, Zakaria Baani, Dalia Sulieman. A Smart Virtual Assistant for Students. PPIS 2020, January 7–9, 2020, Las Palmas de Gran Canaria, Spain.

[3] Kangsoo Kim, Celso M. de Melo, Nahal Norouzi, Gred Bruder, Gregory F. Welch. Reducing Task Load with an Embodied Intelligent Virtual Assistant for Improved Performance in Collaborative Decision Making. 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR).

[4] Vinayak Iyer, Kshitij Shah, Sahil Sheth, Kailas Devadkar. Virtual assistant for the visually impaired. Proceedings of the Fifth International Conference on Communication and Electronics Systems (ICCES 2020) IEEE Conference Record # 48766; IEEE Xplore ISBN: 978-1-7281-5371-1

[5] Rabin Joshi, Supriyo Kar, Abenezer Wondimu Bamud and Mahesh T R, (2023). Personal A.I. Desktop Assistant,2(2),54-60. International Journal of Information Technology, Research and Application)

[6] Movie recommender system using k-nearest neighbors' variants. National Academy Science Letters,45(1), 75-82. Airen, S., & Agrawal, J. (2022).

[7] Movie Recommender system using Sentiment Analysis. In 2021 International Conference on Innovative Practices in Technology and Management (ICIPTM) (pp. 190-193). IEEE. Chauhan, A., Nagar, D., & Chaudhary, P. (2021, February).

[8] Design of an unsupervised machine learning-based movie recommender system. Symmetry,12(2), 185.Cintia Ganesha Putri, D., Leu, J. S., & Seda, P. (2020).

[9] Movie recommendation system using machine learning. International journal of research in industrial engineering, 9(1), 84-98. Furtado, F., & Singh, A. (2020).

[10] A Review of Movie Recommendation System. Goyani, M., & Chaurasiya, N. (2020).