**AI DESKTOP PARTNER**

**A PROJECT REPORT**

***Submitted by***

**Dhruv Arora** (19MIM10014)

**Shrey Shrivastava** (19MIM10056)

**Naman Arora** (19MIM10057)

**Asif Ali Sherani** (19MIM10104)

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

*Specialization in*

***Artificial intelligence and machine learning***



**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**VIT BHOPAL UNIVERSITY**

**KOTHRIKALAN, SEHORE**

**MADHYA PRADESH – 466114**

APRIL 2021

**VIT BHOPAL UNIVERSITY, KOTHRIKALAN, SEHORE**

**MADHYA PRADESH – 466114**

**BONAFIDE CERTIFICATE**

Certified that this project report titled **“AI DESKTOP PARTNER”** the Bonafide work of **Dhruv Arora, Shrey Shrivastava, Asif Ali Sherani and Naman Arora** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here does not form part of any other project / research work on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**PROGRAM CHAIR PROJECT GUIDE**

Dr. V. Pandimurugan Dr. Ashish Kumar Sahu

School of AI & ML division School of AI & ML division

VIT BHOPAL UNIVERSITY VIT BHOPAL UNIVERSITY

**ACKNOWLEDGEMENT**

First and foremost, I would like to thank the Lord Almighty for his presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to Dr. V. Pandimurugan, Head of the Department, School of AI & ML division for much of his valuable support encouragement in carrying out this work.

I would like to thank my internal guide Dr. Ashish Kumar Sahu for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of School of Computer Science and Engineering, who extended directly or indirectly all support.

Last, but not the least, I am deeply indebted to my patents who have been the greatest support while I worked day and night for the project to make it a success.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
|  |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO.** | **TITLE** | **PAGE NO.** |

**ABSTRACT**

The project aims to develop a personal-assistant for Linux-based systems. AI desktop partner draws its inspiration from virtual assistants like Cortana for Windows, Siri for iOS and Google assistant for android. It has been designed to provide a user-friendly interface for carrying out a variety of tasks by employing certain well-defined commands . Users can interact with the assistant either through voice commands or using keyboard input. As a personal assistant, Jarvis assists the end-user with day-to-day activities like general human conversation, searching queries in google, bing or yahoo, searching for videos, retrieving images, live weather conditions, word meanings, searching for medicine details, health recommendations based on symptoms and reminding the user about the scheduled events and tasks.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | List of Abbreviations  List of Figures  List of Tables  Abstract | iv  v  vi  vii |
| 1 | **INTRODUCTION**  1.1 Introduction  1.2 Motivation for the work  1.3 Objective of the work  1.4 Summary |  |
| 2 | **LITERATURE SURVEY**  2.1 Literature review  2.2 Idea of project  2.3 Summary |  |
| 3 | **FRAMEWORK**  3.1 Existing work  3.2 Limitations/disadvantages of existing work  3.3 Basic system requirements  3.4 Software requirements  3.5 Summary |  |
| 4 | **SYSTEM DESIGN AND IMPLEMENTATION**  4.1 Module description  4.1.1 Voice recognition  4.1.2 Weather  4.1.3 Text to speech (pyttsx3)  4.1.4 Wolfram Alpha(wolframalpha)  4.1.5 Music  4.1.6 Wikipedia(wikipedia)  4.2 Module work flow  4.3 Coding |  |
| 5 | **PERFORMANCE ANALYSIS**  5.1 Performance measures  5.2 Performance analysis |  |
| 6 | **FUTURE DEVELOPMENT PLANS AND CONCLUSION**  6.1 Limitations of the application  6.2 Future plans  6.3 Conclusion |  |
| 7 | **BASIC LOOK OF THE PROGRAM**  7.1 Snapshots |  |

**INTRODUCTION**

* 1. **Introduction**

A virtual assistant is a technology based on artificial intelligence. The software uses a device’s microphone to receive voice requests while the voice output takes place at the speaker. But the most exciting thing happens between these two actions. It is a combination of several different technologies: voice recognition, voice analysis and language processing. When a user asks a personal assistant to perform a task, the natural language audio signal is converted into digital data that can be analysed by the software. Then this data is compared with a database of the software using an innovative algorithm to find a suitable answer. This database is located on distributed servers in cloud networks. For this reason, most personal assistants cannot work without a reliable Internet connection. With the increasing number of queries the software’s database gets expanded and optimized, which improves voice recognition and increases the response time of the system.

There are also devices dedicated to providing virtual assistance. The most popular ones are available from Amazon, Google and Microsoft. The technologies that power virtual assistants require massive amounts of data, which feeds artificial intelligence (AI) platforms, including machine learning, natural languageprocessing and speech recognition platforms. As the end user interacts with a virtual assistant, the AI programming uses sophisticated algorithm to learn from data input and become better at predicting the end user's needs.

* 1. **Motivation of the work**

This Software aims at developing a personal assistant for windows-based systems. The main purpose of the software is to perform the tasks of the user at certain commands, provided in either of the ways, speech or text. It will ease most of the work of the user as a complete task can be done on a single command. AI Desktop partner draws its inspiration from Virtual assistants like Cortana for Windows and Siri for iOS. Users can interact with the assistant either through voice commands or keyboard input.

Motivation of the work is to develop a virtual assistant capable of doing simple tasks asked by the user with just one simple command making the work easy for the user. This software addresses problems like security issues and privacy issues.

* 1. **Objective of the work**

Currently, the project aims to provide the windows Users with a Virtual Assistant that would not only aid in their daily routine tasks like searching the web, extracting weather data, vocabulary help and many others but also help in automation of various activities. In the long run, we aim to develop a complete server assistant, by automating the entire server management process - deployment, backups, auto-scaling, logging, monitoring and make it smart enough to act as a replacement for a general server administrator. While eliminating security and privacy issues.

* 1. **Summary**

An **intelligent personal assistant** is a software agent that can perform tasks or services for an individual based on commands or questions. Through this voice assistant, we have automated various services using a single line command. It eases most of the tasks of the user like searching the web, retrieving weather forecast details, vocabulary help and medical related queries. We aim to make this project a completely safe by addressing security and privacy issues.

**Literature Survey**

**2.1 Literature review**

Modern Era of fast-moving technology we can do things which we never thought we could do before but, to achieve and accomplish these thoughts there is a need for a platform which can automate all our tasks with ease and comfort. Thus, we need to develop a Personal Assistant having brilliant powers of deduction and the ability to interact with the surroundings just by one of the materialistic forms of human interaction i.e., Human Voice. The Hardware device captures the audio request through microphone and processes the request so that the device can respond to the individual using in-built speaker module. For Example, if you ask the device ’what’s the weather?’ or ’how’s traffic?’ using its built-in skills, it looks up the weather and traffic status respectively and then returns the response to the customer through connected speaker.

Most famous application of iPhone is “SIRI” which helps the end user to communicate end user mobile with voice and it also responds to the voice commands of the user. Same kind of application is also developed by the Google that is “Google Voice Search” which is used for in Android Phones. But this Application mostly works with Internet Connections. But our Proposed System has capability to work with and without Internet Connectivity. It is named as AI Desktop Partner with Voice Recognition Intelligence, which takes the user input in form of voice or text and process it and returns the output in various forms like action to be performed or the search result is dictated to the end user. In addition, this proposed system can change the way of interactions between end user and the devices. The system is being designed in such a way that all the services provided by the devices are accessible by the end user on the user's voice commands.

New capacities lead to creation of various systems for integration of smart things into Social Networks of the Internet of Things. One of the relevant trends in artificial intelligence is the technology of recognizing the natural language of a human. New insights in this topic can lead to new means of natural human machine interaction, in which the machine would learn how to understand human’s language, adjusting and interacting in it. One of such tools is voice assistant, which can be integrated into many other intelligent systems. In this paper, the principles of the functioning of voice assistants are described, its main shortcomings and limitations are given. The method of creating a local voice assistant without using cloud services is described, which allows to significantly expand the applicability of such devices in the future.

many artificial intelligence applications developed using programming languages like Python, R and so on. Each language comes with its own programming structure and syntactical forms. Programmers are broadly classified into three categories namely, novice users, knowledge intermittent and expert one. For novice users, it is always a challenge to write a code without typographic errors though users know theoretical knowledge of Programming language, its structure and syntax as well as logic of program. Therefore, this paper explores use of voice recognition technique in the field of programming, specifically for writing program with Python programming language. In experimental analysis, it found helpful for new Python programmers and provide new learning curve for programmers wherein beginner can experience hassle free program writing. This paper adds new way of creating interest in beginners for judging their coding paradigm understanding and explore one of the areas for user experience field for better programming Integrated Development Environment Development (IDE).

The problem of user while developing a computer program. Developing a computer program is not an easy task it needs hardware resources which user have to handle. While continuous typing the code there may be possibility of injuries to the fingers of the user. To avoid the problems we are designing a system in which the computer program can developed through the voice. The voice will recognized by the system and that recognized words or word will be compared with the stored keywords in the database and if they are matched then that will be printed on editor and after this again by recognizing the specific keywords the program will be compiled and executed. This system will be easy to use, it reduce human efforts and the use of hardware resources. It would be surely useful for blind as well as novice plus knowledge intermittent users.

**2.2 Idea of the project**

Currently, the project aims to provide the windows Users with a Virtual Assistant that would not only aid in their daily routine tasks like searching the web, extracting weather data, vocabulary help and many others but also help in automation of various activities. In the long run, we aim to develop a complete server assistant, by automating the entire server management process - deployment, backups, auto-scaling, logging, monitoring and make it smart enough to act as a replacement for a general server administrator. While eliminating security and privacy issues.

**2.3 Summary**

**Project aims to aid user in its daily tasks by using a voice/text based virtual assistant that will make user’s work easier and faster while also keeping security and privacy issues in check.**

**Framework**

**3.1 Existing work**

These days the need for human assistants is much lower. Personal assistants are a luxury reserved for the elite. You can thank computer technology for this trend, since it’s now pretty easy to manage your own time and automate many tasks that needed the human touch before.

AI voice assistants mainly accept input by voice. Many of them listen constantly for a trigger word, which makes hands-free use possible. If that’s not possible, your simple tap an icon to say your piece. You’ll find them in smartphones, smart speaker, smart TVs and other “smart” internet-connected devices.

Since almost all assistants are cloud-based, these come with a lot of advantages, not to mention the cutting-edge machine learning algorithms either developed or bought by the big players. You almost certainly already have one of these available on your phone, tablet or smart speaker.

**Siri** is the product that put the whole idea of an AI voice assistant on the map. This assistant first shipped with the iPhone 4S and has featured in every Apple mobile device since then. Obviously, you can only make use of Siri on Apple hardware, but at the same time Siri isn’t the only game in town on iPhones anymore either.

The **Google Assistant**, despite the rather unimaginative name, is a powerful AI that’s available on just about every Android device on the market. Powered by Google’s powerful algorithms, this software is eerily intelligent

**Cortana** is named after the AI character from the Halo series of video games. She’s Microsoft’s answer to Siri and is also integrated with the Windows 10 operating system. Microsoft doesn’t really have a stake in phone hardware anymore with the failure of Windows Phone, so Cortana is available as an app download on both the iOS and Android platforms.

**3.2 Limitations/disadvantages of existing work**

If you were to plot machine intelligence and human intelligence over time on a graph, human intelligence would be a horizontal line. Is the curve that represents machine intelligence going to cross that line? Or is the line corresponding to human intelligence an asymptote that the machine intelligence curve draws ever closer to, but never meets?

Speech recognition technology has come a long way since its mass-market introduction in the 1980s. Cloud-hosted databases now allow for on-the-fly recognition of commands and comments in conversational language. And the virtual assistants which respond to these commands have natural-sounding voices, with nuance and personality. But the technology is still developing, and while there are clear advantages to using speech recognition apps and devices with virtual assistants, there are some downsides too

While it's possible to make both time and cash savings using these virtual assistant devices, they don't always come cheap. The Amazon Echo Plus comes with a price tag of £139.99, while the Google Home is slightly cheaper at £119.

Voice recognition has come a long way in recent years, and Amazon and Google's tech is among the best so far. However, it's still not perfect, and you might find that you still have to occasionally repeat yourself to make the smart speaker understand your command.  
  
The good news is that the AI-based voice assistants learn and adapt to your voice as you go along.

The always-on speakers don't record or send your voice commands until you use the wake word (such as 'Alexa...'). But if having a microphone constantly listening bothers you, it can be easily turned off for privacy.  
What's more, a record of each voice command can be easily reviewed and deleted. And while voice control can make ordering goods from the internet a lot easier, it's wise to set a PIN to confirm any purchases. It's also possible to turn off voice ordering altogether - if the kids are home alone

**3.3 Basic system requirements**

• Intel core i3 2nd gen or higher/ AMD Fx-6350 or higher.

•RAM 8gb ddr3 or higher

•Integrated intel 530 HD or Nvidia GT 710 ddr3 or higher.

**3.4 Software requirements**

•Windows 10 1906 or higher

•Python 3.9 or higher with pip.

•PyCharm or any other IDE with python support

**3.5 Summary**

To summarise in today’s always connected having a virtual assistant who is at your service 24/7 and does so whilst maintaining your privacy is very important. There still is a long way to go but we have also come a long way. As we learn new skills, we will be able to improve on this and implement newer technologies to make our assistant smarter and faster. Given right number of resources and correct guidance we might be able to make it competitive to the products currently available on the market.

**SYSTEM DESIGN AND IMPLEMENTATION**

**4.1 Module description**

A *module* is a collection of source files and build settings that allow you to divide your project into discrete units of functionality. A project can have one or many modules and one module may use another module as a dependency. Each module can be independently built, tested, and debugged.

Additional modules are often useful when creating code libraries within your own project or when you want to create different sets of code and resources for different device types, such as phones and wearables, but keep all the files scoped within the same project and share some code. When various modules are integrated together and binded together, they all together work as a single unit and constitutes our project.

The modules that we have used in our project are described below.

**4.1.1 Voice Recognition**

Voice Recognition Module is a compact easy-control speaking recognition board. It is a speaker-dependent module and supports up to 80 voice commands. Any sound could be trained as command. Users need to train the module first before recognizing any voice commands. Voice commands are stored in one large group like a library. We have used ‘speech\_recognition’ module of python for understanding what the user wants to say and convert the query into text so that it can be used in finding the best solution for it.

**4.1.2 Weather**

This module provides the user with the weather of a particular city which is specified by the user. **openweathermap** is a service that provides weather data, including current weather data, forecasts, and historical data to the developers of web services and mobile applications. It provides an API with JSON, XML and HTML endpoints and a limited free usage tier.

**4.1.3 Text to speech**

Text-to-speech (TTS) is a type of assistive technologythat reads digital text aloud. It’s sometimes called “read aloud” technology. With a click of a button or the touch of a finger, TTS can take words on a computer or other digital device and convert them into audio.

TTS works with nearly every personal digital device, including computers, smartphones, and tablets. All kinds of text files can be read aloud, including Word and Pages documents. Even online web pages can be read aloud.

We have used ‘pyttsx3’ module of python for converting the text into speech.

4.1.4 Wolframalpha

Wolfram Alpha is a computational search engine (sometimes referred to as an "answer engine"). The interface looks similar to that of a regular search engine but queries typed into the search box result answers to specific questions rather than listings of websites that may be relevant to the query.

The Wolfram Alpha search box accepts natural language input in keyword, phrase, or sentence format, as well as mathematical equations. The results are dynamically computed.

We have used the ‘wolframalpha’ module of python for implementing the Wolfram alpha search engine.

4.1.5 Music

This module is responsible for bring the music in our AI desktop partner and making it a bit more interesting. Everyone at some time listens to music no matter of what genre. Music has become an essential part of our lives. We can ask of any song that we would like to hear and the AI desktop partner will search the spotify app for the specific song.

The python module that we have used in our application for integrating music is the ‘pysstx3’ module.

4.1.6 Wikipedia

**Wikipedia is** a free, multilingual open-collaborative online encyclopedia created and maintained by a community of volunteer contributors using a wiki-based editing system. Wikipedia is the largest general reference work on the Internet, and one of the 15 most popular websites as ranked by Alexa; in 2021, it was ranked as the 13th most-visited.

We can literally search for anything on Wikipedia and we expect that we would get the answer to our query. We have used the ‘Wikipedia’ module of python to connect Wikipedia with our AI desktop partner so that we can search anything on wikipedia.

**4.2 Module Work Flow**

**4.3 Coding**

1. virtual.py
2. ask.py

import os  
import time  
import wolframalpha  
from virtual import \*  
def ask():  
 speak('I can answer to computational and geographical questions')  
 question=takeCommand()  
 app\_id="29R7PV-X9K3U8X5ET"  
 client = wolframalpha.Client(app\_id)  
 res = client.query(question)  
 answer = next(res.results).text  
 speak(answer)  
 print(answer)

1. **musicplayer.py**

import os  
from pygame import mixer  
  
def musicplayer(localsong):  
 mixer.init()  
 mixer.music.load(localsong)  
 mixer.music.play()  
 while True:   
   
 print("Press 'p' to pause, 'r' to resume")   
 print("Press 'e' to exit the program")   
 query = input(" ")   
   
 if query == 'p':   
   
 # Pausing the music   
 mixer.music.pause()   
 elif query == 'r':   
   
 # Resuming the music   
 mixer.music.unpause()   
 elif query == 'e':   
   
 # Stop the mixer   
 mixer.music.stop()   
 break

1. **news.py**

import os  
import time  
from virtual import \*  
def news(todaysheadlines):  
 speak("Today's headlines are as follows")

1. **spotifyintegration.py**

import spotipy  
import os  
import json  
import webbrowser  
import spotipy.til as util  
import json.decoder import JSONDecoderError  
from spotipy.oauth2 import SpotifyClientCredentials  
  
#https://open.spotify.com/user/2pc6pedag0kee103hzl4i5h49?si=KIm7Rm3YQIqQR4qReOuC8w  
#spotify:user:2pc6pedag0kee103hzl4i5h49  
#username =sys.argv[1]  
client\_id = os.getenv('SPOTIPY\_CLIENT\_ID')  
#client\_secret=os.getenv('SPOTIPY\_CLIENT\_SECRET')  
album\_url='spotify:artist:5nCi3BB41mBaMH9gfr6Su0'  
  
spotify = spotipy.Spotify(client\_credentials\_manager=SpotifyClientCredentials())  
results=spotify.artist\_top\_tracks(album\_url)  
for track in results['tracks'][:10]:  
 print('track : '+track['name'])  
 print('audio : ' + track['preview\_url'])  
 print('cover art: ' + track['album']['images'][0]['url'])  
 print()

1. **vlcplaylist.py**

import requests  
import vlc  
from time import sleep  
urls = [  
 'http://network.absoluteradio.co.uk/core/audio/aacplus/live.pls?service=acbb',  
 'file:///home/rolf/test.m3u',  
 'file:///home/rolf/happy.mp3',  
 'http://statslive.infomaniak.ch/playlist/energy90s/energy90s high.mp3/playlist.pls',  
 'http://streaming.radio.rtl2.fr/rtl2-1-44-128',  
 ]  
  
playlists = set(['pls','m3u'])  
  
Instance = vlc.Instance()  
  
for url in urls:  
 ext = (url.rpartition(".")[2])[:3]  
 test\_pass = False   
 try:  
 if url[:4] == 'file':  
 test\_pass = True  
 else:  
 r = requests.get(url, stream=True)  
 test\_pass = r.ok  
 except Exception as e:  
 print('failed to get stream: {e}'.format(e=e))  
 test\_pass = False  
 else:  
 if test\_pass:  
 print('Sampling for 15 seconds')  
 player = Instance.media\_player\_new()  
 Media = Instance.media\_new(url)  
 Media\_list = Instance.media\_list\_new([url])  
 Media.get\_mrl()  
 player.set\_media(Media)  
 if ext in playlists:  
 list\_player = Instance.media\_list\_player\_new()  
 list\_player.set\_media\_list(Media\_list)  
 if list\_player.play() == -1:  
 print ("Error playing playlist")  
 else:  
 if player.play() == -1:  
 print ("Error playing Stream")  
 sleep(15)  
 if ext in playlists:  
 list\_player.stop()  
 else:  
 player.stop()  
  
 else:  
 print('error getting the audio')

1. **weather.py**
2. **wolframalpha.py**
3. **sendemail.py**
4. **reminder.py**
5. **facedetection.py**

**PERFORMANCE ANALYSIS**

**5.1Performance measures:**

Performance measurement and program evaluation can both help identify areas of programs that need improvement and determine whether the program is achieving its goals or objectives. They serve different but complimentary functions:

* Performance measurement is an ongoing process that monitors and reports on a program's progress and accomplishments by using pre-selected performance measures.
* Program evaluation, however, uses measurement and analysis to answer specific questions about how well a program is achieving its outcomes and why.

So, performance measurement data describes program achievement, and program evaluation explains why we see those results.

**5.2 Performance analysis:**

Most of the time, developers only measure the time a code takes to run and chooses the one that runs quickly. However, many a time, it becomes imperative to check how much memory is being utilized so that we can manoeuvre the code to meet the system requirement.

We should measure the performance of blocks of python code in a project by recording the execution time and by finding the amount of memory being used by the block. This will help us to know the size of the system required to run the application and also get an idea of the duration of the run.

**FUTURE DEVELOPMENT PLANS AND CONCLUSIONS**

6.1 Limitations of the application:

Currently our application cannot recognize the user’s behaviour and adapt itself to better serve the user and since our program relies heavily on Apis for integration with other services. It can break very easily when the developer of the Apis stop supporting them.

Since our application is written in python which makes it almost platform independent but that platform independence comes at the cost of higher execution times and higher memory usage.

Our application also is unable to process natural language and hence, is unable to understand human language which causes it to not understand commands if the structure of the command is changed. I.e., it is able to respond to “what is the weather like today” but if the user said “is it going to rain today” the application is not able to respond.

**6.2 Future Plans:**

In the future we also plan to develop a GUI for this assistant so that user can have more than one way to interact with the application. We plan on doing this by either using python or by using some other language and loosely coupling them by use of xml or json to make them communicate/interact with each other.

In future we plan to implement full scale NLP (natural language processing) So that the assistant can act more like humans and is able to understand and respond to us.

We also plan to implement some sort of learning technique so that it can learn from interaction and is able to improve itself with each usage.

All of the above-mentioned additions usually require heavy computational power which a personal computer is unable to provide. So, we might require to rent some server space to implement all of this and much more.

**6.3 Conclusion**

There is still a long road ahead of us in regards for the development of the project and making it a viable software

From this project, we improved our skills in programming both hardware language and software language. We used Top-Down Design Methodology, divide the big project into several small task and modules. By this way, we improve our progress and made it easy to locate where the problems are because we used small modules to limit the bug region.

During this procedure, we’ve encountered a lot of problems in implementing and we were able to solve most of them though we still have a lot to improve.