

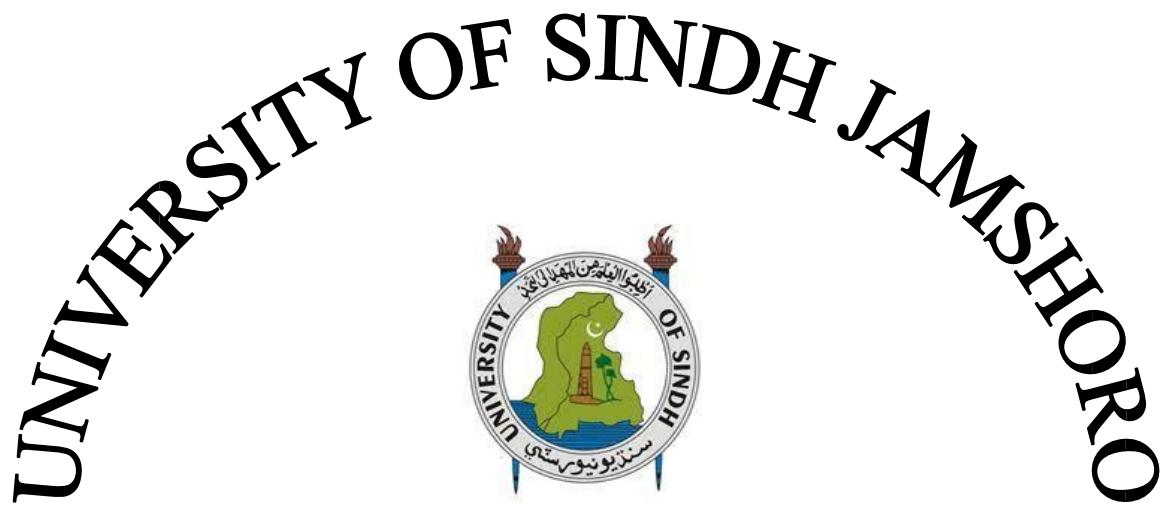
INSTITUTE OF MATHEMATICS AND COMPUTER SCIENCE

BS (CS) THESIS

PERSONAL DESKTOP AI ASSISTANT

THESIS SUBMITTED TOWARDS THE PARTIAL FULFILMENT OF THE
REQUIREMENT OF THE UNIVERSITY OF SINDH FOR THE AWARD OF BS
COMPUTER SCIENCE DEGREE.

DECEMBER, 2023



INSTITUTE OF MATHEMATICS AND COMPUTER SCIENCE

PERSONAL DESKTOP AI ASSISTANT

SUBMITTED BY

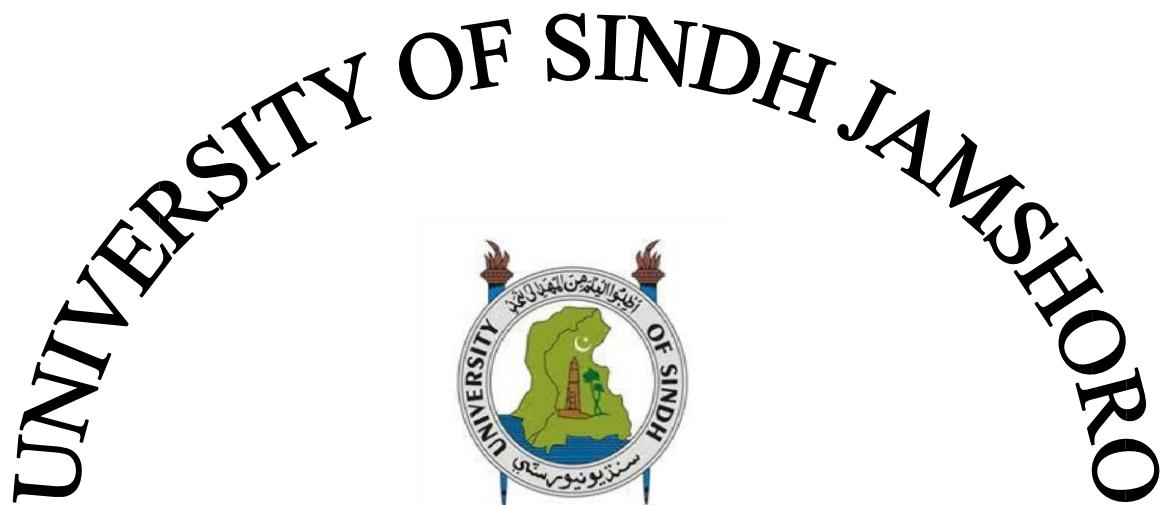
1. Muhammad Sajan (Group Leader) 2K20/CSME/29

2. Hadia Arain 2K20/CSME/20

3. Najeebullah Soomro 2K20/CSME/31

**SUPERVISED BY:
Dr. YASIR ARFAT MALKANI**

DECEMBER, 2023



CERTIFICATE

This is to certify that the project entitled "**Personal Desktop AI Assistant**" has been carried out by **Muhammad Sajan Kalhoro, Hadia Arain and Najeebulah Soomro** during the academic year 2023 as a partial requirement for the degree of Bachelor of Science in Computer Science.

SUPERVISOR

Dr. Yasir Arfat Malkani
Professor
IMCS, University of Sindh, Jamshoro.

DIRECTOR

Prof. Dr. Abdul Wasim Shaikh
Professor
IMCS, University of Sindh, Jamshoro

DECLARATION

This thesis is our original work and has not been ever submitted, in whole or in part, for a degree at this or any other university. It does not contain, to the best of our knowledge and belief, any material published or written by any other person except as acknowledged in the text.

Muhammad Sajan (Group Leader)

2K20/CSME/29

Hadia Arain

2K20/CSME/20

Najeebullah Soomro

2K20/CSME/31

COPYRIGHTS

We, hereby, authorize University of Sindh, Jamshoro to supply our thesis to individuals or libraries for study purposes only.

However, no part of this thesis or the information contained therein may be included in a publication or referred in any publication without prior written permission of the authors. Any reference must be fully acknowledged.

All Copyright Reserved by Muhammad Sajan www.linkedin.com/in/m-sajan-kalhoro

Muhammad Sajan (Group Leader)

2K20/CSME/29

Hadia Arain

2K20/CSME/20

Najeebullah Soomro

2K20/CSME/31

DEDICATION

We dedicate our dissertation work to our family and many of our mentors. This work is also dedicated to our teacher whose efforts made us capable for this achievement and also those friends who helped us.

ACKNOWLEDGEMENTS

We are grateful with the core of our heart to ALMIGHTY ALLAH who made it possible to complete this thesis / project successfully.

Our deepest gratitude goes to our mothers for their unflagging love and support throughout our life. We have no suitable words that can fully describe their everlasting love to us. Similarly, we are grateful to our other family members who always encouraged us in getting this valuable task completed.

We specially thank our supervisor, **Dr. Yasir Arfat Malkani**, for his supervision and support that motivated us throughout the duration of the project with invaluable suggestions time and again.

We wish to thank the Director Institute of Mathematics and Computer Science, **Prof. Dr. Abdul Wasim Shaikh** University of Sindh, Jamshoro for providing required facilities throughout the project duration.

Finally, we thank our friends, colleagues, and well-wishers for their continuous encouragement during the course of this study.

ABSTRACT

"Started out, J.A.R.V.I.S. was just a natural language UI. Now he runs the Iron Legion. He runs more of the business than anyone besides Pepper. And we can get it easy from Google and do it."

The Artificial Intelligence Personal Desktop AI Assistant JARVIS represents an advanced technology aimed at simplifying and streamlining daily tasks for individuals and businesses. It leverages natural language processing and machine learning to comprehend and respond to commands in various languages. The acronym JARVIS stands for "Just rather Mult Intelligence System," drawing inspiration from the fictional advisor with the same name in the Marvel Cinematic Universe.

Remaining true to its namesake, JARVIS is crafted to be intuitive and user-friendly, featuring a straightforward interface for quick and efficient navigation. A notable strength lies in its seamless integration with diverse applications and services, encompassing popular social media platforms, messaging apps, and productivity tools. This consolidated approach enables users to centralize task management and communications effectively. Furthermore, JARVIS is adaptable and flexible, capable of integrating with third-party APIs and services to meet specific requirements.

JARVIS is proficient in a multitude of tasks, ranging from setting reminders, scheduling appointments, and sending emails to making phone calls and controlling smart home devices. Its capabilities extend to providing weather updates, news, and other information tailored to user preferences. The advanced machine learning employed by JARVIS enables it to learn and adapt to user habits over time, enhancing efficiency.

Keywords- associated with JARVIS include bard, Alexa, Mistral, Cortana, Google Assistant, voice assistant, Python speech recognition, and Python text-to-speech library pyttsx3

TABLE OF CONTENT

PERSONAL DESKTOP AI ASSISTANT	i
SUBMITTED BY	ii
CERTIFICATE	iii
DECLARATION	iv
COPYRIGHTS.....	v
DEDICATION.....	vi
ACKNOWLEDGEMENTS	vii
ABSTRACT	viii
CHAPTER 1	1
INTRODUCTION	1
1.1 BACKGROUND.....	1
1.2 ARTIFICIAL-INTELLIGENCE:.....	4
1.3 Virtual assistance:	5
1.4 J.A.R.V.I.S AI.?	5
1.5 The Objectives of the Project	6
1.5.1 Saves Time:.....	6
1.5.2. Conversational Interaction:	6
1.5.3 Reactive Nature:.....	6
1.5.4 Multitasking:	6
1.5.5 No Trigger Phase:	6
1.6 Project scope and direction	7
1.7 Scope of Project	7
1.7.1 Home Automation:.....	7
1.7.2 Personal Assistant:	7

1.7.3 Healthcare:	8
1.7.4 Education:	8
1.7.5 Customer Service:	8
1.7.6 Business Applications:.....	8
1.8 Problem Statement	9
1.9 Speech Recognition:.....	9
CHAPTER 2	11
SYSTEM DESIGN	11
2.1 Hardware Development	11
2.2 Libraries Development.....	12
2.2.1 Speechrecognition==3.10.0:	14
2.2.2 Openai==0.27.8:.....	14
2.2.3 NLTK==3.7:	14
2.2.4 Playsounds==1.3.0:	14
2.2.5 opencv-python== 4.8.0.76:	14
2.2.6 Pywhatkit==5.4:	14
2.2.7 Googletrans==3.1.0a0:.....	14
2.2.8 Flask==3.2.3:	14
2.2.9 Google-auth==2.22:	14
2.2.10 Webdriver-manager==4.0.0:.....	14
2.2.11 GoogleImageScraper==2.3.5:	14
2.2.12 PyPDF2==3.0.1:	14
2.2.13 PyGetwindow==0.0.9:	14
2.2.14 PyAutoGUI==0.9.54:.....	15
2.2.15 Torch==1.12.0:.....	15
2.2.16 Pyaudio==0.2.13:	15

2.2.17 BeautifulSoup==4.12.0:	15
2.2.18 blinker==1.7.0:	15
2.2.19 browser-cookie3==0.19.1:	15
2.2.20 bs4==0.0.1:	15
2.2.21 certifi==2023.11.17:	15
2.2.22 cffi==1.16.0:	15
2.2.23 charset-normalizer==3.3.2:	15
2.2.24 click==8.1.7:	15
2.2.25 colorama==0.4.6:	15
2.2.26 comtypes==1.2.1:	15
2.2.27 curl-cffi==0.5.10:	15
2.2.28 duckduckgo_search==4.1.1:	15
2.2.29 exceptiongroup==1.2.0:	16
2.2.30 fastapi==0.108.0:	16
2.2.31 filelock==3.13.1:	16
2.2.32 Flask==3.0.0:	16
2.2.33 Flask-Cors==4.0.0:	16
2.2.34 g4f==0.1.9.8:	16
2.2.35 huggingface-hub==0.20.2	16
2.2.36 idna==3.6:	16
2.2.37 itsdangerous==2.1.2:	16
2.2.38 keyboard==0.13.5:	17
2.2.39 MarkupSafe==2.1.3:	17
2.2.40 multidict==6.0.4:	17
2.2.41 nest-asyncio==1.5.8:	17
2.2.42 numpy==1.26.3:	17

2.2.43 packaging==23.2:.....	17
2.2.44 pillow==10.2.0:.....	17
2.2.45 platformdirs==4.1.0:	17
2.2.46 pygame==2.5.2:	17
2.2.47 python-pptx==0.6.23:	17
2.2.48 pyttsx3==2.90:	17
2.2.49 requests==2.31.0:	18
2.2.50 selenium==4.16.0:.....	18
2.2.51 sounddevice==0.4.6:	18
2.2.52 undetected-chromedriver==3.5.4:	18
2.2.53 urllib3==2.1.0:	18
2.2.54 webdriver-manager==4.0.1:.....	18
2.2.55 websockets==12.0:.....	18
2.2.56 Werkzeug==3.0.1:.....	18
2.2.57 youtube-transcript-api==0.6.2:.....	18
2.2.58 Click 8.1.3:.....	18
2.2.59 aiohttp==3.9.1:	18
2.2.60 aiosignal==1.3.1:.....	18
2.2.61 annotated-types==0.6.0:.....	18
2.2.62 anyio==4.2.0:	18
CHAPTER 3	19
SOFTWARE DEVELOPMENT.....	19
3.1 Define Features and Requirements:	19
3.2 Choose a Development Platform:	19
3.3 Speech Recognition:.....	19
3.4 Natural Language Processing (NLP):	19

3.5	Task Execution:.....	19
3.6	Text-to-Speech (TTS):	19
3.7	User Interface (UI):	20
3.8	Continuous Improvement:.....	20
3.9	Testing:.....	20
3.10	Deployment:	20
	CHAPTER 4	21
	METHODOLOGY	21
4.1	Speech Recognition Model:.....	23
4.2	Natural Language Processing (NLP):	24
4.3	Skill/Action Integration Module:	24
4.4	Voice User Interface (VUI):	25
4.5	Text-To-Speech Model:	25
4.6	Python Backend:	25
4.7	API Calls:	26
4.8	Context Extraction:	26
4.9	System Calls:.....	26
4.10	Graphical User interface-QtDesigner.....	26
4.11	Hierarchy manner of the face Recognition	27
4.12	Agile methodology	28
4.12.1	Plan:	28
4.12.2	Design:	28
4.12.3	Develop:	28
4.12.4	Test:.....	28
4.12.5	Deploy:.....	28
4.12.6	Review:	28

CHAPTER 5	30
SOFTWARE SYSTEM ANALYSIS.....	30
5.1 Training Model	30
5.2 Neural Networks	30
5.3 Natural Language Processing.....	31
5.4 Speech Recognition System.....	32
5.5 PyTorch - Machine Learning Library	33
5.6 Linear Regression Concept	33
5.7 System Architecture	34
5.8 Sequence Diagram	35
5.9 Use Case Diagram.....	36
5.10 Activity Diagram.....	36
CHAPTER 6	37
MODELING AND ANALYSIS	37
6.1 Selenium Automation Tool.....	37
6.1.1 Selenium Web Driver:	38
6.1.2 Selenium RC (Remote control).....	39
6.1.3 Selenium GRID.....	39
6.1.4 Features of Selenium.....	40
6.2 Beautiful Soup Tool:	40
6.2.1 Uses of Web Scraping.....	41
6.2.2 Steps involved in web scraping:.....	42
6.2.3 Web Scraping libraries/tools	42
CHAPTER 7	43
WORKING OF THE PORJECT.....	43
CHAPTER 8	53

CONCLUSION AND FUTURE SCOPE	53
FUTURE SCOPE.....	53
CONCLUSION.....	54
REFERENCE.....	55

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Our digital life is decided by innovations. Especially in recent years, more innovative technologies were developed to ease our professional lifestyle. An intelligent Personal Assistant is proved to be the most vital innovation in terms of easing our lives and providing a hands-free experience. We are building a PC Personal Assistant that works on voice commands and executes the user query. It does not exist that a person's learns to speak with a system, however currently a computer system learns to speak with a person, looking and traversing their actions, habits, behavior, or nature and creating efforts to become his customized assistant.

Speech recognition is a topic that's extremely useful in many applications and environments in our everyday life.

This system is designed to be used efficiently on desktops. Ai Personal assistant software improves user productivity by managing routine tasks of the user and by providing information from online sources to the user. JARVIS is effortless to use. Call the wake word 'JARVIS' followed by the command. Voice searches have dominated over text search. Web searches conducted via mobile devices have only just overtaken those carried out using a computer and allow your intelligent assistant to make email work for you. Detect intent, pick out important information, automate processes, and deliver personalized responses. This project was started on the premise that there is enough openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities. Speech Recognition library is employed to perform speech to text conversion, Web Search Engine library is employed to urge information from Internet, pytsxs3 library is employed to perform the text to speech, etc. All the tasks are within the textual form which is then converted into an audio signal. A Text-to speech Engine converts the text into pywhatkit

representation, and then it converts the pywhatkit representation to waveforms and multitasking search anything from internet and window which it is give you output.

Intelligent Personal Assistants (IPA) area unit enforced and utilized in operative Systems, net of Things (IOT), and a spread of different systems. Several implementations of IPAs exist these days and corporations like Apple, Google and Microsoft all have their implementations as a serious feature in their operating systems and devices. With the employment of linguistic communication process (NLP), Machine Learning (ML), Artificial Intelligence (AI), and prediction models from these held in applied science (CS), further as theory and techniques from Human-Computer Interaction (HCI), IPAs are getting a lot of intelligent and relevant.

This paper aims to analyze and compare the present major implementations of IPAs so as to work out that implementation is the most developed at this moment in time and is causative to the property way forward for AI. Jarvis could be a system designed to reply to user issued commands to supply convenient management over variety of electronic devices. These devices may be lights, TV's, radios, stereos, etc. The system is going to be designed to figure best among a moderate home with the convenience of a wireless router. The system can take a voice input from a user, match that input to associate in nursing acceptable command among its library of recognized commands or reject the command if it isn't recognized by the system. Associate in Nursing transmit an acceptable message via a router.

The router can then send the operation to the right device therefore the operation may be performed. Assisting users in their tasks is that the main goal of today's personal assistant applications. several such applications square measure being developed, that square measure capable to find the user's habits, abilities, preferences, and goals, even a lot of accurately and predicting the user's actions prior to and perform them while not user's interaction. The assistant agent needs to unceasingly improve its behavior supported previous experiences. Enhancements square measure achieved in personal assistant applications by learning mechanism. Agent's square measure capable of accessing data from databases to guide individuals through completely different tasks, deploying a learning mechanism to accumulate new data on user behavior. Additionally, the resources need to be used in an extremely economical manner resulting in less power consumption.

During this paper we've planned a machine learning approach for learning mechanism of non-public assistant agent.

These types of digital assistants are extremely beneficial to the elderly, the visually and physically challenged, children, and others by ensuring that interacting with the system is no longer a difficult task for humans. Even blind people who are unable to see the system can engage with it by using their voice.

Here are a few of the most important tasks that can be completed with the help of a voice assistant:

1. ChatGpt-4.0 Ai Tool.
2. Mistral7B Ai Tool.
3. Bing Ai Tool.
4. Fake Ai Tool.
5. Image Generation.
6. Create a Presentation.
7. Web Development pages.
8. It Draw Chart.
9. Optimize Code.
10. Error Solving Capability and Store It
11. Summaries/ Translation of Shout Videos.
12. Social Media Search Engine.
13. Search the internet Engine.
14. Multithreading Tasking Support.
15. Setting an alarm.
16. Start any program or utility and Advices.
17. Laptop Sleep Mode, Restart, Shutdown.

These are just a few samples; we will carry out many more tasks in accordance with our requirements. The Voice Assistant that we've developed is suitable for both Windows and Linux users. The voice assistant we've developed is a computer-based gadget built with Python modules and libraries. This assistant is only a basic model that can execute all of the

basic tasks listed above, but modern generation, while true in some ways, still has to be combined with gadget mastery and the internet of things (IoT) for greater results. We used python modules and libraries to create the version, and we used device learning to school our version. Some windows and Linux instructions were also added to the version so that it might operate smoothly on this running computer.

Our approach will essentially work in three modes:

1. Supervised mastering
2. Getting to know each other without being supervised
3. Reward-based learning

Depending on the purpose for which the aid is required for the individual. These can be accomplished with the help of machine learning and deep learning. With the help of Voice Assistant, there may be no need to write down instructions for completing a specific activity over and over again. Once a model is built, it can be utilized an unlimited number of times by an unlimited number of users in the most basic of ways. As a result, with the help of a virtual assistant, we will be able to control numerous things around us on a single platform.

1.2 ARTIFICIAL-INTELLIGENCE:

Artificial intelligence (AI) refers to a computer's or a computer-controlled robot's ability to negotiate tasks normally performed by intelligent beings. The expression is considerably used to relate to a design aimed at creating systems with mortal- suchlike cognitive capacities, analogous as the capability to reason, discern meaning, generalize, and learn from formerly exploits. Since the invention of the digital computer in the 1940s, it has been proved that computers can be programmed to perform extremely complicated jobs with ease, analogous as chancing evidences for fine theorems or playing chess. Despite ongoing increases in computer processing speed and memory capacity, no program can yet match mortal severity across broader fields or in conditioning taking a great deal of common knowledge. still, certain programmes have surpassed the performance situations of mortal specialists and professionals in executing specific tasks, and artificial intelligence in this limited sense can be set up in operations

as different as medical opinion, computer quest machines, and voice or handwriting recognition.

1.3 Virtual assistance:

A virtual adjunct is an independent contractor that works for a customer and provides executive support while working from a position other than the client's office. A virtual adjunct generally works from home, but may pierce important planning accoutrements similar as participated timetables from anywhere. Virtual sidekicks constantly have times of experience working as an executive adjunct or office director. Virtual sidekicks with chops in social media, content operation, blog post jotting, graphic design, and online marketing are chancing new jobs. The demand for educated virtual sidekicks is projected to rise as working from home becomes further accepted by both workers and businesses.

- A virtual assistant is a self-employed professional who provides administrative help to clients from a remote location, usually a home office.
- Scheduling appointments, making phone calls, planning vacations, and managing email accounts are all common responsibilities of a virtual assistant.
- Graphic design, blog authoring, bookkeeping, social media management, and marketing are some of the specialties of virtual assistants.

One benefit of hiring a virtual assistant for an employer is the ability to contract for only the services they require. As small businesses and startups rely on virtual offices to cut expenses, and organizations of all sizes boost their use of the internet for day-to-day operations, virtual assistants have become more popular. A virtual assistant is an independent contractor, therefore a company does not have to give the same benefits or pay the same taxes as a full-time employee.

1.4 J.A.R.V.I.S AI?

- JARVIS is a Personal Desktop AI Assistant which is developed in Python Programming Language.
- It uses Different Technologies to Add New Unique Features.
- It can Automate Tasks with just One Voice Command.
- It is a Desktop Based AI Assistant.

1.5 The Objectives of the Project

1.5.1 Saves Time:

JARVIS is a desktop voice assistant which works on the voice command offered to it, it can do voice searching, voice-activated device control and can let us complete a set of tasks.

1.5.2. Conversational Interaction:

It makes it easier to complete any task as it automatically does 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.

1.5.3 Reactive Nature:

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, the user finds its reaction in an informed and smart way.

1.5.4 Multitasking:

The main application of it can be its multitasking ability. It can ask for continuous instructions one after other until the user “QUIT”, “EXIT” “GOOD BY”, “BY” it.

1.5.5 No Trigger Phase:

It asks for the instruction and listens to the response that is given by user without needing any trigger phase and then only executes the task.

A Security

Security testing mainly focuses on vulnerabilities and risks. As JARVIS is a local desktop application, 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.

B 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.

1.6 Project scope and direction

Helping ourselves is an interesting job. It's much easier to do many other everyday tasks with a single voice command, such as sending an email without typing a word, searching Google and playing music without opening a browser, and opening your favorite IDE. Jarvis differs from other voice assistants in that its desktop specific, users don't need to register to use it, and specifically no internet connection needed to get instructions. we realized that the idea of artificial intelligence in various jobs is to reduce the number of workers and save time. Features of this program include sending emails, reading PDFs, open Whatsapp, opening commands, your favorite IDE, Notepad, etc., opens Google, YouTube etc. in your web browser, provides weather forecast, notifications of your choice. He can make some simple conversations.

1.7 Scope of Project

The term "Jarvis" is often associated with fictional intelligent assistants, like the one featured in the Iron Man movies. In the real world, developing a sophisticated and comprehensive Jarvis-like assistant involves leveraging technologies such as artificial intelligence, natural language processing, machine learning, and more. The scope of Jarvis-like applications is vast and can impact various industries and areas. Here are some potential scopes and applications:

1.7.1 Home Automation:

Jarvis-like systems can control smart home devices, adjust lighting, temperature, and security systems based on user preferences and voice commands.

1.7.2 Personal Assistant:

Assist users with managing tasks, appointments, reminders, emails, and general day-to-day activities.

1.7.3 Healthcare:

Provide health-related information, remind users to take medication, or offer assistance in emergency situations (**Working on Future processing**).

1.7.4 Education:

Assist in learning by providing information on various topics, answering questions, and offering educational resources.

1.7.5 Customer Service:

Enhance customer service by providing automated support, answering common queries, and directing users to relevant resources.

1.7.6 Business Applications:

Assist in business-related tasks such as managing calendars, and providing data analysis by using graph.

1.7.7 Navigation and Travel:

Offer navigation assistance, provide travel recommendations, and help users for planning trips.

1.7.8 Entertainment:

Serve as a source of entertainment by recommending music, movies, or providing news updates.

1.7.9 Accessibility:

Assist individuals with disabilities by offering voice-controlled access to technology and information.

1.7.10 Language Translation:

Facilitate communication by translating languages in real-time.

1.7.11 Financial Assistance:

Provide updates on financial information, manage expenses based Advice.

1.7.12 Real-Time Information Retrieval:

Fetch real-time information such as weather updates, news headlines.

The scope of Jarvis-like systems is continually expanding as advancements in artificial intelligence and related technologies progress. As these systems become more sophisticated, their potential applications across industries and daily life will likely

increase. However, it's important to consider ethical considerations, privacy concerns, and the responsible development and deployment of such technologies.

1.8 Problem Statement

We all know about Personal Desktop AI Assistant and many other virtual assistants designed to help users of Windows, platforms succeed. But surprisingly, there isn't a complete virtual assistant for the Core Windows platform, which is made up of 80% of users. So unstable internet is a big problem for users who may have server issues and places where there is no internet access. The main purpose of creating self-help software (virtual assistant) is to use semantic information found on the web, users create content and provide information from information. The main purpose of Intelligent Personal Desktop AI Assistant is to answer questions that the user may have. This can be done in a business environment, for example a business website with an interactive interface. Intelligent virtual assistants on mobile platforms, your voice to the user "What can I do for you?" It includes call-to-action programs that it asks. and then respond to feedback. A virtual assistant can be a huge time saver.

1.9 Speech Recognition:

Speech recognition, frequently known as speech- totextbook, is the capacity of a machine or program to fete and transfigure spoken words into comprehensible textbook. The vocabulary of rudimentary voice recognition software is confined, and it can only fete words and rulings when pronounced easily. More advanced software can deal with natural speech, multitudinous accentuations, and several languages Computer wisdom, linguistics, and computer engineering exploration are all used in speech recognition. Speech recognition functions are included into numerous current widgets and textbook concentrated program to make using them easier or hands-free. Speech and voice recognition are two distinct technologies that mustn't be confused

- Speech recognition is a technology that recognizes words in spoken language.
- Voice recognition is a biometric technology for relating an existent's voice.

Speech recognition systems process and assay spoken words before converting them to text using computer algorithms.

Following these four processes, a software programmed converts the sound a microphone records into written language that computers and humans can understand

1. Dissect the audio;
2. Break it into corridor;
3. Digitize it into a computer- readable format; and
4. Use an algorithm to match it to the most suitable textbook representation.

CHAPTER 2

SYSTEM DESIGN

We have got a bent to shape our application able to the utilization of gadget voice with the assistance of sapi5 and pyttsx3. Pyttsx3 can also be a text-to-speech conversion library in Python. Now, like distinctive libraries, it works offline and is similar temperament with each Python 2 and three. The Speech Application Programming Interface or SAPI is a diploma API evolved through Microsoft to permit the usage of speech popularity and speech synthesis inside Windows applications. Then we have got a bent to stipulate the talk function to differ this technique to talk the outputs. At that time, we're visiting outline a characteristic to want voice instructions the utilization of the gadget microphone. The foremost function is then made public during which all the competencies of this technique rectangular degree are made public.

To use this program, save it as a Python file (e.g. main.py) and run it. Speak the wake-up keyword ("JARVIS" by default) to activate the program, and then speak a command such as "How are you Jarvis?" or "what time is now?" Jarvis should respond verbally and perform the requested task. Note that this program is a very basic implementation of a virtual assistant and can be expanded upon with more functions and features.

2.1 Hardware Development

- Camera module with good mega pixels.
- System with good specifications.
- 2GB RAM
 - I3 processor system or higher
- 300 Mb ROM or higher
- Window 7,8,10,11 are Supported

2.2 Libraries Development

Following are the libraries which needs to be installed for the program to run.

- BeautifulSoup4==4.12.
- Blinker==1.7.0
- Click==8.1.7
- Browser
- Cookie3==0.19.1
- Bs4==0.0.1
- Certifi==2023.11.17
- Cffi==1.16.0
- Charset
- Normalizer==3.3.2
- Waitress==2.1.2
- WebDriver-Manager==4.0.1
- Websockets==12.0
- Werkzeug==3.0.1
- Win32-Setctime==1.1.0
- Wsproto==1.2.0
- Xlsxwriter==3.1.9
- Yarl==1.9.4
- Youtube-Transcript-Api==0.6.2
- Colorama==0.4.6
- Comtypes==1.2.1
- Curl-Cffi==0.5.10
- Duckduckgo_Search==4.1.1
- Exceptiongroup==1.2.0
- Fastapi==0.108.0
- Filelock==3.13.1
- Flask==3.0.0
- Flask-Cors==4.0.0
- Frozenlist==1.4.1
- Fsspec==2023.12.2
- G4f==0.1.9.8
- Html2text==2020.1.16
- Huggingface-Hub==0.20.2
- Idna==3.6
- Itsdangerous==2.1.2
- Jinja2==3.1.2
- Js2py==0.74
- keyboard==0.13.5

- Loguru==0.7.2
- Lxml==5.0.1
- Lz4==4.3.3
- MarkupSafe==2.1.3
- Mmh3==4.0.1
- Mtranslate==1.8
- Multidict==6.0.4
- Nest-Ayncio==1.5.8
- Numpy==1.26.3
- Outcome==1.3.0.Post0
- Packaging==23.2
- Pillow==10.2.0
- Platformdirs==4.1.0
- Psutil==5.9.7
- Py-Arkose-Generator==0.0.0.2
- Pycparser==2.21
- Pycryptodome==3.19.1
- Pycryptodomex==3.19.1
- Pydantic==2.5.3
- Pydantic_Core==2.14.6
- Pyexecjs==1.5.1
- Pygame==2.5.2
- Pygetwindow==0.0.9
- Pyjsparser==2.7.1
- Pyperclip==1.8.2
- Pypiwin32==223
- Pyrect==0.2.0
- Pysocks==1.7.1
- Python-Dotenv==1.0.0
- Python-Pptx==0.6.23
- Pyttsx3==2.90
- Pywin32==306
- Pyyaml==6.0.1
- Requests==2.31.0
- Selenium==4.16.0
- Six==1.16.0
- Sniffio==1.3.0
- Sortedcontainers==2.4.0
- Sounddevice==0.4.6
- Soupsieve==2.5
- Starlette==0.32.0.Post1
- Tqdm==4.66.1
- Trio==0.23.2
- Trio-Websocket==0.11.1
- Typing_Extensions==4.9.0
- Tzdata==2023.4
- Tzlocal==5.2
- Undetected-Chromedriver==3.5.4
- Urllib3==2.1.0
- Uvicorn==0.25.0

2.2.1 Speechrecognition==3.10.0:

Library for performing speech recognition, with support for several engines and APIs, online and offline.

2.2.2 Openai==0.27.8:

Python client library for the OpenAI API.

2.2.3 Nltk==3.7:

The Natural Language Toolkit (NLTK) is a Python package for natural language processing. NLTK requires Python 3.7,

2.2.4 Playsounds==1.3.0:

Pure Python, cross platform, single function module with no dependencies for playing sounds.

2.2.5 opencv-python== 4.8.0.76:

Wrapper package for OpenCV python bindings.

2.2.6 Pywhatkit==5.4:

PyWhatKit is a Simple and Powerful WhatsApp Automation Library with many useful Features.

2.2.7 Googletrans==3.1.0a0:

Free Google Translate API for Python. Translates totally free of charge.

2.2.8 Flask==3.2.3:

Flask is a lightweight Python web framework that provides useful tools and features for creating web applications in the Python Language.

2.2.9 Google-auth==2.22:

This library simplifies using Google's various server-to-server authentication mechanisms to access Google APIs.

2.2.10 Webdriver-manager==4.0.0:

Library provides the way to automatically manage drivers for different browsers.

2.2.11 GoogleImageScraper==2.3.5:

This is a library for retrieving urls and downloading images from Google Images.

2.2.12 PyPDF2==3.0.1:

A pure-python PDF library capable of splitting, merging, cropping, and transforming PDF files.

2.2.13 PyGetwindow==0.0.9:

A simple, cross-platform module for obtaining GUI information on application's windows.

2.2.14 PyAutoGUI==0.9.54:

PyAutoGUI lets Python control the mouse and keyboard, and other GUI automation tasks.

2.2.15 Torch==1.12.0:

Tensors and Dynamic neural networks in Python with strong GPU acceleration.

2.2.16 Pyaudio==0.2.13:

Cross-platform audio I/O with PortAudio.

2.2.17 BeautifulSoup==4.12.0:

Beautiful Soup is a Python library for pulling data out of HTML and XML files

2.2.18 blinker==1.7.0:

Fast, simple object-to-object and broadcast signaling.

2.2.19 browser-cookie3==0.19.1:

Loads cookies from your browser into a cookiejar object so can download with urllib and other libraries the same content you see in the web browser.

2.2.20 bs4==0.0.1:

Dummy package for Beautiful Soup (beautifulsoup4)

2.2.21 certifi==2023.11.17:

Python package for providing Mozilla's CA Bundle

2.2.22 cffi==1.16.0:

Foreign Function Interface for Python calling C code.

2.2.23 charset-normalizer==3.3.2:

The Real First Universal Charset Detector. Open, modern and actively maintained alternative to Chardet.

2.2.24 click==8.1.7:

Composable command line interface toolkit

2.2.25 colorama==0.4.6:

Cross-platform colored terminal text.

2.2.26 comtypes==1.2.1:

Pure Python COM package

2.2.27 curl-ffi==0.5.10:

libcurl ffi bindings for Python, with impersonation support

2.2.28 duckduckgo_search==4.1.1:

Search for words, documents, images, videos, news, maps and text translation using the DuckDuckGo.com search engine. Downloading files and images to a local hard drive.

2.2.29 exceptiongroup==1.2.0:

Changed handling of exceptions in exception group handler callbacks to not wrap a single exception in an exception group,

2.2.30 fastapi==0.108.0:

i'm using FastAPI a ton these days. [...] I'm actually planning to use it for all of my team's ML services at Microsoft. Some of them are getting integrated into the core Windows product and some Office products.

2.2.31 filelock==3.13.1:

This package contains a single module, which implements a platform independent file lock in Python, which provides a simple way of inter-process communication

2.2.32 Flask==3.0.0:

Flask provides configuration and conventions, with sensible defaults, to get started. This section of the documentation explains the different parts of the Flask framework and how they can be used, customized, and extended. Beyond Flask itself, look for community-maintained extensions to add even more functionality.

2.2.33 Flask-Cors==4.0.0:

By default, submission of cookies across domains is disabled due to the security implications.

2.2.34 g4f==0.1.9.8:

G4F, short for gpt4free, is an innovative artificial intelligence software package on a mission to expand boundaries in AI-powered conversations. Leveraging state-of-the-art models like GPT-3.5 and GPT-4, G4F offers an array of features to facilitate seamless interactive AI experiences.

2.2.35 huggingface-hub==0.20.2

A concurrency issue when using userdata.get to retrieve HF_TOKEN token led to deadlocks when downloading files in parallel.

2.2.36 idna==3.6:

This library provides one such mapping that was developed by the Unicode Consortium.

2.2.37 itsdangerous==2.1.2:

It's possible to customize how data is serialized. Data is compressed as needed. A timestamp can be added and verified automatically while loading a token.

2.2.38 keyboard==0.13.5:

Take full control of your keyboard with this small Python library. Hook global events, register hotkeys, simulate key presses and much more.

2.2.39 MarkupSafe==2.1.3:

MarkupSafe implements a text object that escapes characters so it is safe to use in HTML and XML.

2.2.40 multidict==6.0.4:

Multidict is dict-like collection of key-value pairs where key might occur more than once in the container.

2.2.41 nest-asyncio==1.5.8:

This presents a practical problem: When in an environment where the event loop is already running it's impossible to run tasks and wait for the result.

2.2.42 numpy==1.26.3:

The NumPy leadership has made a strong commitment to creating an open, inclusive, and positive community.

2.2.43 packaging==23.2:

This library provides utilities that implement the interoperability specifications which have clearly one correct behaviour

2.2.44 pillow==10.2.0:

The core image library is designed for fast access to data stored in a few basic pixel formats.

2.2.45 platformdirs==4.1.0:

When writing desktop application, finding the right location to store user data and configuration varies per platform.

2.2.46 pygame==2.5.2:

pygame popular libraries to abstract the most common functions, making writing these programs a more intuitive task

2.2.47 python-pptx==0.6.23:

python-pptx is a Python library for creating, reading, and updating PowerPoint (.pptx) files.

2.2.48 pyttsx3==2.90:

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries,

2.2.49 requests==2.31.0:

Requests allows you to send HTTP/1.1 requests extremely easily. There's no need to manually add query strings to your URLs, or to form-encode your PUT & POST data but nowadays, just use the json method!

2.2.50 selenium==4.16.0:

Selenium is an umbrella project encapsulating a variety of tools and libraries enabling web browser automation.

2.2.51 sounddevice==0.4.6:

this Python module provides bindings for the PortAudio library and a few convenience functions to play and record NumPy arrays containing audio signals.

2.2.52 undetected-chromedriver==3.5.4:

Selenium.webdriver.Chrome replacement with compatibility for Brave, and other Chromium based browsers.

2.2.53 urllib3==2.1.0:

urllib3 is a powerful, user-friendly HTTP client for Python.

2.2.54 webdriver-manager==4.0.1:

Library provides the way to automatically manage drivers for different browsers

2.2.55 websockets==12.0:

websockets is a library for building WebSocket servers and clients in Python with a focus on correctness, simplicity, robustness, and performance.

2.2.56 Werkzeug==3.0.1:

Werkzeug is a comprehensive WSGI web application library.

2.2.57 youtube-transcript-api==0.6.2:

It also works for automatically generated subtitles, supports translating subtitles and it does not require a headless browser, like other selenium based solutions do.

2.2.58 Click 8.1.3:

Click is a Python package for creating beautiful command line interfaces in a composable way with as little code as necessary.

2.2.59 aiohttp==3.9.1:

Async http client/server framework (asyncio)

2.2.60 aiosignal==1.3.1:

aiosignal: a list of registered asynchronous callbacks

2.2.61 annotated-types==0.6.0:

Reusable constraint types to use with typing.Annotated

2.2.62 anyio==4.2.0:

High level compatibility layer for multiple asynchronous event loop implementations

CHAPTER 3

SOFTWARE DEVELOPMENT

Creating Ai Personal Assistant -like voice assistant involves several steps, and it typically requires knowledge in areas such as natural language processing (NLP), speech recognition, and software development. Below is a high-level overview of the process:

3.1 Define Features and Requirements:

Clearly outline the features you want your Jarvis voice assistant to have. For example, do you want it to respond to voice commands, perform tasks, provide information, etc.?

3.2 Choose a Development Platform:

Select a programming language and platform for your voice assistant. Python is commonly used for AI and NLP tasks, but other languages like JavaScript, can also be suitable.

3.3 Speech Recognition:

Implement a speech recognition system to convert spoken words into text. There are various libraries and APIs available for this, such as Google Cloud Speech-to-Text, Microsoft Azure Speech, or open-source options like CMU Sphinx or Mozilla Deep Speech.

3.4 Natural Language Processing (NLP):

Integrate a natural language processing system to understand and interpret user commands. Popular NLP frameworks include spaCy, NLTK, or the Hugging Face Transformers library.

3.5 Task Execution:

Develop the logic for your assistant to perform specific tasks based on user commands. This may involve integrating with other APIs or services for actions like setting reminders, sending messages, or retrieving information.

3.6 Text-to-Speech (TTS):

Implement a text-to-speech system to convert the assistant's responses into natural-sounding speech. Libraries like gTTS (Google Text-to-Speech) or Amazon Polly can be useful.

3.7 User Interface (UI):

If your Jarvis voice assistant will have a graphical user interface, design and implement it. This could be a desktop application, a Desktop based Interface, or even integration with existing platforms like Slack or Discord.

3.8 Continuous Improvement:

Implement mechanisms for continuous learning and improvement. This could involve regularly updating the NLP models, expanding the range of supported commands, or refining the user experience.

3.9 Testing:

Thoroughly test your voice assistant to ensure its accuracy, reliability, and user-friendliness. This includes testing different accents, languages, and potential user scenarios.

3.10 Deployment:

Once satisfied with the testing, deploy your Jarvis voice assistant. This could involve packaging it as a standalone application, a web service, or integrating it into existing platforms.

Remember that building a sophisticated voice assistant like Jarvis involves a combination of various technologies and may require a good understanding of machine learning and artificial intelligence concepts. It's also essential to consider privacy and security aspects, especially when dealing with user data and voice recordings.

CHAPTER 4

METHODOLOGY

Python is a high-level interpreted programming language grounded in Object-Oriented Programming (OOP), emphasizing Rapid Application Development (RAD). It serves as a potent and efficient tool for coding simplicity. Python's distinctive feature lies in its ability to express equivalent logic using only a fraction (up to 1/5) of the code compared to other OOP languages. Widely applicable, Python's rising popularity is closely linked to its role in Artificial Intelligence (AI), Machine Learning (ML), Natural Language Processing, and research papers, enabling its involvement in intricate and captivating processes. Python offers a diverse array of libraries catering to various needs. For JARVIS, specific function libraries include speech recognition for voice interaction, Pyttsx3 for text-to-speech, Selenium for network automation, among others. Python's efficiency is noteworthy, especially for smaller instances where performance is typically not a concern. To enhance Python code efficiency, a common approach involves identifying and optimizing the part that consumes the most time using low-level code. This strategy streamlines programming, making it less time-consuming and more efficient compared to exclusively using a low-level language, as it allows for increased optimization efforts.

We have got a bent to shape our application able to the utilization of gadget voice with the assistance of sapi5 and pyttsx3. Pyttsx3 can also be a text-to-speech conversion library in Python. Now, like distinctive libraries, it works offline and is similar temperament with each Python 2 and Python 3. The Speech Application Programming Interface or SAPI is a diploma API evolved through Microsoft to permit the usage of speech popularity and speech synthesis inside Windows applications. Then we have got a bent to stipulate the talk function to differ this technique to talk the outputs. At that time, we're visiting outline a characteristic to want voice instructions the utilization of the gadget microphone. The foremost function is then made public during which all the competencies of this technique rectangular degree are made public.

The planned system is meant to possess the following functionality:

The Jarvis asked the user for input and keeps listening for orders. The time for a hearing is visiting be set up in step with the user's control

If the assistant fails to grasp the command its visiting keep asking the user to repeat the command once again} and yet again.

This assistant is visiting be bespoken to possess either male or female voice in step with user's demand.

The current version of the assistant supports choices like Checking weather updates, deed and checking emails, Searching Wikipedia, Stream music, Open applications, checking dates and times, taking notes, show notes, Open YouTube, etc.

1. **Open Window Applications:** Jarvis can control Window Application open like: Notepad, cmd, music. Paint, or Paint 3D any Browser, and also restart pc and shutdown pc and etc.
2. **Multitasking and Search engine:** Jarvis can do at a time multitasking such as : at a time he can create a presentation power point and playing music, and at a time give you response about multitasking updates etc.
3. **Financial Adviser:** Jarvis gives you a advice which means that this prodect is best for you or not you can buy it and also Search for you upon your financial requirements
4. **Recommendation System:** Jarvis give you a recommendation about tours or trips and also gives your health advice (health advice future working)
5. **Control Internet Google Chrome:** Jarvis can control your chrome like: open new tab, close last tab, check download file or more.
6. **Creating Presentation:** Jarvis can create a presentation by following your voce and take topic name and Jarvis will be create a presentation in few mints and also save it at location file that was gives to you.
7. **Image Generations:** Jarvis can create a image regarding your requirements and also save location file that was gives to you and show image he is creating image such as: Bing Creating 3D image and Bing ai also Automate backend know.
8. **YouTube video control:** Jarvis can control your YouTube like: play video, pause video, volume increase and decrease and more about YouTube.
9. **Streaming Music:** The user can command Jarvis to play a music track and it's going to execute a command and search into it from the song Folder.

10. **Read the Latest News from Headlines:** Jarvis will examine out latest headlines from the knowledge retailers of the required topics you care about or need information.
11. **Keep Tabs on the Traffic & the Weather:** Jarvis can research the weather forecast or alert you if there is an accident that will delay your morning journey.
12. **Set Reminders/timers:** You'll be able to tell Jarvis to wake you up daily morning at 4 a.m.
13. **Answer the Following Questions:** Jarvis can look up simple information, solve mathematical problems, or tell you a joke.



4.1 Speech Recognition Model:

The system uses Google's online speech recognition system for converting speech input to text. The speech input Users can obtain texts from the special corpora organized on the computer network server at the information center from the microphone is temporarily stored in the system which is then sent to Google cloud for speech recognition. The equivalent text is then received and fed to the central processor.

This module converts spoken words into text through speech recognition algorithms. It analyzes and processes audio input, identifying words and phrases spoken by the user. This technology enables applications and devices to understand and respond to voice commands, making it possible to interact with them through spoken language. Speech Recognition modules are widely used in virtual assistants, voice-controlled systems,

Transcription services, and various other applications

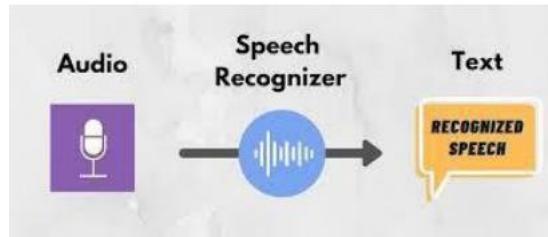


Figure Speech Recognition Model (4.1)

4.2 Natural Language Processing (NLP):

The NLP module interprets the user's input by analyzing the text and extracting its meaning. It applies techniques like language understanding, entity recognition, intent classification, and sentiment analysis to understand user queries. It employs various techniques, such as machine learning and linguistic analysis, to process and derive meaning from text or speech data. NLP modules are used in applications like chatbots, language translation, sentiment analysis, information extraction, and text classification.

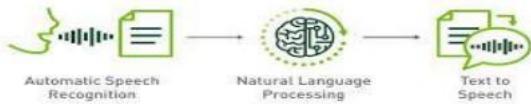


Figure Natural Language Processing (4.2)

4.3 Skill/Action Integration Module:

This module allows integration with third-party applications and services, extending the functionality of the voice assistant. It enables users to perform tasks like ordering food, booking a ride, checking the weather, or accessing specific information from external sources

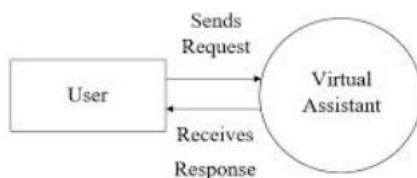


Figure skill action integration (4.3)

4.4 Voice User Interface (VUI):

The VUI module defines the design and interaction patterns for users to communicate with the voice assistant. It includes voice commands, prompts, and feedback mechanisms to facilitate intuitive and user-friendly interactions. A Voice User Interface (VUI) is a technology that allows users to interact with devices or systems using spoken language as the input and auditory feedback as the output. It enables hands-free and eyes-free interaction, making it convenient and accessible for users.

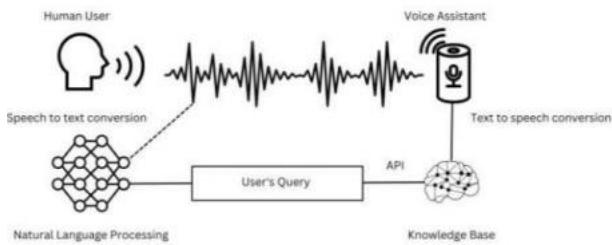


Figure VUI(4.4)

4.5 Text-To-Speech Model:

The TTS module converts the text responses generated by the system into spoken words. It utilizes speech synthesis techniques to generate natural and human-like voices for the voice assistant's output. A Text-to-Speech (TTS) module is a software component that converts written text into spoken audio. It utilizes synthetic speech technology to generate human-like speech output. The TTS module analyzes the text input and applies linguistic and phonetic rules to produce natural-sounding speech.

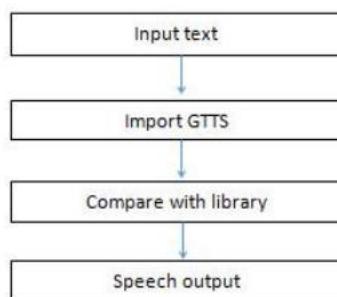


Figure Text-To-Speech Model: (4.5)

4.6 Python Backend:

The python backend parses the voice recognition module's output to determine whether the command or speech output is an API Call, Context Extraction, or System Call. The

output is then sent back to the python backend to provide the user with the desired results.

4.7 API Calls:

The acronym for Application Programming Interface (API) is Application Programming Interface. An API is a software interface that allows two apps to communicate with one another. To put it another way, an API is the messenger that sends your request to the provider and then returns the result.

4.8 Context Extraction:

Context extraction (CE) is the process of obtaining structured data from machine-readable materials that are unstructured or semi-structured. The majority of the time, this activity entails using natural language processing to process human language texts (NLP). TEST RESULTS for context extraction can be seen in recent activities in multimedia document processing, such as automatic annotation and content extraction from images/audio/video.

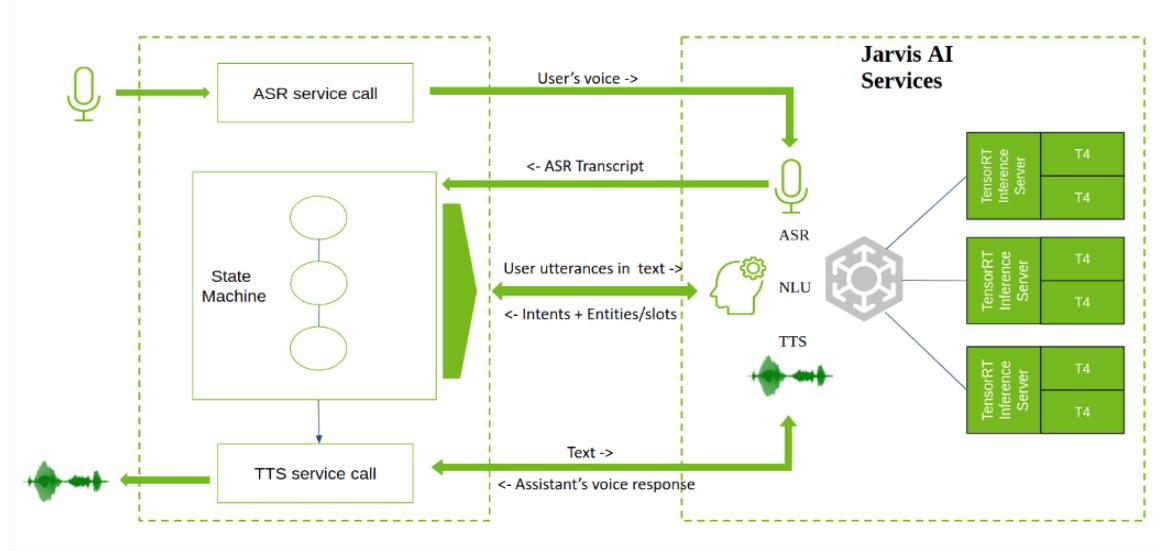
4.9 System Calls:

The mechanism through which a computer software requests a service from the kernel of the operating system on which it is running is known as a system call. Hardware-related services (for example, accessing a hard disc drive), the creation and execution of new processes, and communication with core kernel services such as process scheduling are all examples of this. A process's interface with the operating system is provided by system calls.

4.10 Graphical User interface-QtDesigner

QtDesigner is a program made by the makers of Qt and PyQt that allows you to build GUI applications with drag and drop. Once you have built and saved the application you can run a command line tool that will turn your GUI into python code. This makes it very easy to create GUI's, especially simple ones.

Figure Methodology (4.0)



4.11 Hierarchy manner of the face Recognition

After a successful retrieval of facial images into the respective files created to aid the next process of pumping the face into the recognizer for the train process. Like the following format:

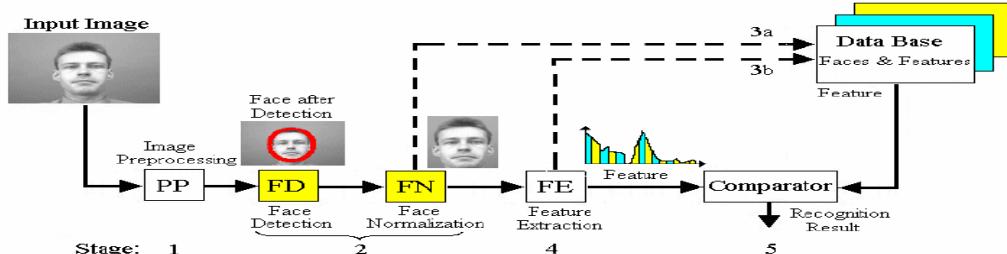


Figure Hierarchy (4.11)

4.12 Agile methodology

4.12.1 Plan:

Agile planning for face recognition and Clap login system is a project of advancement in the login system of user with an incremental, iterative approach.

4.12.2 Design:

Design of Personal Desktop AI Assistant is based of python language and It is designed in way it collects API and Selenium Automation Tools

4.12.3 Develop:

Jarvis Voice-Based-Assistant project is developed in python language and used multiple libraries These libraries make this project quicker and faster.

4.12.4 Test:

An agile methodology for Jarvis Voice-Based-Assistant is an approach to meet requirements and solutions.

4.12.5 Deploy:

Deployment is not part of scrum, because scrum is a framework for developing products, not for delivering them. This means, the deployment in production happens whenever the product owner feels like it. Usually he will decide this after the review, because then the product increment is transparent to him

4.12.6 Review:

In Agile project management, a sprint review is an informal meeting held at the end of a sprint, in which the team shows what was accomplished during this period.

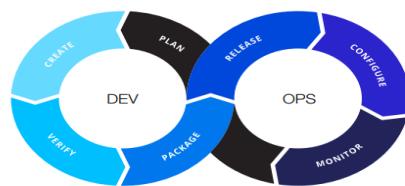
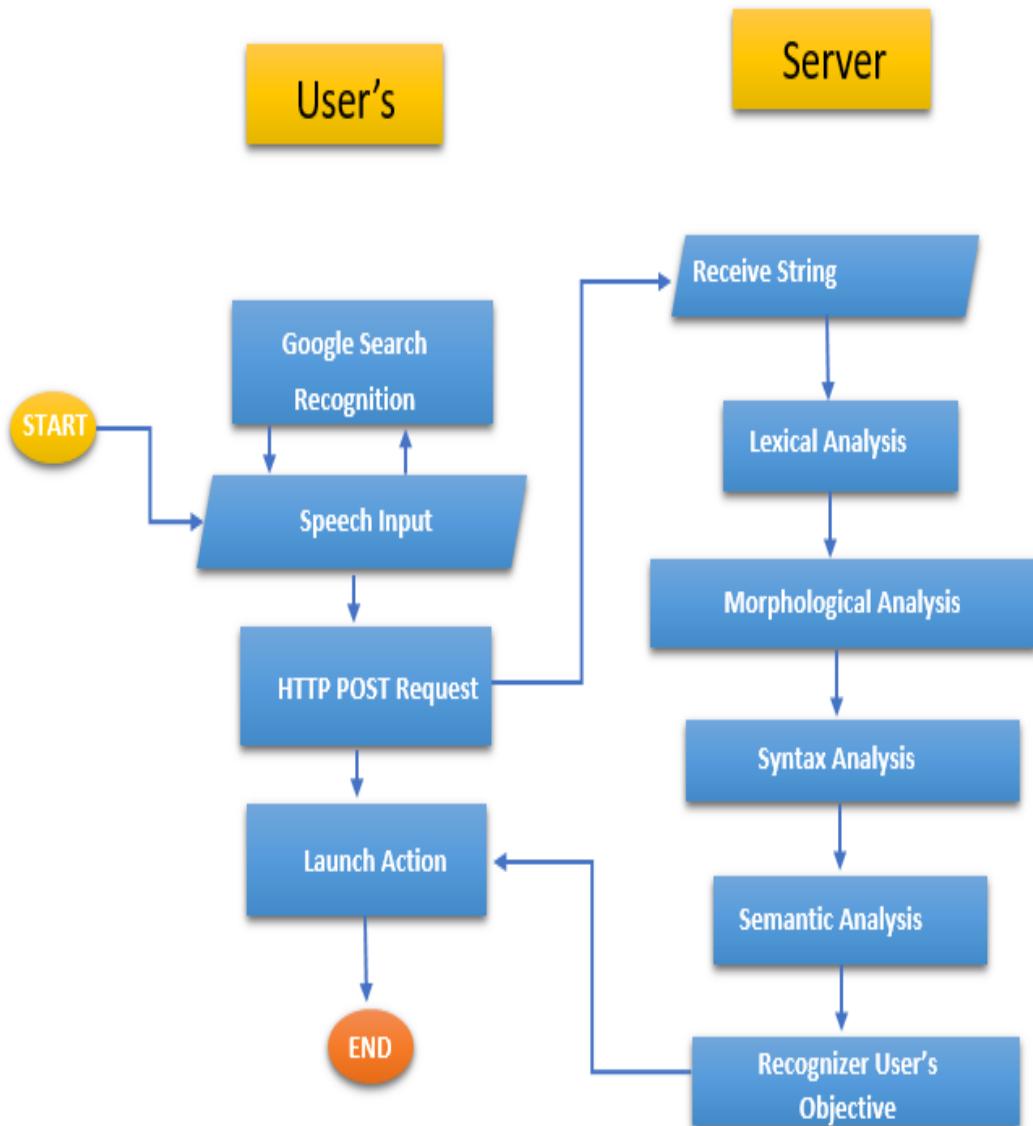


Figure Agile Methodology (4.12)

Project Model



CHAPTER 5

SOFTWARE SYSTEM ANALYSIS

Virtual assistants are typically cloud-based programs that require internet-connected devices and/or applications to function". The technologies that power virtual assistants require vast amounts of knowledge, powering the platforms, as well as machine learning, language communication processes, and speech recognition arena. There are dedicated devices to provide virtual assistance. The most stylish on the market from Amazon, Google and Microsoft having Alexa, Google Siri and Cortana as AI voice assistants respectively given by each company. AI voice assistants often perform simple tasks for end users, such as adding tasks to the calendar; provide information that can usually be searched in an Internet browser; or control and check the health of sensitive devices in the home, send emails, setting up of alarms, getting weather reports, can give your location, perform some basic mathematical calculations, check news, start the music, and open different websites like stack overflow, you tube, Facebook etc.

5.1 Training Model

- With the help of NN as neural network and NLP as natural language processing, create a brain of the model.
- And, with the help of machine learning modules and Deep Learning modules built emotions in the model and dataset to help the model in training.

5.2 Neural Networks

"NN reflects the behavior of the human brain, enabling computer programs to recognize patterns and solve common problems in artificial intelligence and other AI applications". An Artificial Neural Networks (ANNs) consists of a layer of nodes, including an input layer, one or more hidden layers, and an output layer. Each node is connected to another node, with weights and thresholds associated with it. If the output of an individual node is greater than the specified threshold, that node wakes up and sends data to the next layer of the network. Otherwise, the

data will not be sent to the next layer of the network. The network relies on training datasets to learn and improve accuracy over time.

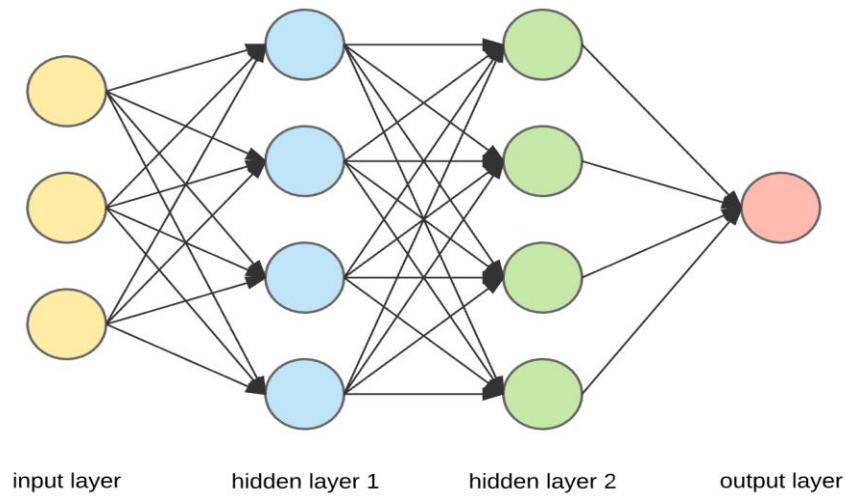


Figure Natural Network (5.2)

Think of each node as a unique linear regression model consisting of input data, weights, bias as thresholds, and outputs.

$$ziri + th = z1r1 + z2r2 + z3r3 + th$$

$$\text{output} = g(r) = 1 \text{ if } z1x1 + c \geq 0; 0$$

$$\text{if } z1x1 + c < 0$$

When an input layer is specified, weight area units are assigned. These weights make it easy to see the importance of a particular variable. Large variables pay a lot of attention to the output for different inputs. Then the units of all input areas are incremented and summed with different weights. Then the output is passed. When this output exceeds a certain threshold, the node is triggered and knowledge is propagated to future layers in the network. This makes the exit of one node the entrance of future nodes. This method of passing knowledge from one layer to the future layer defines this neural network as a feed forward network.

5.3 Natural Language Processing

NLP implies "Natural Language Processing", which is part of the user language of computer science and one of the applications of artificial intelligence. This is a technology used by machines to understand, analyze, manipulate, and interpret human language. This helps developers organize their knowledge to perform tasks such as translation, book reading, speech recognition, and topic segmentation.

1. NLP helps users ask questions about a topic and get a direct answer within seconds.
2. NLP provides accurate answers to your questions. That is, it does not provide unnecessary and unnecessary information.
3. NLP helps computers communicate with people in that language.
4. Most IT industries use natural language processing to improve the efficiency and accuracy of the documentation process and identify information from large databases.

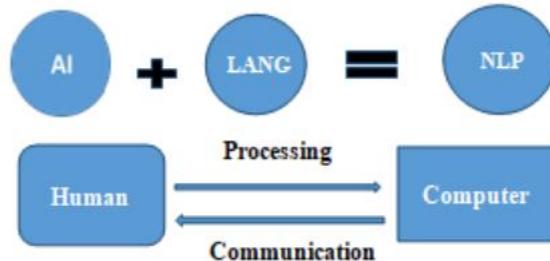


Figure Natural Language Processing (5.3)

5.4 Speech Recognition System

The speech recognition system is the core of the voice application system, which is capable of understanding the voice input given by the user, and at the same time operating the applications efficiently and generating voice feedback to the user. This system is an important component for users as a gateway to use their voice as an input component. (Figure 3.3) .

In a word, in order to clearly recognize the user's speech command and get a response from the system, we should consider that the speech recognition system contains the whole process by which the application system directs the generation. Voice signal to text data and some important meanings, forms of speech.

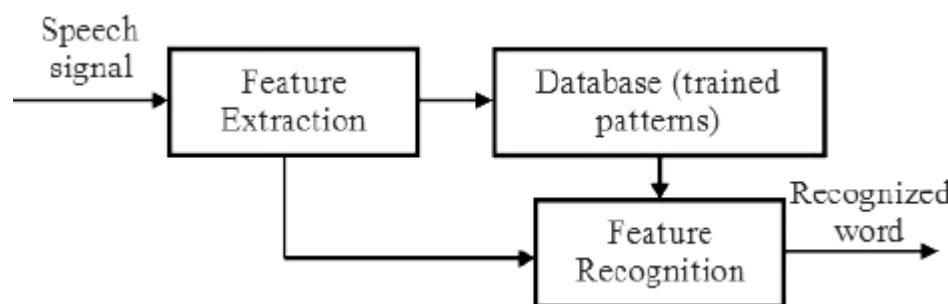


Figure Speech Recognition System (5.5)

5.5 PyTorch - Machine Learning Library

PyTorch allows developers to teach neural network models in a very distributed way. It uses Python's native support for user-machine communication and asynchronous execution of aggregate operations to provide optimized performance in analytics and production environments.

- torch.cuda: supports CUDA tensor types that implement the same function as CPU tensors.
- torch.nn: this package provides many more classes and modules to implement and train the neural network.
- torch.utilis.data: this package is mainly used for creating datasets.

5.6 Linear Regression Concept

This algorithm is a method of finding a linear relationship between a dependent variable and an independent variable by minimizing the distance. This is a supervised algorithm. Here, we use a machine learning supervised algorithmic approach to categorize individual categories. Using this algorithm, we created a voice assistant model that allows users to predict relationships between dependent and independent entities.

$$D = p + qI$$

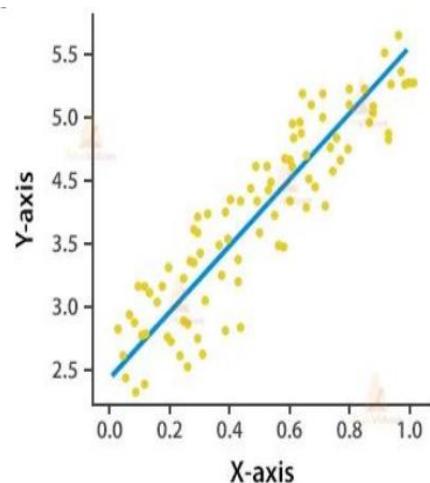
$$p = \frac{1}{n} \sum_{i=1}^n (D_i - \bar{D})(I_i - \bar{I})$$
$$q = \frac{1}{n} \sum_{i=1}^n (I_i - \bar{I})^2$$

$$q = \frac{n(\bar{D}\bar{I}) - (\bar{D})(\bar{I})^2}{n(\bar{I}^2) - (\bar{I})^2}$$

I is the independent variable

D is the dependent variable

p is intercept and q is slope of line here



5.7 System Architecture

Initially the condition is that if the Jarvis voice assistant is active or not, if it is active then it asks for the user input otherwise make Jarvis active (make it on). Then user provides the input in the form of speech or text, after that if the input provided is in text then it goes for the action to be taken or the skills to be executed, else if the input is in speech then it uses the speech recognition feature and converts it into text and goes for the action.

Now after the proceedings if the skills to be executed are adequate to Jarvis then it gives a positive response to the user in form of speech and then executes the commands for operations, hence gives the console output and speech. On the other hand if the skills to be executed are not adequate or inappropriate to Jarvis it gives a negative response and executes no further commands to give Terminal output.

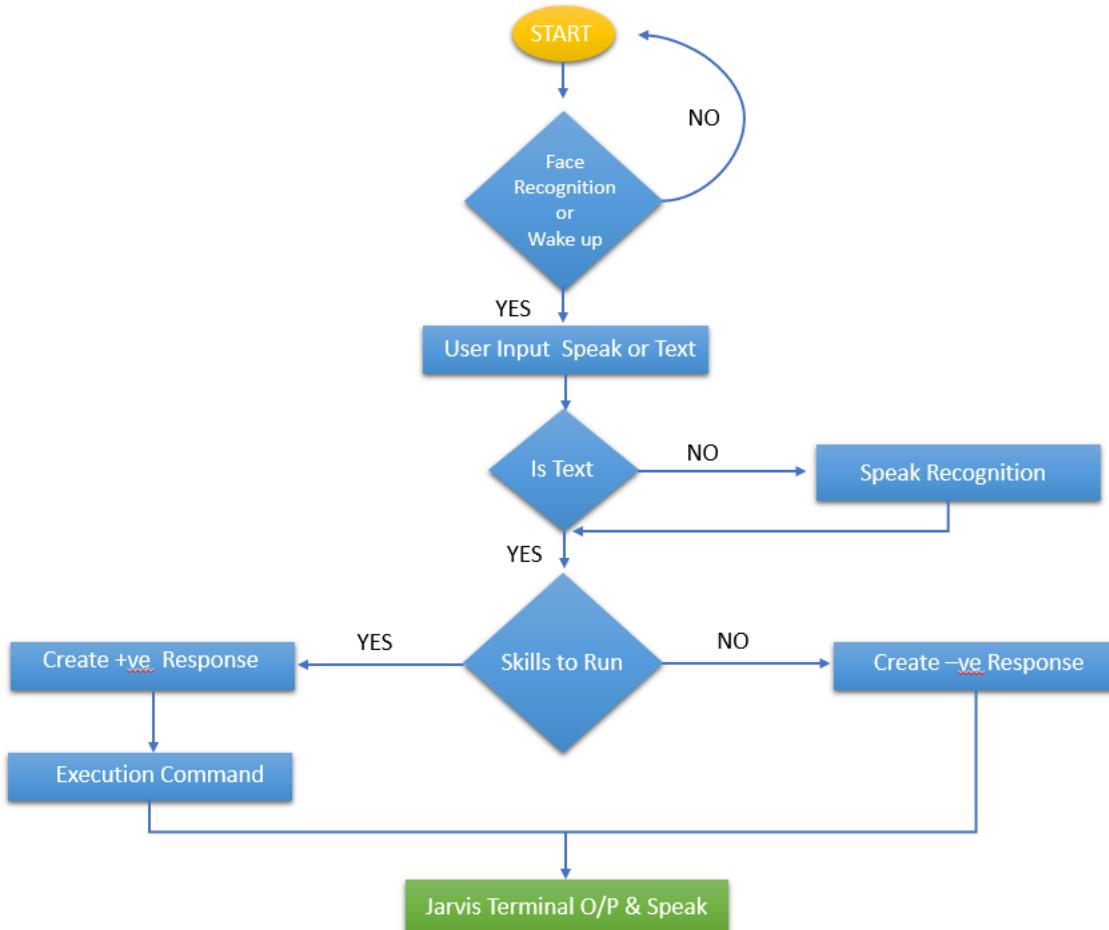


Figure System Architecture (5.7)

5.8 Sequence Diagram

The user sends command to the voice assistant Jarvis then it forwards it to Interpreter i.e. speech recognition feature here and then is directed perform the specific task, after the processing in task model Jarvis executes the task and give the response or feedback to the user

- If after the processing at task model there is some missing information then Jarvis asks for that information, takes the input again, gathers all information and follow the same process as detailed above.

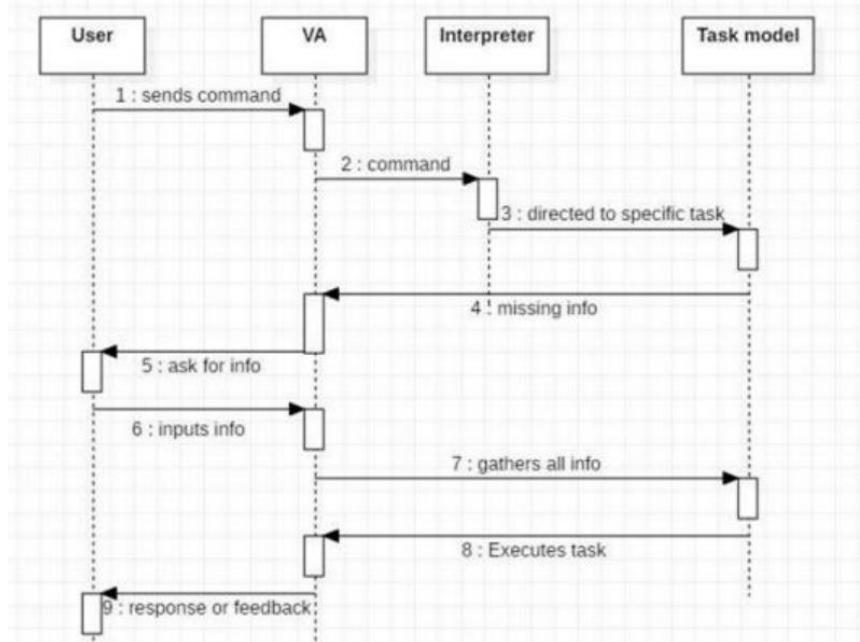


Figure Sequence Diagram (5.8)

5.9 Use Case Diagram

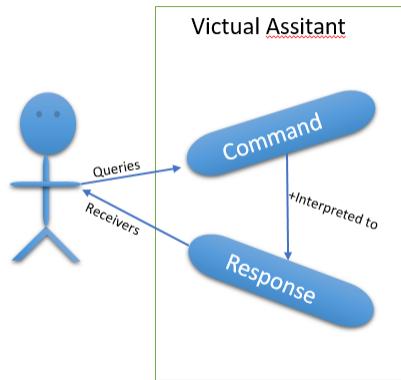


Figure Use Case Diagram (5.9)

5.10 Activity Diagram

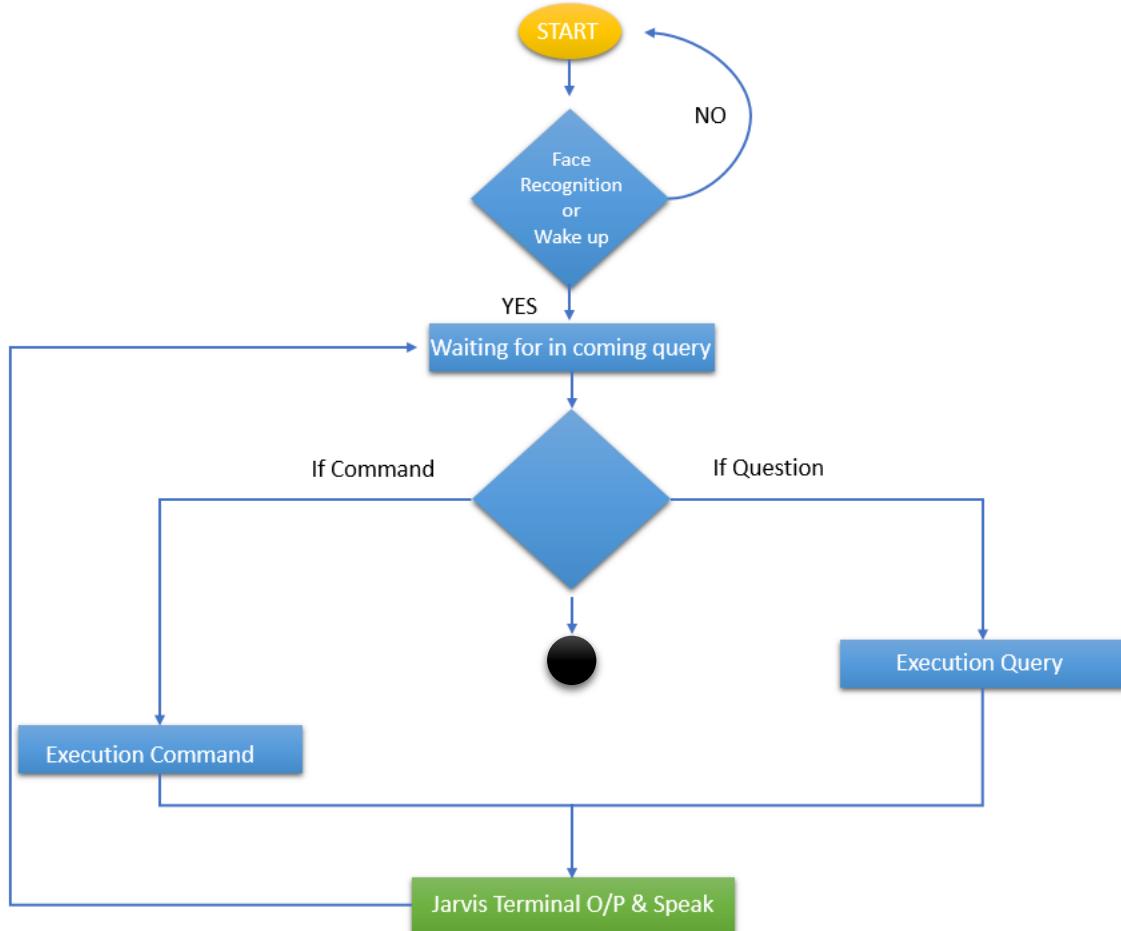


Figure Activity Diagram (5.10)

CHAPTER 6

MODELING AND ANALYSIS

6.1 Selenium Automation Tool

Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. It is quite similar to HP Quick Test Pro (QTP now UFT) only that Selenium focuses on automating web-based applications. Testing done using Selenium tool is usually referred to as Selenium 1, 2 testing. Selenium is not just a single tool but a suite of software, each catering to different testing needs of an organization. The entire Selenium Tool Suite is comprised of four components:

- Selenium IDE, a Google Chrome add-on that you can only use in creating relatively simple test cases and test suites.
- Selenium Remote Control, also known as Selenium 1, which is the first Selenium tool that allowed users to use programming languages in creating complex tests.
- Selenium Grid is also a tool that is used with Selenium RC to execute parallel tests across different browsers and operating systems.

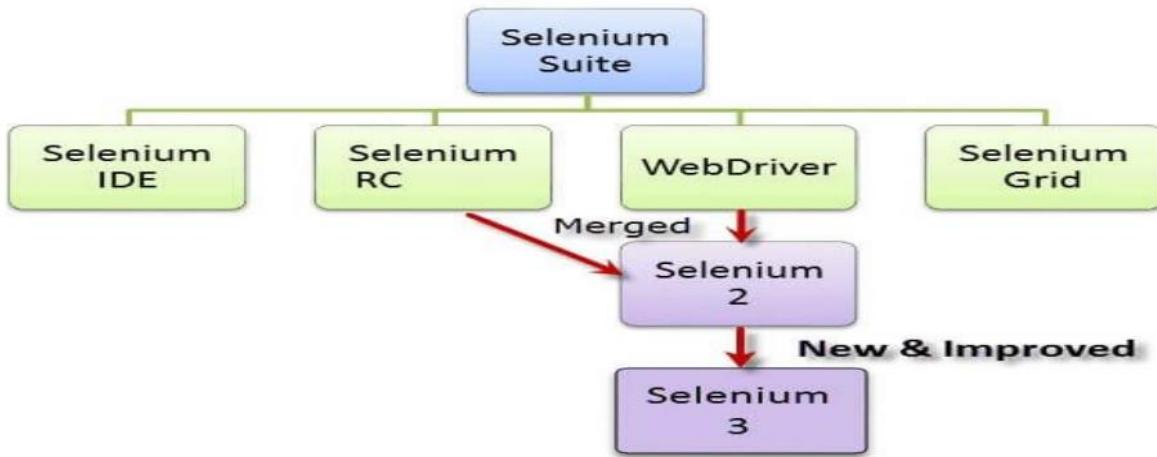


Figure Selenium Automation Tool (6.1)

6.1.1 Selenium Web Driver:

The Web Driver proves to be better than Selenium IDE and Selenium RC in many aspects. It implements a more modern and stable approach in automating the browser's actions. Web Driver, unlike Selenium RC, does not rely on JavaScript for Selenium Automation Testing. It controls the browser by directly communicating with it.

The supported languages are the same as those in Selenium RC.

- Enables simultaneous running of tests in multiple browsers and environments.
- Saves time enormously.
- Utilizes the hub-and-nodes concept. The hub acts as a central source of Selenium commands to each node connected to it.

The entire Selenium Software Testing Suite is comprised of four components:

- Selenium IDE, a Firefox and chrome add-on that you can only use in creating relatively simple test cases and test suites.
- Selenium Remote Control, also known as Selenium 1, is the first Selenium tool that allowed users to use programming languages in creating complex tests.
- Web Driver, is the newer breakthrough that allows your test scripts to communicate directly to the browser, thereby controlling it from the OS level.
- Selenium Grid is also a tool that is used with Selenium RC to execute parallel tests across different browsers and operating systems.

Selenium RC and Web Driver was merged to form Selenium 2.

Selenium is more advantageous than Microfocus UFT One in terms of costs and flexibility.

WebDriver Architecture is made up of four major components:

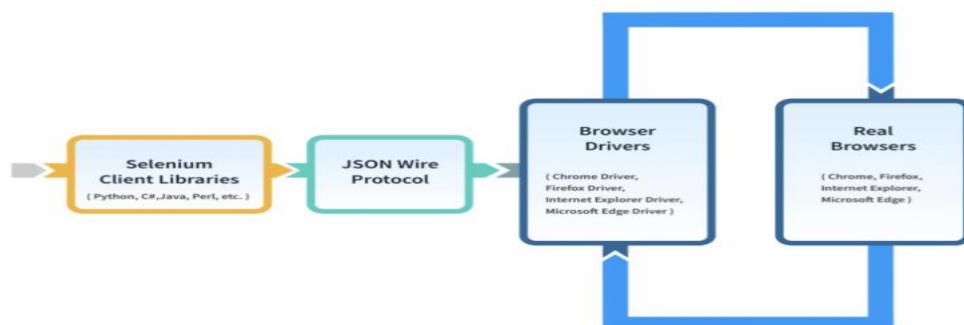


Figure Selenium Web Driver (6.1.1)

6.1.2 Selenium RC (Remote control)

Selenium Remote Control (RC) is a server, written in Java, that accepts commands for the browser via HTTP. RC makes it possible to write automated tests for a web application in any programming language, which allows for better integration of Selenium in existing unit test frameworks. To make writing tests easier, Selenium project currently provides client drivers for PHP, Python, Ruby, .NET, Perl and Java. The Java driver can also be used with JavaScript (via the Rhino engine). An instance of selenium RC server is needed to launch an HTML test case – which means that the port should be different for each parallel run. However, for Java/PHP test case only one Selenium RC instance needs to be running continuously

6.1.3 Selenium GRID

Selenium Grid is a server that allows tests to use web browser instances running on remote machines. With Selenium Grid, one server acts as the hub. Tests contact the hub to obtain access to browser instances. The hub has a list of servers that provide access to browser instances (Web Driver nodes), and lets tests use these instances. Selenium Grid allows running tests in parallel on multiple machines and to manage different browser versions and browser configurations centrally (instead of in each individual test).

The ability to run tests on remote browser instances is useful to spread the load of testing across several machines and to run tests in browsers running on different platforms or operating systems. The latter is particularly useful in cases where not all browsers to be used for testing can run on the same platform.

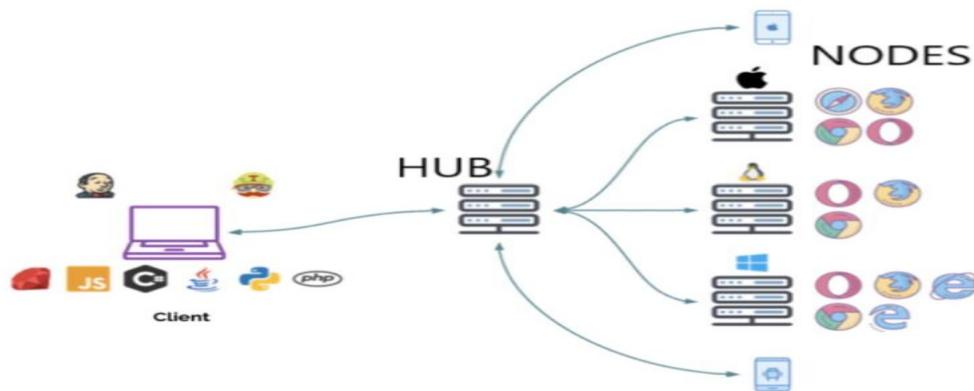


Figure Selenium GRID (6.1.3)

6.1.4 Features of Selenium

Open Source and Portable: Selenium is an open source and portable Web testing Framework.

Combination of tool and DSL: Selenium is combination of tools and DSL (Domain Specific Language) in order to carry out various types of tests.

Easier to understand and implement: Selenium commands are categorized in terms of different classes which make it easier to understand and implement.

Reduce test execution time: Selenium supports parallel test execution that reduce the time taken in executing parallel tests.

Lesser resources required: Selenium requires lesser resources when compared to its competitors like UFT, RFT, etc.

Supports Multiple Programming Languages: C#, Java, Python, PHP, Ruby, Perl, and JavaScript

Supports Multiple Operating Systems: Android, iOS, Windows, Linux, Mac, Solaris.

Supports Multiple Browsers: Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari, etc.

Parallel Test Execution: It also supports parallel test execution which reduces time and increases the efficiency of tests.

6.2 Beautiful Soup Tool:

Web scraping is a technique that is used for extracting HTML content from different websites.

These web scrapers are mainly computer bots that can directly access the World Wide Web using HTTP Protocol and use this information in various applications. The data is obtained in an unstructured format, which is then converted into a structured manner after performing multiple pre-processing steps. Users can save this data in a spreadsheet or export it through an API.

Web scraping can also be done manually for small web pages by simply copying and pasting the data from the web page. But this copy and pasting would not work if we require data at a large scale and from multiple web pages. Here automated web scrapers come into the picture. They use

intelligent algorithms which can extract large amounts of data from numerous web pages in less time.

6.2.1 Uses of Web Scraping

Web scraping is a powerful tool for businesses to gather and analyze information online. It has multiple applications across various industries. Below are some of these that you can check out.

1. **Marketing:** Web scraping is used by many companies to collect information about their products or services from various social media websites to get a general public sentiment. Also, they extract email ids from various websites and then send bulk promotional emails to the owners of these email ids.
2. **Content Creation:** Web scraping can gather information from multiple sources like news articles, research reports, and blog posts. It helps the creator to create quality and trending content.
3. **Price Comparison:** Web scraping can be used to extract the prices of a particular product across multiple e-commerce websites to give a fair price comparison for the user. It also helps companies fix the optimal pricing of their products to compete with their competitors.
4. **Job Postings:** Web Scraping can also be used to collect data on various job openings across multiple job portals so that this information can help many job seekers and recruiters.

Now, we will create a simple web scraper using Python and Beautiful Soup library. We will parse an HTML page and extract useful information from it. This tutorial requires a basic understanding of Python as its only prerequisite.

Web scraping has gained a lot of attention and has a wide range of uses. A few of them are listed below:

1. Social Media Sentiment Analysis
2. Lead Generation in Marketing Domain
3. Market Analysis, Online Price Comparison in eCommerce Domain
4. Collect train and test data in Machine Learning Applications

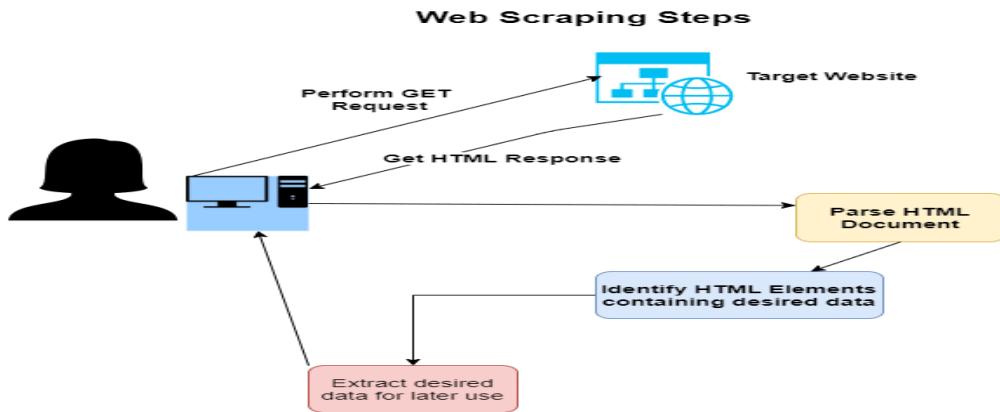


Figure Uses of Web Scraping (6.4.1)

6.2.2 Steps involved in web scraping:

1. Find the URL of the webpage that you want to scrape
2. Select the particular elements by inspecting
3. Write the code to get the content of the selected elements
4. Store the data in the required format

6.2.3 Web Scraping libraries/tools

Selenium: a framework for testing web applications

Beautiful Soup: Python library for getting data out of HTML, XML, and other markup languages

Pandas: Python library for data manipulation and analysis

CHAPTER 7

WORKING OF THE PROJECT

Speech recognition software uses natural language processing (NLP), Speech recognition breaks down into bits it can interpret, converts it into a digital format, and analyzes the pieces of content. From there, the software makes determinations based on programming and speech patterns, after determining what users said, the software transcribes the conversation into text, the text is saved in the variable, as our voice assistance is text dependent the, the given command in the form of text searches for the similar text written in the form of functions, if the text is matched it performs the particular task, if not found it throws an exception. Steps of Working:

The idea behind JARVIS served as a model for current AI initiatives and voice-activated assistants, demonstrating how AI has the ability to improve human-computer interactions. Speech recognition, task execution, text-to-speech conversion, Google Scrap, translation, reminder, YouTube search, automatic whatsapp and Gmail, and interaction with Google Maps are the new capabilities added to the desktop. Speech recognition and text-to-speech conversion employ Python libraries called pytsxs3 library and gTTS, respectively. The web browser library, which enables access to and opening of the website, is utilized for Google scraping. Similar to that, the news functionality is introduced via APIS. The PyQt5 library is used to build a GUI that gives the J.A.R.V.I.S a face.

Fig 7.1 – Following are the python files and Folders of the program. In This folders contains multiple files.

A screenshot of a code editor interface. On the left, there is a file explorer window titled 'EXPLORER' showing a directory structure for a project named 'FYP'. The 'main.py' file is selected in the list. On the right, the main editor window shows the content of 'main.py'. The code is a Python script with imports from various modules like func, llm, and buildin, along with logic for interacting with various APIs and local files. The status bar at the bottom indicates the file is 'main.py' with 29 lines, and the editor is set to Python 3.10.10 64-bit.

```
File Edit Selection View Go Run ... ← → ⌂ FYP
EXPLORER
> FYP
  > audio
  > auth
  > autofunc
  > buildin
  > ChromeFile
  > chromedata
  > customs
  > data
  > func
  > gui
  > imgs
  > injection
  > keys
  > llm
  > output
  > Powerpointer
  > temp
  > templates
  > toolkit
    -Sesis FYP.docx
    app.py
    error.log
    Generation_Of_Images.py
> OUTLINE
> TIMELINE
X 0 △ 0 % 0
main.py x
main.py > ...
1 # from func.Osrc.Chat import Chat
2 # from func.Speak.SpeakHotWord import Speak
3 # from func.Listen.ListenJS import Listen
4 # from func.Osrc.DataOnline import Online_Scraper
5 #from func.XTRA.ExecCode import ExecCode
6 #from func.Social.News import News
7 #from func.Social.SocialMedia import SocialMedia
8
9
10 # from func.XTRA.Clap import MainClapExe
11 from func.Jukebox.YouTube import MusicPlayer
12
13 from llm.Filter import Filter
14 from llm.ChatGpt import ChatGpt,messages as gms
15 from llm.Mistral2 import Mistral7B,messages as mms
16
17 from buildin import GoodMsg
18 from buildin import KnowApps
19
20 from autofunc.youtube import GetTranscript
21
22
23 from Powerpointer.app import get_bot_response
24
25 from Generation_Of_Images import *
26
27
28 # from colorama import Fore, Back, Style
29 import psutil
Ln 19, Col 1 Spaces: 4 UTF-8 CR/LF Python 3.10.10 64-bit Go Live
```

Figure 7.2 – Front page of the program. This menu shows options to select and is saved as main.py in the file.



Fig 7.3 – The assistant can retrieve useful and up-to-date data from websites using web scraping as its foundation.

```

YOU : Jarvis open Google Chrome and search University of Sindh
21
6552
20
6418
19
6274
18
5675
17
5414
```python
import webbrowser

search_query = "University of Sindh"
webbrowser.open('https://www.google.com/search?q=' + search_query)
```
False
19
5854
18
5679
17
5535
16
4862
```python
import os

search_query = "University of Sindh"
os.system(f"start chrome 'https://www.google.com/search?q={search_query}'")
```
This code uses the 'os' module to open Google Chrome and search for "University of Sindh" using the 'start' command
    
```

The DEBUG CONSOLE tab shows Python code for opening a Google search query in a web browser. The code uses the `webbrowser` module to open the URL `https://www.google.com/search?q=University%20of%20Sindh`. It also includes a note at the bottom explaining that it uses the `os` module and the `start` command to open the browser.

Figure 7.4 – Shows the Output of Figure 7.3 Google Scrap or Search on Google.

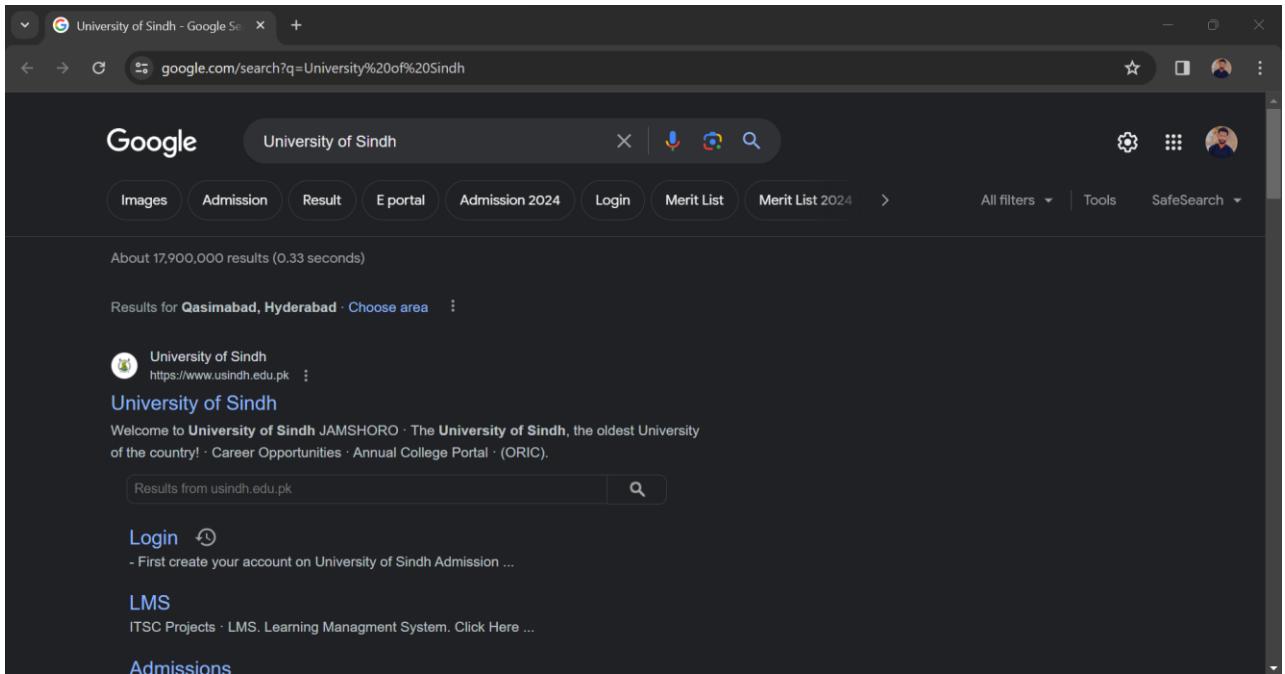


Fig 7.5 – Multitasking for Google Scrap at a time multiple task command.

The screenshot shows a terminal window titled "FYP" with the "TERMINAL" tab selected. The terminal output shows a sequence of commands being run in a Python script. The commands include opening Instagram, Facebook, and YouTube, and searching for "12th movie" on YouTube. The terminal interface includes an Explorer sidebar on the left showing project files like "FYP", "auth", "func", and "gui", and a Problems, Output, Debug Console, and Ports tab bar at the top.

```
YOU : Jarvis open Instagram and after at open Facebook and after that open YouTube and after that such YouTube and after date search 12th movie on YouTube and after that in website open

16
5805
15
5100
```python
import webbrowser
import time
import subprocess

Open Instagram
webbrowser.open('https://www.instagram.com')

Wait for Instagram to load
time.sleep(5)

Open Facebook
webbrowser.open('https://www.facebook.com')

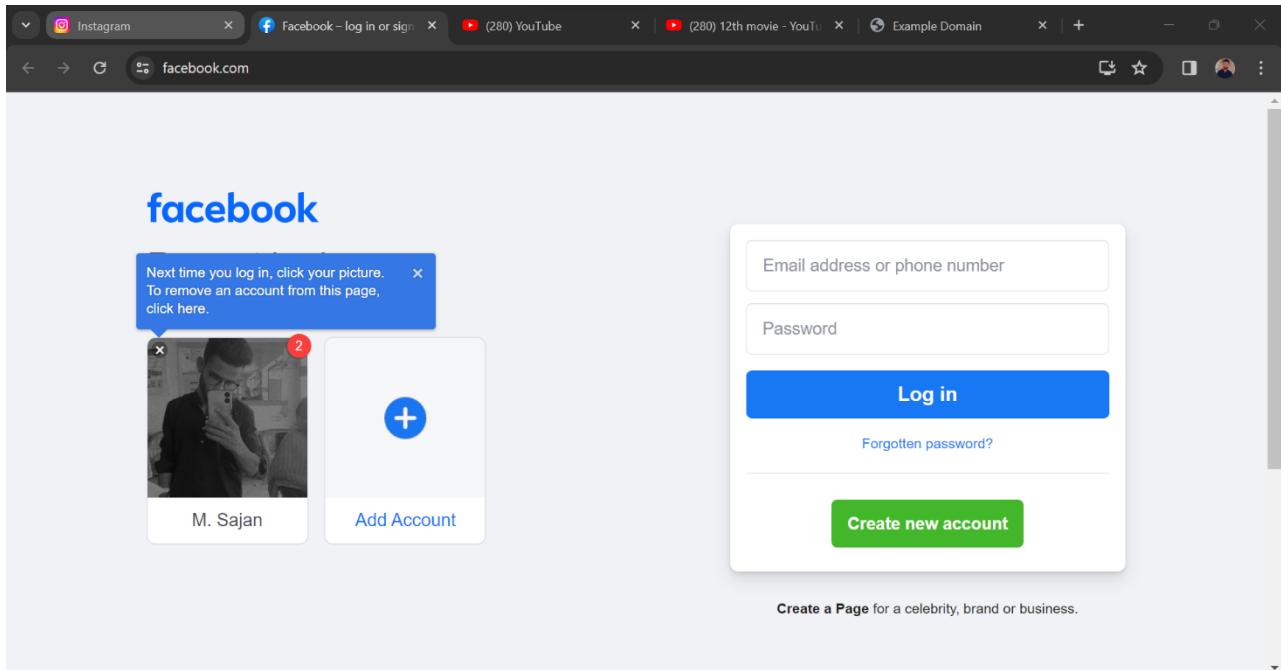
Wait for Facebook to load
time.sleep(5)

Open YouTube
webbrowser.open('https://www.youtube.com')

Wait for YouTube to load
time.sleep(5)

Search for "12th movie" on YouTube
search_query = "12th movie"
webbrowser.open('https://www.youtube.com/results?search_query=' + search_query)
```

**Figure 7.5** – Show the Output of Multitasking.



**Fig 7.6** – Here the Optimize code command, first select a code that was you want to optimize and ask Jarvis command like show on picture.

```
(No symbol) [0x01275818]
BaseThreadInitThunk [0x75AD7BA9+25]
RtlInitializeExceptionChain [0x7747BD2B+107]
RtlClearBits [0x7747BCAF+191]

YOU : optimise this code

18
6575
17
6300
16
5688
15
5544
14
4797
```python
a = str(input())
b = str(input())
answer = ""
len1, len2 = len(a), len(b)
for i in range(len1):
    match = ""
    for j in range(len2):
        if (i + j < len1 and a[i + j] == b[j]):
            match += b[j]
        else:
            if (len(match) > len(answer)):
                answer = match
            match = ""
```
To optimize this code, you can use the concept of dynamic programming to avoid unnecessary calculations. Here's an optimized version:
```

**Fig 7.7 – Optimize code output.**

```

for j in range(len2):
 if (i + j < len1 and a[i + j] == b[j]):
 match += b[j]
 else:
 if (len(match) > len(answer)):
 answer = match
 match = ""
...
To optimize this code, you can use the concept of dynamic programming to avoid unnecessary calculations. Here's an optimized version:

```python
a = str(input())
b = str(input())
answer = ""
len1, len2 = len(a), len(b)
dp = [[0] * (len2 + 1) for _ in range(len1 + 1)]

for i in range(1, len1 + 1):
    for j in range(1, len2 + 1):
        if a[i - 1] == b[j - 1]:
            dp[i][j] = dp[i - 1][j - 1] + 1
            if dp[i][j] > len(answer):
                answer = a[i - dp[i][j]:i]
        else:
            dp[i][j] = 0
print(answer)
```
This optimized code uses a 2D array `dp` to store the lengths of the longest common substrings. It avoids unnecessary calculations by only updating the `dp` array when a match is found. The final result is stored in the `answer` variable.
What's the next move, sir?

```

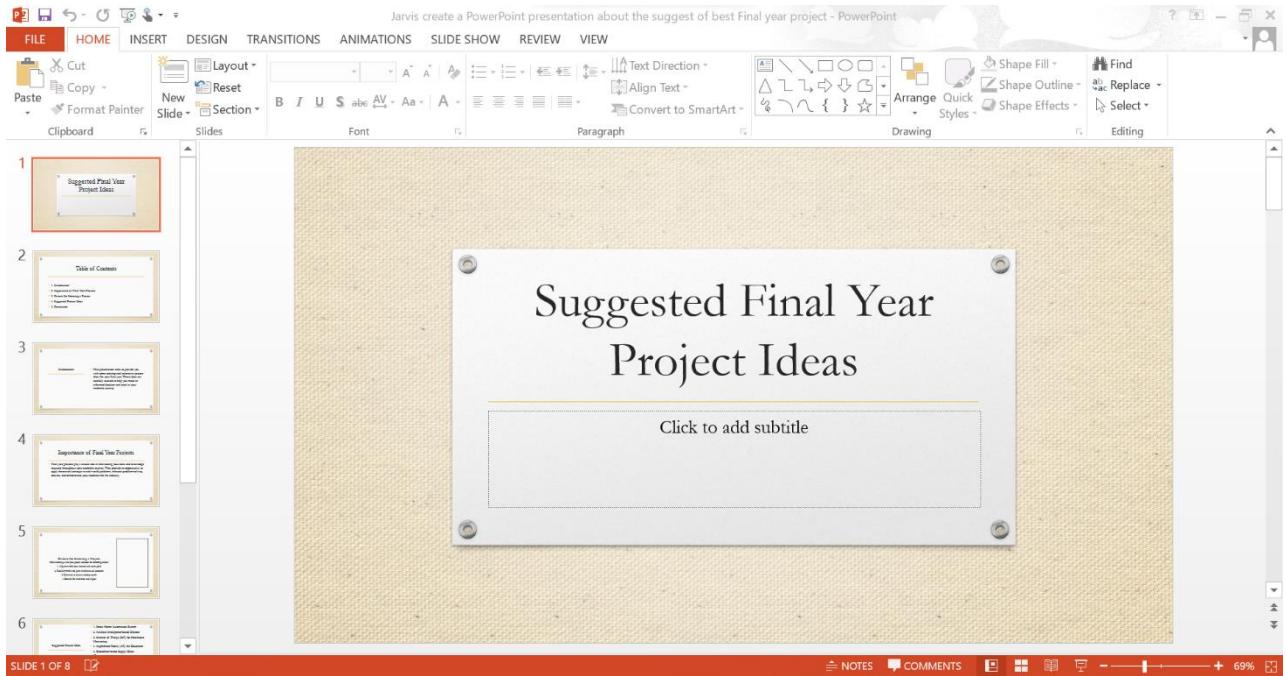
**Fig 7.8 – Here is the Command that you want to create a ppt presentation automatically.**

```

YOU : Jarvis create a PowerPoint presentation about the suggest of best Final year project
No design specified, using default design...
New g4f version: 0.2.0.4 (current: 0.2.0.3) | pip install -U g4f
#Title: Suggested Final Year Project Ideas
#Slide: 1
#Header: Table of Contents
#Content:
1. Introduction
2. Importance of Final Year Projects
3. Criteria for Selecting a Project
4. Suggested Project Ideas
5. Conclusion
#Slide: 2
#Header: Introduction
#Content: This presentation aims to provide you with some exciting and innovative project ideas for your final year. These ideas are carefully curated to help you make an informed decision and excel in your academic journey.
#Slide: 3
#Header: Importance of Final Year Projects
#Content: Final year projects play a crucial role in showcasing your skills and knowledge acquired throughout your academic journey. They provide an opportunity to apply theoretical concepts to real-world problems, enhance problem-solving abilities, and demonstrate your readiness for the industry.
#Slide: 4
#Header: Criteria for Selecting a Project
#Content: When selecting a final year project, consider the following criteria:
1. Alignment with your interests and career goals
2. Feasibility within the given timeframe and resources
3. Relevance to current industry trends
4. Potential for innovation and impact

```

**Fig 7.9** – Here the Output of ppt Presentation that was automatically created.and this title was same as you given a voice command show on top of presentation.



**Fig 7.10** – You want a complete source code of any time of program like face recognition or else you just ask by using voice command or write a line for code and select that line and say like show bellow.

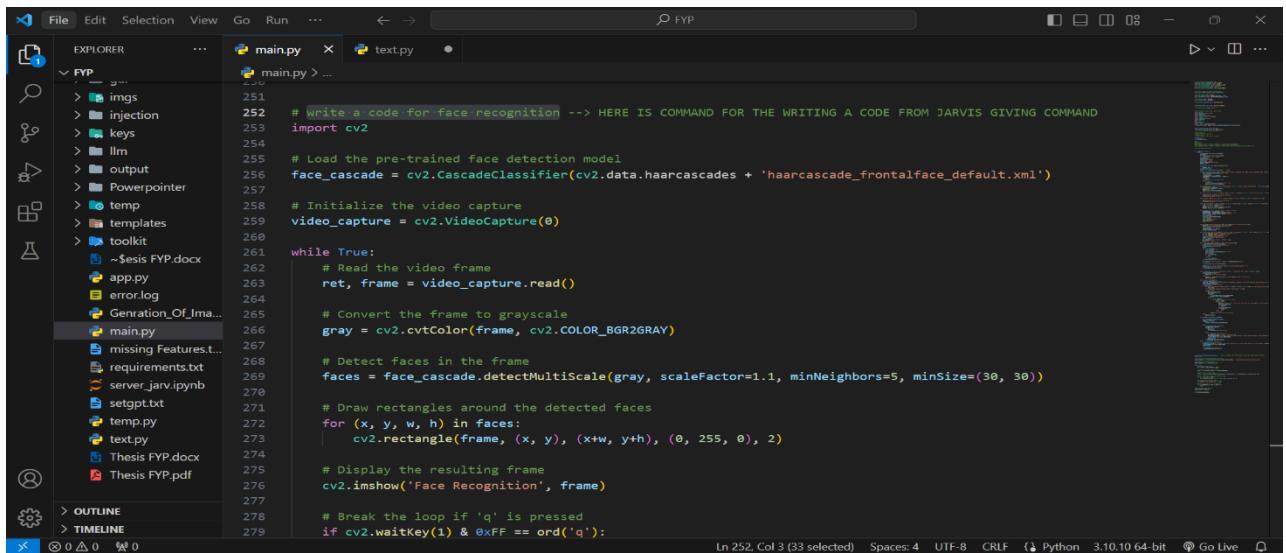
```

File Edit Selection View Go Run ...
FYP
EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + ...
YOU : write code for this
16
7003
15
6859
14
5834
13
5417
Certainly! Here's an example code for face recognition using Python with the OpenCV library:
```python
import cv2
# Load the pre-trained face detection model
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
# Initialize the video capture
video_capture = cv2.VideoCapture(0)
while True:
    # Read the video frame
    ret, frame = video_capture.read()
    # Convert the frame to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    # Detect faces in the frame
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
```
Ln 252, Col 107 Spaces: 4 UTF-8 CRLF Python 3.10.10 64-bit Go Live

```

The screenshot shows a terminal window with the title bar "FYP". The terminal interface includes sections for EXPLORER, PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing a Python session. The user has typed "YOU : write code for this" and received a response with sample Python code for face recognition using the OpenCV library. The code loads a pre-trained face detection model, initializes a video capture, and uses a while loop to read frames, convert them to grayscale, and detect faces using the cascade classifier. The bottom status bar shows file paths and system information.

**Fig 7.11**– Here is the output of that line and you luck at picture **252 line** are selected are command line, and following lines are the code of command line.



```

File Edit Selection View Go Run ... ← → ⌂ FYP
EXPLORER main.py text.py ...
FYP
 __init__.py
 imgs
 injection
 keys
 llm
 output
 Powerpointer
 temp
 templates
 toolkit
 ~Sesis FYP.docx
 app.py
 error.log
 Generation_Of_Ima...
 main.py
 missing Features...
 requirements.txt
 server.jarv.ipynb
 setgpt.txt
 temp.py
 text.py
 Thesis FYP.docx
 Thesis FYP.pdf
 OUTLINE
 TIMELINE
In 252, Col 3 (33 selected) Spaces: 4 UTF-8 CRLF Python 3.10.10 64-bit Go Live
0 0 0 0% 0

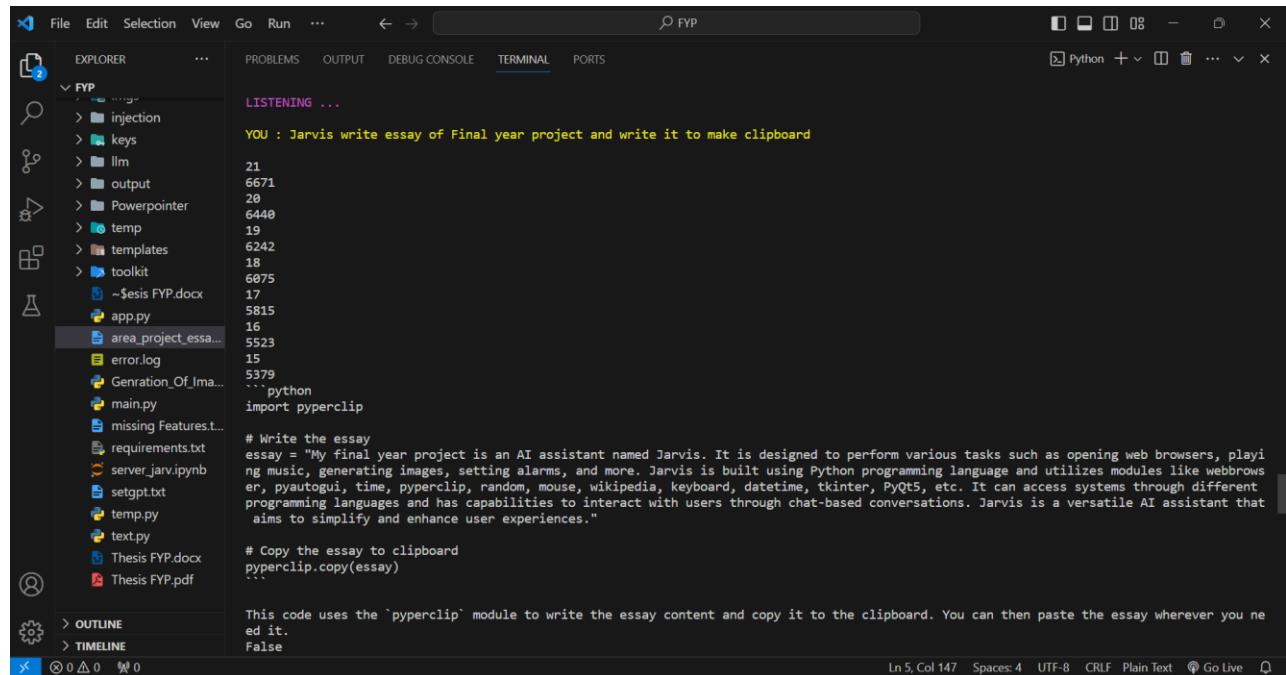
```

```

main.py > ...
251 # write a code for face recognition --> HERE IS COMMAND FOR THE WRITING A CODE FROM JARVIS GIVING COMMAND
252 import cv2
253
254 # Load the pre-trained face detection model
255 face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
256
257 # Initialize the video capture
258 video_capture = cv2.VideoCapture(0)
259
260 while True:
261 # Read the video frame
262 ret, frame = video_capture.read()
263
264 # Convert the frame to grayscale
265 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
266
267 # Detect faces in the frame
268 faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
269
270 # Draw rectangles around the detected faces
271 for (x, y, w, h) in faces:
272 cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
273
274 # Display the resulting frame
275 cv2.imshow('Face Recognition', frame)
276
277 # Break the loop if 'q' is pressed
278 if cv2.waitKey(1) & 0xFF == ord('q'):
279

```

**Fig 7.12** – Know you want to write an assay so just ask him like following command.



```

File Edit Selection View Go Run ... ← → ⌂ FYP
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + - ×
EXPLORER ... LISTENING ...
area_project_essa... YOU : Jarvis write essay of Final year project and write it to make clipboard
21
6671
20
6440
19
6242
18
6075
17
5815
16
5523
15
5379
```
python
import pyperclip
# Write the essay
essay = "My final year project is an AI assistant named Jarvis. It is designed to perform various tasks such as opening web browsers, playing music, generating images, setting alarms, and more. Jarvis is built using Python programming language and utilizes modules like webbrowser, pyautogui, time, pyperclip, random, mouse, wikipedia, keyboard, datetime, tkinter, PyQt5, etc. It can access systems through different programming languages and has capabilities to interact with users through chat-based conversations. Jarvis is a versatile AI assistant that aims to simplify and enhance user experiences."
# Copy the essay to clipboard
pyperclip.copy(essay)
```
This code uses the `pyperclip` module to write the essay content and copy it to the clipboard. You can then paste the essay wherever you need it.
False
In 5, Col 147 Spaces: 4 UTF-8 CRLF Plain Text Go Live
0 0 0 0% 0

```

**Fig 7.13** – Here are the assay of the following Fig 7.12 command line.

```

File Edit Selection View Go Run ... FYP
EXPLORER ... main.py area_project_essay.txt text.py ...
area.project.essay.txt
1 involved analyzing the spatial distribution of a specific species in a given region.
2 e factors influencing the species' distribution and identify potential conservation strategies.
3 data on the species' occurrence and environmental variables such as temperature, precipitation, and habitat character
4 uage and modules like pandas, numpy, and matplotlib, I performed statistical analyses and created visualizations to e
5 ovided valuable insights into the species' habitat preferences and can contribute to informed conservation decisions.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
LISTENING ...
YOU : Jarvis write essay of Final year project and write it to make clipboard
21
6671
Ln 5, Col 147 Spaces: 4 UTF-8 CRLF Plain Text Go Live

```

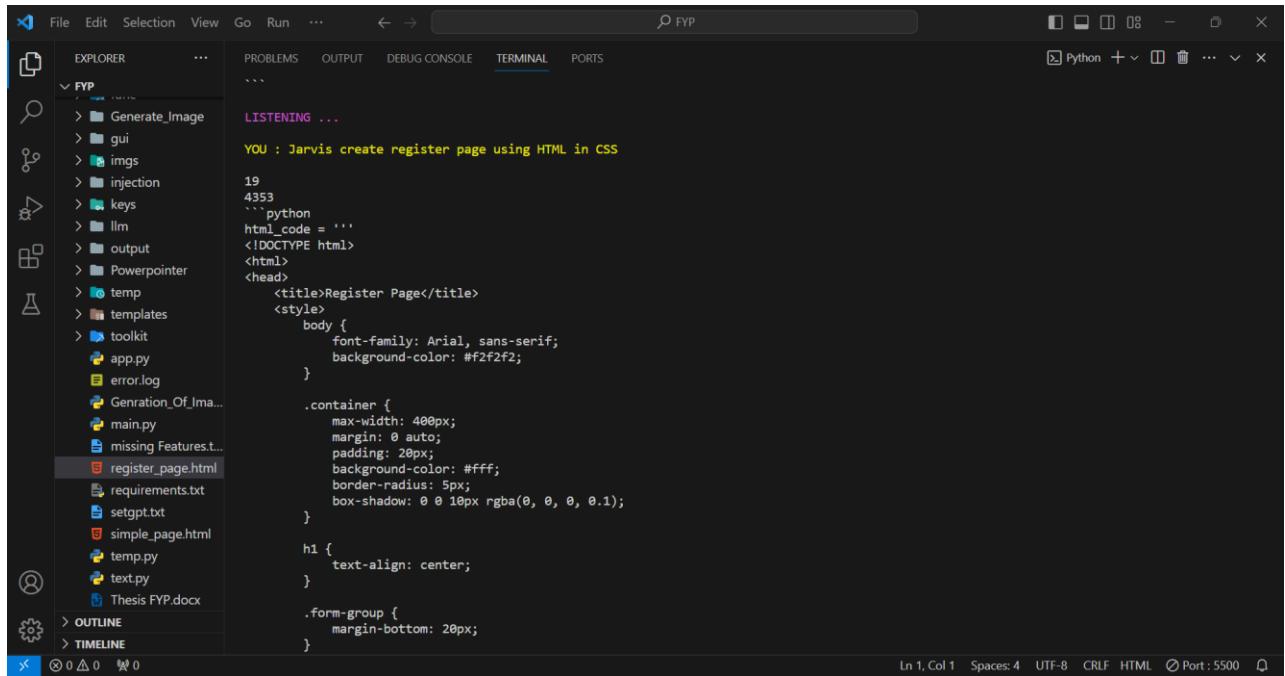
**Fig 7.14** – Bard Image Generation you want to design of any property picture generate by using Jarvis automatically download that picture. Here you want to generate a house design picture also

```

PROBLEMS OUTPUT TERMINAL ...
YOU : Jarvis generate developer background image
16
3485
New g4f version: 0.2.0.4 (current: 0.2.0.3) | pip install -U g4f
```
python
from Generation_Of_Images import Generate_Images, Show_Image
IMGS = Generate_Images(prompt="developer background")
IMGS_TO_SHOW = Show_Image(IMGS)
IMGS_TO_SHOW.open(0)
```
Sending request...
Waiting for results...
.....
Downloading images...
LISTENING ...
YOU : Jarvis show me next image
18
3953
```
python
IMGS_TO_SHOW.open(1)
```
LISTENING ...
YOU : Jarvis show me next image
20
4255
```
python
IMGS_TO_SHOW.open(2)
```

```

**Fig 7.15** – Creating a web page or website based on your Command in short time.



The screenshot shows a terminal window with the following content:

```
File Edit Selection View Go Run ... ⏪ ⏩ 🔍 FYP PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
`--> FYP
 > Generate_Image
 > gui
 > imgs
 > injection
 > keys
 > llm
 > output
 > Powerpointer
 > temp
 > templates
 > toolkit
 app.py
 error.log
 Generation_Of_Ima...
 main.py
 missing Features.t...
 register_page.html
 requirements.txt
 setgpt.txt
 simple_page.html
 temp.py
 text.py
 Thesis FYP.docx
 OUTLINE
 TIMELINE
 ✘ 0 △ 0 ⚡ 0
 LISTENING ...
YOU : Jarvis create register page using HTML in CSS
19
4353
```python
html_code = '''
<!DOCTYPE html>
<html>
<head>
    <title>Register Page</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f2f2f2;
        }

        .container {
            max-width: 400px;
            margin: 0 auto;
            padding: 20px;
            background-color: #fff;
            border-radius: 5px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }

        h1 {
            text-align: center;
        }

        .form-group {
            margin-bottom: 20px;
        }
    </style>

```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF HTML ⚡ Port: 5500

Fig 7.16 – Here the output of fig7.15 create a Web Register page and also code save Name resgitor_page.html that was show bellow in main directory created in Short time.

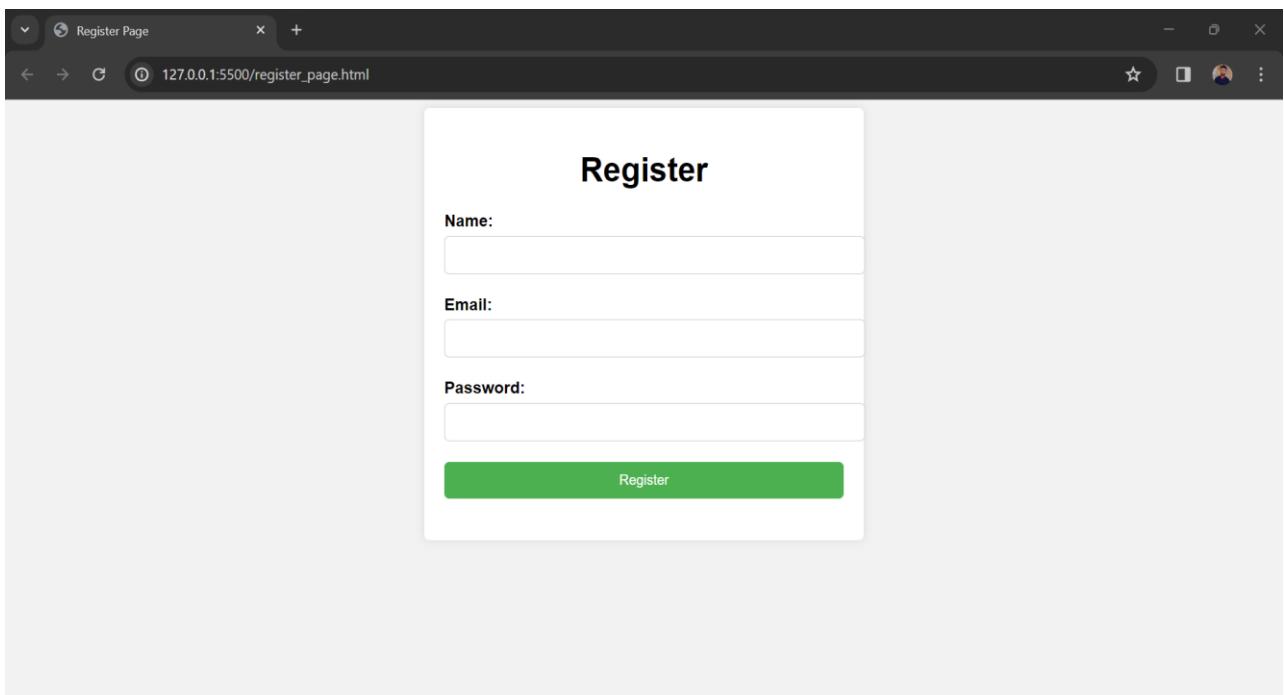


Fig 7.17 – Create a Pie chart or line Graph just simple command. First Select a graph Ans and say read clipboard or read my clipboard and wait for 1 second and then say draw pie chart or line graph.

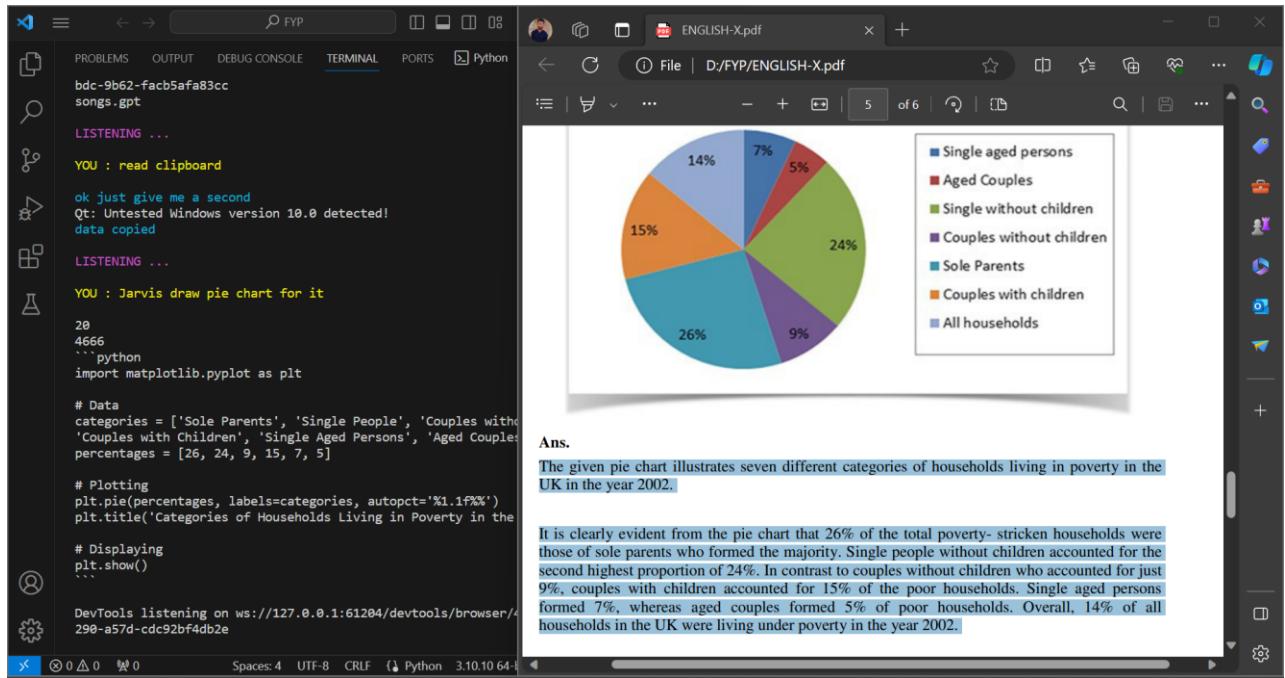
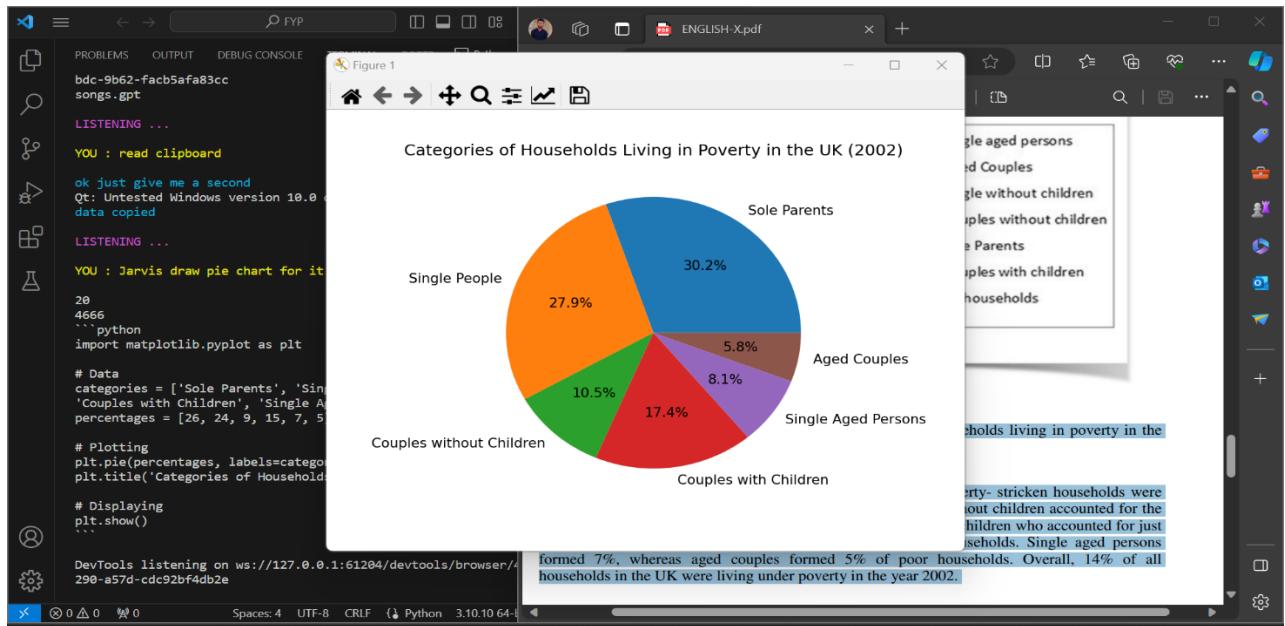


Fig 7.18 – Here the Output of Fig 7.17 to create a pie chart



CHAPTER 8

CONCLUSION AND FUTURE SCOPE

FUTURE SCOPE

In light of boosting technology, the Personal Desktop AI Assistant project offers several opportunities for additional refinement and future scope.

Extending the assistant to enable several modalities of interaction, such as text, speech, and even visual inputs, is one possibility.

Another approach may be to integrate the assistant with other external services and APIs, such as opencv APIs, bing Ai APIs, chatgpt-4.0 APIs, Mistral7B APIs, Creater Presentation APIs, Chart Generator APIs, Image Generation APIs, Web Generation APIs, and others, to provide a broader variety of functionality.

Transforming the assistant into a control center for home automation by connecting it with IoT devices would allow users to manage and control their household appliances conveniently through voice or text commands.

Finally, using machine learning methods to allow the assistant to predict the user's needs or behaviors would be an intriguing project to pursue in the future.

CONCLUSION

The development of a voice assistant project brings forth a range of benefits and opportunities for users in various domains. Voice assistants have witnessed significant advancements in speech recognition, natural language processing, and personalized assistance, leading to improved user experiences. Through integration with devices, applications, and IoT systems, voice assistants have become valuable tools for controlling smart homes, accessing information, and performing tasks hands-free. However, there are still challenges to overcome in the realm of voice assistants. Issues such as accuracy, contextual understanding, personalization, privacy, and bias need to be addressed to enhance the overall user experience. Ongoing research and development efforts are essential to ensure voice assistants continue to evolve and meet user expectations. The proposed voice assistant system should focus on

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 complete server assistant and make it smart enough to act as a replacement for a general server administration. The future plans include integrating Jarvis with mobile using React Native to provide a synchronized experience between the two connected devices. Further, in the long run, Jarvis is planned to feature auto deployment supporting elastic beanstalk, backup files, and all operations which a general Server Administrator does.

The functionality would be seamless enough to replace the Server Administrator with Jarvis.

In this paper we have discussed about Jarvis voice assistance for windows using python. Jarvis voice assistance makes life easier to humans. As like Google assistance and Cortina we make Jarvis voice assistance to be available to all the windows version, We use Artificial intelligence technology for this project, Jarvis voice assistance be able to do all the tasks like other assistance including some special functions like restarting the devices, locking the device, sleeping the device for some particular time and shutdown the device with our voice input. We can expect this Jarvis voice assistance to be permanent

REFERENCE

1. Vishwakarma, Aditya, Ankit Sahu, Nadeem Sheikh, Peeyoosh Payasi, Saurabh Kumar Rajput, and Laxmi Srivastava. "IoT based Greenhouse monitoring and controlling system." In 2020 IEEE Students Conference on Engineering & Systems (SCES), pp. 1-6. IEEE, 2020.
2. G. Bohouta, V. Z. Kępuska, "Comparing Speech Recognition Systems (Microsoft API Google API And CMU Sphinx)", Int. Journal of Engineering Research and Application 2017, 2017.
3. Mohasi, L. and Mashao, D., 2006. Text-toSpeech Technology in Human-Computer Interaction. In 5th Conference on Human Computer Interaction in Southern Africa, South Africa (CHISA 2006, ACM SIGHI) (pp. 79-84).
4. Fryer, L.K. and Carpenter, R., 2006. Bots as language learning tools. Language Learning & Technolognational Conference on Pattern Recognition, Vol. 1, page 1056–1059.
5. Chen, X., Liu, C., & Guo, W. (2020). A Survey on Voice Assistant Systems. IEEE Access, 8, 27056-27070.
6. Kundu, A., & Ekbal, A. (2021). Design and Development of a Voice-Based Intelligent Personal Assistant. In Proceedings of the 10th International Conference on Pattern Recognition and Machine Intelligence (PReMI 2021), pp. 69-81.
7. Sarma, M., Sengupta, S., & Gupta, S. (2019). Voice Assistant Systems: A Review. In Proceedings of the 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence 2019), pp. 167-172.
8. Huang, P. H., & Sharma, S. (2021). Personal Voice Assistants: An Overview of Technologies and Applications. In 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), pp. 1-6.
9. Wang, W., Li, L., Yu, Y., & Zhang, L. (2021). Design and Implementation of Voice Assistant Based on Deep Learning. In Proceedings of the 2021 3rd International Conference on Artificial Intelligence, Automation and Control Technologies (AIACT 2021), pp. 199-204.
10. Bullich, N., & Grivolla, J. (2020). Voice Assistants and Language Technologies: Issues and Opportunities. Language Resources and Evaluation, 54(2), 275-291.

11. Wardaya, D. A., & Sinulingga, K. (2020). Voice Assistant System Using Deep Learning for Indonesian Language. *Journal of Physics: Conference Series*, 1565(4), 042053.
12. Lee, H., Lee, J., & Yoon, S. (2020). A Survey on Voice Interaction for Virtual Assistant and Conversational Agent Technologies. *Journal of Electrical Engineering and Technology*, 15(4), 2019-203

- 1) [Physical Examination and Health Assessment - Canadian E-Book: Physical ... - Carolyn Jarvis - Google Books](#)
- 2) [Human Learning: An Holistic Approach: Jarvis, Peter, Parker, Stella: 9780415340984: Amazon.com: Books](#)
- 3) [Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence \(2021\)](#)
- 4) [Artificial Intelligence For Dummies \(For Dummies \(Computer/Tech\)\) Book Online at Low Prices in India | Artificial Intelligence For Dummies \(For Dummies \(Computer/Tech\)\) Reviews & Ratings - Amazon.in](#)
- 5) [Artificial Intelligence For Dummies \(For Dummies \(Computer/Tech\)\) Book Online at Low Prices in India | Artificial Intelligence For Dummies \(For Dummies \(Computer/Tech\)\) Reviews & Ratings - Amazon.in](#)
- 6) [\(GitHub - PacktPublishing/Mastering-Selenium-WebDriver-3.0-Second-Edition: Mastering Selenium WebDriver 3.0 - Second Edition, published by Packt\)](#)
- 7) [Selenium Simplified: Richardson, Alan John: 9780956733238: Amazon.com: Books](#)
- 8) [Artificial Intelligence Books For Beginners in 2023 - GeeksforGeeks](#)
- 9) [Data Science and Machine Learning using Python : Dr Reema Thareja: Amazon.in: Books](#)
- 10) [Data Analytics using Python : Bharti Motwani: Amazon.in: Books](#)