

# AI Companion

Anushka V<sup>1</sup>, Arnav Kumar<sup>2</sup>, Arun G<sup>3</sup>, Kavya R<sup>4</sup>

<sup>1</sup>Department of Computer Science and Engineering, NIE, Mysore

<sup>2</sup> Department of Computer Science and Engineering, NIE, Mysore

<sup>3</sup> Department of Computer Science and Engineering, NIE, Mysore

<sup>4</sup>Department of Computer Science and Engineering, NIE, Mysore

Dr . Anitha R, Dept. of Computer Science and Engineering, NIE, Karnataka, India

\*\*\*

**Abstract** - An AI Companion can assist users in several ways like, communication (sending emails), fetch weather reports, entertain users by playing music, set alarms and can perform many more functionalities. They play vital role in the day-to-day lives of people. Some famous AI assistants are Siri, Microsoft Cortana, OpenAIGPT-3 and so on. The existing system lacks functionalities such as chatbot, trading bot, stable diffusion, and image scanner. It is platform-dependent, for example, Siri is available only on all Apple devices such as iPad, Mac, etc. the proposed system is platform independent and can perform simple tasks like setting an alarm to performing complex tasks like generating images from text-based input. It needs a good internet connectivity and thus it can be accessed across wide range of platforms. It can also generate text-based output from an image-based input and it can also interact with users like a chatbot and resolve the user queries by providing meaningful output. Functionalities such as chatbot, voice assistant, stable diffusion, image scanner and trading bot have been included in the proposed system.

**Key words:** chatbot, voice assistant, stable diffusion, image scanner and trading bot

## 1. INTRODUCTION

In the modern world the robotic voices the AI assistants play a crucial role. these AI assistants within our smartphones and smart speakers are reshaping the way how we interact with devices by making it simpler, easier and more convenient for the users. A personal assistant is more attentive and always ready to assist the user with the spoken command for example "hey google, play some music" sets everything in motion. These digital assistants provide friendly gestures to effortlessly fulfill our desires. Amid the symphony of AI voices, our own voices—with their inquiries, ideas, and distinctive essence—remain our most formidable tool. So, let us embrace these imperceptible allies with open arms, appreciating their contribution. In our project we are specifically focusing on the five functionalities such as: Voice Assistant, trading bot, Stable diffusion, Text-to-image and Chatbot. The voice assistant in our project includes both technical and theoretical elements to deliver a user experience. the system uses Tkinter as the frontend framework and Python as the backend language. A trading bot is a software application that automates the buying and

selling of financial assets, such as stocks and cryptocurrencies, based on a set of predefined algorithms. Stable diffusion which is developed using an API called Hugging face which converts text into corresponding image as per user requirement. Chatbots are sophisticated computer programs designed to pose human conversation and are engineered using both technical and theoretical principles to deliver responsive interactions with users. chatbot development lies in Natural Language Understanding (NLU), a crucial component enabling the comprehensive of user inputs.

## 2. PROBLEM STATEMENT

The traditional way of multitasking requires a lot of time and leads to decreased productivity in individuals as they may find it challenging to prioritize their tasks due to a long list of tasks, which often leads to decreased job satisfaction and work-related strain.

Developing an AI assistant that enhances productivity, efficiency, and user experience across various domains by assisting in daily tasks like scheduling, information retrieval, and many more tasks, can save time and increase productivity among individuals. It can also help physically challenged individuals solve their queries, by providing solutions in different ways.

## 3. LITERATURE SURVEY

The paper by Shreyashkar Sharma [1] investigates Chatbot development using python : Most advanced and promising methods for human-machine communication, chatbots stand out as particularly promising. The latest trend currently is chatbots, including Slack, Facebook, Siri, Amazon Alexa, Google Assistant, and many more. These are really beneficial, however in this day of constantly evolving technology, consumer expectations are rising along with technology updates. User preference is for the chatbot to be more automated. While there are faults in every system, even the best systems have some, and users have encountered certain issues with certain chatbots.

The paper by Prathyush Jha et al [2] investigates Voice assistant using Python. This paper introduces a novel voice assistant developed using the Python programming language,

aiming to contribute to the evolution of HCI by giving a versatile, user-centric, and efficient solution. Leveraging Python's robust capabilities in Natural Language Processing [NLP] and artificial intelligence, our voice assistant system stands as a tool of open-source technologies in shaping the future of interactive computing.

The paper by author Rohith Tamarkar et al [3] investigates Design and Development of Chatbot. This paper focuses on a newly emerging tool for learning from CHATBOT, A learning and assistance tool, a chatbot is an artificially created virtual entity that interacts with users through text or speech. This chatbot communicates directly with people using artificial intelligence and machine learning concepts. This paper examines the techniques, terminology, and various platforms used to design and develop the chatbot. It also presents some actual practical life typical applications and examples of CHATBOT. The utility of the CHATBOT tool for Computer-Aided Design (CAD) applications is proposed from this review.

The paper by author Stephen Roller et al [4] investigates Recipes for building an open-domain chatbot. They provide recipes for building open domain chatbots that perform well in human evaluations. It has been shown across the field of NLP and in conventional agent in particular that pre-training on large corpora is important. Human evaluation results are highly dependent on the precies set-up one chooses. Model performance can be strongly affected by the specific instruction given to evaluators, such as given topic or not, the overall conversation length, with maybe difficulty to jointly account for. They report performance when employing co-workers in short multi-turn conversations with no prompt.

The paper by author Sainath Patil et al [5] investigates Text to image using Deep Learning. This paper focuses on text to image synthesis refers to the method of generating images from the input text automatically. Deciphering data between picture and text is a major issue in artificial intelligence. Automatic image synthesis is highly beneficial in many ways. Generation of the image is one of the applications of conditional generative models. To generate images, Generative Adversarial Networks (GANs) are used. Recent advancements have been achieved using GANs. The conversion of the text to image is an extremely appropriate example of deep learning.

The paper by author Renas Rajab Asaad et al [6] investigates Image processing with Python Libraries. This paper examines computer vision, a branch of artificial intelligence dedicated to developing technologies that enable computers to see and comprehend visual content like humans do. Python supports this field with libraries designed for processing images and video clips, utilizing machine learning and deep learning to recognize and classify objects accurately. In previously, and each library has an angle in which it is used for all image processing operations, such as the famous OpenC library in OpenCV, which is characterized

by its speed. Although OpenCV is the major image-processing library that has been using in the upcoming future, having a little bit of knowledge about other image-processing libraries will definitely be a good idea.

The paper by author Saksham Sardana et al [7] investigates Trading vice, Trading Bot for Stock. This research paper presents an effort to develop a trading bot designed to autonomously trade stocks in the real-time market. This bot can be developed only after a heavy research in the current market as we intend to take out all the short comings in existing bots available in market to make it league apart.

The paper by author Ajay Kumar Sahu et al [8] investigates Voice assistance using Artificial Intelligence. This paper describes the provocation of applying virtual assistant technology. The paper also introduces the application of virtual assistants that can help open up opportunities for humanity in various fields. Voice control is an important growing feature that will change people's lives. Voice assistants are available for laptops, desktops and mobile phones. Assistant is now available on all electronic devices. A voice assistant is a software agent that can interpret human speech and respond in machine language.

The paper by author Joshua W. Buck et al [9] investigates Natural Language, Mixed-initiative Personal Assistant Agents. In this paper, we explore user support for mixed-initiative interaction with dialogue-based systems using natural language, employing a bag-of-words model and k-nearest-neighbour classifier. We investigate this issue within the framework of a toolkit we developed for creating automated, mixed-initiative dialogue systems. This toolkit includes a dialogue authoring notation and a management engine based on lambda calculus, designed for specifying and implementing task-based, mixed-initiative dialogues. We use ordering at Subway through natural language, human-computer dialogs as a case study.

The paper by author Philip R. Doyle et al [10] investigates Mapping Perceptions of Humanness in Intelligent Personal Assistant Interaction" examines how perceptions of humanness are multidimensional. The analysis identifies eight key themes: partner knowledge set, interpersonal connection, linguistic content, partner performance and capabilities, conversational interaction, partner identity and role, vocal qualities, and behavioural affordances.

The paper by author Samantha Meindl et al [11] investigates Manipulating and Evaluating Levels of Personality Perceptions of Voice Assistants through Enactment-Based Dialogue Design" explores an enactment-based approach to infuse voice assistants with varying levels of personality. In two focus groups, amateur actors were asked to create and perform dialogues between a voice assistant and a user to portray different personalities. These dialogues were implemented using Amazon Alexa and presented to 156 participants in an online survey to determine if the personality levels were effectively

synthesized and to examine whether user personality influences preferences for specific personality levels.

The paper by author Shivam Singh Sikarwar [12] investigates AI Based Voice Assistance. Digital installation provides new opportunities to simplify everyday life using auxiliary technology tools. Voice assistant is a software that utilize artificial intelligence to take input in form of voice and then do the task accordingly. We use Various methods to convert speech into text(STT), then after processing of the text, it converts into speech(TTS). However, the smart personal assistant's study is vast.. It is divided into separate branches, e.g. A computer-related environment, personal interaction with a computer as well Information systems.

#### 4. EXISTING SYSTEM

The existing AI assistants are platform-dependent. A prime example is Siri, available exclusively on Apple devices. Most AI assistants offer solutions in a single format, Whether text-based, image-based, or voice-based.

The existing system doesn't support multiple functionalities like a chatbot, voice assistant, trading bot, image scanner, and stable diffusion, all in a single system. It offers a basic interface with a lack of the stated functionalities.

Its limitations in natural language processing, and adaptability paved the way for the evolution of more sophisticated AI companions that we see today.

#### 5. PROPOSED SYSTEM

The proposed system has the below-stated functionalities.

##### 5.1. Stable Diffusion

The users provide textual input that describes the image they want to generate. The input can range from simple words to detailed sentences. The textual description is divided into individual words or tokens, a process known as tokenization. Then the tokenized input is fed into the pre-trained Stable Diffusion model hosted on the Hugging Face API and the tokenized input is processed by the model and corresponding representations are generated. The model picks points in the latent space using the representations it created and that can be done by introducing random noise or by making small adjustments to the latent space which allows varied outputs. Once the model selects points in the latent space, it uses a decoder to transform the selected points into an actual image that matches the input textual description. The generated image is then returned to the user through the Hugging Face API interface and users can view or download the generated image as needed.

##### 5.2. Voice assistant

The voice assistant in our project aims to provide a smooth user-friendly experience. Our system utilizes Tkinter,

a standard Python interface to the Tk GUI toolkit packaged with Python, for the frontend framework, with Python serving as the backend language. Tkinter is used in the creation of various graphical user interfaces(GUI), and widgets like buttons, text boxes, menus, and it also works with images to deliver user-friendly interaction. On the backend, Python handles the text generation, API calls and functionality of the assistant, using speech recognition libraries such as SpeechRecognition for converting spoken commands into text, and also ensures a proper communication between various parts of the system. Our voice assistant analyzes the text input provided by the user using Natural Language Processing (NLP) tools such as NLTK to understand the context and what the user wants as output. It also uses text-to-speech (TTS) conversion libraries to convert the responses generated into spoken words, so that the communication feel more natural, interactive, and easy to understand. It uses NLP(natural language processing) to overcome the problems of speech recognition such as background noise to efficiently comprehend the user's input, the system breaks it down and extracts relevant information.

##### 5.3. Trading Bot

A Trading bot is a program designed to automate the purchasing and selling of financial assets such as stocks, cryptocurrencies, and others. It is built using Python language and it can access the market data and analyse the market trends for the future. Our trading bot only supports paper trading or "paper" money instead of real money. This will help the users to practice trading strategies without having to face any financial risks. It can serve as a beneficial tool for individual aiming to gain confidence and enhance their trading abilities before engaging in live trading, which entails real-money transactions. The bot accesses the historical data to provide accurate price movements, trading volumes, etc. Users can subsequently establish their trading strategies and parameters, specifying conditions for buying or selling an asset. The bot executes the trades when it finds the right trading opportunities based on the predefined strategies of the user. During order execution, the paper trading bot processes the virtual orders within its simulated environment and carries out transactions based on the simulated market conditions. It also maintains a virtual portfolio to monitor performance, adjust account balance, and compute profit or loss, providing good insights to the users. Thus the users can review their trade history and refine their trading strategies based on the insights provided by the trading bot.

##### 5.4. Chatbot

Our chatbot uses TensorFlow, pandas, numpy and sklearn. These technologies are employed for manipulating data and constructing models. A large dataset which contains question and answer pairs is collected to train our chatbot. Next, the dataset undergoes cleaning and preprocessing using methods such as tokenization, which involves breaking down the

user's input into separate words or tokens to ready the text data for training. We then implement a neural network architecture using TensorFlow. Subsequently, the model undergoes training with the pre-processed dataset, and adjustments are made to the parameters to minimize errors while considering the user's input. Once the chatbot is deployed, the users can interact with it, by sending their queries and the chatbot will in turn process the user input, produce a replay and transmit it to the user.

### 5.5. Image Scanner

An image scanner is a device or software application used to capture digital images of documents, photographs, or other visual content. It converts physical documents or images into digital format, allowing them to be stored, edited, and shared electronically. Image scanners are commonly used in offices, businesses, and homes for tasks such as document scanning, photo digitization, and creating digital copies of important records. They come in various types, including flatbed scanners, sheet-fed scanners, and handheld scanners, each suited for different purposes and requirements. Additionally, there are software applications available that can turn smartphones or tablets into portable image scanners, enabling users to capture and process images on the go. With advancements in technology, modern image scanners often feature high-resolution scanning capabilities, automatic document feeders, and built-in image enhancement tools to improve scan quality. Overall, image scanners play a crucial role in digitizing and managing visual content efficiently in today's digital age.

## 6. SYSTEM DESIGN

Designing an AI assistant that accepts input in the form of voice, text, and image, and runs on any platform involves a multi-modal approach and a flexible architecture. Here's a high-level overview of the system architecture

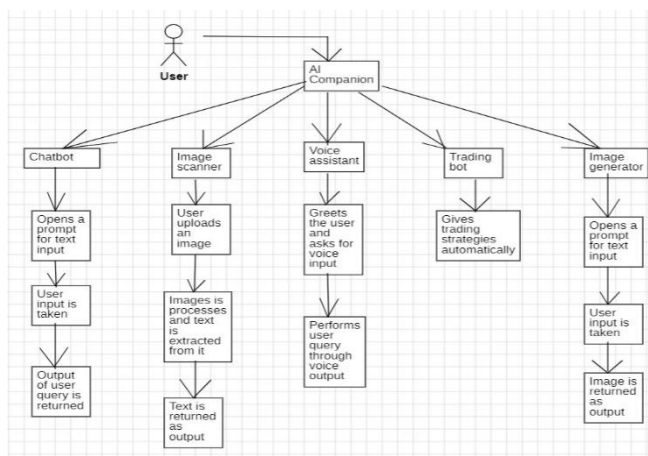


Figure 1: System architecture

*Input Processing:*

*Voice Input:*

Use a Speech Recognition Module: Convert spoken language into text. Commonly used libraries include Google's Speech Recognition API, Microsoft Azure Speech SDK, or open-source options like CMU Sphinx.

*Text Input:*

Directly process the input text.

*Image Input:*

Employ Image Recognition and Processing: Use Computer Vision models (e.g., convolutional neural networks like CNNs) to analyse and understand the content of images. Popular frameworks include TensorFlow and PyTorch.

*Multi-Modal Integration:*

- Combine information from voice, text, and image input for the comprehensive understanding of user queries. A fusion mechanism is employed to merge information from different modalities.

*Natural Language Processing (NLP):*

Use NLP techniques to recognize and interpret user intent from text and voice inputs. This includes activities like Named Entity Recognition (NER), sentiment analysis, and language modelling.

For voice input, convert the recognized speech into text and then apply NLP techniques.

*Dialog Management:*

- Implement a dialog management system to maintain context and manage the flow of conversation.

*1. Task Execution:*

- Carry out the appropriate task based on the understood user intention. This could involve querying a knowledge base, interacting with APIs, or triggering specific actions within the assistant.

*Platform Independence:*

- Use cross-platform development frameworks or technologies to ensure compatibility with various platforms.

*2. Machine learning Models:*

- Develop and deploy machine learning models for voice recognition, image processing, and understanding natural language. This might include pre-trained models for common

tasks or custom-trained models based on specific requirements.

### 3. User Interface (UI):

Develop user interfaces tailored for each platform, ensuring a consistent and user-friendly experience.

### 4. Continuous Learning:

Implement mechanisms for continuous learning and improvement, allowing the AI assistant to adapt to changing user behaviour and preferences.

By adopting a modular and flexible architecture, you can create an AI assistant that seamlessly integrates voice, text, and image inputs across various platforms while providing a responsive and context-aware user experience.

## 7. SYSTEM IMPLEMENTATION

### 7.1. Tools and Technologies Used

**7.1.1. Python:** It is a popular high-level programming language that is very flexible and robust. It works on different platforms (system-independent) and is used in web development to create web applications. Its uncomplicated syntax enables us to write programs in fewer lines, and the code can be executed instantly since this language operated on an interpreter system, facilitating quick and easy prototyping. Its large standard library provides a wide range of modules to perform various tasks as it has huge community support. We have used it as the backend language in our system.

**7.1.2. Tensorflow:** It is an open-source machine learning framework that allows creating dataflow graphs. It's built upon the Python programming language and was launched by Google. It easily integrates with other machine learning frameworks and provides tools and specialized APIs. It is used in our chatbot for lemmatizing.

**7.1.3. Lumibot:** Lumibot is a Python library for algorithmic trading that is used in our system to provide the paper trading functionality. It permits us to create, back test and run trading strategies for various assets such as cryptos and easily converts them into algorithmic trading bots.

**7.1.4. Tkinter:** Tkinter in Python stands out as the quickest and most straightforward method for crafting GUI apps. With Tkinter, crafting a GUI is a breeze, and the visual components

seamlessly blend with the native elements of the operation system, giving the impression that the applications belong to the platform they run on. Tkinter was instrumental in shaping the UI of our AI companion.

**7.1.5. Torch:** It is an open-source machine-learning library and a scientific computing structure. It is used in the trading bot and stable diffusion functionalities of our system.

**7.1.6. Tesseract:** It is an OCR (Optical Character Recognition) for different operating systems. It is used exclusively to extract text data from images and the output is a text file. This has been used in the image scanner functionality of our system.

## 8. TESTING

### 8.1. Unit Testing

Unit testing involves assessing the functionality of individual components within software, such as functions, methods, or procedures, by developers to ensure they operate correctly. These tests are carried using testing frameworks and often involve creating mock objects to simulate dependencies and isolate the unit being tested. The primary goal of unit testing is to validate each unit's correctness and identify defects early in the development cycle, promoting code quality, maintainability, and facilitating easier debugging and refactoring.

### 8.2. Integration Testing

Integration testing is a testing approach in software development that aims to confirm that various modules or systems interact correctly with each other to achieve the desired outcomes. Unlike unit testing, which tests individual components in isolation, integration testing evaluates the interfaces and interactions between integrated components or systems. The primary goal of integration testing is to pinpoint flaws in the connections between components and ensure smooth data transmission between modules or systems. This testing phase can be performed at various levels, such as component integration testing, where individual components are tested together, or system integration testing, where entire systems or subsystems are tested as a whole. Integration testing aims to validate the overall functionality, performance, and reliability of the integrated components, ensuring that they meet the specified requirements and work seamlessly together in a real-world environment.

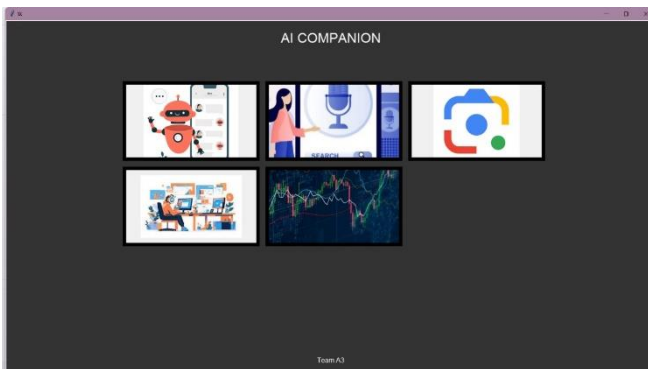


Figure 2: Integration Testing

### 8.3. Yahoo Backtesting

Yahoo Backtesting feature is a tool that lets users evaluate investment strategies using past market data. It allows investors to simulate how a specific portfolio or trading approach would have performed over a chosen timeframe, typically using historical price data for stocks, ETFs, or other financial assets. Users can customize parameters like investment amount, asset allocation, and trading rules to create their own backtesting strategy. The tool calculates hypothetical returns and volatility based on historical data, offering insights into the potential risks and rewards of the tested strategy. However, it's important to recognize the limitations of backtesting, such as not accounting for transaction costs, taxes, and market impact, which may affect the accuracy of the results.

## 9. SCREENSHOTS



Figure 4: Stable diffusion

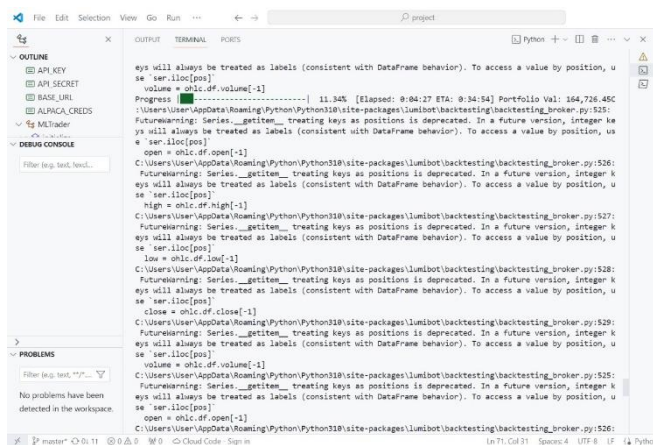


Figure 3: Yahoo BackTesting



Figure 5: Trading Bot showing the MLTrader Compared with SPY

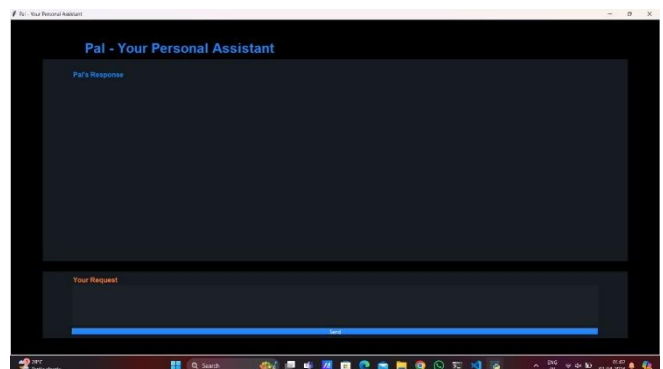


Figure 6: Chatbot



Figure 7: Image to text output using tesseract

## CONCLUSION

The development of an AI assistant capable of processing inputs in the form of voice, text, and images represents a significant stride in human-computer interaction. The integration of Natural Language Processing (NLP), computer vision, and speech recognition technologies enables a versatile and dynamic user experience. The ability to seamlessly transition between these input modalities enhances the adaptability and accessibility of the AI assistant, catering to diverse user preferences and needs.

## REFERENCES

[1] Shreyashkar sharma “Chatbot development using python” International journal of creative research thoughts [IJCRT] Volume 8, 7 July 2020.

[2] Prathysh Jha “Voice Assistance using Python”. In ResearchGate 17 November 2023.

[3] Rohit Tamrakar, Niraj Wani “Design and Development of Chatbot” In ResearchGate 16 May 2021.

[4] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, Jason Weston “Recipes for building an open-domain chatbot” arXiv:2004.13637v2 [cs.CL] 30 April 2020.

[5] Sainath Patil, Akanksha Singh, Ritika Shenoy, Prof. Sainath Patil “Text to Image using Deep Learning” International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 10 Issue 04, April 2021.

[6] Renas Rajab Asaad, Rasan Ismael Ali, Zeravan Arif Ali, Awaz Ahmad Shaban “Image Processing with Python Libraries” Academic Journal of Nawroz University (AJNU), Vol.12, No.2, 2023.

[7] Dr. Nidhi Mishra, Kriti Chandel, Saksham Sardana, Kshittij sharma “Trading vice, Trading Bot for Stock” in International Journal of Novel Research (IJNRD), Volume 8, Issue 2, November 2023.

[8] Ajay Kumar Sahu1, Shivani Dubey, Ashish Kumar Jha, Ritik Bhargava, Priyanshu Priya, Rupa Kumari “Voice Assitant Using Artificial Intelligence”.

[9] Joshua W. Buck, Saverio Perugini, Tam V. Nguyen. “Natural Language, Mixed-initiative Personal Assistant Agents”. International Conference on Ubiquitous Information Management and Communication, January 2018.

[10] Philip R. Doyle, Justin Edwards, Odile Dumbleton, Leigh Clark, and Benjamin R. Cowan. “Mapping Perceptions of Humanness in Intelligent Personal Assistant Interaction”. International Conference on Human-Computer Interaction with Mobile Devices and Services, October 2019.

[11] Sarah Theres Völkel, Samantha Meindl, and Heinrich Hussmann. “Manipulating and Evaluating Levels of Personality Perceptions of Voice Assistants through Enactment-Based Dialogue Design”. Conversational User Interfaces (CUI '21), 27-29, July 2021.

[12] Shivam Singh Sikarwar\*1 “AI Based Voice Assisstant” International Research Journal of Modernization in Engineering Technology and Science Volume:04, Issue:05, May 2022.