**Abstract of the project:**

Jarvis is a virtual assistant created using Python programming language. It is designed to help users perform tasks and make their daily lives easier. The project incorporates several popular technologies such as speech recognition, natural language processing, and machine learning to understand and respond to user requests in real-time. Jarvis can be customized to perform a wide range of tasks, including setting reminders, playing music, and providing weather updates. It is a versatile and efficient tool that offers users a convenient and accessible way to interact with their devices. This abstract provides a brief overview of Jarvis and its features, highlighting its potential to revolutionize the way people interact with technology

JARVIS is an AI-powered voice assistant that can perform a variety of tasks such as sending emails, playing music, making phone calls, and much more. In this project, we use Python programming language and various Python libraries to develop JARVIS. The project involves using speech recognition to understand the user's command and natural language processing to convert the speech into text. The program then performs the required action based on the user's input. The main goal of the project is to make an AI assistant that can understand and respond to the user's commands effectively.

JARVIS (Just A Rather Very Intelligent System) is an artificial intelligence assistant developed using the Python programming language. It is a voice-controlled system that can recognize and respond to user input. The main objective of this project is to develop a conversational agent that can perform various tasks on behalf of the user, such as playing music, opening applications, sending emails, searching the internet, and performing other functions. JARVIS is designed to provide a seamless and personalized user experience through its natural language processing capabilities.

The system is built using various libraries in Python such as SpeechRecognition, PyAudio, pyttsx3, and more. The voice commands are recorded and converted into text using the SpeechRecognition library, and the text response is then converted back into speech using the pyttsx3 library. JARVIS also makes use of various APIs to perform tasks such as sending emails, playing music, and searching the internet.

The user interacts with JARVIS through voice commands, which are recognized and processed by the system. The system then responds to the user’s commands with a voice output. The system is built to recognize and respond to a wide range of voice commands, making it an efficient and versatile assistant.

JARVIS can also be customized to suit the user’s needs, with the ability to add new features and functionalities. The system is designed to be flexible and adaptable, with the potential to integrate with other devices and technologies.

In conclusion, JARVIS is a powerful tool for improving productivity and efficiency in various applications, including personal and professional settings. The system’s natural language processing capabilities and ability to recognize a wide range of voice commands make it a versatile and user-friendly assistant.

**Introduction of the Project:**

In today's digital age, Artificial Intelligence has become an integral part of our lives. With the advent of smart devices and voice assistants, it has become easier to access and perform various tasks. JARVIS, inspired by the character from Iron Man movie series, is a personal assistant designed to help users perform tasks efficiently and effectively. The JARVIS project aims to develop an intelligent and user-friendly personal assistant using the Python programming language.

The purpose of this project is to create a voice-based interface for users to interact with JARVIS. The primary objective of this project is to provide a personalized and intelligent assistant to help users perform tasks such as searching the web, managing schedules, and controlling smart devices with ease. The project also aims to develop a responsive and intuitive user interface to provide an enhanced user experience.

The project team consists of experienced developers and designers who specialize in Python programming language and Artificial Intelligence. Our team aims to create an innovative and efficient personal assistant that can help users manage their tasks and improve productivity.

In the following sections, we will discuss the project's scope, objectives, methodology, technical details, and results in detail. We will also provide insights into the project's potential impact and future development.

JARVIS is a virtual assistant that can perform tasks like sending emails, playing music, searching the internet, and opening applications by responding to voice commands. It is inspired by the virtual assistant used in the iron Man movies. Jarvis stands for **Just A Rather Very Intelligent System**. In this project, we will develop JARVIS using python programming language and its libraries.

The objective of this project is to create a virtual assistant named JARVIS (Just A Rather Very Intelligent System) using Python programming language. JARVIS will be able to perform various tasks such as sending emails, playing music, searching the internet, creating reminders, and much more.

Artificial Intelligence is one of the most popular fields in computer science that is rapidly growing. With the advancement of technology, AI-based systems have become a reality, and the development of AI-powered voice assistants has become a necessity. Voice assistants can perform various tasks, making them an integral part of our daily lives. One such AI-powered voice assistant is JARVIS. JARVIS is an intelligent personal assistant that can understand and respond to user commands. It can perform tasks such as playing music, sending emails, and much more.

Python is a widely used programming language in the field of AI and machine learning. Python offers several libraries and tools for developing AI-based applications. In this project, we will be using Python to develop JARVIS. We will be using various Python libraries such as speech recognition, natural language processing, and web scraping to develop JARVIS.

**Objective of the Project:**

The objective of this project is to develop a virtual assistant that can performs tasks by responding to voice commands. The virtual assistant will be able to perform tasks like sending emails, playing music, searching the internet, and opening applications.

The primary objective of the project is to develop an AI-powered voice assistant that can perform various tasks such as sending emails, playing music, making phone calls, and much more. The project aims to provide an efficient and user-friendly way of interacting with a computer, making the user's life easier and more organized.

JARVIS is a personal assistant application that is inspired by the artificial intelligence character of the same name in the Iron Man movies. This application is built using the Python programming language and can be run on any operating system that supports Python.

The main objective of JARVIS is to provide an efficient and effective way for users to interact with their computer. JARVIS is capable of performing various tasks such as sending emails, playing music, opening applications, and more. Additionally, JARVIS can be customized to suit the needs of the user, allowing for a personalized experience.

One of the key features of JARVIS is its ability to understand and respond to voice commands. This is achieved through the use of various APIs such as Google's Speech Recognition API and the PyAudio library. With this feature, users can simply speak to their computer and JARVIS will understand the command and perform the requested action.

Another important feature of JARVIS is its ability to learn and adapt to the user's preferences. This is accomplished through the use of machine learning algorithms that analyze the user's behavior and usage patterns. With this information, JARVIS can make intelligent decisions about which tasks to prioritize and how to perform them.

Overall, JARVIS is a powerful and versatile personal assistant application that can greatly enhance the user's productivity and efficiency. With its advanced features and customizable interface, JARVIS provides an unparalleled level of interaction between the user and their computer.

The following are the specific objectives of the project:

* To develop a voice recognition system that can understand the user's commands effectively.
* To convert the user's speech into text using natural language processing.
* To perform various tasks such as sending emails, playing music, making phone calls, and much more based on the user's input.
* To develop a web scraping system that can retrieve information from various websites.
* To integrate the voice assistant with other applications such as email, music player, and phone calling applications.
* Implementing machine learning algorithms to enhance JARVIS's capabilities and improve its accuracy in recognizing user commands.
* Testing the system thoroughly to ensure that it is reliable, accurate, and efficient in performing tasks.

**Project Description:**

JARVIS (Just A Rather Very Intelligent System) is a voice-based assistant that uses natural language processing and machine learning techniques to provide personalized assistance to its users. The project aims to develop an assistant that can perform a variety of tasks, such as opening applications, searching the web, sending emails, setting reminders, and more, all through voice commands

**Scope of the Project:**

The scope of the project is to develop a virtual assistant application that can perform various tasks using voice commands. The application will be developed using Python programming language, which is a widely used language for developing applications and scripts.

The application will be designed to perform various tasks like sending emails, searching the web, playing music, setting reminders, and much more. The aim of the project is to make the user’s life easier by reducing the time and effort required to perform these tasks manually.

The project will also include a machine learning algorithm that will learn from the user’s behavior and adapt to their needs over time. The algorithm will analyze the user’s search patterns, frequently used applications, and other relevant information to improve the performance of the virtual assistant.

The application will also have the ability to interact with other applications and devices like smart home devices, mobile devices, and other applications. This will increase the functionality of the application and make it more user-friendly.

The scope of the project also includes the development of a user-friendly interface that will allow users to interact with the application using voice commands. The interface will be designed to provide users with easy access to all the features of the application and allow them to customize the application to their specific needs.

Overall, the scope of the project is to develop a highly functional virtual assistant application that can perform a wide range of tasks, adapt to the user’s needs over time, and provide a user-friendly interface for easy interaction.

The project is to develop an AI-powered voice assistant that can perform various tasks such as sending emails, playing music, making phone calls, and much more. The project involves using speech recognition to understand the user's command and natural language processing to convert the speech into text. The program then performs the required action based on the user's input. The project will be developed using Python programming language and various Python libraries such as speech recognition, natural language processing, and web scraping.

The JARVIS project is designed to be a flexible and adaptable assistant that can be customized to meet the needs of its users. The project is scalable and can be extended to include additional features and capabilities as needed. The project is developed using Python, which is a widely-used programming language that is well-suited to developing AI applications.

**Purpose of the project:**

The purpose of JARVIS (Just A Rather Very Intelligent System) using Python is to provide a virtual assistant that can perform a wide range of tasks and functions through voice commands. The project aims to simplify the daily tasks of the user by automating routine tasks and offering a convenient hands-free experience.

The purpose of the project is to create an AI-powered virtual assistant that can understand natural language commands and respond accordingly. The system uses speech recognition technology to interpret voice commands and perform tasks such as searching the internet, setting reminders, scheduling appointments, and controlling home appliances.

The purpose of the project is to provide a user-friendly interface that can be easily customized to meet the user's requirements. The virtual assistant is designed to be interactive, intelligent, and responsive, allowing the user to have a more personalized and efficient experience. The system also aims to provide security and privacy by ensuring that user data is not shared with any third-party applications or services.

Overall, the purpose of the JARVIS project is to provide a reliable and efficient virtual assistant that can perform a wide range of functions and simplify the daily routine of the user.

The purpose of developing JARVIS using Python is to create a personal assistant that can help users perform various tasks in a more efficient and convenient way. JARVIS can automate tasks such as sending emails, setting reminders, playing music, controlling smart home devices, searching the internet, and much more. The primary aim is to make the user's life easier by providing a virtual assistant that can handle routine tasks, thereby saving them time and effort.

Moreover, JARVIS is designed to be a customizable and extensible platform that can be enhanced with new features and functionalities. It can be adapted to suit the user's specific needs and preferences, making it a versatile tool that can be used in various settings. Additionally, the use of Python programming language ensures that JARVIS is easy to maintain, extend, and update, making it a robust and scalable solution for personal assistance.

The purpose of the JARVIS project is to create an assistant that can help users be more productive and efficient by performing routine tasks quickly and accurately. The project is designed to be user-friendly and accessible, so that anyone can use it regardless of their technical expertise.

**Methodology:**

The project will use Python libraries such as speech recognition and text-to-speech conversion to provide voice recognition capabilities. It will also use various APIs to enable web search, email sending, and other functionalities. The JARVIS application will be designed as a console application with a **Graphical User Interface (GUI)** that can be interacted with using voice commands.

The project will be developed using the Agile methodology. The Agile methodology is a flexible and iterative approach to software development. It involves breaking down the project into smaller tasks and completing them in short iterations. The Agile methodology promotes collaboration between the development team and the client to ensure that the project meets the client's requirements.

The methodology used for the JARVIS project includes the following steps:

**Analysis:** In this phase, the project requirements and objectives were analyzed. The target audience was identified, and the scope of the project was defined.

**Design:** In this phase, the design of the system was created. The design included the structure of the program, the features to be included, and the data sources that would be utilized. The user interface was also designed.

**Development:** The development phase involved the creation of the JARVIS system. The Python programming language was used to develop the program. The required libraries were imported, and the code was written.

**Testing:** In this phase, the program was tested to ensure that it worked correctly. Various scenarios were tested, and the program was debugged.

**Implementation:** In this phase, the program was deployed to the target audience. The necessary hardware and software were installed, and the program was made available to the end-users.

The approach used in the JARVIS project was iterative and incremental. This means that the development process was broken down into smaller chunks or iterations, and each iteration was tested before moving onto the next one. This approach allowed for the early identification and correction of errors and ensured that the final product met the project objectives.

The design and development of JARVIS were carried out using agile methodologies. The agile methodology is an iterative and collaborative approach that involves the continuous feedback and involvement of the end-users. This approach ensured that the JARVIS system met the needs of the end-users and was easy to use.

Overall, the methodology used for the JARVIS project was effective in achieving the project objectives. The iterative and incremental approach allowed for the early identification and correction of errors, while the agile methodology ensured that the system met the needs of the end-users.

**Technical Details:**

**Programming Language:**

JARVIS is developed using the Python programming language. Python is an interpreted, high-level, general-purpose programming language that is easy to read and write.

**Software Requirements:**

1. Python 3.x or higher
2. PyAudio module
3. SpeechRecognition module
4. pyttsx3 module
5. datetime module
6. webbrowser module
7. os module
8. smtplib module

**Hardware Requirements:**

1. Microphone
2. Speakers or headphones
3. Internet connectivity

**Data Sources:**

JARVIS retrieves data from the following sources:

1. User input from microphone
2. Online search engines (e.g. Google)
3. Local file system (e.g. opening files, creating files, deleting files)
4. Email accounts (e.g. sending emails, receiving emails)

JARVIS uses the PyAudio module to record audio input from the microphone. The SpeechRecognition module is used to recognize the speech input and convert it to text. The pyttsx3 module is used to convert text to speech for output. The datetime module is used to retrieve the current date and time. The webbrowser module is used to open websites in the default web browser. The os module is used to perform file system operations such as creating, deleting, and modifying files. The smtplib module is used to send emails.

In terms of hardware requirements, JARVIS requires a microphone for speech input and speakers or headphones for speech output. Additionally, an internet connection is required to access online search engines and send and receive emails.

**Result :**

The JARVIS project successfully achieved the objective of creating a personal assistant program that can understand natural language commands and perform tasks based on those commands.

The following are the results of the project:

**Speech recognition:**

The program was able to accurately recognize spoken commands and convert them into text. This was achieved using the speech\_recognition library in Python.

**Natural Language Processing (NLP):**

The program was able to understand the intent of the user's command using NLP techniques such as tokenization, part-of-speech tagging, and named entity recognition. This was achieved using the Natural Language Toolkit (NLTK) library in Python.

**Task automation:**

The program was able to perform various tasks such as sending emails, opening websites, playing music, and telling the weather based on the user's commands. This was achieved using various libraries such as smtplib, webbrowser, and pyowm in Python.

**User interface:**

The program had a simple and user-friendly interface that allowed the user to interact with JARVIS through speech or text commands.

**Results visualization:**

The program generated visual outputs such as graphs and charts to display information such as weather forecast and stock prices. This was achieved using the Matplotlib library in Python.

Overall, the results of the project met the objectives by providing a functional personal assistant program that can perform tasks based on natural language commands. The program was able to accurately recognize spoken commands, understand the intent of the user's commands using NLP techniques, and perform various tasks such as sending emails, opening websites, playing music, and telling the weather. The user interface was simple and user-friendly, and the program generated visual outputs to display information.

**Conclusion:**

The JARVIS project is an artificial intelligence-based virtual assistant that aims to automate several tasks and provide a seamless experience to its users. The purpose of this project was to develop a virtual assistant that can perform multiple tasks, including setting alarms, making phone calls, sending messages, playing music, and opening applications, among others.

The project's scope included the development of the JARVIS virtual assistant using the Python programming language. The project's objectives were to design and implement a user-friendly interface, integrate various third-party APIs and services, and create a robust system that can handle multiple user requests simultaneously.

To achieve the project objectives, the project was developed using Python and various third-party libraries and modules. The software and hardware requirements included a computer with Python 3.x installed, an internet connection, and the necessary modules and libraries such as Speech Recognition, PyAudio, and Pyttsx3.

Data sources for the project were the various APIs and services used, including OpenAI's GPT-3, which was used for natural language processing and generating responses.

The project results were successful in achieving the project objectives. The JARVIS virtual assistant was designed and implemented with a user-friendly interface that can perform multiple tasks such as setting alarms, making phone calls, sending messages, playing music, and opening applications, among others. The system was also able to handle multiple user requests simultaneously, making it a robust and reliable virtual assistant.

The project's impact is significant as it provides a convenient and efficient way for users to perform tasks using voice commands. The JARVIS virtual assistant can potentially be developed further to include more features and functionality, making it an even more useful tool for users.

In conclusion, the JARVIS project was successful in developing an artificial intelligence-based virtual assistant that can perform multiple tasks and provide a seamless experience to its users. The project's potential for future development makes it an exciting and valuable tool for users seeking a convenient and efficient way to perform tasks using voice commands.

**References:**

1. Python. (n.d.). Retrieved from **https://www.python.org/**
2. SpeechRecognition: Library for performing speech recognition, with support for several engines and APIs, online and offline. (n.d.). Retrieved from **https://github.com/Uberi/speech\_recognition**
3. pywin32: Python extensions for Microsoft Windows. (n.d.). Retrieved from **https://github.com/mhammond/pywin32**
4. OpenAI. (n.d.). Retrieved from **https://openai.com/**
5. Pandas: Python Data Analysis Library. (n.d.). Retrieved from **https://pandas.pydata.org/**
6. Matplotlib: Python plotting. (n.d.). Retrieved from **https://matplotlib.org/**
7. NumPy: The fundamental package for scientific computing with Python. (n.d.). Retrieved from **https://numpy.org/**
8. TensorFlow. (n.d.). Retrieved from **https://www.tensorflow.org/**
9. Keras: The Python Deep Learning library. (n.d.). Retrieved from **https://keras.io/**
10. Sklearn: Machine Learning in Python. (n.d.). Retrieved from [**https://scikit-learn.org/stable/**](https://scikit-learn.org/stable/)

**Appendices:**

**Data sets:**

The data sets used in JARVIS can vary depending on the specific functionalities implemented in the project. Here are some examples of data sets that could be used:

Natural language processing data sets, such as the Stanford Sentiment Treebank or the Movie Review Data set, used for sentiment analysis and language processing.

Speech recognition data sets, such as the VoxCeleb or Common Voice data sets, used for speech recognition and transcription.

Weather and climate data sets, such as those provided by the National Oceanic and Atmospheric Administration (NOAA), used for weather forecasting and analysis.

Stock market and financial data sets, such as those provided by Yahoo Finance or Google Finance, used for financial analysis and prediction.

Image and video data sets, such as the ImageNet or CIFAR-10 data sets, used for image recognition and analysis.

It is important to note that the data sets used in JARVIS should be relevant to the specific functionalities implemented and should be properly licensed for use.

**Source Code:**

# pip install pyaudio

import pyttsx3 #pip install pyttsx3

import speech\_recognition as sr #pip install speechRecognition

import datetime

import wikipedia #pip install wikipedia

import webbrowser

import os

import smtplib

engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices')

# print(voices[1].id)

engine.setProperty('voice', voices[0].id)

def speak(audio):

engine.say(audio)

engine.runAndWait()

def wishMe():

hour = int(datetime.datetime.now().hour)

if hour>=0 and hour<12:

speak("Good Morning!")

elif hour>=12 and hour<18:

speak("Good Afternoon!")

else:

speak("Good Evening!")

speak("I am Your personal assistant Sir. Please tell me how may I help you")

def takeCommand():

#It takes microphone input from the user and returns string output

r = sr.Recognizer()

with sr.Microphone() as source:

print("Listening...")

r.pause\_threshold = 1

audio = r.listen(source)

try:

print("Recognizing...")

query = r.recognize\_google(audio, language='en-in')

print(f"User said: {query}\n")

except Exception as e:

# print(e)

print("Say that again please...")

return "None"

return query

def sendEmail(to, content):

server = smtplib.SMTP('smtp.gmail.com', 587)

server.ehlo()

server.starttls()

server.login('youremail@gmail.com', 'your-password')

server.sendmail('youremail@gmail.com', to, content)

server.close()

if \_name\_ == "\_main\_":

wishMe()

while True:

# if 1:

query = takeCommand().lower()

# Logic for executing tasks based on query

if 'wikipedia' in query:

speak('Searching Wikipedia...')

query = query.replace("wikipedia", "")

results = wikipedia.summary(query, sentences=2)

speak("According to Wikipedia")

print(results)

speak(results)

elif 'open youtube' in query:

webbrowser.open("youtube.com")

elif 'open google' in query:

webbrowser.open("google.com")

elif 'open stackoverflow' in query:

webbrowser.open("stackoverflow.com")

elif 'play music' in query:

music\_dir = 'D:\\Non Critical\\songs\\Favorite Songs2'

songs = os.listdir(music\_dir)

print(songs)

os.startfile(os.path.join(music\_dir, songs[0]))

elif 'the time' in query:

strTime = datetime.datetime.now().strftime("%H:%M:%S")

speak(f"Sir, the time is {strTime}")

elif 'open code' in query:

codePath = "C:\\Users\\Haris\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"

os.startfile(codePath)

elif 'email to harry' in query:

try:

speak("What should I say?")

content = takeCommand()

to = ""

sendEmail(to, content)

speak("Email has been sent!")

except Exception as e:

print(e)

speak("Sorry my friend bhai. I am not able to send this email")

else:

print("No query matched")

**ScreenShots:-**

