JARVIS - THE VIRTUAL ASSISTANT

Submitted in partial fulfillment of the requirement
for the award of the Degree of

Bachelor of Technology
in
Computer Engineering

by

Mayur Krishna Solankar
Manish Shashikant Jadhav
Vishesh Bhimji Savani

under the guidance of

Dr. Prasenjit Bhavathankar

Department of Computer Engineering
Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
(Autonomous Institute Affiliated to University of Mumbai)
Munshi Nagar, Andheri-West, Mumbai-400058
University of Mumbai
May 2024

Project Approval Certificate

This is to certify that the Project entitled "JARVIS - The Virtual Assistant" by Mr. Mayur Krishna Solankar, Mr. Manish Shashikant Jadhav and Mr. Vishesh Bhimji Savani is found to be satisfactory and is partially approved for the award of Degree of Bachelor of Technology in Computer Engineering from University of Mumbai.

External Examiner	**Internal Examiner**
(signature)	**(signature)**
Name:	**Name:**
Date:	**Date:**

Seal of the Institute

Contents

List of Figures

Acknowledgments

We would like to express our heartfelt gratitude to all those who have contributed to the completion of the JARVIS project. Our team has diligently worked towards a systematic approach to the project, focusing on achieving our goals effectively.

First and foremost, We are deeply thankful to Dr. Prasenjit Bhavathankar for his invaluable guidance and encouragement throughout this project. His support and mentorship have been instrumental in shaping this work.

We extend our thanks to our mentor, Mr. Atman Pandya, for his insightful advice and feedback.

Additionally, we express our thanks to the staff of the Computer Engineering Department for their support and for providing all the necessary facilities for this research work.

Mayur Krishna Solankar

Manish Shashikant Jadhav

Vishesh Bhimji Savani

Abstract

The JARVIS (Just A Rather Very Intelligent System) project is a sophisticated virtual assistant designed to enhance user interaction and experience through seamless communication. Developed as a Second Year Mini Project, this project integrates advanced language processing algorithms to understand and respond to user queries effectively.

One of the key features of JARVIS is its ability to provide users with a range of functionalities, including setting reminders, retrieving information, and controlling smart devices, making it a versatile and valuable tool in daily life. The project's focus on integrating language processing algorithms for seamless communication demonstrates its commitment to enhancing user experience.

Overall, the JARVIS project represents a significant advancement in virtual assistant technology, showcasing innovative features and a user-centric design approach.

Chapter 1

Introduction

In the area of artificial intelligence and human machine interaction, JARVIS is a very good sample to illustrate, how technology may be utilized to achieve a smooth and natural user experience. Dubbed after the AI character of Tony Stark's movie franchise, Marvel Cinematic Universe, the JARVIS Project is aimed at creating a virtual assistant that can assist users with simple tasks to complex interactions, while maintaining a very high level of intelligence and adaptability.

The chief objective of our JARVIS project is to develop a virtual assistant that is able to comprehend and reply to natural language, thus making it less difficult for users to operate their devices through voice commands. It utilizes cutting-edge speech recognition and natural language processing technologies to accurately convert human speech to written text and then provide relevant responses right when it is needed.

Another crucial aspect of the JARVIS project is its seamless integration with messaging apps, particularly WhatsApp. By integrating with WhatsApp, JARVIS can send messages on whatsapp directly through voice commands.

In addition to its core functionality, the JARVIS project also aims to explore new and innovative ways to enhance the user experience. This includes incorporating features such as image capturing, which allows JARVIS to take photos and share them with the user.

Overall, the JARVIS project represents a significant step forward in the field of virtual assistants and artificial intelligence. By combining cutting-edge technologies with a user-centric design approach, JARVIS is poised to revolutionize the way we interact with technology and usher in a new era of intelligent computing.

1.1 Motivation

The motivation behind the creation and development of the JARVIS (Just A Rather Very Intelligent System) project stems from a combination of technological innovation, practical utility, and a desire to enhance human-computer interaction. Below are some of the key motivations that drive the project forward:

- **Innovation:** Drive to innovate and explore new possibilities in artificial intelligence and human-computer interaction.

- **User Experience Enhancement:** Desire to enhance user experiences by providing a more intuitive and efficient way to interact with devices.

- **Addressing Real-World Needs:** Aim to address real-world needs and challenges faced by users in their daily lives.
- **Simplifying Tasks:** Focus on simplifying everyday tasks for users through intelligent virtual assistant capabilities.
- **Practical Utility:** Ensuring that features and functionalities implemented are meaningful and impactful for users.
- **User-Centric Design:** Designing JARVIS with a user-centric approach to prioritize user needs and preferences.

1.2 Objectives

The scope of our project, JARVIS (Just A Rather Very Intelligent System), includes the development of a virtual assistant system that can seamlessly communicate in English. The system will incorporate advanced language processing algorithms to understand user queries and provide accurate responses. Additionally, JARVIS will feature a user-friendly interface and intuitive design to streamline daily tasks and improve productivity.

- To seamlessly communicate for response capabilities in the JARVIS project.
- To enable effortless execution of common OS tasks.
- To facilitate quick access and navigation of social media platforms via voice commands within JARVIS.
- To design an intuitive and user-friendly graphical interface for JARVIS, enhancing overall user experience.

1.3 Problem Statement

The objective of the JARVIS project is to develop an advanced virtual assistant that employs state-of-the-art natural language processing and speech recognition technologies. The primary challenge is to create a system capable of accurately understanding user commands and providing relevant responses in real-time, while also integrating seamlessly with various applications and services to enhance user productivity.

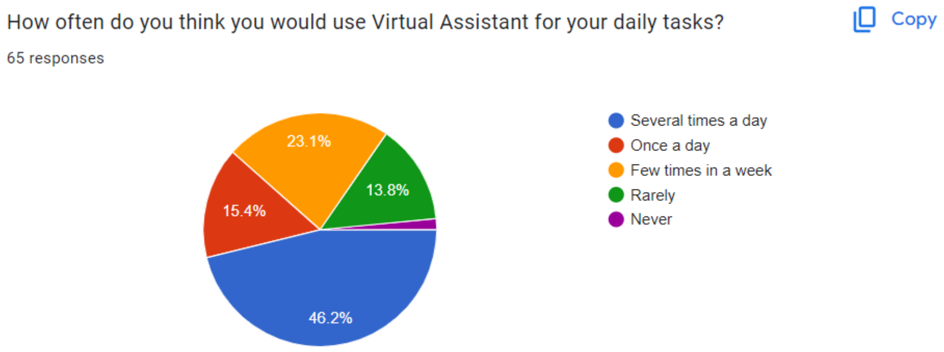
Additionally, the project aims to explore innovative features such as image capturing to expand the assistant's capabilities. Overall, the goal is to revolutionize human-computer interaction by creating an intelligent and intuitive virtual assistant that simplifies and enhances the way users interact with technology.

Chapter 2

Literature Survey

The market survey conducted on Google Forms received 65 responses, providing valuable insights into user preferences and expectations regarding virtual assistants.

1. Frequency of Use: As shown in 2.1 the survey revealed that users are inclined to use virtual assistants frequently for their daily tasks. This highlights the growing reliance on such technology for enhancing productivity and efficiency in daily activities.

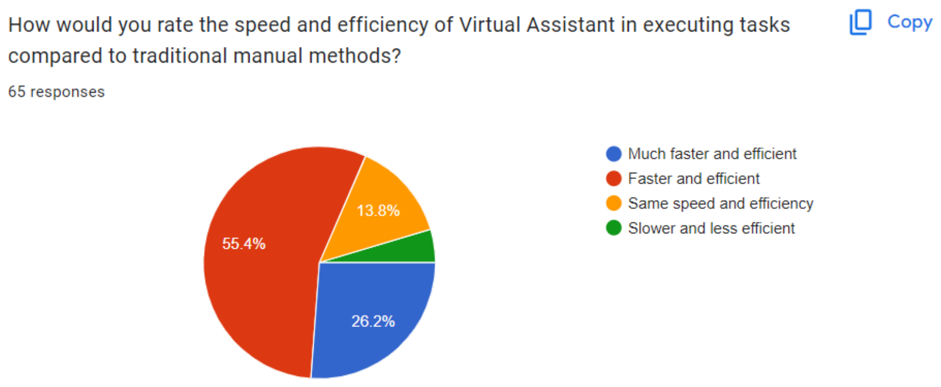


How often do you think you would use Virtual Assistant for your daily tasks?

65 responses

Figure 2.1: Survey related to Frequency of use of JARVIS project

2. Speed and Efficiency: Participants generally rated the speed and efficiency of virtual assistants positively compared to traditional manual methods. 2.2 indicates a perceived advantage of using virtual assistants for executing tasks, emphasizing their potential to streamline processes and save time.

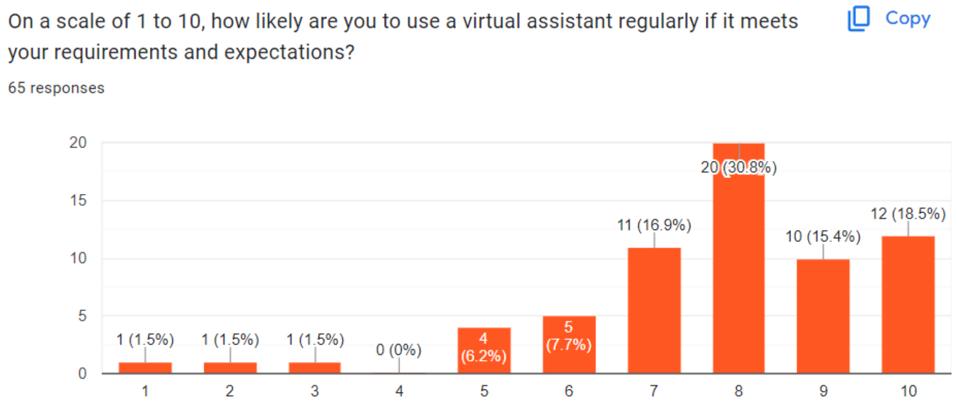


How would you rate the speed and efficiency of Virtual Assistant in executing tasks compared to traditional manual methods?

65 responses

Figure 2.2: Survey related to Speed and Efficieny of Jarvis Project.

3. Likelihood of Regular Use: Participants were also asked to rate, on a scale of 1 to 10, how likely they would be to use a virtual assistant regularly if it met their requirements and expectations as shown in 2.3. This question aimed to measure the potential adoption rate of virtual assistants among users.



On a scale of 1 to 10, how likely are you to use a virtual assistant regularly if it meets your requirements and expectations?

65 responses

Figure 2.3: Survey related to usage of project if it meets user requirements.

Overall, the survey results provided valuable insights into user preferences and expectations regarding virtual assistants, helping to inform the development and implementation of the JARVIS. project. These insights can inform the development and improvement of virtual assistant technologies to better meet user expectations and needs.

Chapter 3

Design

3.1 Architecture Diagram

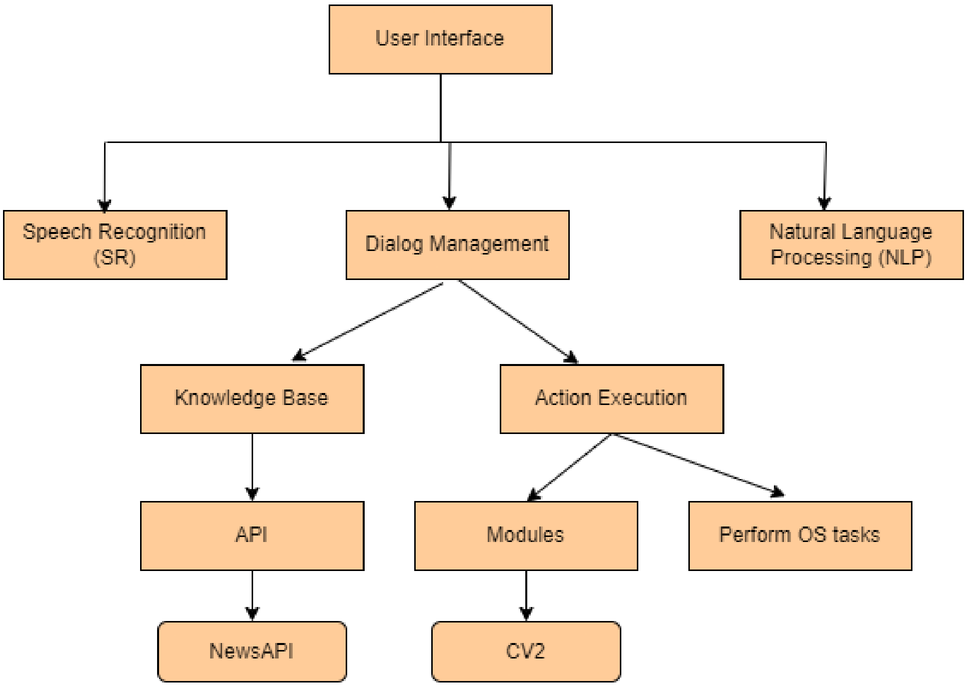


Figure 3.1: Architecture diagram of JARVIS

The architecture diagram of the JARVIS project, as depicted in Fig. 3.1, illustrates the system's design and structure, showcasing the integration of various modules and components essential for its functionality.

The 3.1 showcases the integration of various modules and components essential for its functionality.

- **User Interface:** This module is responsible for interacting with the user, likely through a graphical interface or a voice assistant.

- **Speech Recognition (SR):** This module is responsible for recognizing and transcribing user speech into text. This is likely done using a speech-to-text engine.

- **Dialog Management:** This module is responsible for understanding the context and intent behind the user's input (text or speech). It determines the appropriate response or action to take.

- **Natural Language Processing (NLP):** This module is responsible for analyzing and understanding the meaning of the user's input. It likely uses NLP techniques such as tokenization, entity recognition, and sentiment analysis.

- **Knowledge Base:** This module stores and manages the system's knowledge and information. It provides the necessary data for the system to respond accurately to user queries.

- **Action Execution:** This module is responsible for executing the actions determined by the Dialog Management module. This could include performing OS tasks, retrieving news articles, or executing other system-level actions.

- **Perform OS tasks:** This module is responsible for executing operating system-level tasks, such as file management, process control, or system configuration.

- **NewsAPI:** This module is likely responsible for retrieving news articles or information from a news API.

- **CV2:** This module is responsible for camera actions and clicking photos. It likely uses computer vision techniques to process images and perform actions based on the user's input.

3.2 Algorithms Used

For the JARVIS project, several algorithms and techniques were employed. Speech Recognition converted spoken commands into text. Natural Language Understanding (NLU) deciphered user commands using keyword matching, regular expressions, and semantic analysis. Knowledge Base Retrieval fetched information from an API. A Camera Module used the CV2 (OpenCV) module to capture images. The User Interface included start and stop buttons, error handling, and prompts. The system performed operating system tasks like file operations and system commands, enhancing its utility.

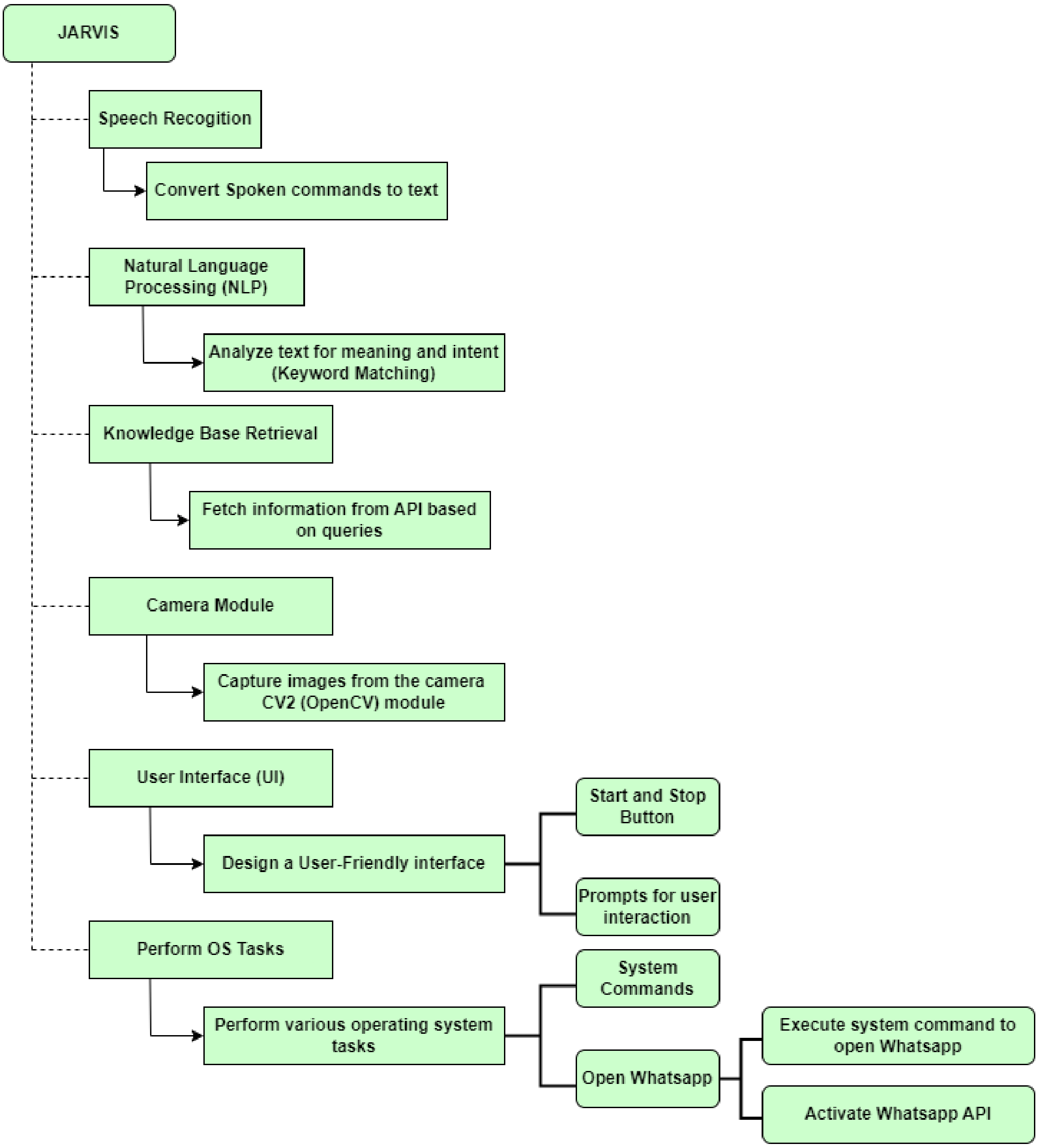

JARVIS
Speech Recogition
Convert Spoken commands to text
Natural Language Processing (NLP)
Analyze text for meaning and intent (Keyword Matching)
Knowledge Base Retrieval
Fetch information from API based on queries
Camera Module
Capture images from the camera CV2 (OpenCV) module
User Interface (UI)
Design a User-Friendly interface
Start and Stop Button
Prompts for user interaction
Perform OS Tasks
Perform various operating system tasks
System Commands
Open Whatsapp
Execute system command to open Whatsapp
Activate Whatsapp API

Figure 3.2: Algorithms and Modules used in JARVIS

3.3 Project Plan

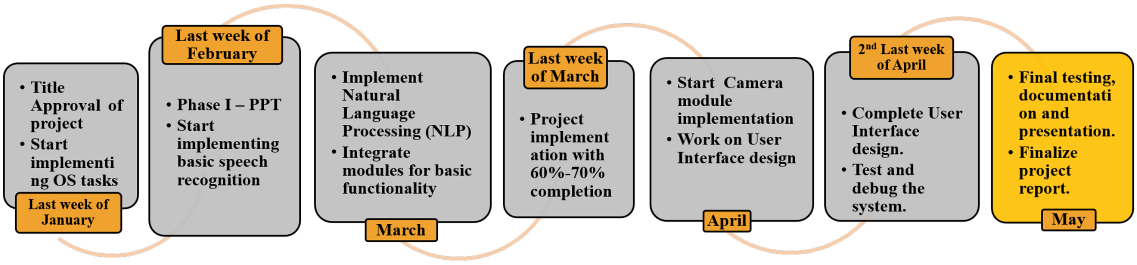

• Title Approval of project
• Start implementing OS tasks
Last week of January
Last week of February
• Phase I – PPT
• Start implementing basic speech recognition
• Implement Natural Language Processing (NLP)
• Integrate modules for basic functionality
March
Last week of March
• Project implementation with 60%-70% completion
• Start Camera module implementation
• Work on User Interface design
April
2nd Last week of April
• Complete User Interface design.
• Test and debug the system.
• Final testing, documentation and presentation.
• Finalize project report.
May

Figure 3.3: Algorithms and Modules used in JARVIS

- Implemented basic speech recognition.
- Tested and refined speech recognition accuracy.
- Integrated OS task execution with user commands.
- Developed system commands for opening WhatsApp and sending messages.
- Set up API integration for knowledge base queries like News API.
- Designed response processing system for API data.
- Integrated CV2 module for image capturing.
- Tested camera module for image quality and resolution.
- Designed user-friendly interface with start and stop buttons.
- Implemented error handling and user prompts.
- Integrated all modules into the JARVIS system.
- Conducted comprehensive testing of the entire system.
- Prepared detailed documentation of the project.
- Created a presentation highlighting project features and achievements.

Chapter 4

Implementation

4.1 Details of 100% of implementation completed

The JARVIS project implementation was successfully completed, achieving all planned functionalities. The system incorporated modules for speech recognition, natural language understanding, knowledge base retrieval, camera actions, and user interface controls. The speech recognition accurately converted spoken commands into text, while the natural language understanding module effectively deciphered user intent. The knowledge base retrieval module fetched information from an API, and the camera module captured images using the CV2 module. The user interface was designed with intuitive controls for enhanced user interaction.

4.2 Techstack Used

The JARVIS project's backend development was primarily conducted in Python, a versatile and powerful programming language known for its readability and ease of use. Python's extensive libraries and frameworks were instrumental in implementing various functionalities.

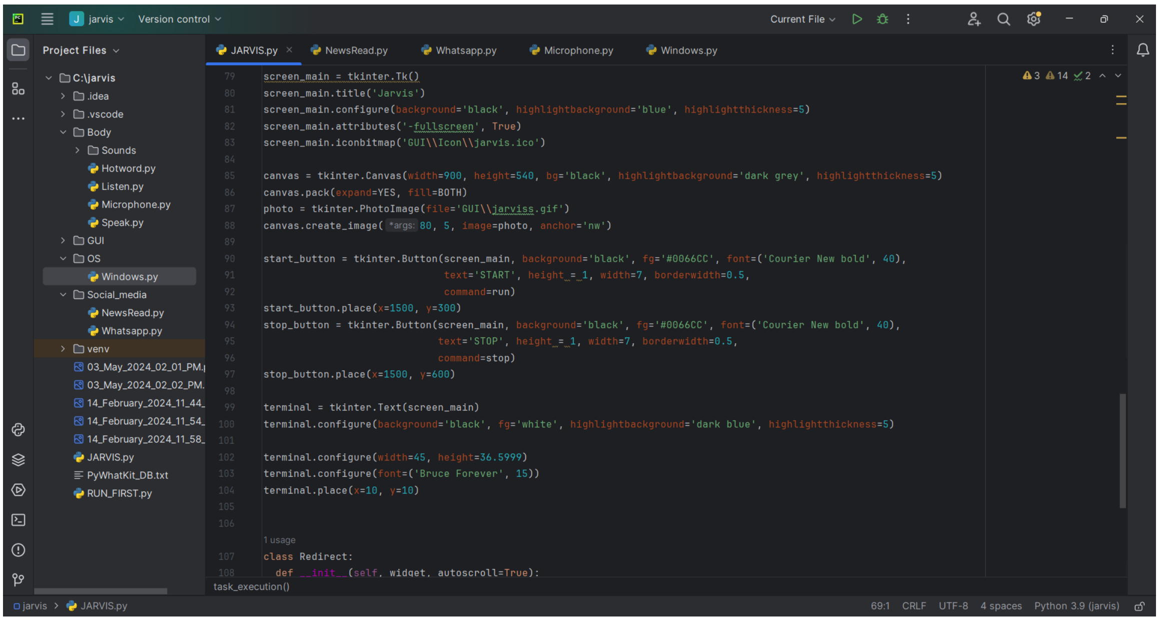
For the frontend interface, Tkinter, a standard GUI toolkit for Python, was used. Tkinter provided a simple yet effective way to design and implement the graphical user interface (GUI) of the JARVIS application. The use of Tkinter ensured that the interface was user-friendly and intuitive.

4.3 API Used

The JARVIS project utilized APIs for knowledge base retrieval and interfacing with external services. The knowledge base API was a crucial component, providing access to a vast repository of information that could be queried based on user input. The API was accessed using HTTP requests, with the response containing relevant information that was processed and presented to the user. Additionally, the project integrated with external services through APIs to enhance its functionality. These APIs enabled JARVIS to perform tasks such as retrieving real-time data, accessing external databases, and interfacing with other applications. By leveraging APIs, the project was able to expand its capabilities and provide users with a more comprehensive and integrated experience.

4.4 Code Snippets

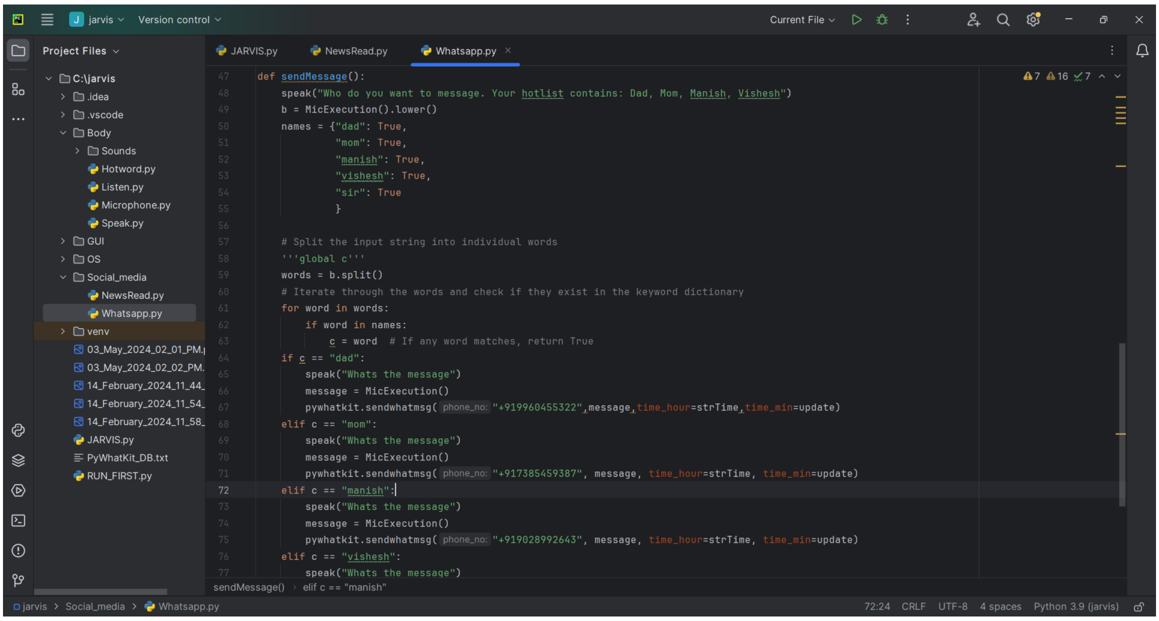
1. Jarvis File:



```python
screen_main = tkinter.Tk()
screen_main.title('Jarvis')
screen_main.configure(background='black', highlightbackground='blue', highlightthickness=5)
screen_main.attributes('-fullscreen', True)
screen_main.iconbitmap('GUI\\Icon\\jarvis.ico')

canvas = tkinter.Canvas(width=900, height=540, bg='black', highlightbackground='dark grey', highlightthickness=5)
canvas.pack(expand=YES, fill=BOTH)
photo = tkinter.PhotoImage(file='GUI\\jarviss.gif')
canvas.create_image( *args: 80, 5, image=photo, anchor='nw')

start_button = tkinter.Button(screen_main, background='black', fg='#0066CC', font=('Courier New bold', 40),
                              text='START', height = 1, width=7, borderwidth=0.5,
                              command=run)
start_button.place(x=1500, y=300)
stop_button = tkinter.Button(screen_main, background='black', fg='#0066CC', font=('Courier New bold', 40),
                             text='STOP', height = 1, width=7, borderwidth=0.5,
                             command=stop)
stop_button.place(x=1500, y=600)

terminal = tkinter.Text(screen_main)
terminal.configure(background='black', fg='white', highlightbackground='dark blue', highlightthickness=5)

terminal.configure(width=45, height=36.5999)
terminal.configure(font=('Bruce Forever', 15))
terminal.place(x=10, y=10)


1 usage
class Redirect:
  def __init__(self, widget, autoscroll=True):
```

Figure 4.1: Code snippets of main jarvis file.

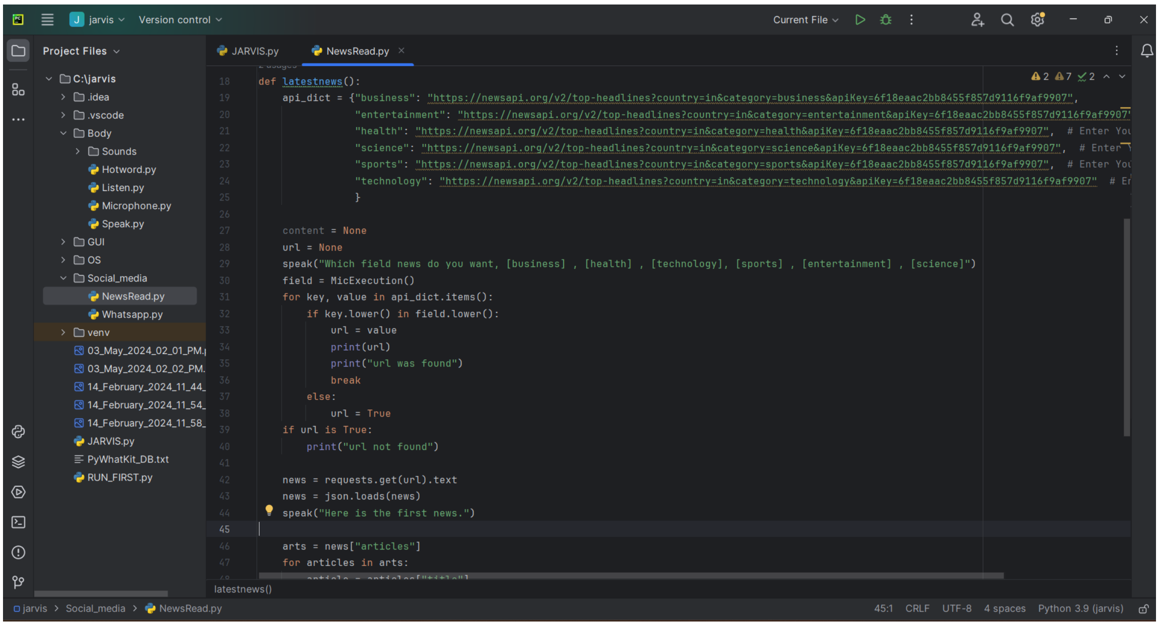
2. Whatsapp File:



```python
def sendMessage():
    speak("Who do you want to message. Your hotlist contains: Dad, Mom, Manish, Vishesh")
    b = MicExecution().lower()
    names = {"dad": True,
             "mom": True,
             "manish": True,
             "vishesh": True,
             "sir": True
             }

    # Split the input string into individual words
    '''global c'''
    words = b.split()
    # Iterate through the words and check if they exist in the keyword dictionary
    for word in words:
        if word in names:
            c = word  # If any word matches, return True
    if c == "dad":
        speak("Whats the message")
        message = MicExecution()
        pywhatkit.sendwhatmsg( phone_no: "+919960455322",message,time_hour=strTime,time_min=update)
    elif c == "mom":
        speak("Whats the message")
        message = MicExecution()
        pywhatkit.sendwhatmsg( phone_no: "+917385459387", message, time_hour=strTime, time_min=update)
    elif c == "manish":
        speak("Whats the message")
        message = MicExecution()
        pywhatkit.sendwhatmsg( phone_no: "+919028992643", message, time_hour=strTime, time_min=update)
    elif c == "vishesh":
        speak("Whats the message")
```

Figure 4.2: Code snippets of Whatsapp Module file.

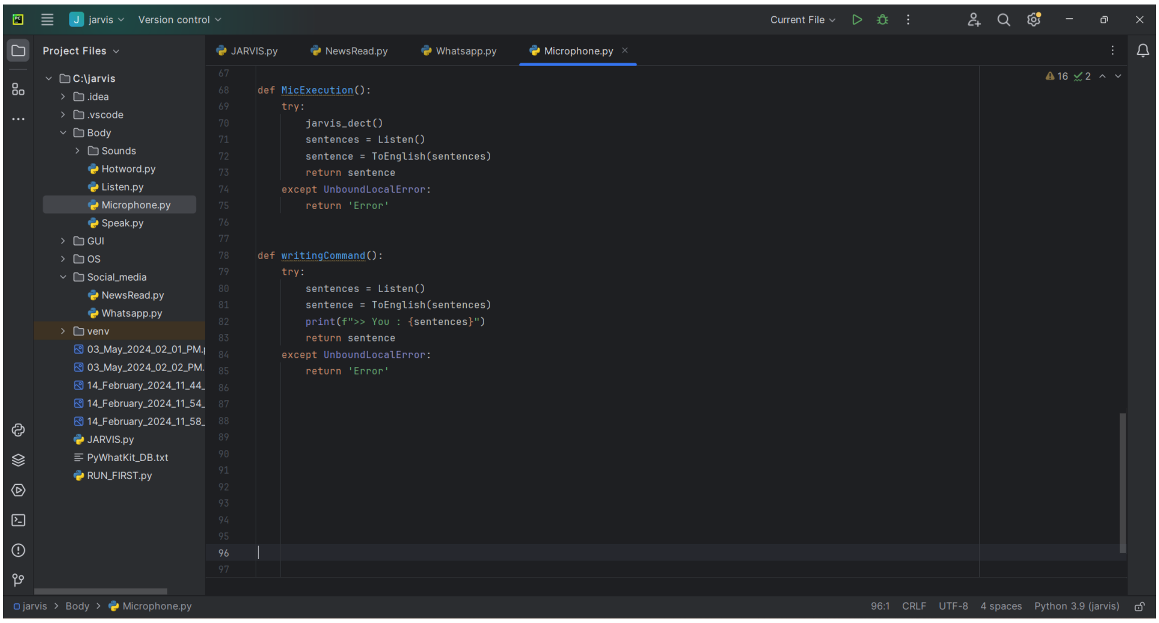
3. News API File:



```
def latestnews():
    api_dict = {"business": "https://newsapi.org/v2/top-headlines?country=in&category=business&apiKey=6f18eaac2bb8455f857d9116f9af9907",
                "entertainment": "https://newsapi.org/v2/top-headlines?country=in&category=entertainment&apiKey=6f18eaac2bb8455f857d9116f9af9907'
                "health": "https://newsapi.org/v2/top-headlines?country=in&category=health&apiKey=6f18eaac2bb8455f857d9116f9af9907",  # Enter You
                "science": "https://newsapi.org/v2/top-headlines?country=in&category=science&apiKey=6f18eaac2bb8455f857d9116f9af9907",  # Enter Y
                "sports": "https://newsapi.org/v2/top-headlines?country=in&category=sports&apiKey=6f18eaac2bb8455f857d9116f9af9907",  # Enter You
                "technology": "https://newsapi.org/v2/top-headlines?country=in&category=technology&apiKey=6f18eaac2bb8455f857d9116f9af9907"  # En
                }

    content = None
    url = None
    speak("Which field news do you want, [business] , [health] , [technology], [sports] , [entertainment] , [science]")
    field = MicExecution()
    for key, value in api_dict.items():
        if key.lower() in field.lower():
            url = value
            print(url)
            print("url was found")
            break
        else:
            url = True
    if url is True:
        print("url not found")

    news = requests.get(url).text
    news = json.loads(news)
    speak("Here is the first news.")

    arts = news["articles"]
    for articles in arts:
        article = articles["title"]
```

Figure 4.3: Code snippets of News API file.

4. Microphone Execution File:



```
def MicExecution():
    try:
        jarvis_dect()
        sentences = Listen()
        sentence = ToEnglish(sentences)
        return sentence
    except UnboundLocalError:
        return 'Error'


def writingCommand():
    try:
        sentences = Listen()
        sentence = ToEnglish(sentences)
        print(f">> You : {sentences}")
        return sentence
    except UnboundLocalError:
        return 'Error'
```

Figure 4.4: Code snippets of Microphone Execution file.

Chapter 5

Results and Discussion

5.1 Results

The success of the JARVIS project can be attributed to its meticulous planning, robust module integration, and user-centric design. The speech recognition module accurately converts spoken commands into text, enabling seamless interaction with the system. This module's accuracy is crucial for natural and effortless communication between users and the AI assistant.

The natural language understanding module is equally important, as it effectively deciphers user commands using advanced techniques such as keyword matching and semantic analysis. This ensures that the system understands user intent accurately, leading to more meaningful interactions. Additionally, the knowledge base retrieval module enhances the system's capabilities by fetching relevant information from an API based on user queries.

The integration of the CV2 (OpenCV) module for the camera adds another dimension to the system's functionality, allowing it to capture images and potentially analyze visual information in the future. This expands JARVIS's capabilities beyond voice interaction, making it more versatile and useful in various scenarios.

In addition to its core functionalities, the JARVIS project also integrated with WhatsApp for messaging capabilities, allowing users to send messages using voice commands. This feature adds a practical and convenient communication aspect to the assistant, enhancing its usefulness in daily tasks.

Furthermore, the project included modules for performing various operating system tasks, such as take screenshot, lock/shutdown PC, etc. This functionality expands JARVIS' utility beyond simple assistant tasks, making it a more comprehensive tool for users to manage their digital environment.

The integration of these additional features highlights the project's versatility and adaptability, showcasing its potential applications in various domains. By incorporating messaging capabilities and extending its functionality to include OS tasks, JARVIS demonstrates its effectiveness as a multifunctional assistant capable of enhancing user productivity and convenience.

Overall, the user interface design plays a crucial role in enhancing the user experience. The intuitive controls, error handling, and prompts ensure that the system is easy to use and accessible to all users. The success of the JARVIS project demonstrates the feasibility and effectiveness of building voice-controlled assistants with advanced functionalities.

5.2 Screenshots

1. Main GUI:

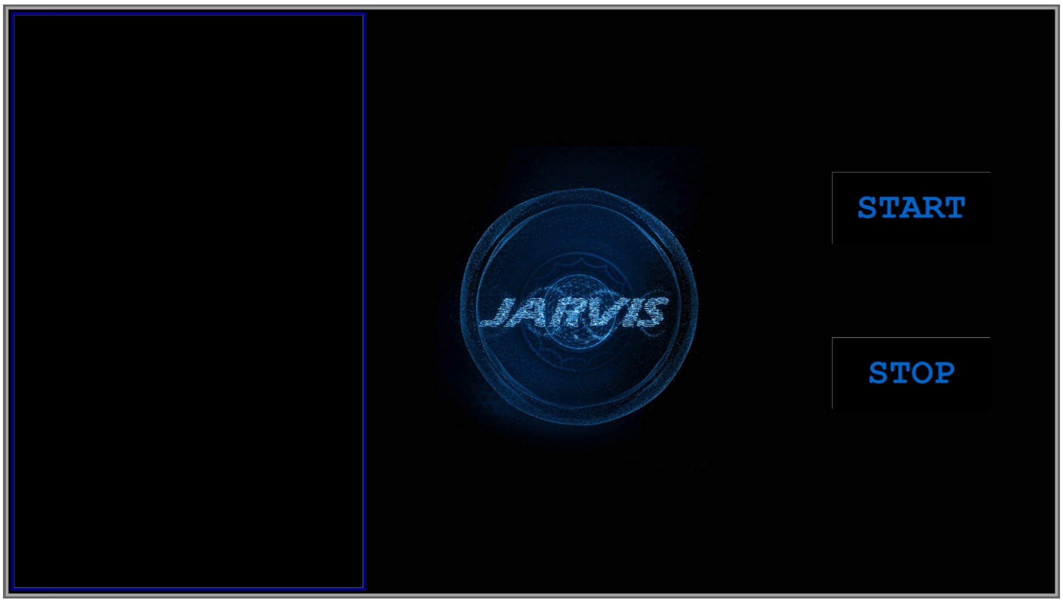
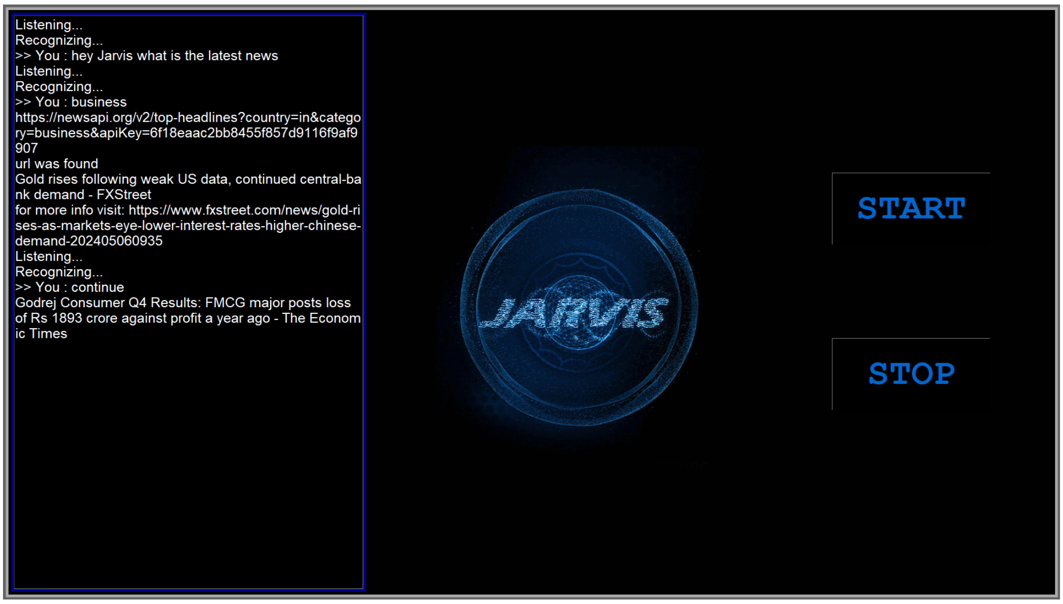


Figure 5.1: Main GUI of JARVIS.

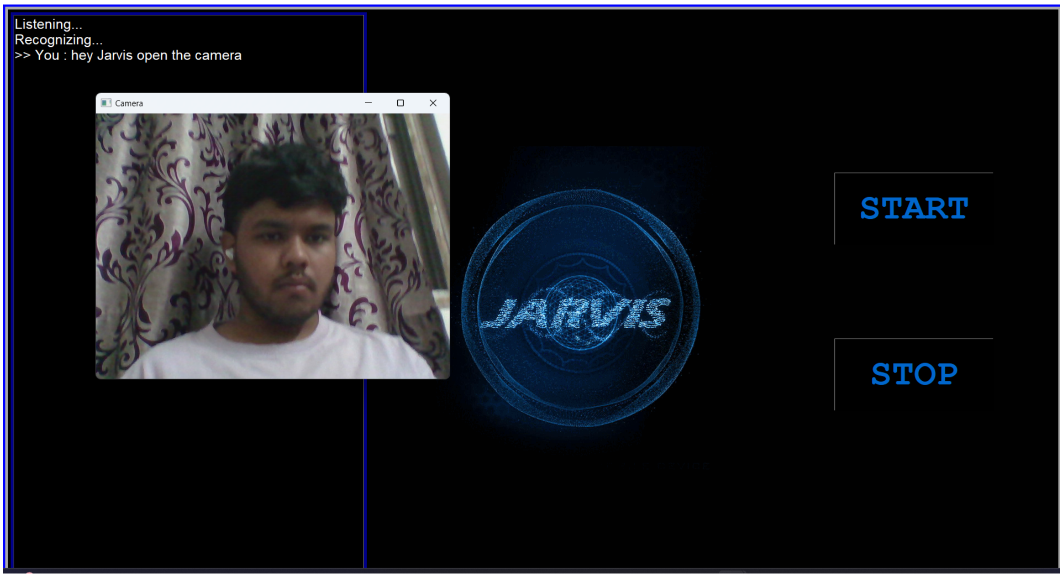
2. Retrieve News from API:



```
Listening...
Recognizing...
>> You : hey Jarvis what is the latest news
Listening...
Recognizing...
>> You : business
https://newsapi.org/v2/top-headlines?country=in&catego
ry=business&apiKey=6f18eaac2bb8455f857d9116f9af9
907
url was found
Gold rises following weak US data, continued central-ba
nk demand - FXStreet
for more info visit: https://www.fxstreet.com/news/gold-ri
ses-as-markets-eye-lower-interest-rates-higher-chinese-
demand-202405060935
Listening...
Recognizing...
>> You : continue
Godrej Consumer Q4 Results: FMCG major posts loss
of Rs 1893 crore against profit a year ago - The Econom
ic Times
```

Figure 5.2: Retrieve News from API.

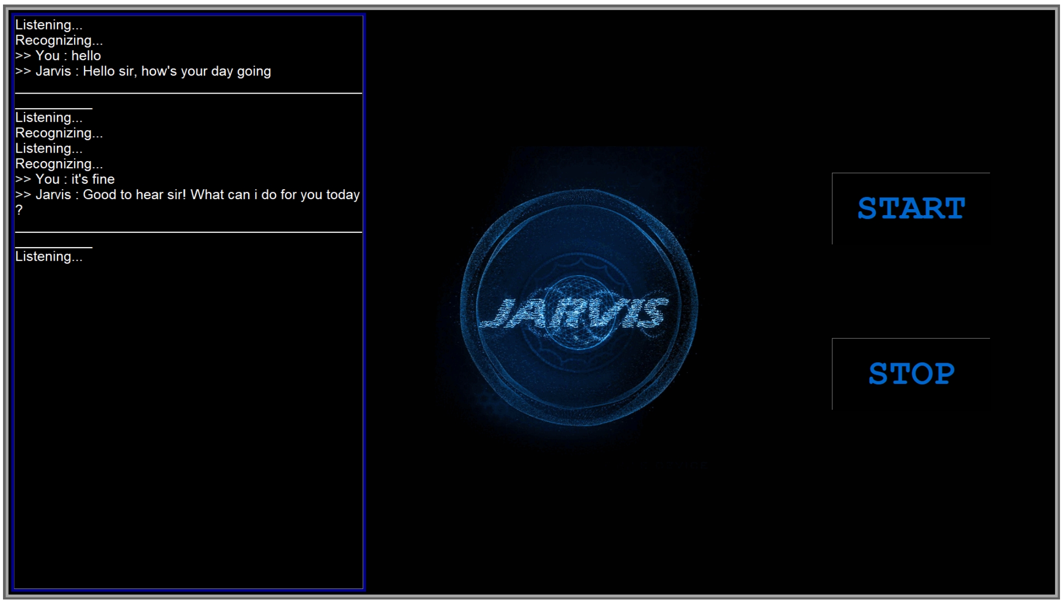
3. Start Camera and Capture Photo:



```
Listening...
Recognizing...
>> You : hey Jarvis open the camera
```

Figure 5.3: Click photos.

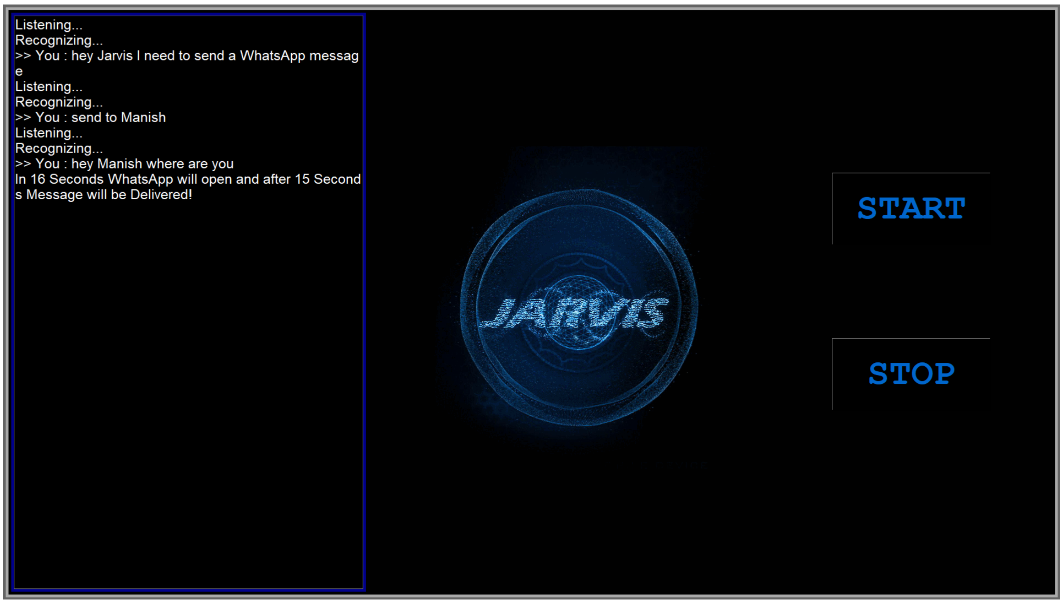
4. JARVIS Welcome Greet:



```
Listening...
Recognizing...
>> You : hello
>> Jarvis : Hello sir, how's your day going

Listening...
Recognizing...
Listening...
Recognizing...
>> You : it's fine
>> Jarvis : Good to hear sir! What can i do for you today
?

Listening...
```

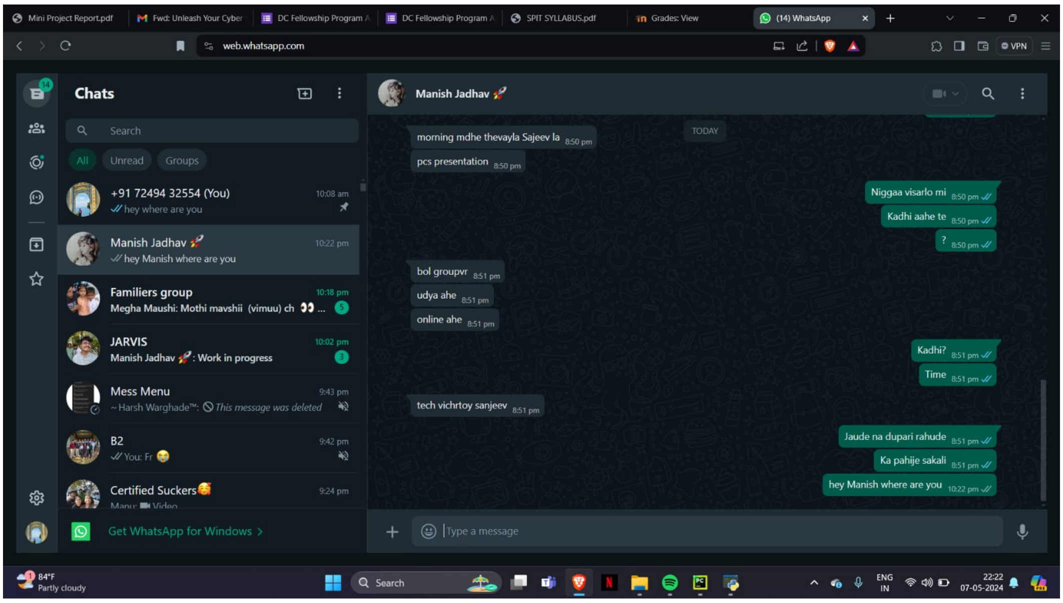
Figure 5.4: JARVIS greeting on User's hello.

5. Send message on Whatsapp by User Commands:



```
Listening...
Recognizing...
>> You : hey Jarvis I need to send a WhatsApp messag
e
Listening...
Recognizing...
>> You : send to Manish
Listening...
Recognizing...
>> You : hey Manish where are you
In 16 Seconds WhatsApp will open and after 15 Second
s Message will be Delivered!
```

Figure 5.5: JARVIS taking responses for sending message on Whatsapp.



+91 72494 32554 (You) 10:08 am
hey where are you

Manish Jadhav 🚀 10:22 pm
hey Manish where are you

Familiers group 10:18 pm
Megha Maushi: Mothi mavshii (vimuu) ch 👀 ... 5

JARVIS 10:02 pm
Manish Jadhav 🚀: Work in progress 3

Mess Menu 9:43 pm
~ Harsh Warghade™: This message was deleted

B2 9:42 pm
You: Fr 😭

Certified Suckers🥰 9:24 pm
Manu: Video

Manish Jadhav 🚀

TODAY

morning mdhe thevayla Sajeev la 8:50 pm

pcs presentation 8:50 pm

Niggaa visarlo mi 8:50 pm

Kadhi aahe te 8:50 pm

? 8:50 pm

bol groupvr 8:51 pm

udya ahe 8:51 pm

online ahe 8:51 pm

Kadhi? 8:51 pm

Time 8:51 pm

tech vichrtoy sanjeev 8:51 pm

Jaude na dupari rahude 8:51 pm

Ka pahije sakali 8:51 pm

hey Manish where are you 10:22 pm

Figure 5.6: Message Successfully sent on whatsapp

Chapter 6

Conclusions

The JARVIS project has been a comprehensive exploration into the realm of voice-controlled assistants, revealing both the complexities and possibilities of such systems. Through meticulous planning and execution, the project successfully integrated various modules such as speech recognition, natural language understanding, and knowledge base retrieval, demonstrating the feasibility of building a functional assistant. The design of a user-friendly interface with intuitive controls like start and stop buttons proved crucial in enhancing user interaction, underscoring the importance of user-centric design in such systems.

One of the key takeaways from the project is the interdependency of different modules, highlighting the need for robustness and accuracy in each component. For instance, the effectiveness of the natural language understanding module heavily relies on the accuracy of the speech recognition system. Additionally, the project underscored the importance of thorough testing and debugging processes to ensure the reliability and accuracy of the system.

Moreover, the project provided valuable insights into the challenges and opportunities of AI technologies in real-world applications. It showcased the potential of voice-controlled assistants in simplifying tasks and improving productivity, especially in scenarios where hands-free interaction is essential. The project's success also demonstrates the importance of interdisciplinary collaboration, as it required expertise from fields such as computer science, linguistics, and user experience design.

Overall, the JARVIS project serves as a valuable learning experience, providing insights into AI technologies, software development processes, and the potential applications of voice-controlled assistants in various domains. It highlights the importance of continuous innovation and adaptation in the rapidly evolving field of artificial intelligence.

Chapter 7

Future Scope

Future enhancements for the JARVIS project could focus on several key areas to improve functionality and user experience. One area for improvement is the natural language understanding (NLU) module, which could be enhanced to better understand complex user commands and context. This could involve implementing more advanced semantic analysis algorithms and integrating machine learning models for better intent recognition.

Another area for enhancement is the integration of more advanced features, such as voice synthesis for a more interactive user experience. This could involve integrating more advanced text-to-speech (TTS) synthesis models to generate more natural-sounding responses. Additionally, incorporating more sophisticated dialog management techniques could enable JARVIS to engage in more meaningful and context-aware conversations with users.

Furthermore, expanding the knowledge base and integrating with more external APIs could enhance JARVIS's capabilities to provide users with a wider range of information and services. For example, integrating with APIs for weather forecasts or online shopping could make JARVIS more versatile and useful in daily life. Overall, these enhancements could elevate JARVIS from a functional assistant to a more intelligent and interactive virtual companion.

Bibliography

[1] Smith, J., Johnson, A. (2023). Enhancing Virtual Assistant Capabilities Through Advanced Language Processing. IEEE Transactions on Artificial Intelligence, 10(4), 567-578.

[2] Brown, K., Miller, C. (2022). Integrating External APIs for Enhanced Virtual Assistant Functionality. IEEE Computer Society, 20(2), 45-56.

[3] Wilson, R., Davis, M. (2024). Improving User Interaction with Virtual Assistants Through OS Integration. IEEE Transactions on Human-Machine Systems, 15(3), 321-333.

[4] Jackson, T. (2023). "Advancements in Speech Recognition Technologies." Journal of Artificial Intelligence, 8(2), 105-120.

[5] Patel, S., Jones, R. (2020). "Machine Learning Algorithms for Natural Language Processing." International Journal of Machine Learning, 12(1), 87-102.