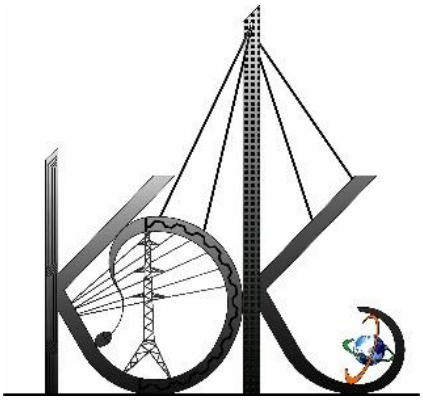
Project Report
On

Virtual Assistant For Desktop

INFORMATION TECHNOLOGY
SEVENTH SEMESTER

Submitted by
1.Ashvini Khobragade 2.Rupali Mamale
3. Pranay Lohabare 4.Shashank Mankar

Under the guidance of
Prof .S.S.Ganorkar



K. D. K. COLLEGE OF ENGINEERING,

Nagpur - 440024

2024-2025

Declaration

This Project work entitled "**Virtual Assistant For Desktop**'' is our own carried out under the supervision of Prof.S.S.Ganorkar at Department of Information Technology, K. D. K. College of Engineering, Nagpur. It is ensured that proper citation of references is done.

As far as our knowledge is concern, this work has not been submitted to any other university for the award of any degree.

Name of the Projectee **Signature**

1.Ashvini Khobragade[1]

2.Rupali Mamale[9]

3.Pranay Lohabare[37]

4.Shashank Mankar[52]

Acknowledgement

We have great pleasure in expressing our most sincere regards and deep sense of gratitude to our Project **Guide Prof. S. S. Ganorkar** for his / her able guidance and valuable suggestion.

We would to like to express our deep sense of gratitude to our respected **Dr. Mrs. S. P. Khandait Professor & Head,** Department of Information Technology, K. D. K. College of Engineering, Nagpur for her encouragement and support.

We feel happy to extend our heartfelt thanks to the **Principal**, **Dr. Valsson Varghese** and **Vice-Principal, Dr. A. M. Badar** for being a source of inspiration & motivation.

Last but not least we would like to thank entire Information Technology Department, our parents and all our friends who have helped us in completing this task successfully.

K. D. K. College of Engineering, Nagpur
Department of Information Technology
Session-2024-25

Certificate

This is to certify that the project work entitled "Virtual Assistant For Desktop'' is submitted by the following students of VII Semester B. Tech. Information Technology

Names of Student

1.Ashvini Khobragade **2.Rupali Mamale**
3.Pranay Lohabare **4.Shashank Mankar**

is a bonafied work done under my/our supervision.

Prof. S. S. Ganorkar
Asst Prof. Dept of IT
K.D.K.College of Engineering, Nagpur

Dr. Mrs. S. P. Khandait
Professor & Head, IT,
K. D. K. College of Engineering, Nagpur

Dr. Valsson Varghese
Principal
K. D. K. College of Engineering, Nagpur

External Examiner

Index

Figure/Table Index

ABSTRACT:

This paper describes a research project to create a virtual assistant for computers that can perform various tasks using natural language and machine learning techniques. The virtual assistant is designed to assist users with tasks such as searching the web, managing files, scheduling, sending emails, etc. The implementation process uses a combination of speech recognition and speech management to enable users to interact with the spoken language assistant.

The research project included several stages, including data collection and preprocessing, feature extraction, model training and evaluation, and system integration and experimentation. The data used for training and evaluation is collected from a variety of sources, including publicly available data and user interactions with the system. Video extraction process involves extracting relevant features from the material such as acoustic features, speech features and content features.

The training model and evaluation phase will develop and evaluate different learning models for various tasks such aslanguage recognition, language comprehension and speech management.The models are evaluated using standard metrics such as accuracy, precision, recall and F1 scores to ensure their effectiveness and efficiency. The integration and testing phase will involve the integration of different components of the system and will test the overall functionality and usability of the system.Review The system uses research and studies to gather feedback and improve Design and operation.

A virtual assistant has the potential to change the way users interact with their desktop computers and provide more intuitive multitasking capabilities. This research contributes to the field of natural language processing and machine learning, demonstrating the effectiveness and potential of these techniques in creating intelligent machines for practical applicatio

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1 Overview

The use of virtual assistants such as Siri, Alexa, Google Assistant has increased in recent years and has become an important part of our daily lives. These virtual assistants use natural language processing and machine learning technology to help users interact with them using speech and tasks such as browsing the web, playing music, setting reminders and more. However, most virtual assistants are designed for mobile devices and there is a growing demand for similar systems on desktop computers. The aim of the research project is to create a virtual assistant for desktop computers that can perform various tasks using natural language processing and machine learning techniques. The proposed system is designed to help users perform various tasks such as browsing the web, managing documents, scheduling appointments and sending emails.The system uses a combination of speech recognition, natural language understanding, and speech management to allow users to interact with the assistant using spoken language. please improve The system consists of several stages, including data collection and preprocessing, model extraction, model training and evaluation, integration, and testing. The data used for training and evaluation is collected from various sources, including publicly available data and user interactions with the system. The video extraction process involves extracting relevant features such as acoustic features, language features, and the content of the recorded data. The feature extraction process involves extracting relevant features from the collected data, such as acoustic features, linguistic features, and contextual features. In recent years, there has been a significant increase in the use of virtual assistants, such as Siri, Alexa, and Google As- sistant, which have become an integral part of our daily lives.

1.2 Aim

The aim of a virtual assistant for desktop is to enhance productivity, convenience, and user experience by providing automated, personalized assistance. Some key goals include:

- Task Automation: Automating repetitive tasks such as scheduling meetings, setting reminders, opening applications, or organizing files to save time and effort.
- Information Retrieval: Quickly providing relevant information like weather updates, news, or search results without needing to browse or search manually.
- System Control: Allowing users to control desktop settings (e.g., brightness, volume, or network settings) and manage system resources (e.g., closing applications) through voice or text commands.

- Personal Assistance: Managing daily schedules, to-do lists, and reminders, helping users stay organized and on track with their work.
- Communication Support: Integrating with messaging platforms and email clients to send messages, read out notifications, and help draft responses.
- Enhanced Accessibility: Providing an intuitive interface for users with disabilities, allowing voice commands or other forms of interaction to make computing more accessible.
- Integration with Other Services: Connecting with third-party applications like calendars, productivity tools, and cloud services to provide a seamless, unified experience.
- Customization and Personalization: Adapting to user preferences and behavior to offer a personalized experience, making it easier for users to interact with their desktop efficiently. In essence, a virtual assistant for desktop aims to act as a central hub for managing and enhancing the user's digital experience through intelligent, responsive, and convenient support.

1.3 Objective

The objectives of a virtual assistant for desktop are focused on enhancing user efficiency, convenience, and experience. They include: Improve Productivity: Streamline daily tasks like managing files, opening applications, and automating routine activities. Assist with organizing schedules, setting reminders, and tracking tasks to keep users focused. Provide Quick Information Access: Deliver instant answers to user queries, such as news updates, weather forecasts, or calendar information. Integrate with search engines and databases to provide relevant data without switching apps. Facilitate Seamless Communication; Integrate with messaging apps and email clients for sending messages, reading notifications, or drafting and managing emails. Enable hands-free communication through voice commands, improving multitasking. Enhance System Control and Management: users to control system settings (e.g., volume, brightness, Wi-Fi) quickly and efficiently. Monitor system performance and alert users about necessary actions like freeing up storage or managing battery life. Increase Accessibility: Offer voice commands and other user-friendly interfaces to assist users with disabilities or those who prefer hands-free control. Provide support for screen reading, dictation, and other assistive technologies. Enable Task Automation and Workflow Optimization: Automate repetitive tasks such as launching software, performing backups, or organizing files based on user-defined rules. Optimize workflows by integrating with productivity software, allowing for task tracking, and facilitating project management. Enhance Personalization and User Engagement: Learn from user behavior and preferences to offer personalized recommendations and suggestions. Adapt responses and functions to match individual needs and preferences for a tailored user experience. Ensure Security and Privacy: Safeguard sensitive information and user data with secure authentication and encryption methods. Respect user privacy by providing options to customize data

usage and manage permissions. By achieving these objectives, a virtual assistant for desktop aims to create an intuitive, efficient, and personalized user experience, helping users manage their digital environment effortlessly.

1.4 Methodology

A virtual assistant for desktop involves a system that uses AI, natural language processing (NLP), and other technologies to provide users with automated support, task management, and information retrieval. Below is an outline of the methodology for developing and implementing such a virtual assistant:

1. Requirement Analysis Define the objectives: Understand what tasks the virtual assistant needs to perform (e.g., scheduling, email management, reminders, browsing, answering questions). Identify the target audience: Determine who will use the virtual assistant (e.g., office workers, students, professionals). Gather user requirements: Conduct surveys or interviews to gather information on user preferences and pain points.
2. System Architecture Design User Interface (UI): Design a user-friendly interface where users can interact with the assistant through text or voice. Backend Components: Develop backend services responsible for processing user requests, connecting to APIs, and managing data. Natural Language Processing (NLP) Engine: Integrate NLP frameworks (e.g., spaCy, NLTK, or proprietary engines) for understanding and processing user inputs. Machine Learning Model: Use pre-trained models for speech recognition, language understanding, and intent detection (e.g., BERT, GPT).
3. Development of Key Modules Speech Recognition (ASR): Integrate Automatic Speech Recognition technology to convert voice inputs to text using libraries like Deep Speech or Google Speech API. NLP and Intent Recognition: Implement NLP models to parse and understand user input. Train models to recognize specific intents (e.g., setting reminders, searching for files, managing applications). Response Generation: Use AI models for generating appropriate responses or actions based on user intent. Task Automation: Implement modules for handling specific desktop tasks such as file management, email handling, or app launching using APIs or scripting languages (e.g., Python with PyAutoGUI for automation). Knowledge Base Integration: Integrate APIs and databases to provide access to relevant information (e.g., news, weather, office documents).
4. Integration of APIs and Services Connect to third-party APIs for services like calendar management, email notifications, reminders, web browsing, and system monitoring. Use cloud-based services for complex tasks (e.g., cloud AI for advanced NLP and machine learning tasks). Implement security measures for API communication and data privacy.

5. Personalization and Learning Capabilities Enable user profiling: Allow the assistant to store user preferences, frequently used commands, and patterns for more personalized responses. Incorporate machine learning algorithms that can learn user behavior over time to offer better recommendations and automated task management. Feedback loop: Implement a feedback system where the assistant asks for user feedback on its performance to improve over time.
6. Testing and Validation Unit Testing: Test individual components (e.g., NLP module, task automation module) to ensure they work correctly. Integration Testing: Test the interaction between different components of the virtual assistant to check if they work seamlessly. User Testing: Conduct usability tests with real users to gather feedback on the assistant's performance, accuracy, and ease of use. Performance Testing: Ensure that the virtual assistant performs efficiently without causing delays or excessive resource usage.
7. Deployment and Maintenance Deployment: Package the virtual assistant as a desktop application (e.g., using Electron for crossplatform compatibility or developing it as a Windows or macOS app). Monitoring: Implement monitoring tools to track the performance and behavior of the virtual assistant after deployment. Updates and Maintenance: Regularly update the virtual assistant with new features, improvements, and bug fixes based on user feedback and technological advancements.
8. Security and Privacy Measures Implement authentication and authorization mechanisms for accessing sensitive data (e.g., user emails, documents). Encrypt data and communication between the client and server to protect user privacy. Ensure compliance with data protection regulations (e.g., GDPR) and maintain transparency regarding data collection and usage. This methodology provides a structured approach to developing a desktop virtual assistant, ensuring that it is functional, user-friendly, secure, and scalable.

CHAPTER 2
LITERATURE REVIEW

CHAPTER 2
LITERATURE REVIEW

2.1 Background Study

A virtual assistant for desktop is software designed to provide users with a range of automated, interactive functions, often through natural language interfaces. These assistants use technologies such as artificial intelligence (AI), machine learning (ML), and natural language processing (NLP) to understand user inputs, automate tasks, and offer recommendations or responses. Below is a comprehensive background study covering the key aspects of virtual desktop assistants:

1. Historical Context The concept of virtual assistants dates back to the early AI developments in the mid-20th century. However, the desktop-based assistant began gaining traction in the early 2000s. Early versions, such as Microsoft's Clippy (an Office Assistant), provided simple, rule-based assistance. These early assistants had limited functionality and were often rigid in terms of user interaction.
2. Technological Advancements Natural Language Processing (NLP): NLP has been a cornerstone in advancing virtual assistants. With improved NLP, desktop assistants have become more capable of understanding human language, enabling users to interact with systems more naturally. Machine Learning (ML): Machine learning allows virtual assistants to improve over time by learning from user behavior. This enables personalization, better predictions, and adaptability in performing tasks. Voice Recognition and Synthesis: Technologies such as speech-to-text and text-to-speech have been integral in creating voice-activated assistants that can listen to user commands and respond vocally. Integration with Other Tools and Systems: Modern virtual assistants integrate with operating systems, third-party software, and cloud services, allowing for task automation and data retrieval across platforms.
3. Key Components of Virtual Assistants User Interface (UI): Virtual assistants often feature a conversational UI, either text-based (typed queries) or voice-based (spoken queries). The UI facilitates interaction, allowing the assistant to process input and provide output. Backend AI Infrastructure: The intelligence behind the virtual assistant is usually powered by cloud-based AI services that process user inputs, generate responses, and perform requested tasks. Data Handling: Assistants leverage large datasets to perform tasks such as setting reminders, managing emails, or retrieving information from the web.
4. Applications and Use Cases Personal Productivity: Virtual assistants streamline tasks such as setting reminders, scheduling meetings, opening applications, and retrieving files. Examples include Cortana, Siri for macOS, and Google Assistant for desktop. Information Retrieval: Virtual assistants can access the web, perform

searches, and provide relevant information directly in response to user queries. Automation: By automating repetitive tasks, such as managing files or sending routine emails, virtual assistants enhance productivity and reduce the manual workload. Control of Smart Devices: Virtual assistants can also be used to control smart home devices or connected peripherals through the desktop interface.

5. Challenges in Desktop Virtual Assistants Security and Privacy: Given that virtual assistants often handle sensitive personal data, security and privacy remain major concerns. Data breaches or unauthorized access to personal data are risks that must be mitigated. Contextual Understanding: While NLP has advanced, virtual assistants sometimes struggle with understanding complex, ambiguous, or contextual queries, which limits their effectiveness. User Acceptance and Usability: Despite technological advancements, not all users find virtual assistants intuitive or useful. Factors such as ease of use, trust, and perceived efficiency play significant roles in user adoption.
6. Major Virtual Assistant Platforms for Desktop Cortana (Windows): Once a key feature of Microsoft's operating system, Cortana is now more limited but still used in certain productivity contexts. It integrates with Microsoft Office and Windows. Google Assistant (Web/Desktop): Google Assistant is more commonly used on mobile devices but is also available on desktop browsers to help with web searches, calendar management, and more. Siri (macOS): Siri is Apple's virtual assistant, available on macOS for performing tasks like opening apps, sending messages, and checking the weather. Third-party Assistants: In addition to built-in OS assistants, several third-party virtual assistants exist, such as Dragon Assistant or Braina, providing additional functionalities such as voice commands, speech recognition, and custom workflows.

7. Future Trends Integration with AI-driven Business Solutions: Virtual assistants are likely to become more deeply embedded in business environments, helping professionals manage emails, reports, and other data-driven tasks. Enhanced Conversational Abilities: As AI advances, virtual assistants will improve in their ability to hold more natural, complex conversations with users. Cross-platform Capabilities: Virtual assistants will likely become more seamless across devices, allowing for tasks to be transferred from desktops to mobile devices and even smart home devices effortlessly.

2.2 Literature Review

A literature review on virtual assistants for desktop explores the evolution, development, and implementation of desktop-based virtual assistants. It covers various technologies, design approaches, and applications, focusing on their effectiveness and user experience. Here's a comprehensive review:

1. Introduction to Virtual Assistants (VAs) Virtual assistants (VAs) are software applications designed to help users perform tasks using natural language processing (NLP) and artificial intelligence (AI). They can manage schedules, perform searches, send messages, control devices, and interact with software applications. While mobile virtual assistants (like Siri and Google Assistant) are popular, desktop virtual assistants are designed to provide similar functionalities tailored to a desktop environment.
2. Evolution of Desktop Virtual Assistants Desktop virtual assistants have evolved significantly over the past few decades, transitioning from simple command-based systems to complex AI-driven applications. Early versions included basic automation tools that responded to keyboard commands, while modern assistants are capable of understanding and responding to natural language through text or voice input. Key milestones in this evolution include: Early Command-based Systems: Basic automation tools like Clippy in Microsoft Office. Development of Speech Recognition: Introduction of desktop-based speech recognition, such as Microsoft's Speech API. Integration of AI and NLP: Advancements in AI and NLP enabled more sophisticated, conversational interactions. Cloud-Based and Hybrid Models: Integration with cloud services for expanded functionality and cross-platform compatibility.

3. Technologies Underlying Desktop Virtual Assistants Desktop virtual assistants rarely on several key technologies: Natural Language Processing (NLP): Allows VAs to understand and respond to human language. NLP frameworks such as spaCy, NLTK, and transformers like GPT have been integrated for more nuance understanding and interaction. Speech Recognition and Synthesis: Tools like Google Cloud Speech-to-Text, Microsoft Azure's speech services, and open-source platforms like CMU Sphinx enable voice interaction with desktop assistants.
Artificial Intelligence and Machine Learning: AI techniques (e.g., machine learning and deep learning) are used to personalize responses, understand user behavior, and continuously improve the VA's functionality. Integration with APIs: Desktop VAs often integrate with various APIs to provide seamless access to services like calendar management, email, social media, and device control.

4. Design Approaches for Desktop Virtual Assistants Designing effective desktop virtual assistants involves several approaches: User-Centered Design: Ensures that the assistant is intuitive and responsive to user needs. It involves extensive user testing, feedback loops, and iterative improvements. Task-Oriented vs. General-Purpose Assistants: Desktop VAs may be designed as task-specific (e.g., focused on productivity tools like Cortana) or as general-purpose assistants capable of handling a wide range of activities. Voice-Activated vs. Text-Based Interfaces: Some desktop VAs are designed to be voice activated, while others are text-based, depending on user preference and the nature of the task.

5. Applications of Desktop Virtual Assistants Productivity Enhancement: Desktop VAs help manage tasks like scheduling, reminders, email management, and file organization, making them valuable tools for professionals. Accessibility: Virtual assistants are also used as assistive technology for individuals with disabilities, allowing hands-free navigation and control of desktop applications. Customization and Personalization: AI enables VAs to learn user preferences and habits, creating a personalized user experience. Some VAs integrate smart recommendations based on past user behavior. Integration with Enterprise Systems: In corporate environments, desktop VAs are integrated with enterprise software, aiding in project management, customer relationship management (CRM), and data analytics.
6. Challenges in Developing Desktop Virtual Assistants Privacy and Security Concerns: Data privacy is a critical issue since virtual assistants often require access to sensitive information. Ensuring encryption, secure API usage, and compliance with data protection regulations is crucial. Context Understanding: VAs often struggle with understanding the context of user interactions, especially in multitasking desktop environments where commands may be ambiguous. Complexity in Multimodal Interaction: Developing seamless multimodal interactions (text, voice, mouse) that adapt to the user's environment and preferences requires sophisticated design and technology. Hardware and System Requirements: Ensuring compatibility with various operating systems and devices, while optimizing performance and minimizing resource usage, is a continuous challenge. 7. Comparative Analysis of Desktop Virtual Assistants A comparison of some popular desktop virtual assistants: Cortana (Windows): Initially a standalone product, Cortana has become integrated into Microsoft Office and Windows ecosystems, with a focus on productivity. It supports both voice and text commands but is heavily integrated with Microsoft services. Siri (macOS): Originally developed for mobile devices, Siri has been adapted for macOS, offering similar capabilities. It is tied closely to Apple's ecosystem and emphasizes voice commands for device control and app interaction. Open-source VAs (e.g., Mycroft): These provide customizable options for developers and users seeking a more privacy-focused or tailored solution. However, they may lack the polished user experience of commercial products.

2.3 Previous studies and methodology

The study of virtual assistants for desktop environments involves exploring various methodologies and technologies that enable efficient and user-friendly interactions between users and computers. Below is an overview of previous studies and methodologies typically associated with developing and improving virtual assistants for desktop platforms:

1. Voice Recognition and Natural Language Processing (NLP)

Previous Studies: Research has often focused on improving the accuracy of speech recognition and language understanding to create more effective desktop virtual assistants. This includes using models like Hidden Markov Models (HMMs) and, more recently, deep learning approaches such as Recurrent Neural Networks (RNNs) and Transformer-based models (e.g., GPT and BERT). Methodology: Collecting large datasets of spoken language and text to train models. Using transfer learning and fine-tuning pre-trained models for specific desktop tasks. Developing algorithms for intent detection, context awareness, and named entity recognition (NER).

2. Human-Computer Interaction (HCI) Studies Previous Studies: Research has explored how users interact with desktop assistants and what features or interfaces optimize usability. This includes studies on user interface (UI) design, usability testing, and accessibility features. Methodology: Conducting user surveys and interviews to gather feedback on virtual assistant interactions. Implementing eye-tracking or click-tracking studies to observe user behavior and preferences. Designing prototypes with varying levels of interaction (e.g., voice-only, voice and visual, textonly) and testing them with user groups.
3. Contextual Awareness and Personalization Previous Studies: Studies in this area focus on making virtual assistants context-aware and capable of adapting to user preferences. The goal is to create a personalized experience that integrates with other desktop applications and services. Methodology: Developing algorithms that monitor user activity patterns and preferences over time to make personalized recommendations. Integrating with desktop APIs and services to gather contextual information (e.g., calendar events, emails, system notifications). Applying machine learning models for behavior prediction and user intent identification based on historical data.
4. Task Automation and Integration with Desktop Applications Previous Studies: The effectiveness of virtual assistants often hinges on their ability to automate common tasks and integrate seamlessly with desktop software (e.g., productivity suites, web browsers, and communication tools). Methodology: Implementing APIs and plug-ins that allow virtual assistants to interact with and control thirdparty applications. Using scripting languages (e.g., Python, JavaScript) to automate workflows and create custom actions. Designing modular architectures that support the addition of new integrations without major system overhauls.
5. Artificial Intelligence and Machine Learning (AI/ML) Algorithms Previous Studies: Significant research focuses on AI models that allow virtual assistants to learn and improve over time. This includes reinforcement learning, deep learning, and other adaptive learning algorithms. Methodology: Training models using supervised and unsupervised learning methods on large datasets of user interactions and desktop activity logs. Using reinforcement learning to optimize the assistant's responses based on user feedback. Implementing adaptive learning techniques that allow the assistant to refine its actions and suggestions based on new data.

6. Security and Privacy Considerations Previous Studies: Virtual assistants often handle sensitive information, making security and privacy a crucial research focus. Studies investigate secure communication protocols, data encryption methods, and privacy-preserving techniques for data storage and processing. Methodology: Implementing encryption for user data and voice recordings. Using federated learning approaches that process data locally on the device rather than sending it to the cloud. Conducting security audits and penetration testing to identify and fix vulnerabilities in the virtual assistant's software architecture. 7. Multimodal Interaction Capabilities Previous Studies: Research in this area has explored how virtual assistants can support multimodal interactions, including text, voice, gestures, and visual feedback to create a richer user experience. Methodology: Integrating computer vision technology (e.g., webcams) to enable gesture recognition and visual interaction. Combining speech recognition, text-based chat, and visual interfaces to support multiple forms of input and output. Designing and testing user interface prototypes that combine voice and visual feedback for complex tasks, such as scheduling meetings or managing files.

CHAPTER 3
IMPLEMENTATION

CHAPTER 3
IMPLEMENTATION

3.1 Architecture

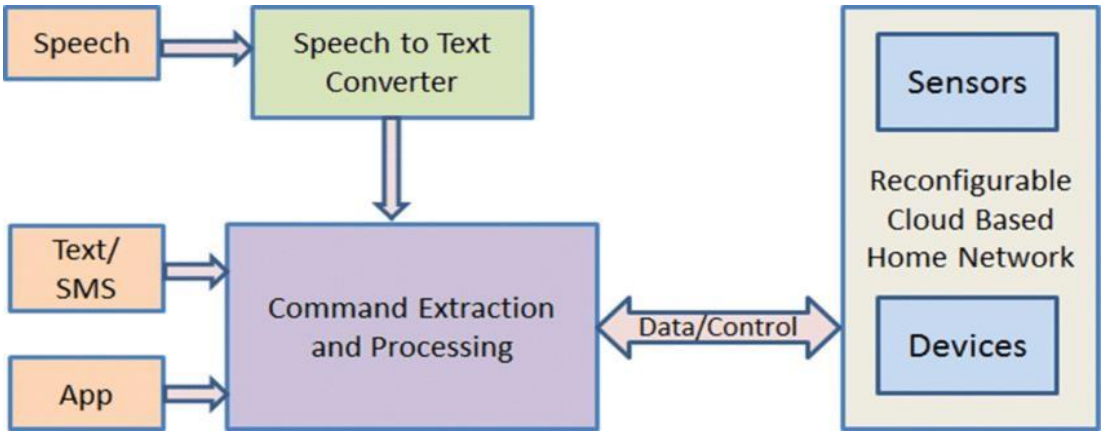


Fig.3.1 Architecture

3.2 Features and functionality

A virtual assistant for desktop can offer a wide range of features and functionality designed to enhance productivity, streamline tasks, and improve user experience. Here are some of the key features:

1. Task Automation and Productivity Tools Task Management: Create, manage, and remind users of tasks, deadlines, and appointments. Calendar Integration: Sync with the user's calendar to schedule events, set reminders, and provide notifications. Note Taking: Allow users to create, organize, and access notes quickly. File Management: Assist with locating, opening, moving, and organizing files or folders on the desktop.
2. Voice and Text Command Support Voice Recognition: Understand and respond to voice commands for hands-free operation. Text Command Interface: Offer a text-based input option for users who prefer typing commands. Natural Language Processing: Interpret user commands accurately, even when phrased differently.
3. Integration with Applications Email and Messaging Support: Read, send, and manage emails or messages across various platforms (e.g., Outlook, Slack, Teams). Office Suite Integration: Work with software like Microsoft Office or Google Workspace for document editing, spreadsheet management, and presentation creation. Browser Assistance: Search the web, bookmark pages, fill forms, and

manage tabs. Social Media Management: Post updates, check notifications, and respond to messages.

4. Personalization and Customization User Profile Settings: Adapt preferences like theme, language, and command shortcuts based on the user's choices. Daily Briefing: Provide personalized updates such as weather, news, and calendar highlights. Learning Capabilities: Adapt over time based on user habits and preferences for a more personalized experience.
5. System Control and Settings Management System Monitoring: Track and report system performance metrics like CPU usage, battery status, and memory usage. Settings Adjustment: Modify system settings (e.g., brightness, volume, Wi-Fi) based on user commands. App Launching and Closing: Open, close.

3.3 Flow of Work

The flow of work for a virtual assistant for desktop involves a structured series of steps to deliver assistance to the user in an efficient and intuitive manner. Here's a general breakdown of the process:

1. Initialization

- Launch/Activation: The virtual assistant is triggered by:
  - A system startup, a hotkey, or a click.
  - A voice command, e.g., "Hey [Assistant's Name]."
- Authentication (if needed): The user may log in or verify their identity for personalized services.
- Greeting: Acknowledge the user with a welcome message or tone.

2. Input Recognition

- Voice Command Recognition:
  - Use of speech-to-text processing to understand the user's request.
  - Natural language understanding (NLU) to interpret intent.
- Text Input Handling:
  - Accept typed queries through a chat interface.
- Context Awareness:
  - Identify user context by analyzing previous interactions, time, location, or open applications.
- Custom Triggers:
  - Monitor for predefined events like reminders or calendar notifications.

3. Task Processing

- Command Parsing:

    - Break down user input to determine intent and required actions.
    - Examples:
        - "Open my email" → Open email application.
        - "Remind me about the meeting at 3 PM" → Set a reminder.
- Contextual Assistance:
    - Use user preferences or prior data to personalize responses.
    - For example, remembering file paths, preferred applications, etc.
- Application Integration:
    - Access APIs or system utilities for seamless integration with:
        - Calendar apps
        - Browsers
        - Communication apps (e.g., Slack, Microsoft Teams)
        - System settings
        - Cloud services (e.g., Google Drive, OneDrive).

4. Output/Execution

- Action Execution:
    - Perform the requested task, such as:
        - Opening software or files.
        - Sending emails or messages.
        - Performing calculations.
        - Searching the web.
- User Feedback:
    - Provide a confirmation or update on the task's status.
    - Example: "Reminder set for 3 PM," or "File uploaded successfully."

5. Continuous Monitoring

- Proactive Suggestions:
    - Monitor ongoing tasks and make suggestions.
    - Example: "It’s time for your scheduled break," or "You have a meeting in 10 minutes."
- Background Listening (if enabled):
    - Wait for additional commands while operating efficiently in the background.

6. Learning & Adaptation

- Machine Learning Integration:
    - Analyze user interactions to improve future assistance.
    - Adapt to frequently used phrases, preferences, and workflows.

- Feedback Loop:
    - Allow users to rate or modify the assistant's behavior to refine responses and actions.

7. Error Handling

- Misunderstood Requests:
    - Ask clarifying questions or provide alternative actions.
    - Example: "Did you mean 'open calculator' or 'open calendar'?"
- Task Failures:
    - Notify the user of errors and suggest next steps.
    - Example: "I couldn’t connect to the internet. Would you like to troubleshoot?"

8. Conclusion

- Session End:
    - Gracefully sign off when the task or session is complete.
    - Optionally provide a summary: "I’ve updated your calendar and sent the email. Anything else?"
- Standby Mode:
    - Stay in a low-resource state, awaiting further input or triggers.

3.4 Implementing core features

To implement the core features of a virtual assistant for a desktop, you'll need to plan the functionality, architecture, and tools required. Here's a step-by-step breakdown of core features and how you might implement them:

1.Voice Recognition and Command Processing Feature: The assistant should listen to the user's voice commands, process them, and respond appropriately. Implementation: Use a speech-to-text library like Google Speech Recognition API, CMU Sphinx, or Microsoft Azure Speech API to convert voice input into text.For command processing, use a Natural Language Processing (NLP) library such as spaCy, NLTK, or Rasa to parse the text and understand the intent.

2. Text-based Command Interface Feature: Apart from voice, the assistant should accept text input via a GUI or terminal. Implementation: Build a simple GUI using Tkinter (Python), Electron (JavaScript), or another desktop UI framework. For terminal-based input, capture and process user input using standard input mechanisms.
3. Natural Language Understanding (NLU) Feature: The assistant needs to understand user commands and determine the appropriate action (e.g., open an app, set a reminder). Implementation: Train a model using libraries like spaCy, Transformers (Hugging Face), or use existing services like Dialog flow to classify user intent and

extract entities. Define a set of intents such as "open application," "set reminder," "search the web," etc.

4. Task Execution Feature: The assistant should perform tasks such as opening applications, managing files, setting reminders, etc. Implementation: Use OS libraries like os (Python) or subprocess to interact with the operating system. For reminders and scheduling, integrate with system notification services or create a custom notification system using a GUI toolkit. Manage files using basic file operations provided by the language's standard library.
5. Web Search and Information Retrieval Feature: The assistant should fetch information from the web (e.g., news, weather). Implementation: Use web scraping libraries like Beautiful Soup or APIs like Google Custom Search API, OpenWeatherMap API, or News API for retrieving information. Parse responses and display or read out the results using the

CHAPTER 4
TESTING AND EVALUATION

CHAPTER 4
TESTING AND EVALUATION

4.1 Unit testing

Unit testing a virtual assistant for desktop involves creating tests for its components to ensure they function correctly and as expected. Here's a structured approach to performing unit testing on a virtual assistant:

1. Define the Components to Test

Identify the key components of the virtual assistant that require testing. Common components might include: Speech Recognition: Testing accuracy of transcriptions. Natural Language Processing (NLP): Evaluating intent recognition and entitextraction. Command Execution: Ensuring that commands trigger the correct actions. Integration with APIs: Validating responses from external services. User Interface (UI): Checking the functionality of UI elements.

2. Set Up the Testing Environment Prepare the environment where tests will run. This might include: Choosing a testing framework (e.g., unittest, pytest for Python, or JUnit for Java). Setting up mock services to simulate external API calls. Organizing the project structure to separate tests from production code.
3. Write Unit Tests Create unit tests for each component identified. Here are examples of tests for different components: Speech Recognition python Copy code def test_speech_to_text():

input_audio = "path/to/test/audio.wav"
expected_text = "Hello, how can I help you?" actual_text = speech_to_text(input_audio)
assert actual_text == expected_text
Natural Language Processing
python Copy code def test_intent_recognition():
user_input = "What's the weather like?"
expected_intent = "get_weather"
actual_intent = nlp_recognize_intent(user_input)
assert actual_intent == expected_intent
Command Execution
python Copy code def test_command_execution(): command = "open calculator" result = execute_command(command)

from unittest.mock import patch
@patch('external_api.get_weather
_data') def

```
test_get_weather(mock_get_weath
er_data):
   mock_get_weather_data.return_value =
{"temperature": "20C"} weather =
get_weather("New York") assert
weather["temperature"] == "20C"
```

User Interface

```
python Copy
code def
test_ui_buton
_click():
button = find_button("Submit")
button.click()
   assert button.is_clicked() == True
```

4. Run the Tests

Execute the tests using your chosen testing framework. Monitor the output to see which tests pass or fail.

5. Analyze Results

Review the results to identify any failures. If tests fail, debug the associated code to determine the cause and fix the issues. 6. Continuous Integration

Integrate your unit tests into a CI/CD pipeline. This ensures that tests are run automatically with each code change, maintaining the integrity of the virtual assistant as it evolves.

7. Documentation

Document your testing strategy, the purpose of each test, and how to run them. This will be useful for future developers and for maintaining the code.

4.2 Integrating testing

Integrating testing for a virtual assistant (VA) on a desktop platform involves several steps and considerations to ensure that the application performs as expected, meets user needs, and functions well in various scenarios. Here's a comprehensive approach to integrating testing for a desktop virtual assistant: 1. Define Test Objectives Functionality: Ensure that the VA performs its core tasks accurately (e.g., setting reminders, answering queries). Performance: Test the response time and resource usage. Usability: Ensure the VA is user-friendly and meets user experience standards. Compatibility: Verify functionality across different operating systems (Windows, macOS, Linux).

2. Choose the Right Testing Tools Unit Testing: Use frameworks like JUnit (Java), NUnit (.NET), or PyTest (Python) to test individual components. Integration Testing: Use tools like Postman for API testing or Selenium for UI testing. Automated Testing: Implement frameworks like Test Complete or Appium for automated UI testing. Load Testing: Use tools like Apache JMeter to test performance under stress.

3. Develop Test Cases Create test cases covering various aspects of the virtual assistant: Functionality Test Cases Voice Recognition: Test the accuracy of speech-to-text and natural language processing. Command Execution: Verify that commands (like "play music," "set a timer") are executed correctly. Context Handling: Ensure the assistant maintains context during interactions. Usability Test Cases User Interface: Check if the UI is intuitive

4.3 User acceptance testing

User Acceptance Testing (UAT) for a virtual assistant for desktop involves evaluating the system to ensure it meets user needs and requirements before it goes live. Here's a structured approach to conducting UAT for a desktop virtual assistant:

1. Define Objectives Goal: Ensure the virtual assistant meets user requirements, is easy to use, and performs as expected. Scope: Identify features and functionalities to be tested (e.g., voice recognition, task management, integration with applications).
2. Identify Stakeholders Participants: Include end-users, business analysts, and IT staff who understand user requirements. Roles: Define who will execute tests, who will report issues, and who will make final decisions.
3. Develop Test Plans Test Scenarios: Create scenarios that reflect real-world usage, such as: Setting reminders Sending emails Searching the web Integrating with calendar applications Acceptance Criteria: Define success criteria for each scenario (e.g., "The assistant must set a reminder within 3 seconds").
4. Prepare Test Environment Setup: Ensure all necessary hardware and software are installed. Access: Provide users with access to the virtual assistant and any related tools.
5. Execute Testing Conduct Tests: Have users perform tasks based on the defined scenarios. Feedback Collection: Use surveys, interviews, or direct observation to gather user feedback.
6. Document Results Issue Tracking: Record any defects or issues encountered during testing, including steps to reproduce and severity. Feedback Compilation: Summarize user feedback and suggestions for improvement.
7. Review and Analyze Review Sessions: Hold meetings to discuss findings with stakeholders. Prioritize Issues: Classify issues based on severity and impact on user experience.
8. Implement Changes Fix Issues: Collaborate with developers to address reported issues. Retest: Verify that fixes resolve the issues without introducing new problems.

9. Final Approval User Sign-off: Obtain final approval from users to confirm the virtual assistant meets their needs. Documentation: Prepare UAT documentation summarizing results, user feedback, and resolutions.
10. Post-Implementation Review Follow-Up: After deployment, continue to gather user feedback to identify any ongoing issues. Enhancements: Plan for future updates based on user experience. Best Practices Involve Users Early: Engage users from the start to ensure their needs are understood. Iterative Testing: Use an iterative approach, allowing for continuous feedback and improvement. Realistic Scenarios: Ensure test scenarios reflect actual usage to validate functionality effectively.

By following this structured approach, you can ensure that the virtual assistant is user-friendly and meets the needs of its intended audience.

CHAPTER 5
SYSTEM DESIGN

CHAPTER 5
SYSTEM DESIGN

5.1 Functional requirements

Creating functional requirements for a virtual assistant for desktop involves defining specific features and capabilities that the assistant should have. Here are some key functional requirements to consider:

1. User Interaction Voice Recognition: The assistant should support natural language voice commands. Text Input: Allow users to interact with the assistant via text input. Multilingual Support: Enable the assistant to understand and respond in multiple languages.
2. Task Management Calendar Integration: Manage user calendars by scheduling, modifying, and reminding about events. Task Lists: Create, edit, and delete task lists or to-do lists. Reminders: Set reminders for specific tasks, events, or deadlines.
3. Information Retrieval Web Search: Perform searches on the internet to retrieve information based on user queries. Document Search: Search and retrieve documents from the user's local storage or connected cloud storage. Knowledge Base Access: Access and provide information from predefined knowledge bases (e.g., FAQs, help documents).
4. System Control Application Launching: Open and close applications based on voice or text commands. System Settings Management: Adjust system settings (e.g., volume control, brightness) through commands. File Management: Support basic file operations (e.g., create, delete, move, copy files).
5. Communication Email Management: Send, read, and organize emails through voice commands. Messaging: Integrate with messaging platforms (e.g., Slack, Microsoft Teams) to send and receive messages. Notifications: Provide alerts for incoming messages, emails, or calendar events.
6. Personalization User Profiles: Allow users to create and manage profiles for personalized experiences. Preferences Management: Enable users to set preferences for language, voice, and other settings. Learning Capability: Implement machine learning to adapt to user behavior and improve assistance over time.
7. Integration Third-party Service Integration: Integrate with external APIs (e.g., weather, news) for additional functionalities. Smart Home Control: Interface with smart home devices for control and automation.
8. Security and Privacy User Authentication: Require user authentication for accessing sensitive information or performing critical tasks. Data Encryption: Ensure that user data is encrypted and handled securely. Privacy Settings: Provide options for users to manage their data and privacy preferences.

9. User Feedback Feedback Mechanism: Include a feature for users to provide feedback on assistant performance and accuracy.

5.2 Non functional requirements

Non-functional requirements (NFRs) are critical to ensure that a virtual assistant for desktop applications performs well and meets user expectations. Here are some key non-functional requirements to consider:

1. Performance Response Time: The virtual assistant should respond to user queries within a specified time frame (e.g., less than 2 seconds). Throughput: The system should handle multiple requests concurrently without significant degradation in performance. Scalability: The assistant should be able to handle an increasing number of users or requests without compromising performance.
2. Usability User Interface: The interface should be intuitive and user-friendly, minimizing the learning curve for users. Accessibility: The virtual assistant should comply with accessibility standards (e.g., WCAG) to support users with disabilities. Customization: Users should be able to customize the assistant's settings, including voice, language, and response style.
3. Reliability Availability: The system should have high uptime (e.g., 99.9%) to ensure it is accessible when needed. Error Handling: The assistant should handle errors gracefully, providing informative messages and fallback options to users. Data Integrity: The assistant should ensure that user data is consistent and accurate, preventing data loss or corruption.
4. Security Authentication: The assistant should implement strong user authentication methods to protect sensitive information. Data Privacy: User data should be stored securely and comply with relevant data protection regulations (e.g., GDPR). Secure Communication: All data exchanged between the assistant and external services should be encrypted.

5. Maintainability Modularity: The architecture should support modular development, allowing for easy updates and feature additions. Documentation: Comprehensive documentation should be provided for both users and developers to facilitate maintenance and troubleshooting. Logging and Monitoring: The system should implement logging and monitoring capabilities to track usage patterns and detect issues.

5.3 Use case Diagram

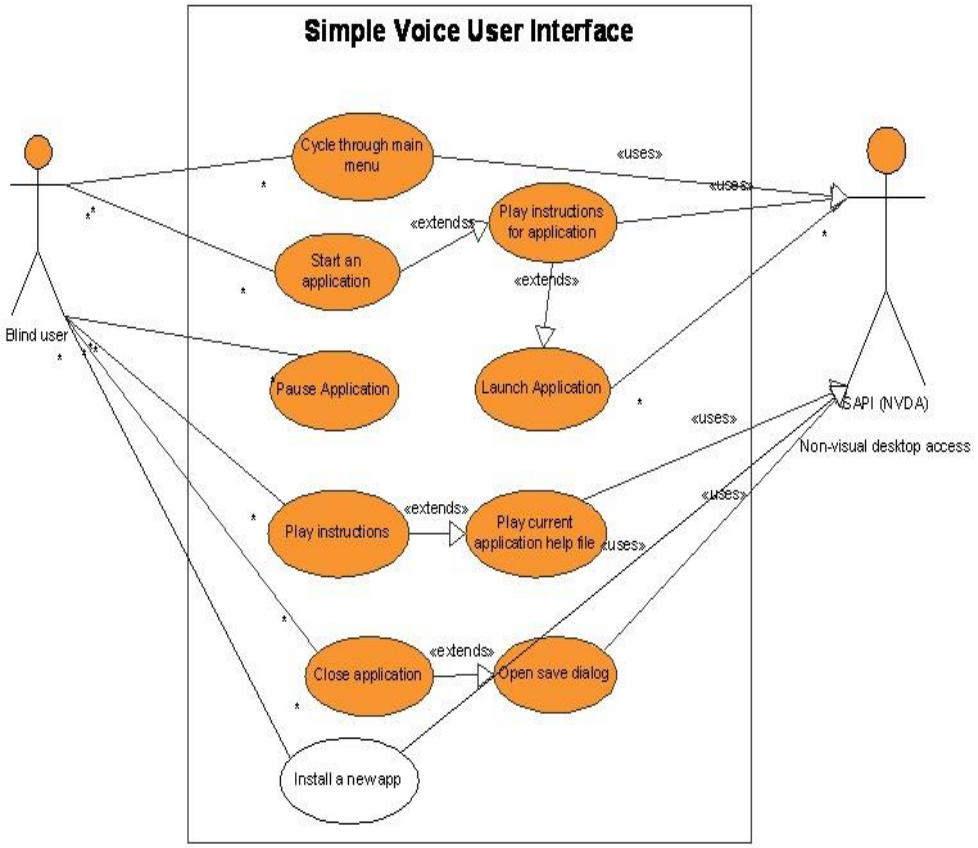


Fig.5.3 Use Case Diagram

CHAPTER 6
RESULTS & CONCLUSION

CHAPTER 6
RESULTS & CONCLUSION

6.1 Results & Conclusion

Our original methodology was limited because it only considered the implementation side of things in terms of the components and the architecture. We talked about two stages of chatbot development, which were really two components of the software system: knowledge abstraction and response generation. The former is subdivided into: data gathering, which is providing the raw data; data manipulation, which manages and classifies data for design purposes; and data augmentation, which is an extra step for increasing the number of examples available for the training process of the machine learning model. The later involved response generation, which manages how entities and intents are treated to show useful content depending on the type of interaction between the user and the chatbot. It is also important to highlight the relevance of capturing this information as nodes of a decision tree and choosing a convenient flow of conversation. Future work is going to outline the influence of virtual assistants in the performance of students. Collaboration with other institutions and replication within similar and different educational contexts is needed in order to make further studies and obtain conclusive results about the impact of virtual assistants in education. For instance, the case of study was one particular implementation (for a very specific set of needs, in a narrow context), what we need is a broader set of contexts from which to capture the data generated by users and analyze it.

6.2 Scope of work

The scope of work for a virtual assistant can vary depending on the needs of the employer or client, but generally includes tasks such as: Administrative Support: Managing emails, scheduling appointments, organizing files, and handling correspondence. Calendar Management: Scheduling meetings, appointments, and events, and coordinating with relevant parties. Data Entry: Inputting, updating, and maintaining databases, spreadsheets, and other documents. Travel Arrangements: Booking flights, hotels, and transportation, and preparing travel itineraries. Research: Conducting internet research on various topics, gathering data, and compiling reports. Customer Support: Responding to inquiries, resolving issues, and providing assistance to customers or clients. Media Management: Creating and scheduling posts, monitoring accounts, and engaging with followers. Content Creation: Writing, editing, and proofreading documents, articles, blog posts, and other content. Basic Bookkeeping: Managing invoices, expenses, and basic financial records. Personal Tasks: Assisting with personal errands, shopping, and other miscellaneous tasks. It's important for both parties to clearly define the scope of work to ensure expectations are metand tasks are completed efficiently.

References

1. *M. Bapat, H. Gune, and P. Bhattacharyya*, "A paradigm-based finite state morphological analyzer for marathi," in Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), pp. 26–34, 2010.
2. *B. S. Atal and L. R. Rabiner*, "A pattern recognition approach to voiced unvoiced-silence classification with applications to speech recognition," Acoustics, Speech and Signal Processing, IEEE Transactions on, vol. 24, no. 3,pp. 201–212, 1976.
3. *V.Radha and C. Vimala*, "A review on speech recognition challenges and approaches," doaj.
   org, vol. 2, no. 1, pp. 1–7, 2012.
4. *T. Schultz and A. Waibel*, "Language- independent and language adaptive acousticmodeling for speech recognition", SpeechCommunication, vol. 35, no. 1, pp. 31–51, 2001.
5. *J. B. Allen*, "From *lord rayleigh to shannon*: How do humans decode speech," in InternationalConference on Acoustics, Speech and Signal Processing, 2002.
6. *M. Bapat, H. Gune, and P. Bhattacharyya*, "A paradigm-based finite state morphological analyzer for marathi," in Proceedings of the 1st Workshop on South and Southeast Asian NaturalLanguage Processing (WSSANLP), pp. 26–34, 2010.
7. *G. Muhammad, Y. Alotaibi, M. N. Huda*, et al., pronunciation variation for asr: A survey of the"Automatic speech recognition for bangla digits," literature," Speech Communication, vol. 29, no.in Computers and Information Technology, 2009.2, pp. 225–246, 1999.
8. *S. R. Eddy, "Hidden Markov models*," Current opinion in structural biology, vol. 6, no. 3, pp.361–365, 1996.
9. Excellent style manual for science writers is "Speech recognition with flat direct models," IEEEJournal of Selected Topics in Signal Processing, 2010.
10. *Srivastava S., Prakash S.* (2020) Security Enhancement of IoT Based Smart Home Using

Hybrid Technique. In: Bhattacharjee A., *Borgohain S., Soni B., Verma G., Gao XZ.* (eds) MachineLearning, Image Processing, Network Security and Data Sciences. MIND 2020. Communications in Computer and Information Science, vol 1241. Springer, Singapore.
https://doi.org/10.1007/978-981-15-6318-8_44

Annexure I
Copy of certificate

Annexure I
Copy of certificate

International Journal of
Innovative Research in Technology
An International Open Access Journal Peer-reviewed, Refereed Journal
www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology (ISSN 2349-6002) is hereby awarding this certificate to

RUPALI LAXMAN MAMALE

In recognition of the publication of the paper entitled

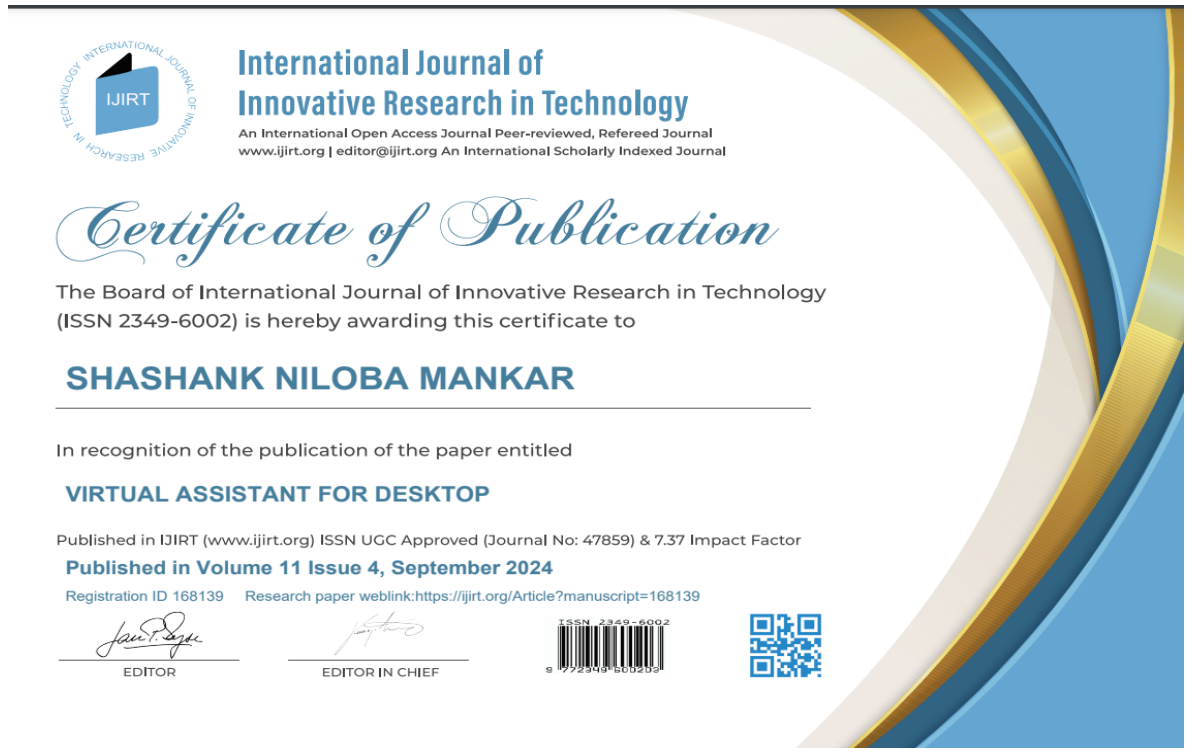
VIRTUAL ASSISTANT FOR DESKTOP

Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 4, September 2024

Registration ID 168139 Research paper weblink:https://ijirt.org/Article?manuscript=168139

EDITOR EDITOR IN CHIEF



International Journal of
Innovative Research in Technology
An International Open Access Journal Peer-reviewed, Refereed Journal
www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology (ISSN 2349-6002) is hereby awarding this certificate to

SHASHANK NILOBA MANKAR

In recognition of the publication of the paper entitled

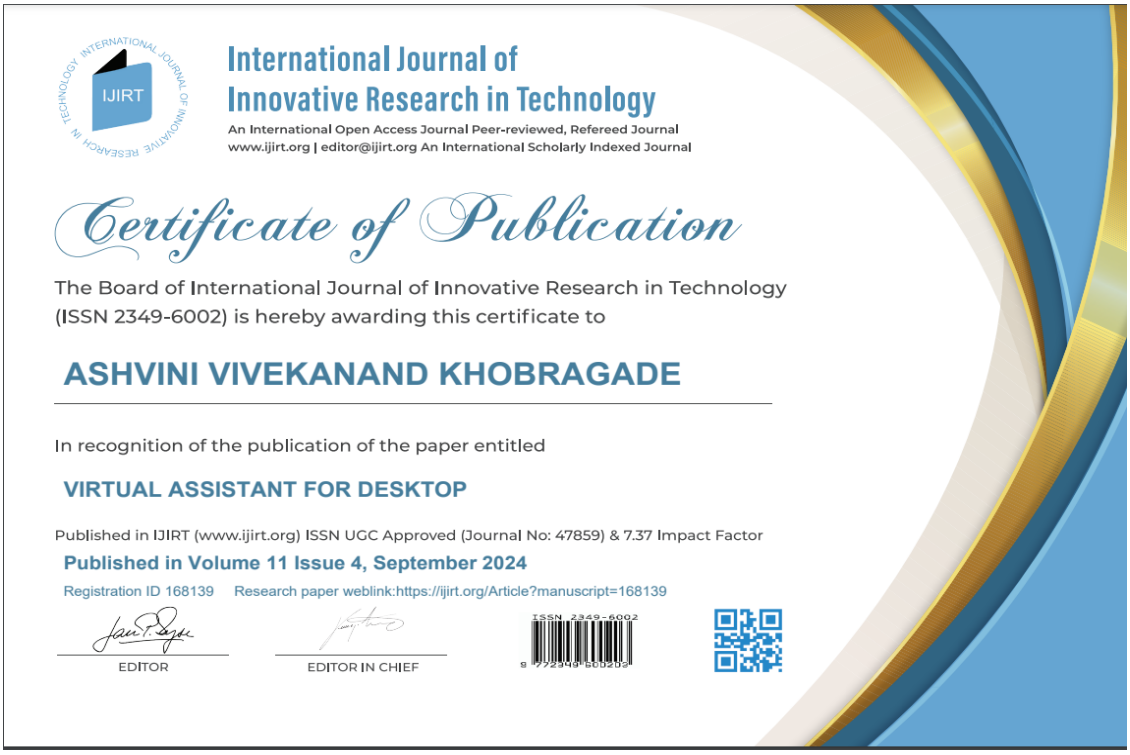
VIRTUAL ASSISTANT FOR DESKTOP

Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 4, September 2024

Registration ID 168139 Research paper weblink:https://ijirt.org/Article?manuscript=168139

EDITOR EDITOR IN CHIEF



International Journal of
Innovative Research in Technology
An International Open Access Journal Peer-reviewed, Refereed Journal
www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology
(ISSN 2349-6002) is hereby awarding this certificate to

ASHVINI VIVEKANAND KHOBRAGADE

In recognition of the publication of the paper entitled

VIRTUAL ASSISTANT FOR DESKTOP

Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 4, September 2024

Registration ID 168139 Research paper weblink:https://ijirt.org/Article?manuscript=168139

EDITOR EDITOR IN CHIEF

International Journal of
Innovative Research in Technology
An International Open Access Journal Peer-reviewed, Refereed Journal
www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology
(ISSN 2349-6002) is hereby awarding this certificate to

PRANAY ANANTA LOHABARE

In recognition of the publication of the paper entitled

VIRTUAL ASSISTANT FOR DESKTOP

Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 4, September 2024

Registration ID 168139 Research paper weblink:https://ijirt.org/Article?manuscript=168139

EDITOR EDITOR IN CHIEF

Annexure – II
PO, PSO Attainment

Annexure – II
PO, PSO Attainment

POs	KEYWORDS	PROGRAM OUTCOMES
PO1	Engineering knowledge	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. By applying the knowledge engineering we have made a Calculator. In this we have provided two Calculator they are as follows:- •Public Charging Calculator •Journey Cost Calculator
PO2	Problem analysis	Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. By applying the problem Analysis, we have provided a booking system for the user. The system works by select the charging station according to their current location and book a time slot for charging EV vehicle as per their convenience.
PO3	Design/development of solutions	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. By applying this we are

		providing the slot booking system and location finding so the user should be easily accessible to the EV.
PO4	Conduct investigations of complex problems	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. By applying this we have studied some research paper on our project through which we have analysis the data and we have make our system on that way where every user can easily use it at any time.
PO5	Modern tool usage	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations. By applying this we have provided the various resources to the user and we have also provided the search nearby CS technique from which the user can easily check the nearby CS according to its current location and in this we have also make two Calculator as a modern tool for getting the calculate cost of the electric vehicle easily.
PO6	The engineer and society	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. By using

		the engineering knowledge to build up this system which has uprooting the hazardous impact on the society as well as environment.
PO7	Environment and sustainability	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable Development. By applying it reduces the global warming, impact on the environment. We encourage the user to use the EV by providing the Calculator which gives the understanding between the Ev and conventional vehicles and also we provide some rewards to the user so to attract more customer which will be having an significant impact on the environment. It also reduces the carbon emission which is mostly occurs due to the conventional vehicles running on the road.

PO Attainment:-

PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

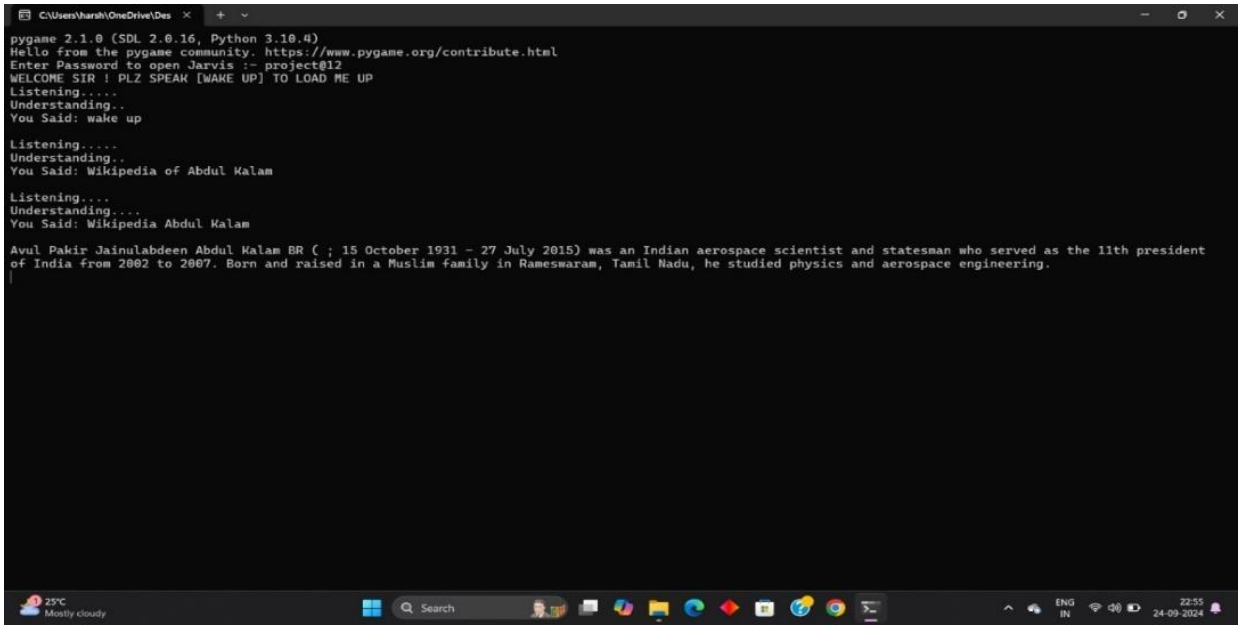
PROGRAMME SPECIFIC OUTCOMES (PSO):-

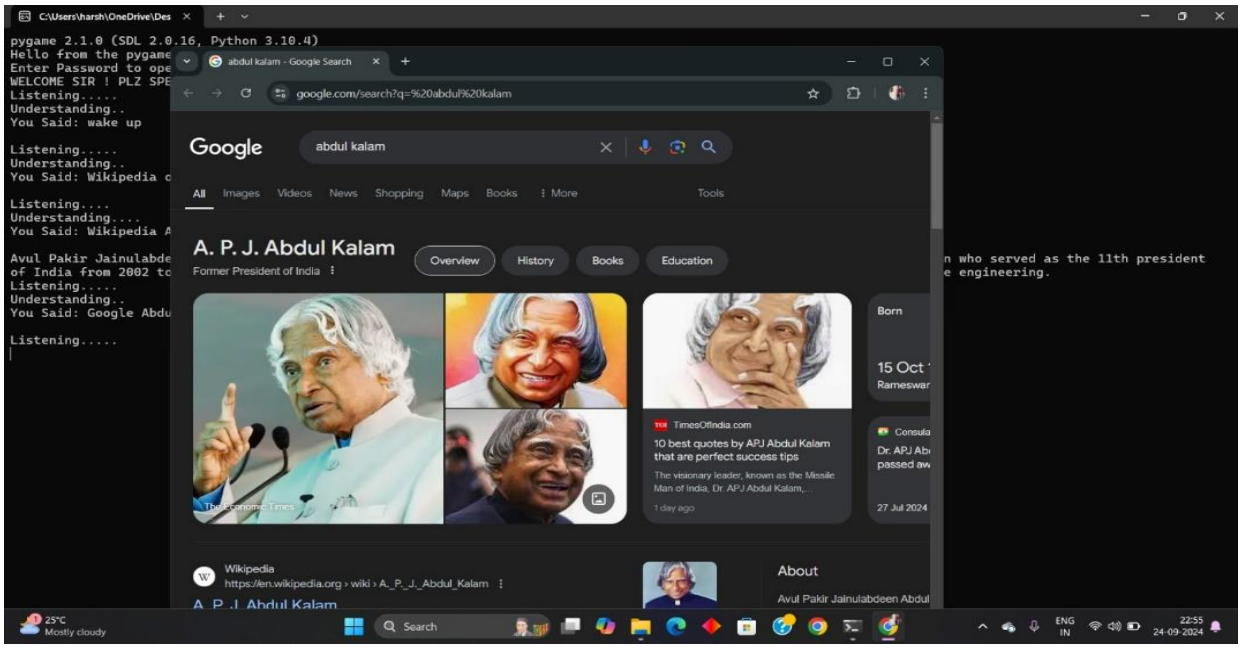
PSO	STATEMENT
PSO1	Design and deployment of hardware and software based computer systems meeting societal needs. As our system is the software system which is made for the user and they can

	easily access the services provided by the system.
PSO2	Skills towards development of web based systems, mobile apps, big data, data analytics and data mining towards sustainable solutions. Through this program we have develop the skills of programming like Html, CSS, JavaScript as a scripting language and databases as PHP. By learning this we have enhance our knowledge and gain hands on practice on it.

Annexure – III
Bulk Calculation/Printout of Output

Annexure – III
Bulk Calculation/Printout of Output





Annexure – IV

Resume

Annexure – IV
Resume

SHASHANK MANKAR

B.Tech. - Information Technology

Ph: +91-7770072377
Email: shashanknmankar.it22d@kdkce.edu.in
Nagpur, Maharashtra, India - 440027

EDUCATION

Karmaveer Dadasaheb Kannamwar College Of Engineering 2021 - 2025
B.Tech. - Information Technology | Percentage: **66.47 / 100**

Somayya Polytechnic, Chandrapur 2020
Diploma | Diploma - Computer Science and Engineering - Computer Science and Engineering | MSBTE |
Percentage: **81.03 / 100**

06717-KENDRIYA VIDYALAYA OF CHANDA CHANDRAPUR, Bhadravati 2015
10th | CBSE | CGPA: **6.60 / 10**

INTERNSHIPS

Jija IT Solutions PVT. LTD. | IT / Computers - Software 13 Mar, 2023 - 21 Apr, 2023
Data Analyst
Data Analyst Internship

Overview:
Completed a comprehensive Data Analyst internship, gaining hands-on experience in data collection, analysis, and visualization to support business decisions.

Key Responsibilities:

- Data Collection and Cleaning: Gathered and preprocessed data from multiple sources to ensure accuracy and reliability.
- Data Analysis: Utilized statistical tools (Excel, SQL, Python) to analyze datasets, identifying trends and patterns.
- Reporting and Visualization: Created detailed reports and visualizations using Tableau and Power BI to present insights to stakeholders.
- Collaboration: Worked with cross-functional teams to support data-driven decision-making processes.

Key Achievements:

- Improved data quality by implementing effective cleaning techniques.
- Delivered actionable insights that informed strategic business decisions.
- Developed and presented reports that enhanced data comprehension among team members.

PROJECTS

Virtual Assistant Using Python
Mentor: Prof s s Ganorkar | **Team Size:** 5
Key Skills: Development | Technical Analysis | Soft Skills | Software Development | Technical Documentation
Virtuala Assistant Project

Overview
The Virtual Assistant Project aims to develop an intelligent, conversational AI to assist users with tasks through natural language interactions. Leveraging advanced machine learning and NLP, the assistant provides real-time help, integrates with various services, and offers personalized experiences.

Key Objectives:
- Manage reminders, schedules, and tasks.
- Connect with third-party platforms.
- Adapt to user habits and preferences.
- Continuously learn from interactions.

Features:
- Natural language understanding with context awareness.
- Voice and text interaction modes.
- Task organization and information retrieval.
- Secure integration with external services.

Technology Stack:
- Advanced ML/NLP models, Cloud Computing, and secure APIs.

Benefits:
- Boosts productivity and convenience.
- Offers a personalized and engaging experience.

Future Plans:
- Incorporate emotional intelligence, multilingual support, and advanced analytics..

PERSONAL DETAILS

Gender: Male
Marital Status: Single
Current Address: Infinity clinics, Omkar nagar , nagpur, Nagpur, Maharashtra, India - 440027
Date of Birth: 18 Nov, 1999
Known Languages: English, Hindi, Marathi
Phone Number: +91-7770072377
Emails: shashanknmankar.it22d@kdkce.edu.in , Shashankmankar412@gmail.com

Pranay Ananta Lohabare

At.Post-Tarsa Sub_Dist-Mouda Dist-Nagpur 441106
9607871176 | pranayloh14@gmail.com

Objective

To work in an environment which encourages me to succeed and grow professionally where I can utilize my skills and knowledge appropriately.

Education

- **Janta High School,Mouda** 2017
  SSC
  83.20%
- **Janta Junior College,Mouda** 2019
  HSC
  57.08%
- **Government Polytechnic,Nagpur** 2022
  Diploma
  81.10%
- **KDK College Of Engineering, Nagpur** -
  B.TECH-Information Technology(Pursuing-Forth Year))
  61.08%(7th Semester)

Key Skill

- Basic Python & Advance Python My SQL

Projects

- **Virtual Assistant**
  To create a virtual assistant capable of assisting users with tasks such as managing schedules, providing information, controlling smart devices, or offering personalized recommendations.
  Technologies Used
  Programming Languages: Python
  APIs: Integration with external services (e.g., Google Calendar API, Spotify API).
  Speech Recognition: Tools like Google Speech-to-Text or IBM Watson.

Personal Details

- Name-Pranay Ananta Lohabare
  DOB-14-06-2001
  Gender-Male
  Nationality-Indian
  Language- English,Hindi, Marathi

I hereby declare that the details and information given above are complete and true to the best of my knowledge.

Date:- Signature
Place:-

RUPALI MAMALE

B.Tech. - Information Technology
Ph: +91-9322035423
Email: rupalilmamale.it22d@kdkce.edu.in
Nagpur, Maharashtra, India - 440002

EDUCATION

Karmaveer Dadasaheb Kannamwar College Of Engineering 2021 - 2025
B.Tech. - Information Technology | Percentage: 72.33 / 100

Purnamal Lahoti Government Polytechnic Latur 2022
Diploma | Diploma - IT | MSBTE | Percentage: 79.81 / 100

Swami vivekanand vidyalaya Trikoli, Umarga 2019
10th | Maharashtra State board of Secondary and higher Secondary education Pune | Percentage: 80.40 / 100

INTERNSHIPS

K D K College of Engineering Nagpur 13 Mar, 2023 - 30 Apr, 2023
Nagpur
Is Certified in Salesforce Admin and has successfully completed his/her 1.5 month Training/Intership Program administrated by Technical expert as CSM IT services Pvt.Ltd.

PROJECTS

Virtual Assistant Using Python
Mentor: Prof.Priyanka Dewalkar | **Team Size:** 5
Key Skills: Natural language processing ,Speech recognition | Artificial intelligence, Modules of python

PUBLICATIONS / RESEARCH / WHITE PAPERS

Virtual Assistant Using Python 15 Mar, 2024
XX National Conference on Emerging Trends in Engineering and Technology | **No. of Authors:** 6
Key Skills: Speech Recognition Natural language processing Artificial intelligence Modules of python Action integration module

ACHIEVEMENTS

- MS-CIT certified with grade 82

ASSESSMENTS / CERTIFICATIONS

Journey to Cloud: Envisioning your Solution

Introduction to C++ and data Structure

XX National Conference on Emerging Trends in Engineering and Technology

For participating state level poster presentation by information Technology in puranmal Lahoti Government Polytechnic Latur

EXTRA CURRICULAR ACTIVITIES

- I have participated in state level poster presentation in Latur

PERSONAL DETAILS

Gender: Female
Marital Status: Single
Current Address: Darshan Colony Nandanvan Nagpur, Nagpur, Maharashtra, India - 440002

Email: rupalilmamale.it22d@kdkce.edu.in

Date of Birth: 05 Jun, 2003
Known Languages: Marathi ,Hindi, English
Permanent Address: at post wagdari taluka-omerga dist-osmanabad, Umarga, Maharashtra, India - 413606
Phone Number: +91-9322035423

ASHWINI KHOBRAGADE

B.Tech. - Information Technology

Ph: +91-9021408904

Email: ashwinivkhobragade.it22d@kdkce.edu.in

Nagpur, Maharashtra, India - 440009

EDUCATION

Karmaveer Dadasaheb Kannamwar College Of Engineering 2021 - 2025

B.Tech. - Information Technology | Percentage: **63.44** / **100**

Government Polytechnic, Bramhapuri, Bramhapuri 2022

Diploma | MSBTE | Percentage: **68.50** / **100**

Shri Dnyanesh Mahavidyalaya, Nawargaon, Nawargaon 2020

12th | State board | Percentage: **54.46** / **100**

Bharat vidyalaya Nawargaon 2018

10th | Nagpur | Percentage: **73.60** / **100**

PERSONAL DETAILS

Gender: Female

Marital Status: Single

Current Address: Darshan colony Nandanvan Nagpur, Nagpur, Maharashtra, India - 440009

Date of Birth: 21 Mar, 2002

Known Languages: English, Marathi, Hindi

Phone Numbers: +91-9021408904, +91-7798713728

Emails: ashwinivkhobragade.it22d@kdkce.edu.in , ashu.kho03@gmail.com

Annexure V
Plagiarism Report

Annexure V
Plagiarism Report

Annexure VI
Poster

K.D.K COLLEGE OF ENGINEERING NAGPUR
DEPARTMENT OF INFORMATION TECHNOLOGY
"VIRTUAL ASSISTANT FOR DESKTOP"

Name of Student:- 1.Ashvini Khobragade 2.Rupali Mamale 3.Pranay Lohbare 4.Shashank Mankar
Name of Guide:-Assist.Prof.Ganorkar

Abstract:-his paper describes a research project to create a virtual assistant for computers that can perform various tasks using natural language and machine learning techniques.The virtual assistant is designed to assist users with tasks such as searching the web, managing files,scheduling, sending emails, etc.The implementation process uses a combination of speech recognition and speech management to enable users to interact with the spoken language assistant. The research project included several stages, including data collection and preprocessing, feature extraction, model training and evaluation, and system integration and experimentation. The data used for training and evaluation is collected from a variety of sources, including publicly available data and user interactions with the system.Video extraction process involves extracting relevant features from the material such as acoustic features, speech features and content features. The training model and evaluation phase will develop and evaluate different learning models for various tasks such aslanguage recognition, language comprehension and speech management.

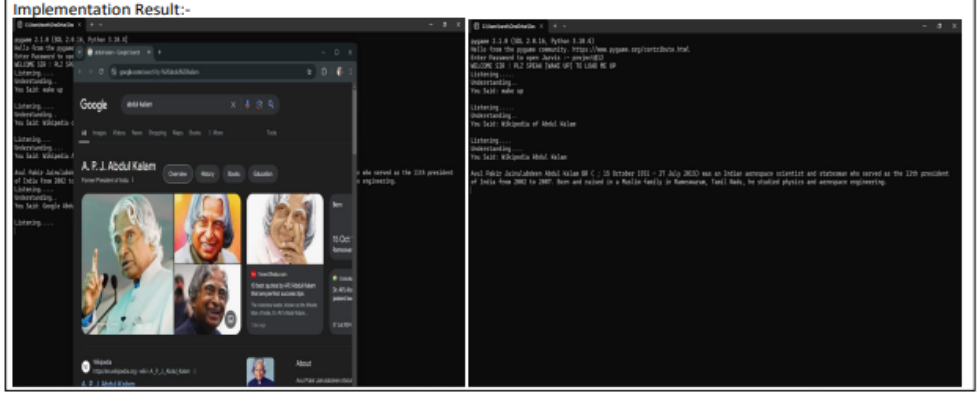
Introduction:-The use of virtual assistants such as Siri, Alexa, Google Assistant has increased in recent years and has become an important part of our daily lives. These virtual assistants use natural language processing and machine learning technology to help users interact with them using speech and tasks such as browsing the web, playing music, setting reminders and more. However, most virtual assistants are designed for mobile devices and there is a processing growing demand for similar systems on desktop computers. The aim of the research project is to create a virtual assistant for desktop computers that can perform various tasks using natural language and machine learning techniques. The proposed system is designed to help users perform various tasks such as browsing the web, managing documents, scheduling appointments and sending emails.The system uses a combination of speech recognition, natural language understanding, and speech management to allow users to interact with the assistant using spoken language. please improve The system consists of several stages, including data collection and preprocessing, model extraction, model training and evaluation, integration, and testing. The data used for training and evaluation is collected from various sources, including publicly available data and user interactions with the system.

Flowchart:-

Result:-Our original methodology was limited because it only considered the implementation side of things in terms of the components and the architecture. We talked about two stages of chatbot development, which were really two components of the software system: knowledge abstraction and response generation. The former is subdivided into: data gathering, which is providing the raw data; data manipulation, which manages and classifies data for design purposes; and data augmentation, which is an extra step for increasing the number of examples available for the training process of the machine learning model.

Conclusion:- The later involved response generation, which manages how entities and intents are treated to show useful content depending on the type of interaction between the user and the chatbot. It is also important to highlight the relevance of capturing this information as nodes of a decision tree and choosing a convenient flow of conversation. Future work is going to outline the influence of virtual assistants in the performance of students. Collaboration with other institutions and replication within similar and different educational contexts is needed in order to make further studies and obtain conclusive results about the impact of virtual assistants in education. For instance, the case of study was one particular implementation (for a very specific set of needs, in a narrow context), what we need is a broader set of contexts from which to capture the data generated by users and analyze it.

Implementation Result:-



References

1. *M. Bapat, H. Gune, and P. Bhattacharyya*, "A paradigm-based finite state morphological analyzer for marathi," in Proceedings of the 1st Workshop on South and Southeast Asian NaturalLanguage Processing (WSSANLP), pp. 26–34, 2010.
2. *B. S. Atal and L. R. Rabiner*, "A pattern recognition approach to voiced unvoiced-silence classification with applications to speech recognition," Acoustics, Speech and Signal Processing,IEEE Transactions on, vol. 24, no. 3,pp. 201–212, 1976.
3. *V.Radha and C. Vimala*, "A review on speech recognition challenges and approaches," doaj. org, vol. 2, no. 1, pp. 1–7, 2012.
4. *T. Schultz and A. Waibel*, "Language- independent and language adaptive acousticmodeling for

Program Outcomes

Po1	Po2	Po3	Po4	Po5	Po6	Po7	Po8	Po9	Po10
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Name and Sign of student

1.Ashvini Khobragde
2.Rupali Mamale
3.Pranay Lohabare
4.Shashank Mankar