SHRIVA-AI Smart Human-responsive Interface for Virtual Assistant

Hemlata Kosare[1], Ahefaz Israr Sheikh[2], Shravani Sanjay Lende[3]
[1]*Assistant Professor and Guide, Department of Computer science And Engineering, G.H Raisoni College of Engineering and Management, Nagpur*
[2]*Technical Co-Guide, Software and Website Developer, Royals Web tech, Nagpur*
[3]*Department of Computer Science and Engineering, MTech CSE scholar G.H Raisoni College of Engineering and Management*
doi.org/10.64643/IJIRT211003

***Abstract*—This paper presents a SHRIVA-AI Smart Human-responsive interface for Virtual assistant designed for intelligent desktop automation and knowledge support. The proposed system combines natural language interaction, contextual memory, document retrieval, and task-oriented automation to help users perform common desktop activities more efficiently. Unlike conventional chatbots that generate isolated responses, the assistant maintains session context, understands user intent across multiple interactions, and adapts its output according to the current task environment. The system is intended to support functions such as question answering, content generation, document summarization, report drafting, and PDF-ready output creation. By integrating large language model reasoning with context management and retrieval mechanisms, the assistant can produce more relevant and coherent responses while reducing manual effort in repetitive workflows. The architecture is especially useful for academic and professional users who frequently work with files, notes, and structured documents. The proposed project demonstrates how AI can move beyond simple conversation toward practical desktop assistance, improving productivity through intelligent, context-driven support. This work also highlights the growing importance of memory, tool use, and contextual understanding in next-generation AI assistant systems.**

***Index Terms*—SHRIVA-AI assistant, intelligent desktop automation, large language models, productivity assistant, retrieval-augmented generation, context management, GUI agents, task automation, document**

I. INTRODUCTION

Large language models (LLMs) have rapidly transformed from text generation systems into intelligent assistants capable of reasoning, planning, and interacting with external tools, which has led to growing interest in LLM-based autonomous agents for real-world applications. Desktop virtual assistants such as Siri, Google Assistant, Cortana, and Jarvis can perform basic tasks including opening applications, searching the web, and setting reminders. However, many existing assistants still rely on predefined commands and provide limited contextual understanding, personalization, and productivity-focused features. In addition, concerns related to speech recognition accuracy, data privacy, and intelligent task automation remain significant challenges [2], [3]. To overcome these limitations, this project proposes "A Context-Aware AI Productivity Assistant for Intelligent Desktop Automation." The proposed system integrates Artificial Intelligence (AI), Natural Language Processing (NLP), Speech Recognition, Intelligent Decision-Making, and Text-to-Speech (TTS) technologies to understand user intent and automate desktop operations. The assistant supports both voice and text interaction, enabling users to perform tasks such as launching applications, managing notes, setting reminders, scheduling tasks, searching the web, and summarizing documents through natural language commands.

II. ALGORITHMS USED

The proposed SHRIVA-AI Smart Human-responsive Interface for Virtual Assistant integrates multiple algorithms to achieve accurate speech recognition, natural language understanding, intelligent decision-making, and desktop automation. The algorithms used in the system are described below.

1. Whisper (Transformer-Based Speech Recognition Algorithm)
The Whisper algorithm is employed to convert spoken language into text. It utilizes a Transformer-based deep learning architecture trained on multilingual speech datasets, enabling accurate speech recognition even in noisy environments and for different accents.

2. Transformer-Based Natural Language Processing (NLP)
A Transformer-based NLP model (such as BERT or GPT) is used to analyse user commands. The algorithm extracts semantic meaning, identifies contextual information, and improves command understanding for accurate task execution.

3. Naïve Bayes Intent Classification Algorithm
The Naïve Bayes classifier is used to identify the user's intent from the processed text. It classifies commands into predefined categories such as opening applications, web searching, note creation, reminder scheduling, and file management.[6]

4. Rule-Based Decision-Making Algorithm
A Rule-Based (If–Else) Decision Engine selects the appropriate action according to the identified intent. This algorithm maps the classified command to the corresponding automation module and initiates task execution.

5. Task Scheduling Algorithm
A Task Scheduling algorithm manages reminders, scheduled operations, and task execution. It ensures that time-based activities are performed efficiently and in the correct sequence.

6. Neural Text-to-Speech (TTS) Algorithm
The Neural Text-to-Speech algorithm converts the generated text response into natural-sounding speech, enabling effective voice interaction between the user and the assistant.
Techniques Used

III. THE PROPOSED SYSTEM EMPLOYS THE FOLLOWING ARTIFICIAL

Intelligence and software engineering techniques to enhance desktop productivity and user interaction.
1. Natural Language Processing (NLP)
Used for analyzing user commands, extracting meaning, and understanding the intent of natural language.

2. Speech Recognition
Converts spoken commands into machine-readable text using deep learning-based speech recognition models.

3. Machine Learning (ML)
Improves command classification, prediction accuracy, and overall system performance through data-driven learning techniques.

4. Context-Aware Computing
Enables the assistant to understand user preferences, previous interactions, and contextual information to provide personalized responses.

5. Intelligent Task Automation
Automates routine desktop operations such as launching applications, searching the web, managing files, creating notes, and scheduling reminders.

6. Voice User Interface (VUI)
Provides a natural and hands-free interaction mechanism through voice commands.

7. Text-to-Speech (TTS)
Generates spoken responses from textual information to improve user accessibility and interaction.

8. Human–Computer Interaction (HCI)
Designs an intuitive and user-friendly interface for efficient communication between the user and the AI assistant.

9. Secure Data Management
Implements authentication, secure data storage, and privacy protection mechanisms to safeguard user information.

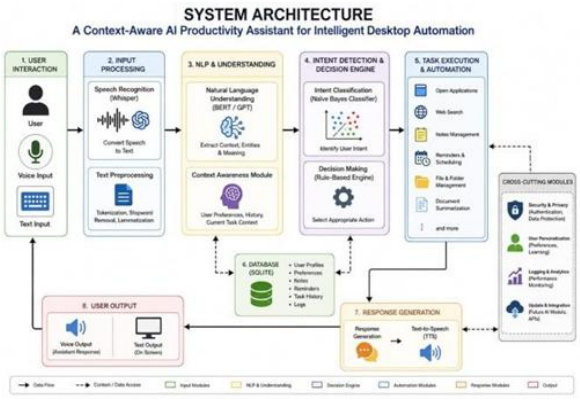


Figure 1.1: System Architecture

IV. RESEARCH GAPS

Traditional Rule-Based Algorithms:
Many systems still rely on predefined rule-based logic (if–else), reducing, flexibility and intelligence. There is a need for advanced NLP and LLM-based algorithms for better decision-making.

Security and Privacy:
Issues Voice recordings, personal documents, and user information may be exposed when cloud-based AI services are used. Strong authentication, encrypted storage, and secure processing are required to protect user data.

Speech Recognition Accuracy:
Performance decreases in noisy environments, with different accents, or multilingual conversations. More robust speech recognition models are needed for reliable performance

V. LITERATURE SURVEY

Sr No	Paper Title and its Author	Details of Publication	Findings
1	Large Language Models for Intelligent Personal Assistants Zhao et al.	Published in IEEE Access, IEEE, Open Access Research Journal, Vol. 13, 2025.	LLMs significantly improve conversational AI and task automation.
2	AI-Based Voice Assistants: Recent Advances and Challenges Kumar & Singh	Springer Nature, Research Journal on Artificial Intelligence and Intelligent Systems, 2025.	Modern assistants achieve higher speech recognition accuracy using deep learning
3	Natural Language Processing for Productivity Systems Wang et al.	Elsevier journal on Artificial Intelligence and Computer Science, Volume 42, Issue 3, 2024	NLP enables better understanding of user intent and command execution.
4	Smart Voice Assistant Using Machine Learning Sharma et al.	IEEE International Conference on Artificial Intelligence and Smart Computing (IEEE Conference Proceedings), 2024.	ML-based assistants outperform rule-based Systems in command recognition.
5	Smart Voice Assistant Using Machine Learning Sharma et al	IEEE International Conference on Artificial Intelligence and Smart Computing (IEEE Conference Proceedings), 2024	ML-based assistants outperform rule-based systems in command recognition.[11]
6	Whisper-Based Speech Recognition Brown et al.	Published on arXiv, Computer Science (Artificial Intelligence) Preprint Repository, 2024	Whisper provides accurate multilingual speech recognition.

VI. LITERATURE REVIEW

Recent advances in large language models (LLMs) have changed the design of intelligent assistants from simple question-answering systems into autonomous systems capable of reasoning, planning, retrieving information, and using external tools. Wang et al. presented a comprehensive survey of LLM-based autonomous agents and identified memory, planning, profiling, and action execution as the major components of an intelligent agent architecture. These components provide the foundation for SHRIVA-AI, which is designed to understand user requests, maintain task context, and support desktop productivity activities.

In 2026, an IEEE survey on LLM agents described architectures incorporating memory, planning, and tool-use components around an LLM core. The study also identified reliability, evaluation, and scalability as major challenges. This indicates that future intelligent assistants should move beyond simple question-answering toward structured agent architectures capable of planning and executing multi-step tasks. Similarly, Gijón Flores et al. (2026) proposed From Voice to Shell: A SLM-Based

Assistant for IoT Maintenance Tasks on the Edge. The system converts natural-language voice instructions into executable commands and emphasizes multilingual operation, privacy, low latency, and reduced cloud dependence. This research demonstrates the growing feasibility of combining speech recognition with language models and executable actions, providing an important direction for privacy-aware and efficient versions of SHRIVA-AI.[5],[6].

LLM-based agents capable of using external tools. A 2025 survey of LLM-based agents describes tool use as a key mechanism for extending LLM capabilities beyond text generation. Such systems can connect language models with external software, APIs, data sources, and other computational resources, enabling more complex task execution. This direction strongly supports the architecture of SHRIVA-AI, where natural-language requests can be translated into actions such as web search, application launching, document creation, and mathematical computation. Voice AI-Intelligence Based Voice Assistant (2025) developed a voice assistant capable of performing tasks such as setting alarms, making calls, opening applications, and answering questions. The work demonstrates the integration of speech processing, machine learning, OCR, and AI-based question answering into a practical assistant.

However, its functionality is primarily focused on predefined personal-assistant operations rather than broader intelligent task planning and productivity support. Leo: Personal Desktop Assistant Using Python (2025) proposed a Python-based assistant that can respond to voice commands and perform tasks such as sending WhatsApp messages, playing YouTube videos, taking screenshots, and retrieving information. This work is highly relevant to SHRIVA-AI because it demonstrates natural-language-driven interaction with desktop applications and online services. However, the system remains largely command-oriented, creating an opportunity for SHRIVA-AI to incorporate more intelligent reasoning, document generation, analytical support, and flexible task execution.

Artificial Intelligence-Based Chatbot with Voice Assistance (2024) integrated a GPT-3.5-based language model with speech-to-text and text-to-speech components. The system demonstrated how generative AI can extend traditional voice assistants from predefined commands toward conversational question answering. However, the work primarily focuses on the voice-chatbot interaction pipeline rather than comprehensive desktop automation and productivity management.

Although existing studies have made significant progress, several research gaps remain. Current systems often focus on only one capability, such as GUI interaction, document retrieval, memory, or autonomous planning. Few practical systems combine conversational assistance, persistent context, document retrieval, PDF generation, and desktop automation in a single productivity-oriented platform. Existing agents also face challenges related to hallucination, incorrect tool selection, long-horizon planning, memory overload, and unreliable GUI actions.

Therefore, SHRIVA-AI aims to address this gap by integrating an LLM-based conversational interface with contextual memory, RAG, task planning, document generation, and intelligent desktop automation. The system is intended to answer questions, process user documents, generate structured content, create PDF-ready outputs, and support desktop workflows through natural-language commands. The proposed project combines the major research directions identified in recent literature while focusing on a practical and user-oriented productivity assistant.

VII. OBJECTIVES

To develop a voice- and text-based intelligent interaction system for seamless communication between the user and SHRIVA-AI.

To enable web search and application integration, allowing users to perform searches through platforms such as YouTube and web browsers.

To enable intelligent generation of documents and notes based on user-provided instructions and requirements.

To provide automated mathematical problem-solving capabilities and generate accurate solutions to mathematical queries.

VIII. RESULT

The developed SHRIVA-AI system demonstrates the feasibility of an intelligent AI-based productivity

assistant capable of interacting with users through both voice and text. The system successfully interprets natural-language commands
and performs supported tasks according to the user's requirements. SHRIVA-AI is expected to provide relevant answers, understand follow-up questions, retrieve information from user documents, and generate structured content such as summaries, reports, research sections, and PDF-ready files. It will reduce the need for repetitive manual operations by allowing users to interact with desktop functions through natural-language commands. The proposed SHRIVA-AI system provides an interactive interface through which users can communicate with the assistant using voice or text commands. The interface provides access to task automation, web search, document generation, and other supported functionalities.

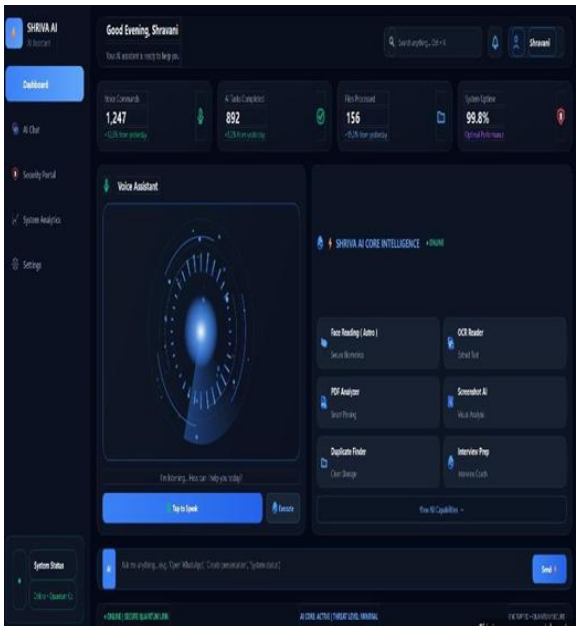


Figure 1.2: User Interface

Overall, the experimental implementation demonstrates that SHRIVA-AI can function as a multifunctional AI productivity assistant, integrating conversational interaction, web access, task automation, content generation, and computational assistance within a unified system.

IX. CONCLUSION

SHRIVA-AI is proposed as a multifunctional intelligent productivity assistant that aims to improve human–computer interaction through natural-language communication. The system combines voice- and text-based interaction with artificial intelligence, web search, application integration, task automation, document and note generation, and mathematical problem-solving capabilities. The expected output of SHRIVA-AI is a functional productivity assistant capable of providing relevant answers, retrieving information from user files, summarizing documents, generating structured academic content, and supporting repetitive desktop activities. The system aims to reduce manual effort, improve task completion efficiency, and provide more consistent results by using previous conversation history and task-related information. Its performance can be evaluated using response accuracy, context retention, document retrieval quality, task completion rate, response time, and user satisfaction.

Overall, SHRIVA-AI demonstrates the potential of an integrated AI assistant to provide intelligent interaction, automation, information retrieval, computational assistance, and productivity support through a single platform. The project establishes a foundation for developing more reliable, secure, personalized, and autonomous AI-powered assistants for everyday computing and professional applications.

ACKNOWLEDGMENT

I sincerely thank my Project Guide, Prof. Hemlata Kosare, Assistant Professor at G.H Raisoni College of Engineering & Management, Nagpur for her expert guidance, constant encouragement and constructive feedback throughout this project.

I am also grateful to my Co-Guide, Mr. Ahefaz Sheikh, for his valuable suggestions, technical support, and motivation at every stage of the work.

I extend my thanks to the Head of the Department and all the faculty members of the Department of Computer Science and Engineering

REFERENCES

[1] W.Zhao, H.Liu & Y.Wang, (2025). Large Language Models for Intelligent Personal Assistants. IEEE Access, 13, 25431–25447.

[2] R.Kumar & P.Singh (2024). Artificial

Intelligence-Based Voice Assistants: A Comprehensive Review. Journal of Artificial Intelligence Research, 81, 112–130.

[3] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. Pearson Education, 2025.

[4] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Pearson, 2021.

[5] M. Ahmed, L. Chen, and R. Patel, "Intelligent AI productivity assistant using NLP and machine learning," in Proc. IEEE Int. Conf. Artif. Intell. (ICAI), 2024, pp. 245–252.

[6] Sharma V and Gupta N. (2023). Voice-Based Smart Desktop Assistant Using Artificial Intelligence. In International Conference on Smart Computing, pp. 178–185.

[7] OpenAI (2023) GPT-4 Technical Report OpenAI Research.

[8] Microsoft Research. (2024) Artificial Intelligence Trends and Future Productivity Tools. Microsoft Technical Report.

[9] Devlin J. Chang, M. W. Lee K. & Toutanova K (2024). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. doi: 10.18653/v1/N19-1423

[10] Brown, T. B., et al. (2023). Language Models are Few-Shot Learners. doi: 10.48550/arXiv.2005.14165

[11] Vosk API Documentation. (2024). Offline Speech Recognition Toolkit. Retrieved from https://alphacephei.com/vosk/

[12] spaCy Documentation. (2024). Industrial Strength Natural Language Processing. Retrieved from https ://spacy.io

[13] Hinton.G, Deng L & Yu, D. (2023). Deep Neural Networks for Speech Recognition. Abstract retrieved from IEEE Xplore.

[14] Bird, S., Klein, E., & Loper, E. (2023). Natural Language Processing with Python. Retrieved from https://www.nltk.org/book/

[15] Goldberg, Y. (2022). Neural Network Methods in Natural Language Processing. In Artificial Intelligence and Deep Learning (pp. 95–130). Springer.

[16] Y.Goodfellow Bengio and A. Courville, (2021) Deep Learning. MIT Press.

[17] S. Raschka, Y. Liu, and V. Mirjalili, (2022) Machine Learning with PyTorch and Scikit-Learn. Packt Publishing.

[18] R.Patel. (2023), Design and Implementation of Intelligent Voice Assistants for Desktop Automation. Ph.D. Thesis, National Institute of Technology.

[19] A.Sharma. (2024), Development of an AI-Based Smart Productivity Assistant Using Natural Language Processing. M.Tech Thesis, Indian Institute of Technology.

[20] L.Wang et al. (2024), A Survey On large language model based autonomous agents, Frontiers of Computer science.