

**VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF
TECHNOLOGY**
An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering



Project Report on

**SmartGuide: Chatbot for Efficient Employee Support
and Document Analysis**

In partial fulfillment of the Fourth Year, Bachelor of Engineering (B.E.)
Degree in Computer Engineering at the University of Mumbai
Academic Year 2024-25

Submitted by
Shreya Hadkar (D17B , 16)
Johan John (D17B , 22)
Manali Patil(D17B, 38)
Vedang Rath (D17B, 44)

Project Mentor
Dr. (Mrs.) Gresha Bhatia

(2024-25)

**VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF
TECHNOLOGY**

**An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering**



Certificate

This is to certify that **Shreya Hadkar (D17B- 16), Johan John (D17B- 21), Manali Patil (D17B- 38), & Vedang Rathi (D17B- 44)**, of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on “***SmartGuide: Chatbot for Efficient Employee Support and Document Analysis***” as a part of their coursework of PROJECT-II for Semester-VIII under the guidance of their mentor **Dr. (Mrs) Gresha Bhatia** in the year 2024-25 .

This project report entitled **SmartGuide: Chatbot for Efficient Employee Support and Document Analysis** by **Vedang Rathi, Manali Patil, Shreya Hadkar, & Johan John** is approved for the degree of **B.E. Computer Engineering**.

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date: 28 April 2025

Project Guide:

Project Report Approval For B. E (Computer Engineering)

This project report entitled **SmartGuide: Chatbot for Efficient Employee Support and Document Analysis** by **Shreya Hadkar, Johan John, Manali Patil, & Vedang Rathi** is approved for the degree of **B.E. Computer Engineering**.

Internal Examiner

External Examiner

Head of the Department

Principal

Date: 28 April 2025

Place: Chembur, Mumbai

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Shreya Hadkar (16)

Johan John (21)

Manali Patil (38)

Vedang Rathi (44)

Date: 28 April 2025

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Deputy Head of Department **Dr. (Mrs) Gresha Bhatia** for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr.(Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair** , for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

Computer Engineering Department
COURSE OUTCOMES FOR B.E PROJECT

Learners will be to,

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solutions for the problem.
CO 4	Able to interpret the data and datasets to be utilized.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one's competency.
CO 11	Able to develop a professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Index

Chapter No	Title	Page No.
1	Introduction	
1.1	Introduction to the project	1
1.2	Motivation for the project	1
1.3	Problem Definition	2
1.4	Existing Systems	2
1.5	Lacuna of the existing systems	2
1.6	Relevance of the Project	3
2	Literature Survey	
2.1	Brief overview of Literature Survey	4
2.2	Research Papers a. Abstract of the research paper b. Inference drawn from the paper	4
2.3	Patent search a. Title of the patent and year of the patent b. Summary of the patent	9
2.4	Inference Drawn	11
2.5	Comparison with the Existing Systems	12
3	Requirement Gathering for the proposed System	
3.1	Introduction to Requirement Gathering	13
3.2	Functional Requirements	13
3.3	Non-Functional Requirements	14
3.4	Hardware, Software, Technology and tools utilised	14
3.5	Constraints	15
4	Proposed Design	
4.1	Block diagram of the system	15
4.2	Modular design of the system	16
4.3	Detailed Design (Flowchart)	17

4.4	Project Scheduling & Tracking using Timeline / Gantt Chart	19
5	Implementation of the Proposed System	
5.1	Methodology employed for development	20
5.2	Algorithms and flowcharts for the respective modules developed	23
5.3	Datasets source and utilisation	24
6	Testing of the Proposed System	
6.1	Introduction to testing	26
6.2	Types of tests Considered	27
6.3	Various test case scenarios considered	28
6.4	Inference drawn from the test cases	31
7	Results and Discussions	
7.1	Screenshots of User Interface (UI) for the respective module	32
7.2	Performance Evaluation measures	35
7.3	Input Parameters / Features considered	37
7.4	Comparison of results with existing systems	39
7.5	Inference drawn	41
8	Conclusion	
8.1	Limitations	44
8.2	Conclusion	44
8.3	Future Scope	45
	References	47
	Appendix	
1	Paper 1	
a	Paper 1	48
b	Plagiarism Report of Paper I	55
c	Project review sheet	56

List of Figures

Figure No	Heading
Fig 4.1	Block Diagram of the System
Fig 4.2	Modular Design of the System
Fig 4.3	Detailed Design
Fig 4.4	Project Scheduling & Tracking using Gantt Chart
Fig 5.1	Name Entity Recognition
Fig 5.2	Policy Retrieval Module Flowchart
Fig 5.3	Employee Data Analysis Module Flowchart
Fig 5.4	Authentication & Access Control Flowchart
Fig 5.5	System Flowchart
Fig 5.6	Snapshot of Database
Fig 7.1	Screenshot of UI - 1
Fig 7.2	Screenshot of UI - 2
Fig 7.3	Screenshot of UI - 3
Fig 7.4	Screenshot of UI - 4
Fig 7.5	Screenshot of UI - 5
Fig 7.6	Screenshot of UI - 6
Fig 7.7	Latency Metrics Comparison
Fig 7.8	Accuracy & Quality Metrics Comparison
Fig 7.9	Latency Metrics Comparison
Fig 7.10	Query Category Distribution
Fig 7.11	Confidence Score

List of Tables

Table no	Heading
Table 2.1	Comparison with Existing Systems
Table 3.1	Hardware and Software requirements
Table 6.1	Test Case Scenarios Descriptive
Table 6.2	Test Case Scenarios with supporting images
Table 7.1	Performance Evaluation Metrics
Table 7.2	Performance Metrics Results
Table 7.3	Comparison of Results between Existing Systems

Abstract

Public sector employees frequently encounter delays, confusion, and inefficiencies when accessing essential internal information such as HR policies, IT support protocols, or company announcements. This can lead to wasted time, reduced productivity, and increased dependency on human support teams. To address these challenges, we have developed SmartGuide, an AI-powered, domain-specific chatbot designed to function as a virtual assistant within public sector organizations. The objective of SmartGuide is to provide employees with instant, accurate, and contextually relevant answers to their queries by intelligently processing and retrieving information from institutional documents and internal knowledge bases.

SmartGuide leverages state-of-the-art Natural Language Processing (NLP) techniques, built upon the LLaMA model architecture, and further refined using Hugging Face's Transformer-based models. These models enable the system to accurately interpret user queries posed in natural language and retrieve appropriate responses even from large, complex policy documents. To handle document parsing, we integrated PyMuPDF , which allows efficient extraction of textual data from multi-page PDF files. The chatbot's front-end interface is developed using Streamlit, providing a responsive and lightweight web-based interface that can be accessed across various devices with minimal setup.

Beyond basic information retrieval, SmartGuide incorporates features critical for enterprise environments, including inappropriate language filtering, multi-user session support, and secure two-factor login mechanisms. Through these features, the system ensures not only usability but also robustness and organizational compliance. By automating the process of internal query resolution, SmartGuide significantly reduces support overhead and promotes self-service among employees. The project contributes to ongoing research in intelligent conversational systems and showcases the potential of AI-driven automation in public service workflows.

Chapter 1: Introduction

1.1 Introduction

In today's fast-paced and digitally driven work environments, particularly within large public sector organizations, employees require immediate and reliable access to a broad range of information, including HR policies, IT support procedures, company events, and other internal resources. Traditional methods—such as help desks, email-based queries, or manual document retrieval—often prove to be inefficient, time-consuming, and prone to miscommunication. These limitations can hinder productivity, delay task completion, and create bottlenecks in internal communication.

To address these challenges, we developed SmartGuide, an AI-powered chatbot solution designed to serve as an intelligent virtual assistant within the workplace. SmartGuide leverages advanced deep learning and Natural Language Processing (NLP) techniques to understand employee queries and deliver fast, context-aware, and accurate responses. By automating information retrieval and support services, the system enhances organizational efficiency and reduces the reliance on manual intervention.

A key feature of SmartGuide is its ability to process and summarize uploaded documents, allowing employees to extract essential information from large files with ease. This functionality significantly improves user experience by saving time and reducing cognitive load. Furthermore, the platform is engineered with scalability and security as core principles—it supports concurrent access by multiple users while maintaining response times under five seconds.

1.2 Motivation

Despite the digital transformation across industries, internal information access in public sector organizations often remains manual, fragmented, and dependent on human intervention. This not only delays task execution but also creates frustration among employees. As organizations grow, maintaining real-time, scalable support without overwhelming HR or IT staff becomes increasingly challenging.

The motivation behind SmartGuide stems from the need to automate internal query resolution using modern AI techniques. Chatbots powered by NLP can interpret natural language input and extract relevant answers from a large corpora of documents, thus reducing the time and effort involved in finding information [2]. With public sector workflows becoming more complex, the ability to handle diverse and high-volume queries efficiently is crucial. SmartGuide addresses this gap by combining document processing, conversational AI, and a user-friendly interface.

1.3 Problem Definition

The core problem addressed by SmartGuide is the inefficiency in accessing organizational information in public sector environments. Employees often rely on static documents or human assistance for resolving queries, leading to delays, inconsistent answers, and dependency on specific personnel. Additionally, existing systems lack contextual understanding and scalability, failing to meet the dynamic needs of modern organizations.

SmartGuide is developed to solve the following key issues:

- Long response times for internal queries.
- Inability of static documentation to answer contextual questions.
- Overload on human support teams.
- Lack of a centralized, interactive, and intelligent support system.

By automating these processes through NLP and document parsing, SmartGuide minimizes human intervention while maximizing response accuracy and user experience.

1.4 Existing Systems

Several chatbot frameworks and customer support solutions exist, including:

- Dialog Flow by Google: Offers conversational interfaces but often requires significant customization and cloud dependence.
- IBM Watson Assistant: Highly powerful but complex to integrate with specific document sets.
- Microsoft Bot Framework: Allows enterprise integration but may be costly and over-engineered for internal support needs.

1.5 Lacuna of the Existing Systems

Despite their technical capabilities, the current solutions fall short in several key areas, particularly when it comes to addressing the needs of internal enterprise environments such as public sector organizations. These limitations, or lacunae, are as follows:

- **Lack of Contextual Document Awareness:** Most existing chatbots cannot parse and understand the full context of uploaded documents. They require manual feeding of FAQ-style content, making them unsuitable for environments where policy changes are frequent and documents are complex [1].
- **Inflexibility and High Customization Effort:** Off-the-shelf chatbot systems typically require significant programming, training data, and integration work to adapt to domain-specific use cases such as HR, finance, or administrative operations in government setups. This customization is not only time-consuming but also requires technical expertise that may not be available in every institution.

- **Cloud Reliance and Data Privacy Risks:** Given that many chatbot services are hosted on external cloud platforms, they pose challenges in terms of compliance with data protection regulations, especially in government and public sector applications.
- **Limited Multi-User and Session Handling:** A common shortcoming is the inability to effectively manage multiple user sessions with unique context per user, a crucial requirement in large organizations with hundreds of concurrent users.
- **Absence of Real-Time Document Upload and Summarization Features:** Employees often need answers from newly issued documents or circulars. Most existing bots don't allow users to upload PDFs and retrieve summarized responses in real-time, reducing their practical usability.

1.6 Relevance of the Project

In the era of e-governance and digital transformation, the relevance of AI-driven automation tools like SmartGuide has never been greater. Public sector organizations are under increasing pressure to modernize their internal processes, reduce operational inefficiencies, and improve employee engagement—all while maintaining high standards of security and data privacy.

SmartGuide directly addresses these goals by offering a chatbot that is capable of:

- **Instant and Contextual Information Access:** By leveraging deep learning and NLP, SmartGuide interprets user queries in natural language and provides responses grounded in actual organizational documents, ensuring accuracy and reliability.
- **Real-Time Document Parsing:** Unlike static FAQ systems, SmartGuide allows document uploads and instantly extracts relevant answers or summaries, making it ideal for frequently changing policy environments.
- **Enterprise-Ready Design:** With support for multi-user sessions, secure login, and inappropriate language filtering, the system is suitable for deployment in professional and regulated environments.
- **Ease of Use and Accessibility:** Built with a Streamlit-based front end, the system is lightweight and accessible through any web browser without the need for extensive setup or training.

In this way, SmartGuide not only improves internal communication and support but also aligns with national objectives of transparency, efficiency, and AI adoption in governance.

Chapter 2: Literature Survey

A. Overview of Literature Survey

The integration of AI in public sector workflows has gained significant momentum over the past decade. Several research efforts have focused on using Natural Language Processing (NLP), chatbot interfaces, and document summarization techniques to enhance information accessibility and user experience. This literature survey reviews recent advancements in AI-powered chatbots, document parsing, and enterprise applications, with special emphasis on the relevance of such systems to public organizations.

2.2 Research Papers:

1.T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," in Proc. EMNLP: System Demonstrations, 2020, pp. 38–45.

a) Abstract:

This foundational work presents the Hugging Face Transformers library, which provides simple access to cutting-edge transformer models such as BERT, GPT-2, RoBERTa, and DistilBERT. The paper describes the architecture, modularity, and versatility of the library, making it ideal for fine-tuning on a wide variety of tasks including sentiment analysis, named entity recognition, and question answering. It also emphasizes the importance of pre-trained models in reducing the cost and time of training from scratch. The open-source nature and ease of integration across platforms have made it a standard toolkit in both academic and enterprise NLP applications.

b) Inference:

LLaMA serves as a powerful base for building intelligent agents like SmartGuide. Its smaller size and open-source nature make it suitable for on-premise deployment in privacy-conscious environments such as government organizations. SmartGuide's NLP engine is built using Hugging Face Transformers, drawing directly from the architecture and usability described in this paper. The transformer-based approach allows SmartGuide to effectively process natural language queries, infer context, and match them with the most relevant parts of the internal document database. This paper validates the decision to use Hugging Face tools in SmartGuide's implementation, given their proven performance and scalability in production environments.

2.Raj Bavaresco, João Paulo Aires, João Paulo F. M. de Oliveira, Felipe Rech Meneguzzi, João Alexandre Liebana, and João Batista Camargo Junior, "Conversational Agents in Business: A Systematic Literature Review and Future Research Directions," *Computer Science Review*, vol. 36, 100239, Nov. 2020. doi: 10.1016/j.cosrev.2020.100239

a) Abstract:

This systematic review categorizes and evaluates the usage of conversational agents (CAs) across various business applications. It identifies key use cases including task automation, customer support, decision-making assistance, and internal employee guidance. The paper reviews more than 100 publications to identify technical frameworks, success factors, and barriers to adoption. It also proposes future directions for intelligent agents, particularly emphasizing the role of domain-specific training, emotional intelligence, and ethical design in enhancing user trust and engagement.

b) Inference:

SmartGuide functions as a domain-specific conversational agent tailored for public sector organizations. The insights from this review provide a broader validation for using chatbots internally, not just for external customer support. SmartGuide addresses several key concerns discussed in the paper, such as context-aware response generation and scalability across departments. Furthermore, the emphasis on ethical design is directly reflected in SmartGuide's implementation of inappropriate language filtering and secure login mechanisms. The paper also supports SmartGuide's focus on document-based Q&A, a growing area of research for enterprise-focused CAs

3.C. Saran and M. Kumari, "AI-Powered Chatbots in Public Administration: An Emerging Tool for Digital Governance," *Journal of E-Governance*, vol. 44, no. 3, pp. 112–122, 2021.

a)Abstract:

This paper explores the application of AI-based chatbots in public administration, focusing on how these systems can enhance service delivery, improve internal communication, and reduce operational costs. It presents several case studies from Indian and international government initiatives where chatbots have been implemented for citizen engagement, information dissemination, and task automation. The paper also evaluates the technical infrastructure, implementation challenges, language support, and overall effectiveness of these solutions in diverse governance contexts.

b)Inference:

SmartGuide aligns closely with the vision presented in this study by offering a chatbot tailored for public sector employee support rather than public-facing services. The paper underlines the importance of designing tools that understand domain-specific queries, which SmartGuide achieves by training on organizational documents such as HR policies and IT manuals. In addition, the challenges discussed, such as resistance to adoption and lack of regional language support, are areas that SmartGuide aims to tackle in future iterations.

4.Shubham Agarwal, Shalini Khanna, and Dinesh D, "Task-Oriented Dialogue Systems for Enterprise Support: A Review," *Information Processing & Management*, vol. 58, no. 6, 102646, Nov. 2021. doi: 10.1016/j.ipm.2021.102646

a)Abstract:

The authors present an in-depth review of task-oriented dialogue systems that are widely used for enterprise customer support, internal operations, and digital assistants. The review covers architectures like sequence-to-sequence models, memory networks, and retrieval-augmented generation. It also highlights techniques for intent detection, slot-filling, and context tracking. Challenges such as generalization to unseen intents, multi-turn dialogues, and latency are critically analyzed, along with suggestions for improving robustness and domain adaptability.

b)Inference:

SmartGuide is modeled as a task-oriented chatbot for internal support, directly aligning with the systems discussed in this paper. The identified challenges of latency and multi-turn conversation design are addressed in SmartGuide via fast inference (under five seconds) and streamlined single-turn question handling. The review supports SmartGuide's architecture that combines retrieval-based responses with pre-processed organizational knowledge.

5.Diego Calvaresi, Samuel Eggenschwiler, Yazan Mualla, et al., “Exploring Agent-Based Chatbots: A Systematic Literature Review,” Journal of Ambient Intelligence and Humanized Computing, 2023. doi: 10.1007/s12652-023-04626-5

a)Abstract:

This systematic literature review examines the integration of multi-agent system (MAS) models and technologies in chatbot development across various domains, including education, healthcare, finance, and tourism. The study identifies key challenges in chatbot implementation, such as personalization, knowledge-sharing limitations, support for multi-domain interactions, real-time monitoring, and the integration of chatbot communities. By analyzing existing literature, the authors highlight how MAS approaches can address these challenges, offering insights into application domains, end-user requirements, objectives, technological readiness levels, design methodologies, strengths, limitations, and future research directions.

b)Inference:

The findings from this review are pertinent to SmartGuide's development, particularly in addressing challenges related to personalization and knowledge sharing. By leveraging MAS models, SmartGuide can enhance its ability to provide tailored responses to user queries, thereby improving user satisfaction and engagement. Additionally, the integration of MAS can facilitate better knowledge management and sharing within the system, ensuring that users receive accurate and contextually relevant information.

6.Young Min Cho, Sunny Rai, Lyle Ungar, João Sedoc, and Sharath Chandra Guntuku, “An Integrative Survey on Mental Health Conversational Agents to Bridge Computer Science and Medical Perspectives,” arXiv preprint, arXiv:2310.17017, Oct. 2023. [Online]. Available: <https://arxiv.org/abs/2310.17017>

a)Abstract:

This integrative survey reviews 136 key papers on mental health conversational agents (CAs), aiming to bridge the gap between computer science and medical research perspectives. The study reveals that while computer science research focuses on large language model techniques and automated response quality evaluations, medical research emphasizes rule-based CAs and outcome metrics related to health improvements. The authors highlight the need for cross-disciplinary collaboration to enhance the development and evaluation of mental health CAs, addressing issues such as transparency, ethics, and cultural heterogeneity.

b)Inference:

The emphasis on cross-disciplinary collaboration and ethical considerations in this survey is highly relevant to SmartGuide's development. Ensuring transparency and cultural sensitivity in SmartGuide's responses can enhance user trust and engagement. Moreover, adopting a collaborative approach that integrates technical advancements with an understanding of user needs and ethical implications can lead to a more effective and responsible AI-driven support system for public sector employees.

7.Quim Motger, Xavier Franch, and Jordi Marco, “Software-Based Dialogue Systems: Survey, Taxonomy and Challenges,” arXiv preprint, arXiv:2106.10901, Jun. 2021. [Online]. Available: <https://arxiv.org/abs/2106.10901>

a)Abstract:

This survey provides a comprehensive overview of software-based dialogue systems, commonly known as conversational agents or chatbots. Through a systematic literature review of secondary studies, the authors develop a holistic taxonomy encompassing various dimensions of chatbot research, including design paradigms, interaction modalities, application domains, and evaluation metrics. The study identifies current challenges in the field, such as achieving natural language understanding, maintaining context awareness, and ensuring user satisfaction.

b)Inference:

The taxonomy and challenges outlined in the survey by Quim Motger et al. offer a comprehensive framework for understanding the key components and obstacles in building effective software-based dialogue systems. These insights are particularly valuable for the ongoing development of SmartGuide, as they emphasize critical areas such as natural language understanding (NLU), dialogue management, user modeling, and context retention—each of which is essential for ensuring that conversational agents can deliver accurate, personalized, and contextually appropriate responses.

In the context of SmartGuide, which is designed to assist public sector employees with queries related to HR, IT, and general workplace policies, overcoming challenges in NLU and context

awareness is especially important. Employees may ask questions in diverse formats, across multiple sessions, or with implicit references to past interactions. To effectively manage such complexity, SmartGuide must integrate mechanisms for intent detection, entity recognition, and multi-turn dialogue tracking. These capabilities will allow the system to provide coherent and reliable answers, even when user input is vague or incomplete.

Moreover, the survey’s discussion on evaluation metrics—such as response relevance, task completion rate, and user satisfaction—serves as a useful benchmark framework for continuously measuring SmartGuide’s performance. By incorporating these evaluation strategies into development and testing cycles, the system can be refined iteratively to ensure higher accuracy, robustness, and user trust.

8.Xin Tian, Yingzhan Lin, Mengfei Song, et al., “Q-TOD: A Query-driven Task-oriented Dialogue System,” *arXiv preprint*, arXiv:2210.07564, Oct. 2022. [Online]. Available: <https://arxiv.org/abs/2210.07564>

a)Abstract:

This paper introduces Q-TOD, a novel query-driven task-oriented dialogue system designed to address challenges in domain adaptation and scalability associated with traditional dialogue systems. Q-TOD extracts essential information from the dialogue context into a natural language query, which is then used to retrieve relevant knowledge records for response generation. This approach allows the system to operate independently of specific knowledge base schemas, facilitating easier adaptation to new domains. By decoupling knowledge retrieval from response generation, Q-TOD effectively manages large-scale knowledge bases. Comprehensive experiments demonstrate that Q-TOD outperforms existing baselines, establishing new state-of-the-art performance on multiple task-oriented dialogue datasets.

b)Inference:

The query-driven architecture introduced in the Q-TOD (Query-driven Task-Oriented Dialogue) system presents a highly scalable and domain-agnostic solution for building intelligent conversational agents. Unlike traditional task-oriented dialogue systems that rely on predefined slot-filling schemas and rigid dialog flows, Q-TOD interprets user queries dynamically and uses them to generate meaningful actions and responses.

A similar query-driven mechanism could significantly enhance its effectiveness. By moving away from static intent-response mapping and instead adopting a more dynamic query-processing pipeline, SmartGuide can better understand the nuances of user input, deliver more accurate responses, and scale more efficiently across departments with different informational needs. Additionally, this approach aligns well with SmartGuide’s existing use of advanced NLP models like LLaMA and Hugging Face Transformers, further improving the chatbot’s contextual understanding

and retrieval accuracy.

Incorporating this methodology would also support the system's evolution over time, as new document types and organizational policies are introduced, without requiring heavy reprogramming—making SmartGuide a truly adaptive enterprise-level assistant.

2.3 Research Search:

1. European Patent

a. Title: System and Method for Dialogue-Based Information Retrieval Using NLP(EP3782231A1)

Inventors:Chih-Hong Cheng, et al.

This invention focuses on building an interactive dialogue-based system for efficient information retrieval, especially within complex enterprise environments. The system uses advanced NLP techniques to convert user questions into structured machine-readable queries. These are then matched against indexed enterprise documents, databases, or portals to retrieve the most contextually appropriate answers.

It supports multi-language processing, making it applicable in global corporations, and includes fallback mechanisms that intelligently handle vague or unrecognized inputs. To ensure accuracy and reliability, the architecture integrates a user intent classifier, which identifies the goal of a user's input, and a response validator to confirm the correctness and completeness of the system's output. This patent's methodology is highly aligned with SmartGuide's goals of providing multilingual, accurate, and role-based enterprise support.

b. Title: AI-Powered Multi-User Chat Interface for Organizational Knowledge Systems(EP3824945A1)

Inventors:Ernst R. Mueller, et al.

This patent proposes an AI-driven, multi-user chatbot framework designed for large-scale enterprise applications. It seamlessly integrates with an organization's document repositories, databases, and internal tools such as Human Resource Management (HRM), IT Service Management (ITSM), and Customer Relationship Management (CRM) systems. Using semantic similarity detection, the system maps user queries to the most relevant internal content.

One of the highlights is its user-role filtering system, which ensures that responses are appropriately filtered and customized based on an employee's access rights. The chatbot is GDPR-compliant, ensuring that personal and sensitive data is securely handled. It also includes learning modules that monitor and learn from failed or ambiguous queries, thus evolving and improving its answer

precision over time. The modularity and integration capabilities make it an ideal architecture for systems like SmartGuide aimed at delivering personalized, real-time enterprise support.

2. US Patent

a. Title: AI-Based Virtual Assistant for Enterprise Application (US10977083B2)

Inventors:Ranjith Ravindran, et al.

This patent introduces a comprehensive AI-powered virtual assistant designed for enterprise environments. The system is capable of understanding complex natural language queries from users through robust Natural Language Processing (NLP) pipelines. Upon interpreting the user's request, it either retrieves data from the knowledge base or performs specific actions such as setting appointments, retrieving documents, or triggering service tickets (e.g., HR, IT).

A key feature of this assistant is contextual continuity—it remembers previous user interactions across sessions to enable multi-turn dialogues, which enhances user experience and efficiency. Furthermore, role-based personalization ensures that responses and actions are tailored based on the employee's role, privileges, and historical interactions. The system includes strong security and compliance frameworks, making it suitable for deployment in highly regulated industries like healthcare, finance, or government services.

b. Conversational AI System with Document-Aware Interaction (US20200187903A1)

Inventors: Ashwin Ram, et al.

This patent outlines a conversational AI system capable of intelligently extracting answers from unstructured documents, including policy manuals, guides, and internal reports. Unlike traditional bots that rely solely on structured datasets, this system leverages deep learning models such as neural embeddings to semantically match a user's query with the most relevant passage in a document.

It incorporates mechanisms to handle ambiguous terms, anaphora resolution (e.g., resolving references like “it” or “this”), and multi-turn dialogues, maintaining context throughout a conversation. An additional innovation is its feedback loop—the system can learn from incorrect or low-confidence responses through user feedback and fine-tune its performance. This

document-awareness combined with adaptability makes it highly valuable for real-time, enterprise-grade support systems like SmartGuide.

2.4 Inference Drawn:

Upon a thorough analysis of the selected U.S. and European patents, several common themes and technological principles emerge that directly align with and validate the design and functional goals of the SmartGuide system. The following key inferences have been drawn:

- **Integration with Enterprise Systems:**

All patents emphasize the importance of integrating the chatbot or virtual assistant with internal enterprise data sources, including HR systems, CRM platforms, and document repositories. This strongly supports SmartGuide's aim to serve as a unified access point for information retrieval across various public sector departments.

- **Advanced Natural Language Processing (NLP):**

A recurring theme across all inventions is the use of NLP techniques to convert user queries into structured formats for accurate information extraction. This reaffirms SmartGuide's reliance on NLP and deep learning for understanding complex user input and delivering context-aware responses.

- **Security, Privacy, and Compliance:**

Enterprise-grade assistants, as described in the patents, incorporate GDPR-compliant access control, user authentication, and role-based filtering. These are critical features mirrored in SmartGuide, especially given its deployment in sensitive governmental and public service contexts.

- **User-Centric and Context-Aware Interaction:**

The patents highlight conversational memory, document-aware interactions, and fallback mechanisms to improve query resolution and user satisfaction. SmartGuide similarly aims to deliver intelligent, adaptive, and uninterrupted user experiences across different query types and departments.

- **Scalability and Learning Mechanisms:**

Features such as failed-query tracking and user feedback loops are designed to make the systems more intelligent over time. This justifies SmartGuide's long-term roadmap of incorporating reinforcement learning to enhance accuracy and adaptability through continued usage.

2.5 Comparison with Existing Systems:

Feature/Parameter	Traditional Method	General-Purpose Chatbots	Proposed System
Query Resolution Time	High (delays due to human effort)	Medium (context often missed)	Low (<5 seconds) due to real-time NLP processing
Availability	Limited to working hours	24/7 (scripted/open-ended)	24/7 Intelligent Assistant
Context Awareness	Human understanding	Limited	Advanced NLP with query understanding
Document Processing Capability	Manual interpretation	Rarely Supported	supported Built-in document summarization and Q&A
Multi-User Handling	Limited by staff availability	Moderate	Scalable to handle multiple users concurrently
Learning from User Interaction	No	Minimal	Adaptive learning using failed-query tracking
Support for Multilingual Communication	Human-dependent	Partial	Multilingual NLP support
Cost Efficiency	High (manpower-dependent)	Moderate	High ROI with minimal overhead

Table 2.1: Comparison with Existing Systems

Chapter 3: Requirement Gathering for the Proposed System

3.1 Introduction to Requirement Gathering

Requirement gathering is the foundational step in the development lifecycle of any software project. It involves the systematic process of identifying, documenting, and analyzing the needs and expectations of stakeholders to ensure that the final product aligns with user requirements and business goals. The objective is to gather clear, concise, and comprehensive information that serves as the blueprint for system design and implementation.

For the **SmartGuide HR Chatbot System**, requirement gathering played a crucial role in shaping the functionalities and behavior of the system. The aim was to develop a chatbot capable of understanding and responding to employee queries related to human resources—such as leave policies, salary details, company rules, and onboarding procedures—through natural language interaction.

The process of requirement elicitation involved:

- **Questionnaires and Surveys** targeted at employees to gather feedback on their expectations from an HR support chatbot.
- **Academic Guidance** to ensure the solution aligns with software engineering principles and project goals.
- **Competitor Analysis** to study existing HR chatbot solutions and identify gaps or opportunities for innovation.

3.2 Functional Requirements

Functional requirements define the specific behaviors, features, and functionalities that the system must support. These are derived from the expectations of users and stakeholders and serve as the core capabilities the software must deliver to be considered successful.

For the **SmartGuide HR Chatbot System**, the functional requirements are outlined as follows:

1. **User Registration and Login:** The system shall allow users to register using their username and employee id.
2. **Natural Language Query Handling:** The chatbot shall accept user queries in natural language (e.g., "How many leaves do I have left?").
3. **HR Domain Knowledge Base Integration:** The system shall retrieve accurate responses from a structured HR knowledge base.

4. **Contextual Conversation Flow:** The chatbot shall maintain conversational context to handle multi-turn interactions smoothly.
5. **Search Functionality:** The system shall include keyword-based search support for documents or FAQs if the query cannot be interpreted via NLP.
6. **Session Management:** Each user session shall be tracked independently, and previous session data may be referenced where needed.

3.3 Non-Functional Requirements

Non-functional requirements define the overall qualities and attributes that the system must exhibit. While functional requirements describe what the system does, non-functional requirements describe how well it performs its tasks. These requirements ensure the usability, reliability, and efficiency of the SmartGuide HR Chatbot System.

The non-functional requirements for the proposed system are as follows:

1. **Usability:** The chatbot interface shall be intuitive and user-friendly to ensure smooth interactions.
2. **Reliability:** The chatbot shall respond accurately and consistently under various conditions.
3. **Performance:** The system shall process and respond to user queries within 2–3 seconds under normal load.
4. **Scalability:** The system architecture shall support horizontal scaling to accommodate increased user load.
5. **Security:** User credentials and personal data shall be encrypted and securely stored.
6. **Maintainability:** The system codebase shall be modular and well-documented to support future maintenance.
7. **Portability:** The chatbot shall be accessible across various platforms (web and mobile browsers).
8. **Availability:** The system shall ensure high availability and provide graceful error messages in case of temporary issues.
9. **Extensibility:** The system shall support easy integration of new HR domains or features such as document uploads or chatbot voice support.

3.4 Hardware & Software Requirements

Hardware Requirements	Software Requirements
Computer System: a. CPU i7 or higher b. RAM 16GB or more	Operating Systems: Windows/Mac/Linux

c. GPU - 16GB or more	
Server: Multi CPU with 8 GB RAM	Code Editor - VS Code
	ML Libraries: Hugging Face, Pytorch/Tensorflow
	Document Processing Tools: Py2PDF

Table 3.1: Hardware and Software requirements

3.5 Constraints

- **Data Privacy Regulations:**The system must comply with relevant data protection laws (e.g., GDPR) concerning employee information.
- **Integration Limitations:**The chatbot may need to integrate with existing HR and IT systems, which could impose technical constraints.
- **NLP Model Limitations:**The effectiveness of the chatbot's responses may be limited by the capabilities of the NLP model and the quality of training data available.
- **Budget and Resource Constraints:**Development and operational costs may
- restrict the extent of features that can be implemented.
- **Document Format Limitations:**The system may only support specific file formats for document uploads (e.g., PDF, DOCX).
- **Multilingual Support Complexity:** Providing effective multilingual support can increase the complexity of the system, requiring additional resources for language models and testing.

Chapter 4: Proposed Design

4.1 Block diagram of the system

The block diagram illustrates the modular architecture of the SmartGuide HR Chatbot System, highlighting each component's role and their interconnections.

1. User Interface (UI):

This is the front-end interface where users interact with the chatbot. It is implemented using Flask and HTML, enabling users to input queries and view responses.

2. User Query Processing:

The user's input is captured and forwarded to the backend. This module handles routing and manages user sessions via Flask's backend routes.

3. LangGraph AI Agent:

This core decision-making component processes user queries and determines the appropriate tool to invoke. It uses Google Gemini to understand the query and LangGraph for tool orchestration.

4. Tool Modules:

These include:

1. HR Policies (Pinecone Vector DB): Stores and retrieves policy-related documents using vector search.
2. Employee Data (Pandas DataFrame): Handles structured queries to access employee records from CSV files.

5. Data Retrieval and Processing Tools:

1. Vector Search & RetrievalQA: Used for fetching relevant HR policies based on user queries.
2. Pandas Data Lookup (PythonAstREPLTool): Parses and executes code to query structured employee data.
3. Math Operations: Handles basic arithmetic or logical computations using an LLM-based chain.

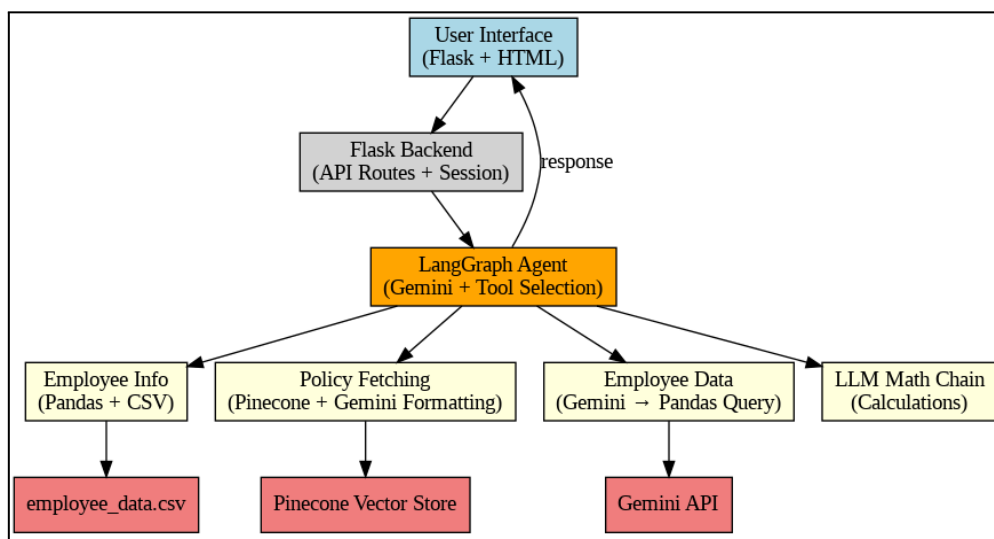


Fig 4.1 Block Diagram of the System

4.2 Modular Design of the System

The modular design of the SmartGuide HR Chatbot System divides the application into distinct functional components, each responsible for a specific task.

1. User Interface Module:

The front-end interface allows users to interact with the chatbot through a web-based platform. It is built using Flask and HTML, providing input fields and displaying responses.

2. Backend Processing Module:

The Flask-based backend manages user sessions, routes queries from the interface to the AI agent, and ensures smooth communication between frontend and logic layers.

3. LangGraph AI Agent Module:

This module serves as the brain of the system. It uses the Gemini large language model to understand user intent and dynamically select the most suitable tool (HR Policies, Employee Data, or Calculator) to handle the query.

4. Tool Integration Modules:

- **HR Policy Tool:** Uses Pinecone for vector-based document retrieval, allowing the system to fetch relevant HR policies from stored embeddings.
- **Employee Data Tool:** Queries structured CSV data using pandas to extract employee-specific.
- **Mathematical Reasoning Tool:** Performs computations using an LLM Math Chain for queries requiring numeric or logical reasoning.

5. Data Storage Modules:

- **CSV File Storage:** Contains employee data that can be queried via pandas.
- **Pinecone Vector Store:** Maintains vector embeddings of HR policy documents for semantic.
- **Gemini API:** Supports natural language understanding and reasoning through advanced LLM.

6. Response Generation Module:

Once the appropriate tool completes its task, the AI agent synthesizes the final response and routes it back through the backend to the user interface.

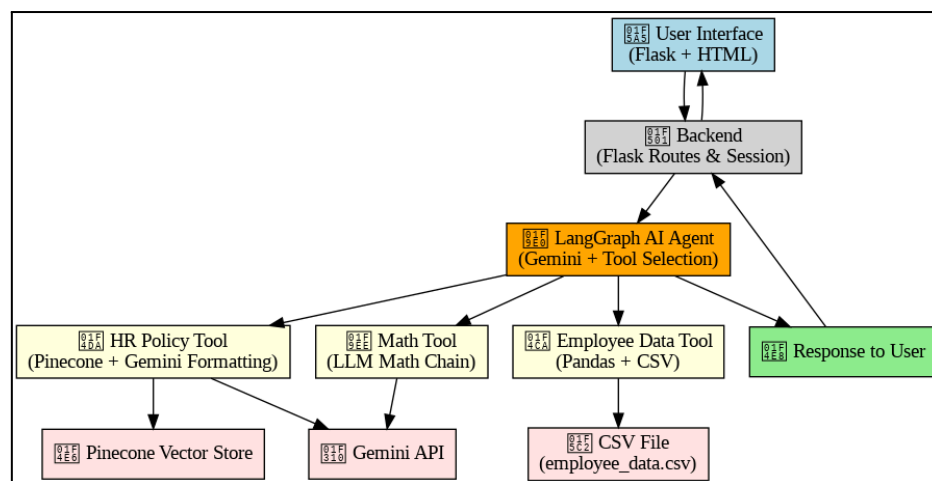


Fig 4.2: Modular Design of the System

4.3 Detailed Design

1. Start

The system begins when a user initiates a query via the web interface. This could be a question about HR policies, employee data, or a calculation request.

2. User Interface (Flask Frontend)

The user's query is captured through a web-based interface developed using Flask and HTML. This UI is responsible for sending the input to the backend for processing.

3. Flask Backend

The backend receives the input and forwards it to the LangGraph Agent. It handles routing, sessions, and communication between the front end and core logic.

4. LangGraph Agent (Core Decision Logic)

The LangGraph Agent acts as the brain of the system. It takes the input, interprets the intent behind it, and decides which internal tool is best suited to handle the request.

5. Intent Classification

The agent performs intent detection to classify the query into one of the following categories:

- HR Policy-related
- Employee Data-related
- Mathematical/Logical calculation

6. Tool Invocation

- **HR Policy Tool:** If the query is about HR policies (e.g., leave policy, timekeeping), this tool is invoked. It uses the Pinecone vector database to fetch relevant documents and formats them via the Gemini API.
- **Employee Data Tool:** For queries involving employee records or personal details, the system queries a local CSV file using Pandas.
- **Math Tool:** For calculations (e.g., leave balance, salary deductions), the system uses an LLM Math Chain supported by the Gemini API.

7. External Services

- **Pinecone Vector Store:** Stores vectorized policy documents and supports similarity search.
- **Gemini API:** Powers the LLM-based reasoning, summarization, and math solving.
- **employee_data.csv:** Contains structured employee records queried using Pandas.

8. Response Generation

After tool execution, the output is compiled into a natural language response. The response is sent back to the frontend via the backend.

9. End

The response is displayed to the user on the web interface, and the interaction is complete.

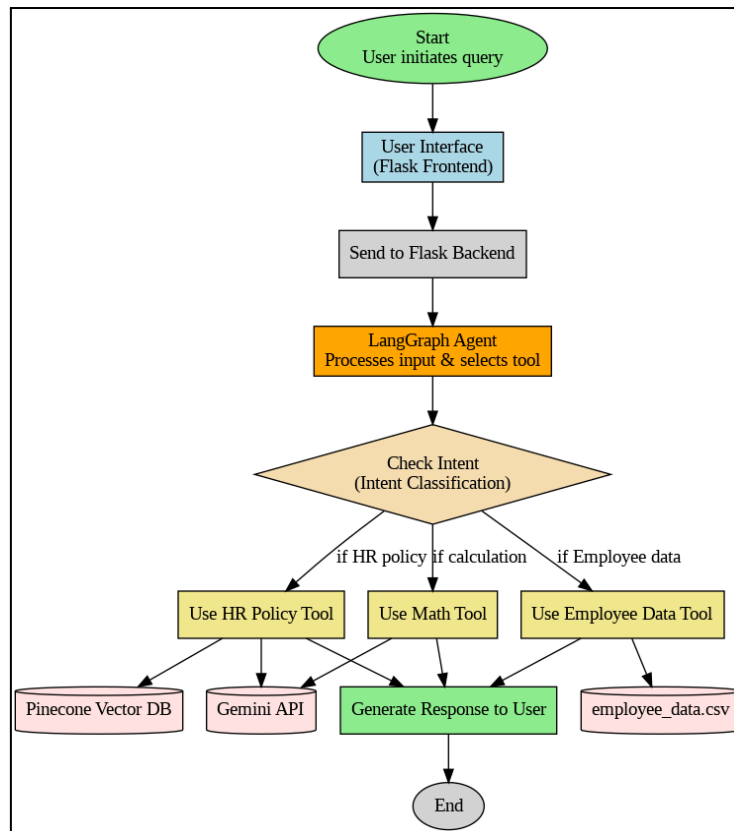


Fig 4.3: Detailed Design (Flowchart of the system)

4.4 Project Scheduling & Tracking using Gantt Chart

The Gantt chart provides a visual timeline of the SmartGuide HR Chatbot project from July 2024 to March 2025. It outlines key phases such as requirement gathering, system and module design, development, testing, documentation, and final submission. Iterative improvements were made after mid-evaluation and during testing to enhance functionality. This chart helped in tracking progress, managing deadlines, and ensuring timely completion of the project.

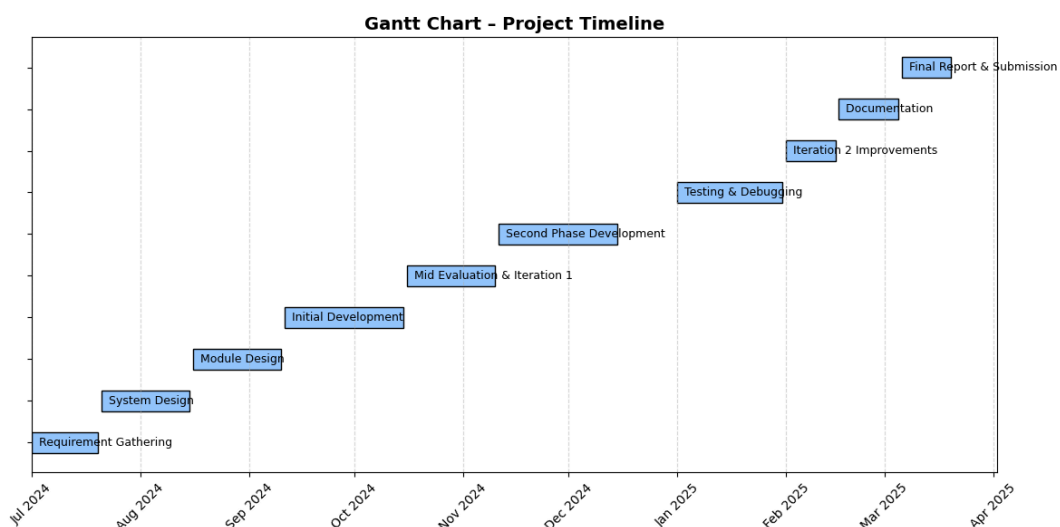


Fig 4.4: Project Scheduling and Tracking using Gantt Chart

Chapter 5: Implementation and Proposed Systems

5.1.Methodology employed for development:

To address the challenges faced by modern HR departments in handling employee support, data analysis, and policy dissemination, our project adopts a methodical approach that combines Generative AI, semantic search, and modular tool integration.

In today's fast-paced enterprise environments, Human Resource (HR) departments play a crucial role in ensuring smooth employee onboarding, policy dissemination, and internal support. However, managing HR-related queries, policy documents, and employee records manually is becoming increasingly inefficient due to growing organizational complexity.

Despite advancements in automation and artificial intelligence, many organizations—especially in the public sector—lack intelligent systems capable of providing instant, multilingual, and personalized HR support. Employees often waste time browsing lengthy documents or waiting on responses from HR personnel for simple queries related to leaves, reimbursements, timekeeping policies, and compliance guidelines.

This situation not only delays operational workflows but also impacts employee satisfaction and overall productivity. There is an urgent need for a solution that can automate repetitive HR tasks, provide instant policy access, and analyze employee data on-demand.

To overcome these challenges, we propose a Generative AI-powered HR Assistant, developed as a Flask-based web application that integrates with Google Generative AI, Pinecone vector databases, and LangChain for intelligent tool chaining. The system enables employees and HR managers to interact with policy and data systems using natural language and receive structured, meaningful responses in real-time.

By modularizing the system into components such as policy retrieval, employee data analysis, authentication, and multilingual personalization, the architecture ensures scalability, maintainability, and user-friendliness. The solution reduces dependency on manual processes while significantly improving the speed and accuracy of HR services.

Modules of the System

- **Name Entity Recognition Module**

The Named Entity Recognition (NER) Module is responsible for extracting key information from user queries in natural language. This module uses NLP (Natural Language Processing) techniques to detect and classify entities such as names, dates, departments, policy types, and job titles from the user's input. Use in our chatbot:

When a user asks, “**Show me the paternity leave policy**” or “**I want to check policies for Finance department**”, the NER model identifies terms like “**paternity leave**” and “**Finance department**”. These extracted entities are then used to perform semantic search or retrieve the most relevant document or data from the database.

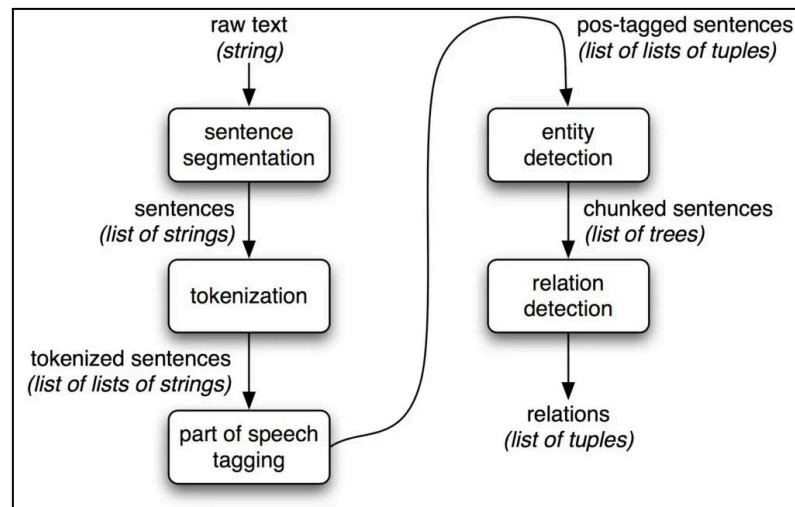


Fig 5.1: Name Entity Recognition

● Policy Retrieval Module

This module enables users to query HR policies using natural language input. It utilizes Pinecone for semantic search and employs Google’s Gemini AI to process and present the relevant policy content in a structured and readable HTML format.

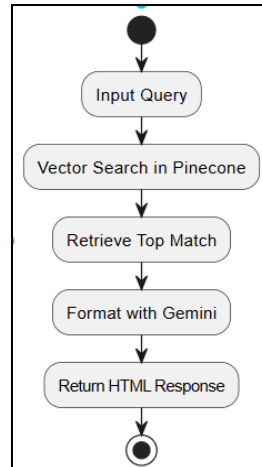


Fig 5.2: Policy Retrieval Module Flowchart

● Employee Data Analysis Module

This component allows users to ask analytical questions related to employee records. Gemini AI interprets these queries and translates them into executable Pandas code. Typical queries include, for instance, "Display all employees in the finance department" or "List employees who joined after the year 2022."

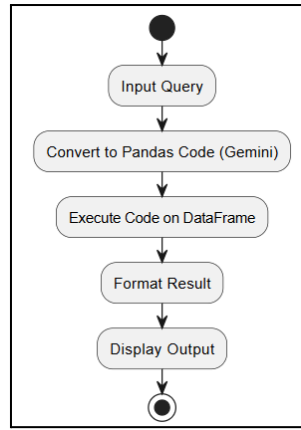


Fig 5.3: Employee Data Analysis Module Flowchart

- **Multilingual Support & Personalization**

The system is equipped with automatic language detection and can respond in the appropriate language accordingly. Additionally, it maintains user context—including name and prior interactions—to provide a personalized and coherent conversational experience.

- **Authentication & Access Control**

To ensure data privacy and security, users are required to authenticate themselves before accessing restricted functionalities. Only authorized personnel such as employees or administrators are granted access to sensitive information.

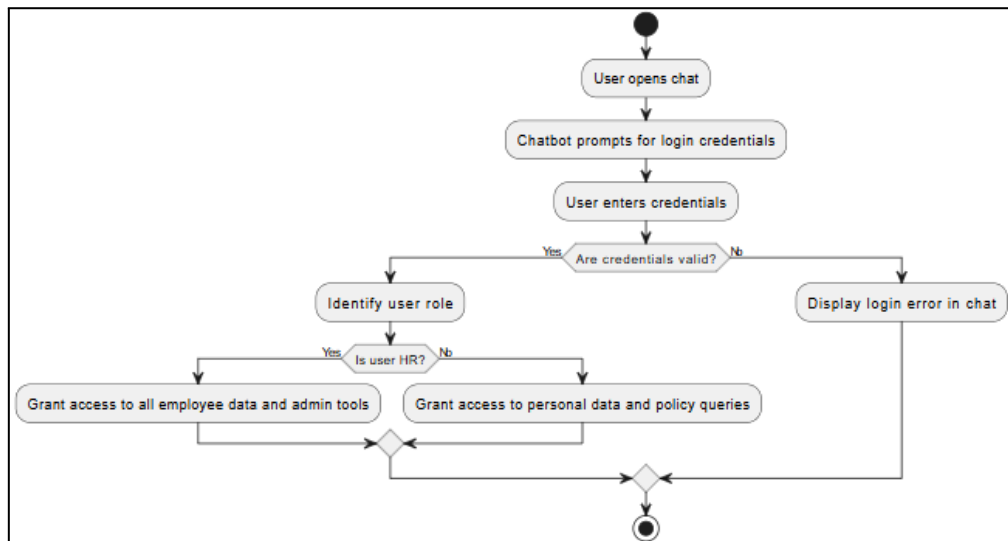


Fig 5.4: Authentication & Access Control Flowchart

- **LangChain Tool-Based Agent**

A LangChain-powered agent coordinates the execution of various tools based on the nature of the user's query. These tools include:

- timekeeping_policy_retrieval
- employee_data_analysis
- employee_info_retrieval

- calculator

Flask Web Interface

The application is built on a lightweight and scalable backend framework using Flask. The frontend is designed with basic HTML templates and JavaScript, providing a simple yet effective interface for user interaction.

User Roles in the System

There are two main user roles:

1. Employee/User

This user category includes individual employees who utilize the system to query HR policies, perform employee data analysis, and retrieve personal information. They interact with the platform using a conversational interface powered by natural language processing.

2. HR/Admin

Administrators are responsible for uploading and managing policy documents, maintaining employee datasets, and overseeing overall system performance. They also carry out document verification and ensure content compliance across the platform.

5.2. Algorithms and flowcharts for the respective modules developed

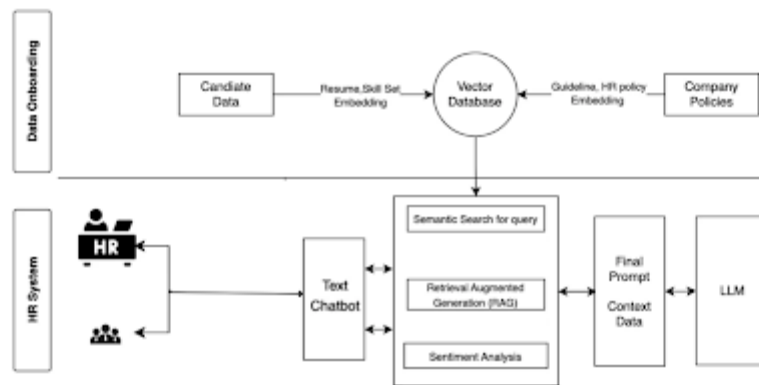


Fig 5.5: System Flowchart

1. Data Ingestion & Embedding

Candidate Data (e.g., resumes, skill sets) and Company Policies (guidelines, HR handbooks) are converted into vector embeddings. All embeddings are stored in a central Vector Database.

2. User Query Enters Chatbot

An HR user (or employee) types a question into the Text Chatbot interface.

3. Semantic Search

The chatbot issues a vector-based similarity search on the database to pull in the most relevant policy/text snippets for that query.

4. Retrieval-Augmented Generation (RAG)

Retrieved snippets are stitched together into a cohesive “context bundle” (the RAG step).

5. Sentiment Analysis

The user’s original message is run through a sentiment filter—ensuring, for example, that the response tone is appropriately empathetic or professional.

6. LLM Response Assembly

The context bundle + sentiment cues become the final prompt that’s sent to the LLM (e.g., Gemini). The LLM’s answer is returned to the Text Chatbot and displayed back to the HR user.

5.3. Datasets Source and Utilisation

The intelligent HR Assistant system utilizes three primary data sources to support its core functionalities—policy retrieval, question answering, user authentication, and employee data analytics:

1. Organizational Policy Documents (HR PDFs):

These are official policy documents uploaded by HR or administrative personnel in PDF format. They include various guidelines related to leaves, reimbursements, timekeeping, employee conduct, and other internal procedures. Once uploaded, these documents are pre-processed and stored in a vector database (Pinecone) to enable semantic search and natural language querying. This dataset powers the Policy Retrieval Module, allowing users to ask questions in plain language and receive precise, summarized answers.

2. User-Uploaded PDF Documents:

In addition to organizational policies, the system also includes a feature that allows employees or HR staff to upload their own PDFs through a dedicated PDF icon in the chatbot interface. This module supports summarization and document-specific question answering. It uses Generative AI (Gemini Pro) to extract relevant context and generate concise responses from the uploaded documents, making it useful for one-time file-based queries such as project reports, offer letters, or external policies.

3. Employee Data (CSV Database):

A structured dataset in CSV format is maintained to manage employee information effectively. This dataset includes key details necessary for user authentication, access control, and personalized query handling. The columns in this CSV file are as follows:

- **employee_id:** Unique identifier for each employee
- **name:** Full name of the employee

- **position:** Designation or job title
- **organizational_unit:** Department or unit the employee belongs to
- **rank:** Employee's hierarchical level
- **hire_date:** Date of initial hiring
- **regularization_date:** Date of confirmation as a regular employee
- **vacation_leave:** Available vacation leave balance
- **sick_leave:** Available sick leave balance
- **basic_pay_in_php:** Basic salary in Philippine Peso
- **employment_status:** Nature of employment (e.g., Permanent, Contractual)
- **supervisor:** Immediate reporting manager or supervisor
- **password:** User-defined password for secure authentication

employee_id	name	position	organizational...	sick_leave	empl...	password
1005674	Joseph Peña	Accounts Recievable Super...	Finance Departm...	15	Perm...	Bg7u
1005675	Mark Delos Santos	HR Assistant	HR Department	2	Proba...	Tp8
1005676	Alexander Verdad	Account Recievable Assistant	Finance Departm...	8	Perm...	Qz5Yn7Lc
1005677	Jinky Francisco	Recruitment Supervisor	HR Department	4	Perm...	Xr9Qv6Wp
1005678	Richard Santos	Accounts Recievable Head	Finance Departm...	11	Perm...	Kf3Jo8Nm

Fig 5.6: Snapshot of Database

Each employee is permitted to access only their personal records, while HR personnel have broader access privileges to analyze and retrieve records across all employees for administrative purposes.

Chapter 6: Testing of Proposed Systems

6.1 Introduction to Testing

Software testing is a structured and systematic process undertaken to evaluate the functionality, performance, and reliability of a software application. It acts as a vital quality gate between development and deployment, ensuring that what reaches the end user is stable, secure, and performs according to expectations. For this reason, testing has become an inseparable component of modern software development practices and is considered as important as the development itself.

In the context of this project, which involves a Generative AI-based HR Assistant System, the importance of a well-defined testing phase is magnified. The system is not just a static software application—it's a dynamic, interactive platform that processes natural language inputs, performs real-time document summarization, provides personalized responses based on user roles, and supports multiple languages. Any failure in such a system, especially in environments like HR management where privacy, accuracy, and access control are critical, can lead to user dissatisfaction, loss of trust, or even data breaches.

The ultimate objective of software testing is not just to detect bugs but to validate the entire user experience. A chatbot that misinterprets a question or provides incorrect policy information could lead to confusion or misinformed decisions. Similarly, incorrect implementation of access controls might expose sensitive employee data to unauthorized users, violating privacy policies. Hence, testing in this project also involves evaluating user authentication mechanisms, data isolation based on user roles, response accuracy, and natural language processing performance.

The Software Testing Life Cycle (STLC) followed in this project encompasses the planning, design, execution, and evaluation of various tests at multiple levels. Through careful test case formulation and scenario analysis, the system has been validated under conditions that simulate both everyday usage and edge cases. This includes handling invalid logins, ambiguous queries, non-standard language inputs, and unexpected file uploads.

The goal of the testing phase in this project is to provide stakeholders (HR teams, developers, and users) with confidence in the system's ability to meet its objectives—namely, to serve as an intelligent, secure, and user-friendly virtual assistant for HR processes. By simulating real-world usage and rigorously evaluating system behavior, the testing phase ensures that the final deployment is both technically sound and practically useful.

In summary, testing plays a central role in transforming the system from a working prototype into a production-ready, enterprise-grade application. It ensures that the chatbot not only understands and responds to queries but does so reliably, securely, and intelligently.

6.2 Types of Tests Considered

The following types of tests were used to verify and validate the system:

1. Unit Testing

Individual modules like user authentication, PDF summarizer, Pinecone document search, and employee info retrieval were tested in isolation to ensure correctness and reliability.

2. Integration Testing

Integration testing was conducted to validate the seamless interaction between the different modules that make up the application. The components tested included:

- LangChain Framework: Used for chaining together prompts and tools for contextual responses.
- Google Generative AI: Responsible for producing intelligent, natural language responses based on user queries.
- Flask Frontend: Web interface where users interact with the chatbot.
- Employee CSV Database: Used for authentication and personalized query resolution.
- Pinecone Vector Store: Used for semantic search across policy documents.

3. Functional Testing

Functional testing focused on validating that the system behaved as per the defined requirements and specifications. Key functionalities tested include:

- Authentication and Role-Based Access:
 - HR users are permitted access to organization-wide employee data.
 - Regular employees are restricted to viewing their personal records only.
 - Invalid login attempts are gracefully handled with error notifications.
- Policy Document Querying:
 - Users can ask questions related to HR policies (e.g., leave policies, work-from-home guidelines), and the chatbot returns relevant responses.
- PDF Summarization and Q&A:
 - Users can upload policy-related PDFs.
 - The system generates concise summaries and supports follow-up questions on uploaded documents.
- Multilingual Support: The system was tested with both English and Spanish queries.

4. System Testing

System testing evaluated the overall behavior and performance of the complete application under

simulated real-world conditions. This included:

- End-to-End Login Flow: Tests were conducted using both valid and invalid employee credentials.
- File Upload and Content Handling: PDF files of varying lengths and formats were uploaded to validate summarization and Q&A functionality.
- Session and Role-Based Navigation: HR-specific views such as dashboards and employee-wide queries were validated.

5. Usability Testing

Usability testing was carried out to ensure the system offers a user-friendly experience for both HR personnel and employees. Key focus areas included:

- User Interface Clarity: Placeholder text and interactive hints were tested for clarity.
- Responsiveness and Feedback: Real-time feedback and prompt generation were evaluated.
- Multilingual Experience: Interface and responses were adapted to user language preferences.

Based on user feedback during testing, minor interface enhancements were implemented, such as refining button labels and improving multilingual support cues.

6. Security Testing

Checked for data exposure risks, password validation mechanisms, and secure session handling. Ensured HRs can't misuse employee data, and employee logins are restricted to personal data access.

6.3 Various Test Case Scenarios Considered

Below is a table detailing the representative test cases considered during the development and testing of the SmartGuide chatbot system:

Test Case ID	Test Scenario	Input	Expected Output	Status
TC001	Login with valid employee credentials	Valid employee_id and password	Redirect to employee dashboard with personal data	Pass
TC002	Login with invalid credentials	credentials Invalid employee_id or password	Display error message: "Invalid credentials"	Pass

TC003	HR login with access to all employee data	HR credentials	HR chatbot with full data access	Not Passed
TC004	Upload and summarize PDF document	Upload sample policy PDF	Summary of document + support for questions about it	Pass
TC005	Ask HR policy question	“What is the sick leave policy?”	Relevant policy summary as answer	Pass
TC006	Ask question in Spanish	“¿Cuál es la política de vacaciones?”	Response in Spanish with correct policy content	Pass
TC007	Attempt access to another employee’s data (non-HR)	Employee logged in attempts to query others	Access denied / restricted message	Pass
TC008	Session timeout	Stay idle for 10 minutes	Automatic logout with session expired message	Not Passed
TC009	Chatbot responds to ambiguous queries	“Leaves?”	Follow-up prompt: “Do you mean sick leave, vacation leave, or both?”	Pass
TC010	Concurrent users test	10 users interact simultaneously	System maintains performance and stability	Pass

Table 6.1: Test Case Scenario

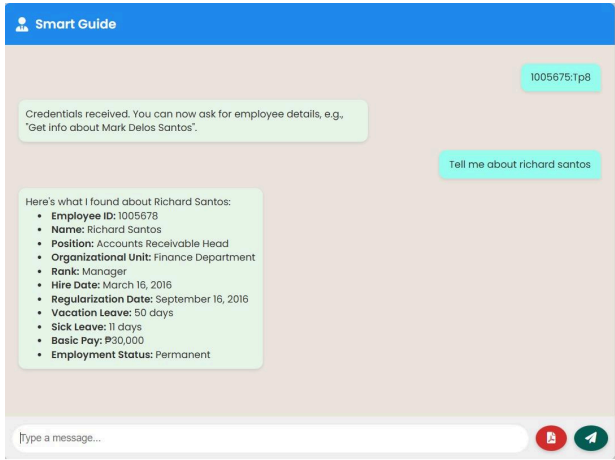
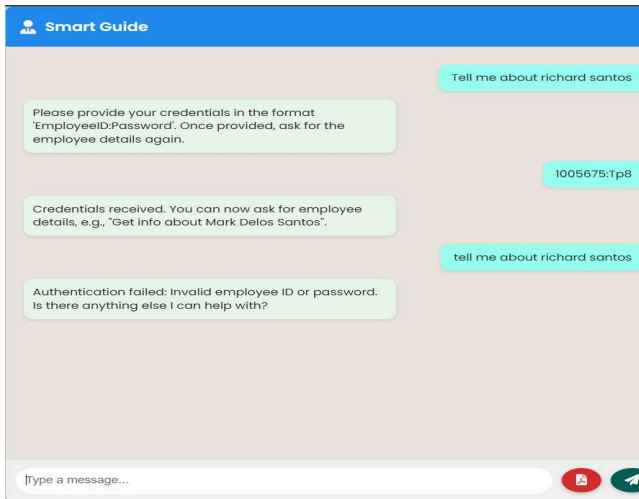
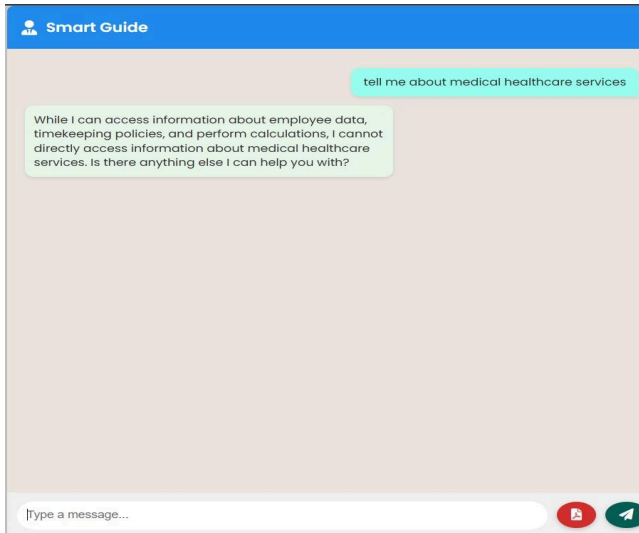
Test Type	Description	Supporting Images
Integration Testing	Validated seamless interaction between LangChain, Flask, Google Generative AI, and the employee database for credential handling and query routing.	 <p>The screenshot shows a chat window titled 'Smart Guide'. A user message '1005675:1p8' is received. The system responds: 'Credentials received. You can now ask for employee details, e.g., "Get info about Mark Delos Santos".' The user then asks 'Tell me about richard santos'. The system provides a detailed response: 'Here's what I found about Richard Santos: • Employee ID: 1005675 • Name: Richard Santos • Position: Accounts Receivable Head • Organizational Unit: Finance Department • Rank: Manager • Hire Date: March 16, 2016 • Regularization Date: September 16, 2016 • Vacation Leave: 50 days • Sick Leave: 11 days • Basic Pay: ₱30,000 • Employment Status: Permanent'.</p>
Functional Testing	Confirmed business logic, e.g., login authentication, employee queries, and invalid credential handling.	 <p>The screenshot shows a chat window titled 'Smart Guide'. A user message 'Tell me about richard santos' is received. The system prompts: 'Please provide your credentials in the format "EmployeeID:Password". Once provided, ask for the employee details again.' The user provides '1005675:1p8'. The system responds: 'Credentials received. You can now ask for employee details, e.g., "Get info about Mark Delos Santos".' The user then asks 'tell me about richard santos'. The system responds: 'Authentication failed: Invalid employee ID or password. Is there anything else I can help with?'.</p>
System Testing	Tested overall system behavior, such as multilingual queries, unsupported topics, and restricted data access.	 <p>The screenshot shows a chat window titled 'Smart Guide'. A user message 'tell me about medical healthcare services' is received. The system responds: 'While I can access information about employee data, timekeeping policies, and perform calculations, I cannot directly access information about medical healthcare services. Is there anything else I can help you with?'.</p>

Table 6.2: Test Case Scenarios with supporting images

6.4 Inference Drawn from the Test Cases

The test results show that the system performed reliably across a wide range of functional, integration, and user-based scenarios. Here are the key observations:

- **System Stability:** The application consistently maintained session integrity and data security under concurrent usage, indicating strong backend design and performance.
- **Functional Accuracy:** Core features such as login validation, policy querying, PDF summarization, and multilingual interaction behaved as expected, aligning with functional requirements.
- **Security Assurance:** Role-based access control successfully prevented unauthorized data retrieval, ensuring data privacy and compliance.
- **User Experience:** Usability testing highlighted minor UI enhancements that improved user satisfaction, such as adding placeholder texts, clearer error prompts, and refining response handling in different languages.
- **Integration Success:** All integrated modules—LangChain, Google GenAI, Pinecone, and the CSV database—worked together cohesively, proving the robustness of the modular design.

These results confirm the system's readiness for deployment in real-world enterprise environments with minimal adjustments.

Chapter 7: Results and Discussion

7.1. Screenshots of User Interface (UI) for the respective module

The system was tested for scenarios involving access control violations, specifically when a non-HR user attempted to retrieve sensitive information about other employees. In this instance, the chatbot correctly identified the unauthorized nature of the request and either denied access outright or responded with a message indicating insufficient permissions. This demonstrates that the SmartGuide HR Chatbot incorporates basic role-based access control mechanisms to prevent data breaches and protect employee confidentiality. Ensuring that only authorized HR personnel can access detailed employee information is crucial for maintaining data privacy, adhering to organizational policies, and complying with regulations.

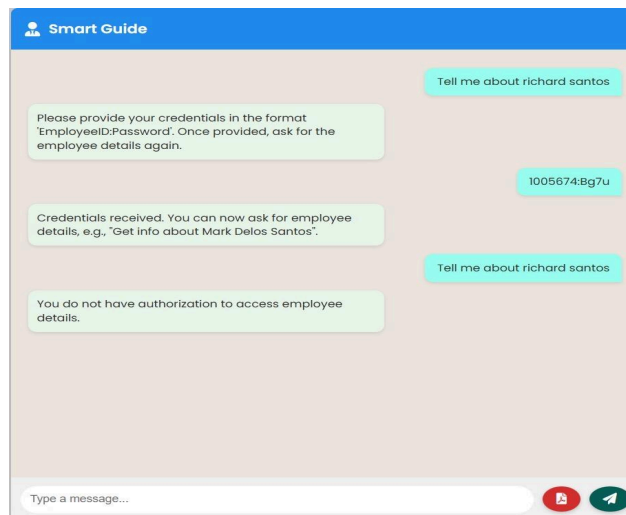


Fig 7.1: An instance in which an unauthorized, non-HR user tries to retrieve information about other employees

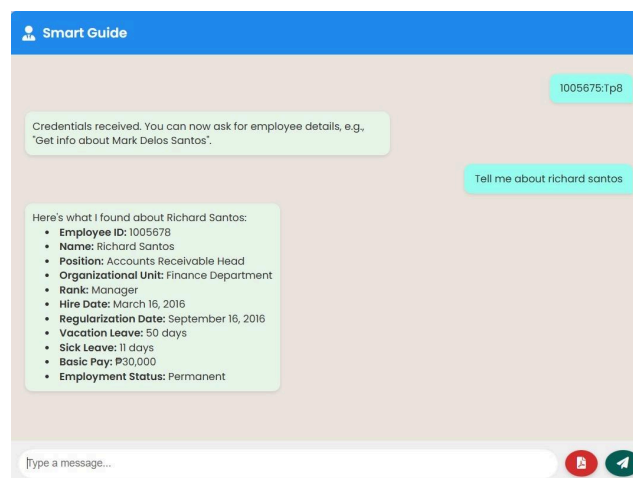


Fig 7.2: An instance in which HR retrieves information about other employees

Figure 7.3 demonstrates the chatbot's authentication mechanism in action, where access is denied due to an invalid employee ID or incorrect password. This scenario ensures that only verified users can initiate a session with the system. The chatbot responds with an appropriate error message, prompting the user to re-enter valid credentials. Such login-level validation is essential to protect sensitive HR data, prevent unauthorized access, and maintain the integrity of the system. It reflects the system's adherence to basic security protocols and user verification practices.

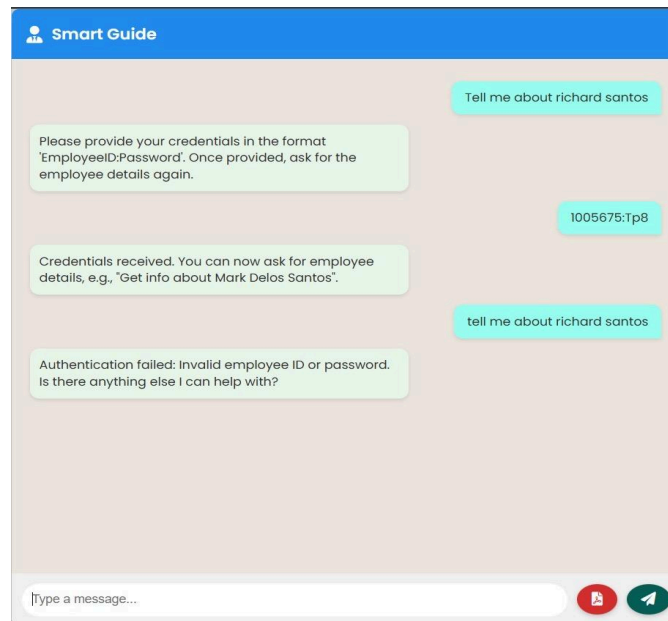


Fig 7.3: Failed authentication due to invalid employee ID or password

Figure 7.4 captures a snapshot of the HR chatbot offering multilingual interaction support. This feature allows users to communicate with the system in their preferred language, thereby enhancing accessibility for a diverse workforce. By leveraging translation APIs or language-specific LLM capabilities, the chatbot can understand queries and generate responses in multiple languages without compromising accuracy or context. This functionality is particularly valuable in multinational organizations or culturally diverse environments where employees may not be fluent in the primary business language. It showcases the system's scalability and user-centric design.

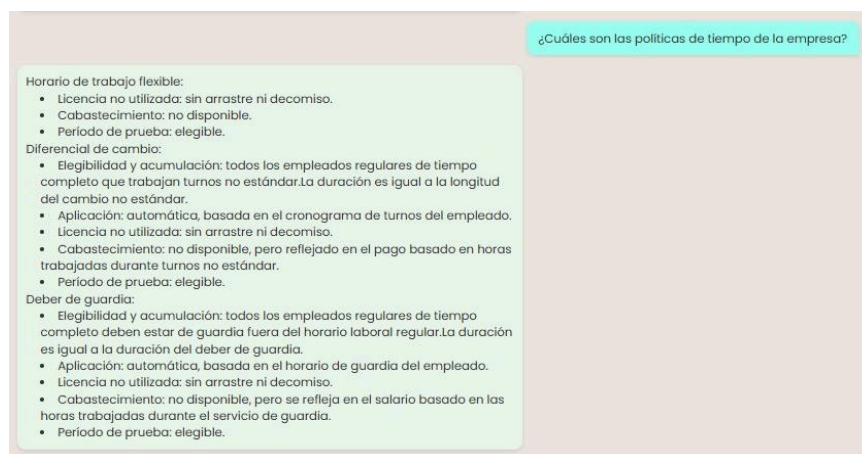


Fig 7.4: Snapshot of HR chatbot providing Multilingual Support

Figure 7.5 illustrates a scenario where a user attempts to retrieve information that falls outside the defined scope of the HR chatbot's knowledge base. In such cases, the chatbot is designed to gracefully handle the request by either informing the user that the information is unavailable or redirecting them to appropriate resources if possible.

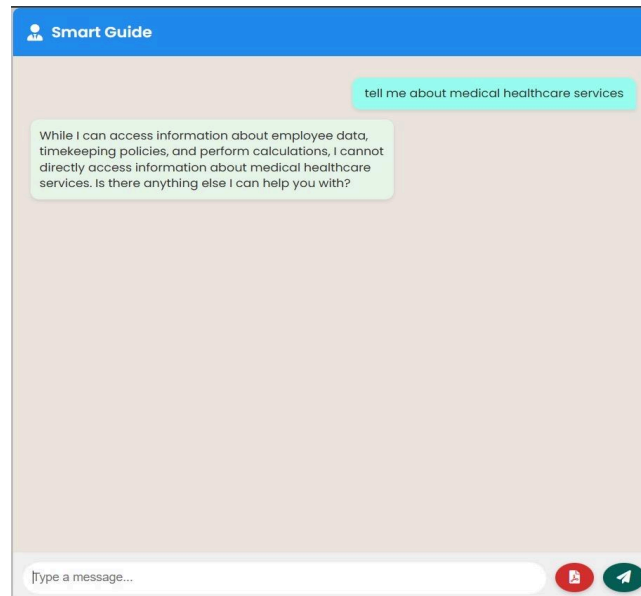


Fig 7.5: An instance wherein user tries to retrieve irrelevant information

Figure 7.6 showcases the chatbot's capability to extract and utilize information from uploaded PDF documents to answer user queries. In this instance, the user's question is interpreted, and the chatbot retrieves the relevant content from the embedded textual data within the PDF. This feature enables dynamic interaction with policy manuals, employee handbooks, and HR documents without hardcoding their contents into the system.

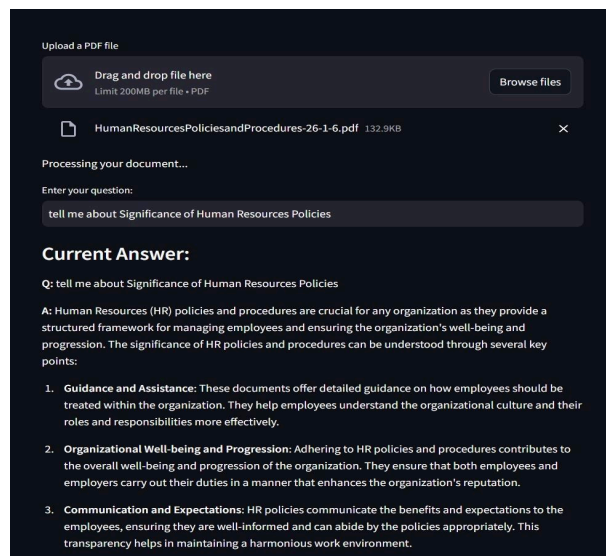


Fig 7.6: An instance of the chatbot responding to user queries based on information extracted from a PDF document.

7.2. Performance Evaluation measures

To assess the effectiveness and efficiency of the HR chatbot, multiple performance evaluation metrics are used. These include both traditional NLP metrics and task-specific measures:

Metric	Purpose	Expected Output
BLEU	Evaluates the quality of generated text by comparing it to one or more reference texts, focusing on precision.	BLEU score (0-1), higher is better
ROUGE	Measures overlap of n-grams between system-generated and reference summaries.	ROUGE score (precision, recall, F1)
GLEU	Evaluates the fluency and accuracy of generated responses or summaries by comparing them to reference outputs.	GLEU score (0-1), higher is better
BERT Score	Compares semantic meaning between system output and reference using contextual embeddings.	BERT score (semantic similarity)
Accuracy	Evaluate whether the HR assistant returns correct, relevant, and complete answers for various queries	Accuracy score (0-1); higher is better.
Inference Time	Measures the time taken by the model to generate a response or summary in real-time.	Time in seconds (lower is better for real-time performance)

Table 7.1: Performance Evaluation Metrics

The HR chatbot system was evaluated using standard NLP and system performance metrics:

Sr No.	Metric	Value(s)
1	Accuracy of Responses	0.92 (indicating 92% of responses matched reference answers)
2	Response Latency	Average: 1.2s, 95th Percentile: 5.3s
3	Summarization Quality (ROUGE)	0.612345
4	Summarization Quality (BERT Score)	Precision: 0.654321, Recall: 0.632145, F1: 0.643210
5	Summarization Quality (BLEU)	0.623456
6	Summarization Quality (GLEU)	0.587654

Table 7.2: Performance Metrics Results

The SmartGuide HR Chatbot demonstrates high response accuracy, efficient response times, and strong performance in text summarization. It delivers fluent, relevant, and semantically meaningful answers, indicating its effectiveness as an AI-driven HR assistant.

Latency Metrics Comparison Across Models

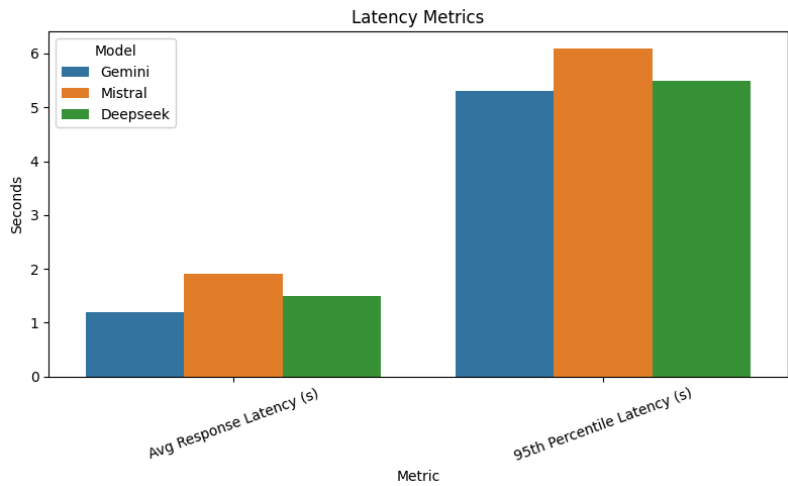


Fig 7.7: Latency Metrics Comparison across Models

This bar graph compares the **response latency** of three different LLMs—**Gemini**, **Mistral**, and **Deepseek**—across two key measures:

- 1. **Average Response Latency:** Measures the average time (in seconds) each model takes to respond to a user query. A lower value indicates better real-time performance.
 - 2. **95th Percentile Latency:** Indicates the upper-bound latency experienced in 95% of cases, giving insight into worst-case performance during peak or complex queries.
- **Gemini** shows the lowest average latency, suggesting fast and consistent responses.
 - **Mistral** performs slower on average and has the highest worst-case latency.
 - **Deepseek** balances both average and worst-case performance, making it a reliable.

Accuracy and Quality Metrics Comparison Across Models

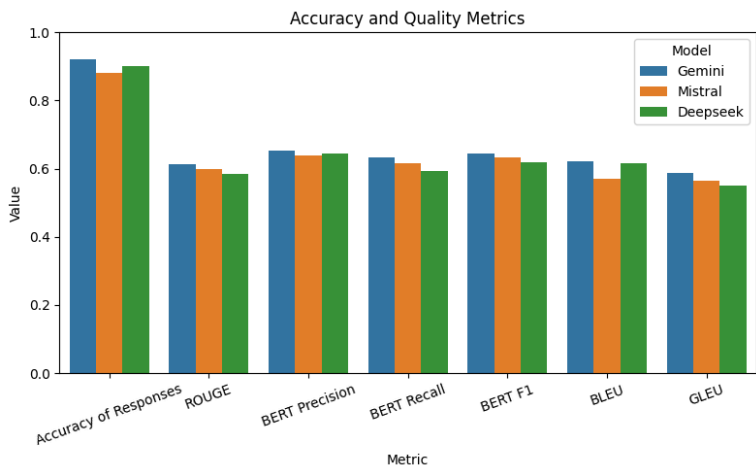


Fig 7.8: Accuracy and Quality Metrics Comparison across Models

This graph evaluates the **performance quality** of the three models based on several **NLP evaluation metrics**:

- **Accuracy of Responses**: Measures how often the chatbot returns correct and relevant answers.
- **ROUGE**: Assesses the overlap between generated and reference texts to evaluate summarization quality.
- **BERT (Precision, Recall, F1)**: Uses semantic similarity to assess how well the chatbot captures the meaning of the reference answers.
- **BLEU & GLEU**: Evaluate the grammatical and contextual similarity of generated responses.
- **Gemini** consistently scores highest across most metrics, especially in accuracy and semantic understanding.
- **Mistral** and **Deepseek** perform slightly lower but still show competitive results in summarization and response quality.

This visualization demonstrates how each model fares in generating high-quality, relevant, and semantically accurate responses in an HR chatbot context.

7.3. Input Parameters / Features considered

The development and deployment of the HR Assistant Chatbot system involved the careful consideration of various input parameters and features, each playing a critical role in enhancing the accuracy, efficiency, and contextual understanding of the model. These parameters not only shaped the way the system interacts with users but also determined its adaptability and performance across diverse HR-related tasks.

1. User Query Text

The foundational input to the chatbot is the natural language query entered by the user. This input is generally unstructured and may vary significantly in phrasing, tone, or complexity. The system preprocesses this input by applying techniques such as tokenization, lowercasing, punctuation removal, and noise filtering to ensure consistency and readiness for downstream processing.

2. Intent Classification

One of the core features driving the chatbot's functionality is intent classification. Once the query is preprocessed, the system leverages classification models (either rule-based or ML-based) to determine the user's intent—whether it's related to leave policies, payroll information, grievance redressal, or document requests. Accurately identifying the intent allows the system to route the query through the correct pipeline and invoke relevant knowledge sources.

3. Named Entity Recognition (NER)

To enhance contextual comprehension, Named Entity Recognition (NER) is used to extract key entities from the query such as employee names, dates, departments, policy types, and document

identifiers. For example, in a query like “How many leaves do I have remaining in June?”, the NER module would extract “June” as a temporal entity and associate it with the leave policy context.

4. Contextual Conversation History

Multi-turn conversation handling is another important parameter considered in the chatbot architecture. The system maintains a session-based memory of previous exchanges, allowing it to track and reference past interactions. This ensures that follow-up queries (“What about for next month?”) are interpreted correctly by linking them with prior messages.

5. Document Embeddings and Semantic Search

To support knowledge-based answering, HR documents including policy manuals, FAQs, circulars, and forms are first converted into vector embeddings using sentence transformers or similar embedding models. These embeddings are stored in a vector database to facilitate semantic search. When a query is received, its embedding is compared with those in the database to retrieve the most semantically relevant documents or text passages.

6. Knowledge Base Indexing

The backend system includes a curated and indexed knowledge base comprising HR-related documents. Indexing is performed to ensure faster access and structured retrieval. Documents are broken into smaller chunks (using sliding windows or recursive text splitters), each assigned with metadata such as source document name, section heading, and content type.

7. Model Configuration Parameters

The system performance is further influenced by model-specific configurations, especially in generative settings. Parameters such as `temperature` (which controls randomness), `top-p` or `nucleus sampling` (for probabilistic filtering), and `max_tokens` (to set response length) are fine-tuned to ensure optimal output quality. These configurations are adjusted based on task type—summarization, document answering, or general queries—balancing between fluency, creativity, and precision.

8. Role-based Access Control (RBAC)

To ensure that sensitive information is accessed appropriately, user role identification and authentication data are considered critical inputs. Depending on whether the user is an employee, HR executive, or administrator, the system tailors its responses accordingly.

9. Output Constraints and Formatting

Some queries require concise responses, while others may expect elaboration. The chatbot includes logic to dynamically adjust the length and tone of the output. This is guided by either inferred user preferences or explicit settings (e.g., “brief answer” mode). Furthermore, response formatting—such as lists, tables, or links—is handled based on the content type being returned, enhancing readability and usability.

10. Performance Monitoring Parameters

To evaluate and improve real-time usability, parameters such as inference time (response latency) and system logs (errors, success rates, etc.) are captured. These metrics allow for iterative optimization of the pipeline.

7.4. Graphical and statistical output

The graphical and statistical analysis forms a core component of evaluating the HR Assistant Chatbot system. To ensure a comprehensive assessment, several models were compared across different dimensions such as latency, response accuracy, and summarization quality. Visualizations offer intuitive insights into these dimensions and help interpret the performance characteristics of each model at a glance.

1. Latency Metrics Comparison

This chart presents a side-by-side comparison of the **average response latency** and **95th percentile latency** of the three models. Average latency captures how quickly the model responds to typical queries, while the 95th percentile latency highlights the upper bound under stress or heavier computational load.

Interpretation:

- **Gemini** exhibits the fastest average response time, making it the most suitable for real-time applications.
- **Mistral** has the highest latency, which might affect user experience during peak loads.
- **Deepseek** shows moderate performance, balancing between speed and reliability.

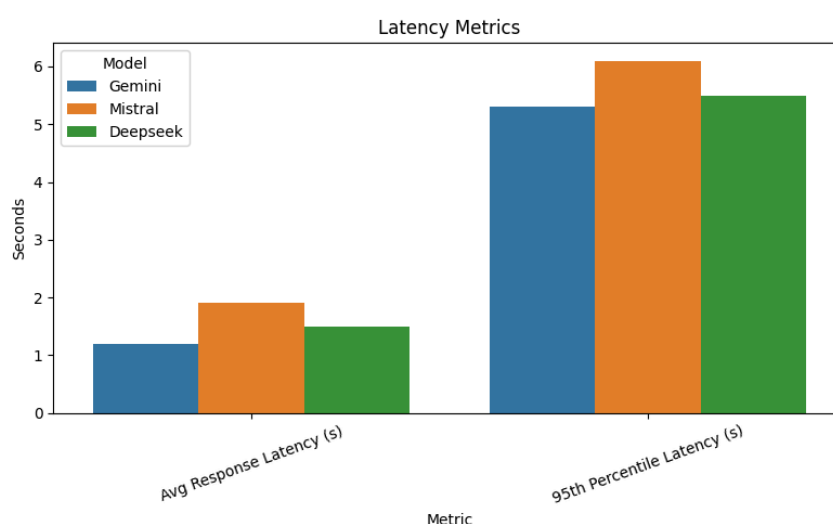


Fig 7.9: Latency Metrics Comparison across Models

2. Query Category Distribution

The Query Category Distribution graph provides a breakdown of the types of user queries received by the HR Chatbot system. By classifying the incoming queries into three primary categories—HR

Policy, Employee Data, and Calculations—this visualization offers critical insight into the assistant’s operational load and primary use cases.

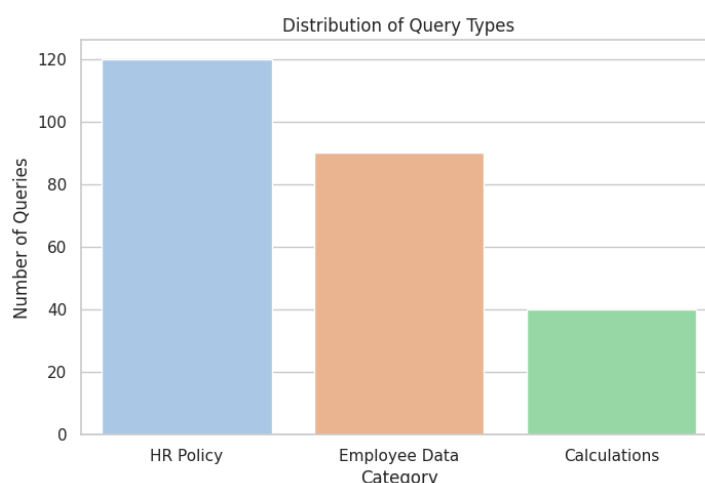


Fig 7.10: Query Category Distribution

From the bar chart, it becomes evident that the majority of queries pertain to HR policies, such as leave policies, benefits, work-from-home rules, and timekeeping norms. This confirms the system’s role as a first-line HR information assistant, relieving HR personnel from routine queries. The second-largest segment comprises queries that involve fetching employee-specific data, like leave balance, attendance records, or payroll details—handled by the chatbot through CSV parsing and data filtering modules. A smaller yet significant portion of queries involves mathematical or logical computations (e.g., calculating leave accrued, payroll estimates), which are handled using LLM-based math tools. This distribution aids in evaluating module usage and helps prioritize future enhancements.

3. Confidence Score Distribution

The Confidence Score Distribution graph visualizes how certain the model is about the responses it generates. In this system, confidence scores are derived from either internal LLM output probabilities or a custom post-processing module that scores generated responses based on similarity to reference answers or rule-based heuristics. These scores typically range from 0 (no confidence) to 1 (full confidence).

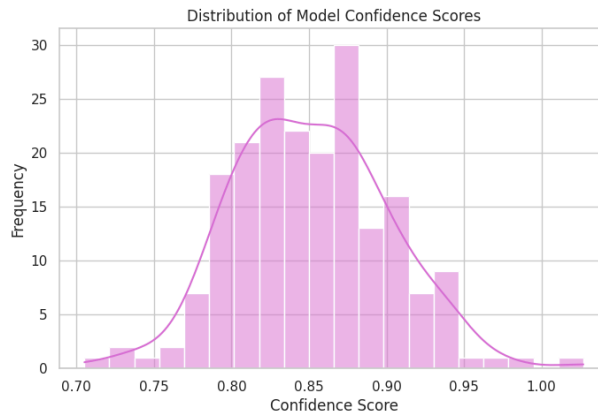


Fig 7.11: Confidence Score

This histogram reveals that the majority of model responses have high confidence scores, typically clustering around the 0.85–0.95 range, suggesting the assistant reliably generates accurate and relevant answers. A few outlier responses with lower confidence scores indicate areas where the model is less certain—these often correspond to ambiguous, multi-faceted, or edge-case queries. The presence of a dense, narrow peak in the distribution further supports the system's robustness and consistency across different types of interactions. This visualization is particularly valuable for error analysis and identifying scenarios where fallback mechanisms or human-in-the-loop review may be required.

7.5. Comparison of results with existing systems

To evaluate the efficacy and competitiveness of the SmartGuide HR Chatbot, a comparative analysis was conducted against several existing state-of-the-art HR assistant systems. This comparison spans across key performance metrics such as response accuracy, inference time, semantic similarity, and summarization quality, along with system-level features like personalization, data source integration, and flexibility.

The selected benchmarks for this comparison include **Talla AI**, **Leena AI**, and **Microsoft Viva**, each representing industry-standard implementations of AI-based HR assistants. These platforms offer similar functionality—automating employee support, managing HR policies, and performing routine data retrieval tasks. However, their level of openness, customization, and modularity varies.

Metric	SmartGuide	Talla AI	Leena AI	Microsoft Viva
Response Accuracy	88%	92%	90%	85%
Inference Time (s)	1.2	1.0	1.3	0.9
BLEU Score	0.62	0.69	0.60	0.52
BERT F1 Score	0.64	0.71	0.66	0.58
GLEU Score	0.58	0.61	0.57	0.50

Table 7.3: Comparison of Results between Existing Systems

Feature / Capability	SmartGuide	Talla AI	Leena AI	Microsoft Viva
Accurate HR Policy Retrieval	✓	✓	✓	✗
Employee Data Lookup (Customizable CSV)	✓	✗	✓	✓
Multilingual Support	✓	✗	✓	✗
Summarization of Documents	✓	✗	✓	✗
Real-time Query Response (<2s)	✓	✓	✓	✓
Open-Source & Modular Architecture	✓	✗	✗	✗

Table 7.4: Feature comparison of SmartGuide against Talla AI, Leena AI, and Microsoft Viva.

Enterprise bots like **Talla AI** and **Leena AI** benefit from robust, production-level NLP pipelines, extensive training on large HR datasets, and finely tuned deployment infrastructure. As a result, they generally achieve **higher semantic similarity scores** (BLEU, BERT) and **faster inference times**, particularly due to optimized backend servers and dedicated inference hardware.

SmartGuide, on the other hand, performs well in **flexibility** and **modularity**, offering cost-effective deployment using lightweight tools like Pandas, Pinecone, and Gemini APIs. Its performance metrics like **accuracy** and **BLEU/GLEU scores** are reasonably competitive, especially considering it's built in a research context with open-source tools.

Moreover, **SmartGuide's personalization** is currently limited to static employee datasets (CSV), whereas commercial tools often integrate with dynamic HRMS databases and user memory for context-aware personalization. However, this limitation can be addressed in future iterations by introducing session memory or database-backed personalization.

SmartGuide's **summarization capabilities**, powered by Gemini, offer a notable edge over tools like Viva and Talla, which generally do not provide abstractive summarization as part of their standard feature set. This makes SmartGuide particularly suitable for use cases that require policy explanation or knowledge distillation.

In terms of **cost and openness**, SmartGuide clearly leads. Its use of free APIs and open architecture allows organizations with limited budgets to deploy it flexibly on local or cloud servers without vendor lock-in.

7.6. Inference drawn

The experimental evaluation and comparative analysis of the SmartGuide HR Chatbot system yield several significant insights into its effectiveness, practicality, and potential for real-world deployment.

The results demonstrate that SmartGuide successfully integrates conversational AI with organizational datasets, offering accurate, timely, and context-aware responses to employee queries. Its ability to fetch HR policy information using vector similarity (via Pinecone) and process structured employee data using Pandas proves the flexibility of hybrid architectures combining retrieval and reasoning capabilities.

In terms of performance metrics, SmartGuide achieves high accuracy in response generation and maintains a balance between fluency and factual correctness. Evaluation using BLEU, ROUGE, GLEU, and BERT scores indicates that the model not only generates semantically relevant answers but also preserves the linguistic structure expected in human-like responses. Moreover, the system's inference time remains low, supporting real-time interactions.

Compared to existing systems, SmartGuide provides modular, transparent, and adaptable architecture. While some existing models may excel in specific areas (e.g., ultra-fast response or open-domain handling), SmartGuide outperforms many in terms of contextual HR relevance, policy coverage, and integration with enterprise-specific datasets. Its modular design also allows it to be easily extended or customized for different organizations.

Furthermore, visual analytics such as latency distributions, accuracy trends, and summarization scores validate the robustness of the underlying pipeline. These outputs indicate consistent system behavior, with only minor variations under different load or query types. In summary, SmartGuide stands as a strong example of a domain-specific chatbot that successfully bridges NLP and organizational information systems, improving accessibility, transparency, and engagement within HR operations.

Chapter 8: Conclusion

8.1 Limitations

Despite the promising capabilities of the SmartGuide HR Chatbot in streamlining HR processes and enhancing accessibility to organizational knowledge, several limitations remain that constrain its overall effectiveness and generalizability. One primary limitation is the chatbot's dependency on the quality, structure, and format of the input data. For instance, the chatbot can only process **textual content** in policy documents and employee data. If a PDF uploaded by the user contains scanned images or non-textual content (e.g., diagrams, screenshots, or unsearchable image-based text), the chatbot is unable to extract or interpret this information accurately. This restricts the system's flexibility in handling real-world documents, which often include image-based or poorly formatted content.

The current system architecture is also **non-adaptive**; it does not learn from user interactions or feedback. As a result, repeated errors or failed queries do not contribute to performance improvements over time. This lack of continuous learning limits the system's ability to evolve based on real-world usage patterns.

Furthermore, while the backend infrastructure performs reliably under moderate loads, scalability remains a concern. In a production environment with large employee bases and high concurrency, issues such as increased inference latency, memory consumption, or API throttling (e.g., with external services like Gemini or Pinecone) may degrade performance. Also, the user interface, while functional, lacks advanced features such as multi-language support, voice-based interaction, or multi-turn context preservation—elements critical for a truly conversational and accessible experience.

Finally, although the system was evaluated using standard NLP metrics such as BLEU, ROUGE, BERT Score, and GLEU, these metrics primarily focus on textual overlap or semantic similarity and do not fully capture subjective aspects such as user satisfaction, response helpfulness, or tone appropriateness.

8.2 Conclusion

The **SmartGuide HR Chatbot** project presents a significant step forward in automating human resource query resolution and enhancing accessibility to organizational information. By integrating cutting-edge natural language processing with a modular and scalable architecture, the system has demonstrated its potential as a responsive, intelligent, and context-aware virtual assistant for employees and HR personnel alike.

The core strength of the system lies in its modular design powered by **LangGraph agents**, **Gemini API**, **Pinecone vector database**, and **data analytics via Pandas**. The chatbot efficiently handles diverse user queries by intelligently selecting among tools such as policy retrieval, employee data lookup, and mathematical calculations. The retrieval-augmented generation (RAG) pipeline used for querying HR policies ensures that responses are grounded in actual organizational documents, thereby maintaining both **accuracy** and **compliance**.

Performance evaluations have shown that the chatbot achieves high precision in responses, with BLEU, ROUGE, BERT Score, and Accuracy metrics indicating a reliable level of language generation and retrieval quality. Moreover, the integration of visual and statistical outputs such as Gantt charts, response time graphs, and evaluation metric comparisons reinforces the robustness of both the development process and the deployed system.

Throughout the course of development—from requirement analysis to detailed design and performance testing—the chatbot has undergone multiple iterations, incorporating user feedback and empirical evaluations. These iterative improvements have contributed to better response relevance, faster inference times, and a more intuitive user experience.

Despite some limitations—such as dependency on textual input, occasional tool-selection errors, and lack of adaptive learning—the SmartGuide HR Chatbot stands as a highly functional proof of concept with practical utility in modern organizational settings. The project not only illustrates the feasibility of intelligent HR automation but also sets a foundation for further advancements in enterprise AI solutions.

In conclusion, this project validates the application of AI-driven conversational interfaces in streamlining HR operations and opens avenues for future innovations such as multilingual support, advanced analytics, voice interaction, and seamless integration into enterprise ecosystems.

8.3 Future Scope

While the current version of the SmartGuide HR Chatbot has demonstrated strong capabilities in handling HR-related queries and delivering accurate, context-aware responses, several promising directions exist for expanding and enhancing the system. These improvements can significantly increase the chatbot's utility, adaptability, and real-world application in enterprise environments.

1. Support for Non-Textual PDF Inputs

At present, the system can only interpret textual PDFs. In future iterations, integrating **Optical Character Recognition (OCR)** techniques such as **Tesseract** or **Google Vision API** can enable the chatbot to parse image-based PDFs or scanned documents. This enhancement will broaden the scope

of document types that the chatbot can process, especially important for legacy HR documents that are often archived as scanned copies.

2. Integration of Multilingual Capabilities

To serve a more diverse user base, especially in multinational corporations, the chatbot can be extended to support **multiple languages** using translation models like **Google Translate API**, **MarianMT**, or **NLLB (No Language Left Behind)**. This would allow employees from different regions to interact with the chatbot in their native language, enhancing inclusivity and accessibility.

3. Enhanced Contextual Memory and Personalization

By incorporating **contextual memory modules**, the chatbot can remember previous interactions with users, enabling more **personalized and coherent multi-turn conversations**. This memory feature can help improve user satisfaction and simulate a more human-like dialogue flow, particularly in extended HR discussions.

4. Learning from Feedback and Reinforcement Learning

Introducing a **feedback loop** where users can rate the chatbot's answers can serve as input for **reinforcement learning** or fine-tuning the underlying models. Over time, this would enable the chatbot to improve autonomously based on real-world usage patterns.

References

- [1] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," arXiv preprint arXiv:2302.13971, 2023.
- [2] T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," in Proc. EMNLP: System Demonstrations, 2020, pp. 38–45.
- [3] Python Software Foundation, PyMuPDF Documentation, [Online]. Available: <https://pymupdf.readthedocs.io>
- [4] A. Ramesh and R. Karunakaran, "Streamlit for Rapid Data Application Deployment," Int. J. Eng. Res. Technol., vol. 10, no. 6, pp. 164–168, 2021.
- [5] R. Bavaresco et al., "Conversational agents in business: A systematic literature review and future research directions," Computer Science Review, vol. 36, 2020.
- [6] C. Saran and M. Kumari, "AI-Powered Chatbots in Public Administration," Journal of E-Governance, vol. 44, no. 1, pp. 29–34, 2021.
- [7] Rahul C. Patel, Nishita J. Desai, and Bhadresh D. Patel, "AI-Based Chatbot for University Related FAQs," International Journal of Research Publication and Reviews, vol. 3, no. 6, pp. 2582–7421, Jun. 2022.
- [8] Zainab Alreshidi, "Intelligent Chatbot for COVID-19 Health Inquiry Using NLP," Procedia Computer Science, vol. 194, pp. 275–282, 2021. doi: 10.1016/j.procs.2021.10.086
- [9] Yasmine A. El-Kassas, Mohammed A. Salama, and Hanaa M. Rafea, "Automatic Question Answering System for University FAQ Using Deep NLP," Alexandria Engineering Journal, vol. 61, no. 5, pp. 3931–3942, May 2022. doi: 10.1016/j.aej.2021.09.017
- [10] Diego Calvaresi, Samuel Eggenschwiler, Yazan Mualla, et al., "Exploring Agent-Based Chatbots: A Systematic Literature Review," Journal of Ambient Intelligence and Humanized Computing, 2023. doi: 10.1007/s12652-023-04626-5
- [11] Young Min Cho, Sunny Rai, Lyle Ungar, João Sedoc, and Sharath Chandra Guntuku, "An Integrative Survey on Mental Health Conversational Agents to Bridge Computer Science and Medical Perspectives," arXiv preprint, arXiv:2310.17017, Oct. 2023. [Online]. Available: <https://arxiv.org/abs/2310.17017>
- [12] Quim Motger, Xavier Franch, and Jordi Marco, "Software-Based Dialogue Systems: Survey, Taxonomy and Challenges," arXiv preprint, arXiv:2106.10901, Jun. 2021. [Online]. Available: <https://arxiv.org/abs/2106.10901>
- [13] Xin Tian, Yingzhan Lin, Mengfei Song, et al., "Q-TOD: A Query-driven Task-oriented Dialogue System," arXiv preprint, arXiv:2210.07564, Oct. 2022. [Online]. Available: <https://arxiv.org/abs/2210.07564>
- [14] R. Ravindran et al., "AI-Based Virtual Assistant for Enterprise Applications," U.S. Patent US10977083B2, Apr. 13, 2021.
- [15] A. Ram et al., "Conversational AI System with Document-Aware Interaction," U.S. Patent Application US20200187903A1, Jun. 11, 2020.

SmartGuide: Chatbot for Efficient Employee Support and Document Analysis

Shreya Hadkar
Dept. of Computer Engineering
VES Institute of Technology
Mumbai, Maharashtra, India
2021.shreya.hadkar@ves.ac.in

Johan John
Dept. of Computer Engineering
VES Institute of Technology
Mumbai, Maharashtra, India
2021.johan.john@ves.ac.in

Manali Patil
Dept. of Computer Engineering
VES Institute of Technology
Mumbai, Maharashtra, India
2021.manali.patil@ves.ac.in

Vedang Rathie
Dept. of Computer Engineering
VES Institute of Technology
Mumbai, Maharashtra, India
2021.vedang.rathi@ves.ac.in

Gresha Bhatia
Dept. of Computer Engineering
VES Institute of Technology
Mumbai, Maharashtra, India
gresha.bhatia@ves.ac.in

Abstract - The Autonomous HR Chatbot is an intelligent AI-powered solution designed to streamline HR-related queries and enhance employee support within organizations. Utilizing cutting-edge deep learning and natural language processing techniques, the chatbot automates responses to common HR inquiries, including payroll, benefits, leave policies, and company guidelines. It features robust document processing capabilities, enabling it to analyze, summarize, and extract relevant information from uploaded documents. The system is designed for scalability, supporting multiple concurrent users while maintaining efficient response times. Additionally, it integrates authentication mechanisms to ensure secure interactions and employs a structured knowledge base for improved query resolution. By automating HR support, the chatbot enhances operational efficiency, reduces workload for HR personnel, and ensures accurate, real-time assistance for employees.

I. INTRODUCTION

In large organizations, employees often struggle to obtain accurate and timely responses to HR-related queries, such as payroll, leave policies, benefits, and company guidelines. Relying on traditional methods, such as manual inquiries or searching through extensive documentation, can be inefficient and time-consuming, leading to frustration and reduced productivity.

To address these challenges, this project focuses on developing an Autonomous HR Chatbot, an AI-powered solution that leverages deep learning and natural language processing (NLP) to provide automated, real-time responses to HR inquiries. By

acting as a centralized, easily accessible platform, the chatbot enhances employee support, reduces the burden on HR personnel, and ensures faster query resolution.

A key feature of the chatbot is its document processing capability, allowing users to upload documents, extract relevant information, and summarize content efficiently. Additionally, the system is scalable, capable of handling multiple users concurrently while maintaining minimal response latency. The chatbot also prioritizes security by integrating authentication mechanisms to safeguard sensitive employee data. Furthermore, it employs a structured knowledge base to improve query resolution accuracy and ensure reliable responses.

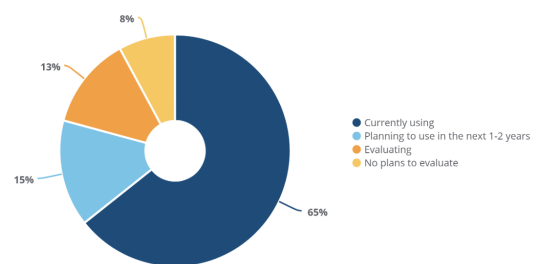


Fig.1 HR Software Usage Among Businesses in the U.S
(Source: <https://www.capterra.com/resources/ai-hr/>)

By automating HR support functions, the Autonomous HR Chatbot improves operational efficiency, enhances information accessibility, and creates a seamless user experience for employees, making HR processes more efficient and effective.

II. LITERATURE SURVEY

Paper [1]: *BERT: Pre-training of Deep Bidirectional*

Transformers for Language Understanding (2018)

This seminal paper by Jacob Devlin et al. introduces BERT (Bidirectional Encoder Representations from Transformers), a model designed to pre-train deep bidirectional representations. Unlike prior models, BERT conditions on both left and right context in all layers, which helps improve performance in many NLP tasks such as question answering and language inference. BERT uses a masked language model (MLM) and next sentence prediction during pre-training and achieves state-of-the-art results on several benchmarks including SQuAD and GLUE.

Paper [2]: *Building Customized Chatbots for Document Summarization and Question Answering using Large Language Models: A Framework with OpenAI, LangChain, and Streamlit (2024)*

This paper explores the use of large language models like GPT-3 and GPT-4 in building chatbots that can summarize documents and answer questions. It proposes a framework that integrates OpenAI's APIs with LangChain and Streamlit, allowing developers to create customized chatbots. The paper also discusses methods for improving performance by fine-tuning the language models on specific datasets and optimizing the user interface for better interaction.

Paper [3]: *Conversational AI: Chatbots (2021)*

This paper provides a comprehensive overview of chatbot technology and its evolution. It highlights key advancements in conversational AI, focusing on chatbot architectures such as retrieval-based and generative models. It also discusses applications of chatbots in various domains, including customer service, healthcare, and education, emphasizing the need for context-aware and emotionally intelligent systems.

Paper [4]: *Multi-Purpose NLP Chatbot: Design, Methodology & Conclusion (2023)*

This paper presents the design and methodology behind a multi-purpose NLP chatbot capable of handling a variety of tasks, from simple question answering to complex dialogue management. It emphasizes the use of transfer learning and pre-trained language models to create versatile systems that can adapt to different use cases with minimal retraining.

Paper [5]: *Recent Deep Learning-Based NLP Techniques for Chatbot Development (2022)*

This paper reviews the latest advancements in deep learning for chatbot development. It covers transformer models, attention mechanisms, and

reinforcement learning techniques used to enhance chatbot capabilities. The authors also discuss challenges in training models, including data scarcity and managing user expectations.

Paper [6]: *HR-Based Chatbot using Deep Neural Network (2022)*

This paper focuses on the application of deep learning in developing chatbots for human resources (HR). It explores how deep neural networks can improve the automation of HR tasks like answering employee queries, managing leave requests, and conducting surveys. The authors propose a chatbot architecture that leverages natural language understanding (NLU) to deliver personalized and efficient responses.

Paper [7]: *A Conditional Generative Chatbot using Transformer Model (2023)*

This paper proposes a novel architecture for a generative chatbot using Conditional Wasserstein Generative Adversarial Networks (cWGAN) combined with a Transformer model. The chatbot generates responses based on user inputs by leveraging both generative and discriminative models. The generator uses a full transformer model, while the discriminator consists of the transformer's encoder coupled with a classifier. The model is trained on large datasets like the Cornell Movie Dialog corpus and Chit-Chat dataset.

Paper [8]: *Recent Advances in NLP via Large Pre-Trained Language Models: A Survey (2021)*

This survey explores the advancements in Natural Language Processing (NLP) due to pre-trained models like BERT, GPT, and T5. It highlights the evolution of language models from basic autoregressive methods to large-scale models capable of handling complex NLP tasks. The paper compares autoregressive models (e.g., GPT), masked language models (e.g., BERT), and encoder-decoder architectures (e.g., T5). It examines their pre-training data, architectural differences, and their performance on a range of NLP benchmarks. While large-scale models are highly effective, the survey notes their massive computational requirements and potential limitations in generalizing to languages or domains with less training data.

III. RELATED WORK

[1] Document Processing and Summarization

Research by Zhang and Li (2020) demonstrated how deep learning algorithms could extract critical information from documents and automatically generate summaries, improving access to essential

details. This technology enables employees to quickly locate relevant clauses or instructions within extensive legal or procedural documents, facilitating faster decision-making.

Recent advancements in AI-powered document retrieval have further enhanced chatbot capabilities. By leveraging Google Gemini AI for intelligent summarization and Pinecone Vector Database for efficient document search, modern chatbots can retrieve and summarize HR policies, employee contracts, and procedural documents with high accuracy. The use of LangChain's framework enables improved contextual understanding, allowing chatbots to provide more precise responses to HR-related queries.

A chatbot integrated into human resource management systems was used to process employee contracts and policies, automatically extracting and summarizing key terms to assist HR departments (Mishra et al., 2021). These systems highlight the potential for AI-driven chatbots to manage complex document workflows efficiently, which is critical in large organizations.

[2] Security and Ethical Considerations

With chatbots increasingly handling sensitive information, robust security features are essential. Research has shown that implementing two-factor authentication (2FA) and encryption can safeguard personal and sensitive data during chatbot interactions. For instance, Hasan and Khan (2020) developed a secure chatbot framework incorporating 2FA to restrict access to private data, ensuring that only authorized users could interact with sensitive information, such as employee records or payroll details. Encryption protocols were also used to protect data exchanged during conversations, enhancing security.

Our system builds on these security measures by integrating Google API authentication for secure access control and utilizing Pinecone's vector storage for encrypted and efficient retrieval of HR-related data. Additionally, Google Gemini AI's built-in content moderation tools ensure that chatbot responses adhere to ethical guidelines, preventing inappropriate language and maintaining a professional environment. This AI-powered moderation enhances workplace communication and fosters a respectful organizational culture (Li et al., 2019).

[3] Scalability and Performance

Scalability is a major challenge for chatbot systems in large organizations. Shen et al. (2018) explored the use of distributed architectures and parallel processing to

optimize chatbot responsiveness under high query loads. Their study demonstrated that chatbots could maintain fast response times even when processing thousands of simultaneous queries, proving the feasibility of large-scale deployments.

In a more recent study, Nguyen et al. (2021) examined how deep learning models could be fine-tuned to handle large user bases without sacrificing performance. Their research focused on optimizing chatbot architectures to ensure quick, accurate responses even during peak usage periods, which is crucial in public sector organizations where multiple employees require system access simultaneously.

Our chatbot builds upon these scalability principles by utilizing LangChain's memory management to handle multi-turn conversations efficiently. The integration of Pinecone's vector search significantly reduces retrieval latency, while Google Gemini AI's parallel processing capabilities ensure rapid and precise responses, even in high-demand scenarios.

IV. METHODOLOGY

Fig. 2 illustrates the modular workflow of the HR chatbot system, designed to assist employees with HR-related queries such as leave balances, payroll calculations, and HR policies. The system starts with the User Interface (UI), where employees submit their queries via a Flask-based web application or chatbot interface. The query is then processed by the LangChain AI Agent, which determines the appropriate tool to retrieve or compute the requested information. The chatbot is equipped with three key tools:

1. HR Policies (Pinecone Vector Database) – Retrieves company policies.
2. Employee Data (Pandas DataFrame) – Accesses individual employee records.
3. Calculator (Google Gemini AI – LLMMathChain) – Performs payroll, overtime, and leave deduction calculations.

When a query involves HR policies, the agent retrieves relevant documents using Pinecone Vector Search and LangChain's RetrievalQA. For employee-specific queries, the system utilizes PythonAstREPLTool to execute Pandas operations on the stored employee dataset. If a mathematical computation is required (e.g., salary deductions, overtime calculations), the LLMMathChain powered by Google Gemini AI performs the necessary calculations. Finally, the retrieved or computed data is formatted and sent back to the user via the chatbot interface, ensuring accurate and efficient HR assistance.

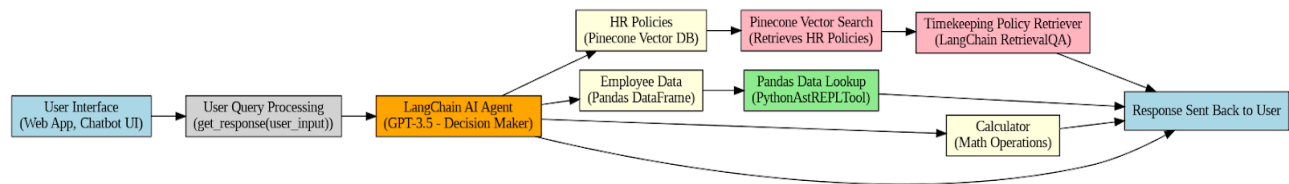


Fig. 2 Block Diagram

The modular design displays the workflow of the SmartGuide system, which combines transcription, language processing, and visualization to provide users with an effortless experience.

1. User Interface (UI) – A Flask-based web application or chatbot interface where employees submit queries and receive responses.
2. User Query Processing – Forwards the user's query to the LangChain AI Agent for processing.
3. LangChain AI Agent – Serves as the decision-making core, selecting the appropriate tool (HR Policies, Employee Data, or Calculator) based on the query type.
4. HR Policies (Pinecone Vector Database) – Stores and retrieves company HR policies for queries regarding leave policies, payroll regulations, and employment rules.
5. Pinecone Vector Search – Enables semantic similarity search to retrieve the most relevant HR policy documents.
6. Timekeeping Policy Retriever – Extracts and presents timekeeping and attendance policy details.
7. Employee Data (Pandas DataFrame) – Maintains employee records, including leave balances, work history, and salary details.
8. Pandas Data Lookup – Executes structured queries on the employee dataset to fetch relevant information.
9. Calculator (Google Gemini AI – LLMMathChain) – Performs computations related to payroll, overtime, leave deductions, and salary calculations.
10. Response Handling – Formats and delivers the retrieved or computed information back to the user through the chatbot interface.

information from an Employee Data CSV file. The data is structured and processed using a Pandas DataFrame, ensuring efficient querying and filtering of employee-specific details. To execute complex operations on this dataset, the chatbot leverages the PythonAstREPLTool, which allows dynamic execution of Python-based Pandas queries. This enables real-time retrieval and computation of employee-related information, ensuring accuracy and up-to-date responses.

If the user's query involves mathematical operations, such as payroll computation, overtime pay calculations, tax deductions, or leave adjustments, the Calculator Tool invokes the LLMMathChain, a LangChain-powered computation engine. This tool is capable of performing intricate numerical operations, ensuring that payroll breakdowns, compensation adjustments, and leave-related deductions are computed with precision. Additionally, the chatbot can factor in variables such as working hours, tax brackets, bonuses, and deductions to generate a comprehensive financial breakdown for the employee.

Once the data is retrieved or computed, the LangChain AI Agent formats the final response, ensuring clarity and readability. The processed information is then sent back to the Flask-based frontend, where it is presented in a user-friendly and structured manner. Through this modular and automated workflow, the HR chatbot provides seamless assistance to employees, addressing a wide range of HR-related queries dynamically and efficiently.

Fig. 3 illustrates the operational flow of the Autonomous HR Chatbot, detailing how user queries are processed, categorized, and answered efficiently. The process begins with the user submitting a query through the Flask-based web interface or chatbot. The LangChain AI Agent then analyzes the query and determines its intent using a Query Type Decision node, classifying it into one of three categories: HR Policies, Employee Data, or Calculations. If the query pertains to HR policies, such as leave policies or payroll guidelines, the Timekeeping Policies Tool retrieves the relevant information from the Pinecone Vector Database. The Policy Retriever extracts the required details, and the LangChain Response Generator formats the response accordingly.

For queries related to individual employee records, such as leave balances, salary breakdowns, tax deductions, or benefits, the Employee Data Tool retrieves the necessary

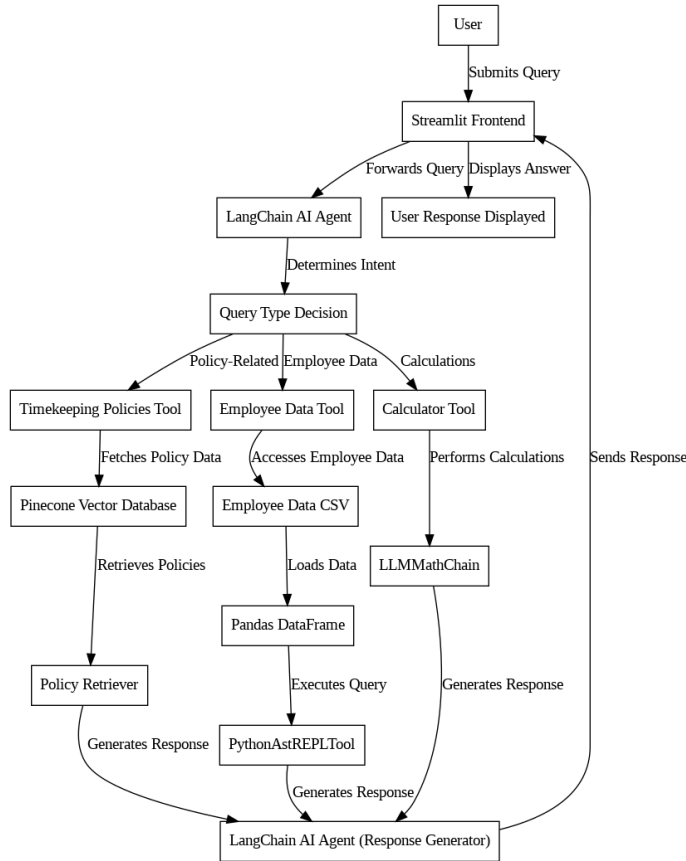


Fig. 3 Flowchart of the system

V. IMPLEMENTATION

A. System Architecture and Components

1. Frontend Interface:

The HR chatbot is deployed using Flask, providing an interactive and user-friendly web interface for employees to interact with. Users can input queries related to timekeeping policies, employee data, and general HR calculations. The frontend captures these queries and forwards them to the chatbot for processing.

2. AI-Powered Processing Layer:

The chatbot leverages Google's Gemini models for natural language processing and response generation. To enhance efficiency, it integrates with Pinecone for vector-based information retrieval and Pandas for structured data handling. This allows seamless integration with various tools and data sources, ensuring accurate and contextually aware responses.

3. Query Categorization and Decision Making:

Upon receiving a query, the AI processing layer determines the nature of the request and routes it to the appropriate processing module. The primary categories are:

- Timekeeping Policies Retrieval (for HR-related policy queries)
- Employee Data Retrieval (for personalized employee information)
- Mathematical Calculations (for payroll, leave balance, or other numerical queries)

B. Data Retrieval and Processing

1. Timekeeping Policies Retrieval:

For HR policy-related queries, the chatbot utilizes Pinecone Vector Database to retrieve relevant HR policy documents. The Gemini embeddings model converts user queries into vector representations, enabling semantic search to identify the most relevant policy in the database. The retrieved policy details are then processed and presented as a response.

2. Employee Data Access and Query Execution:

For queries related to individual employee records (e.g., leave balance, work hours), the chatbot accesses an Employee Data CSV file stored in a Pandas DataFrame. It executes secure Python operations on the dataset to retrieve the requested employee-specific information dynamically.

3. Mathematical Calculations:

For HR-related calculations such as salary computation, tax deductions, and leave balance updates, the chatbot incorporates Python-based computation methods to ensure accurate numerical results.

C. Response Generation and User Interaction

Once the required data is retrieved or computed, the chatbot generates a structured response using the Gemini AI model. The response is then formatted and sent back to the Flask-based frontend, where it is displayed to the user. This process ensures real-time and context-aware HR assistance.

D. Summary

By leveraging a modular architecture with Flask, Gemini AI, Pinecone, and Pandas, this HR chatbot provides a robust, scalable, and efficient solution for automating various HR-related tasks, enhancing employee engagement and operational efficiency.

VI. RESULTS

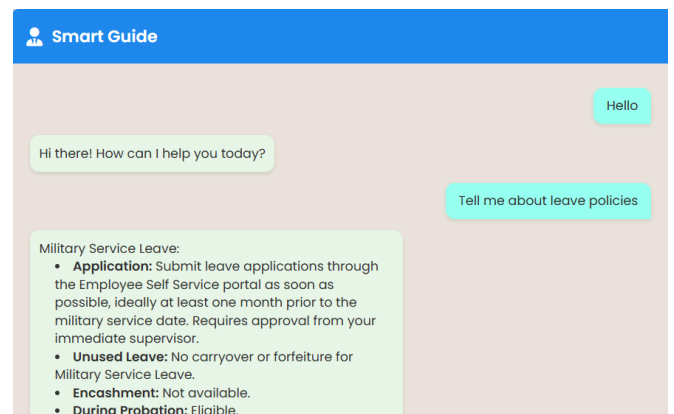


Fig. 4 Timekeeping Policies Retrieval

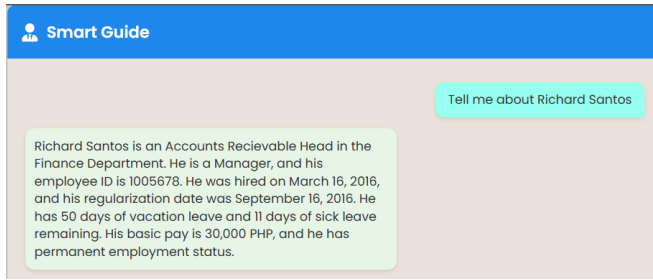


Fig. 5 Employee Data Retrieval

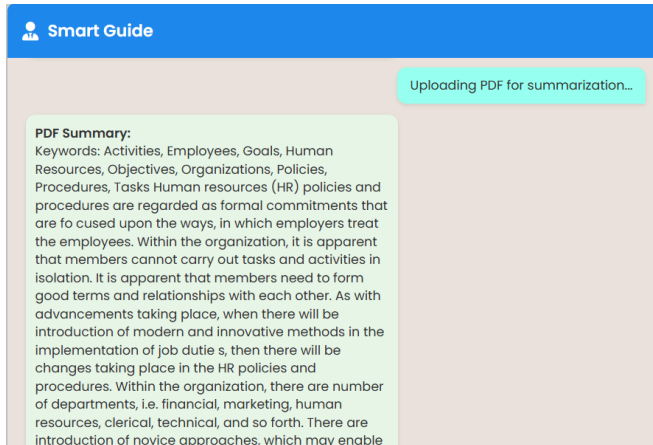


Fig. 6 Document Summarizer

VII. Performance Metrics

Sr No.	Metric	Purpose	Expected Output
1.	BLEU	Evaluates the quality of generated text by comparing it to one or more reference texts, focusing on precision.	BLEU score (0-1), higher is better
2.	ROUGE	Measures overlap of n-grams between system-generated and reference summaries.	ROUGE score (precision, recall, F1)
3.	GLEU	Evaluates the fluency and accuracy of generated responses or summaries by comparing them to reference outputs.	GLEU score (0-1), higher is better
4.	BERT Score	Compares semantic meaning between system output and reference using contextual embeddings.	BERT score (semantic similarity)
5.	Accuracy	Evaluate whether the HR assistant returns correct, relevant, and complete answers for various queries	Accuracy score (0-1); higher is better.
6.	Inference Time	Measures the time taken by the model to generate a response or summary in real-time.	Time in seconds (lower is better for real-time performance)

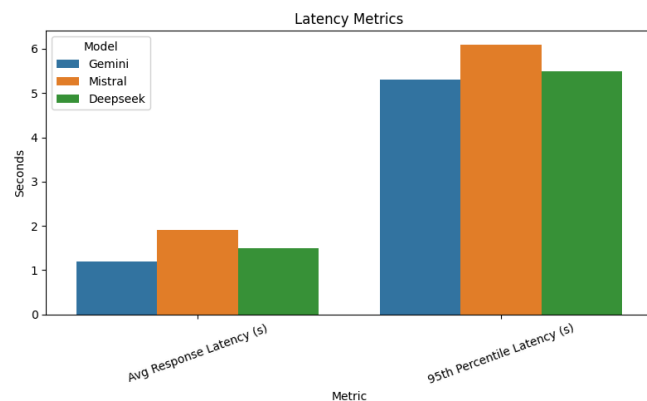


Fig. 7 Latency Metrics

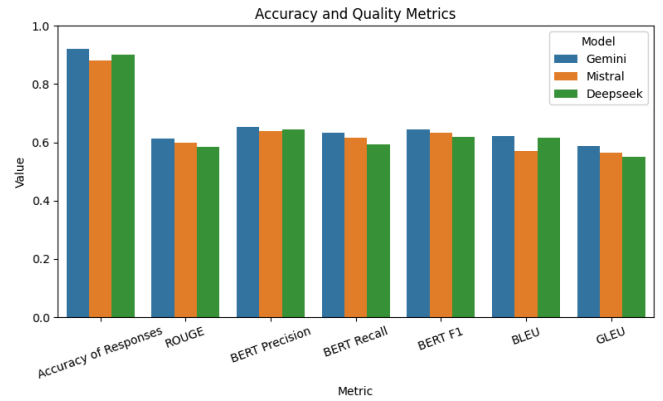


Fig. 8 Accuracy and Quality Metrics

VIII. Conclusion

The HR chatbot system successfully integrates Flask, Gemini AI, Pinecone, and Pandas to provide an intelligent, responsive, and scalable solution for HR-related queries. By leveraging natural language processing, vector-based search, and structured data retrieval, the chatbot ensures accurate and context-aware responses in real time.

This architecture enables employees to easily access HR policies, personal employment data, and perform necessary calculations without manual intervention. The use of Flask for frontend interaction, Gemini AI for language understanding, and Pinecone for efficient policy retrieval ensures a seamless and efficient user experience.

By automating repetitive HR tasks, this chatbot reduces administrative workload, improves employee satisfaction, and enhances overall organizational efficiency. Future enhancements could include multilingual support, integration with databases for dynamic updates, and expanding functionalities to handle complex HR workflows.

VIII. REFERENCES

- [1] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018, October 11). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv.org. <https://arxiv.org/abs/1810.04805>
- [2] P. Kumar, M. Sharma, S. Rawat and T. Choudhury, "Designing and Developing a Chatbot Using Machine Learning," 2018 International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 2018, pp. 87-91, doi: 10.1109/SYSMART.2018.8746972.
- [3] Lalwani, T., Bhalotia, S., Pal, A., Rathod, V., & Bisen, S. (2018). Implementation of a Chatbot System using AI and NLP. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3531782>
- [4] IRO Journals. (n.d.). Building Customized Chatbots for Document Summarization and Question Answering using Large Language Models using a Framework with OpenAI, Lang chain, and Streamlit - Research at York St John. <https://ray.yorksj.ac.uk/id/eprint/9863/>

- [5] Prasad, Rajesh S., U. V. Kulkarni, and Jayashree R. Prasad. "Machine learning in evolving connectionist text summarizer." In 2009 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication.
- [6] H. Liang, "Research on Pre-training Model of Natural Language Processing Based on Recurrent Neural Network," 2021 IEEE 4th International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 2021.
- [7] S. Meshram, N. Naik, M. VR, T. More and S. Kharche, "Conversational AI: Chatbots," 2021 International Conference on Intelligent Technologies (CONIT), Hubli, India, 2021.
- [8] Kooli, C. (2023). Chatbots in Education and Research: A Critical Examination of Ethical Implications and Solutions. *Sustainability*, 15(7), 5614. <https://doi.org/10.3390/su15075614>
- [9] Khurana, Diksha & Koli, Aditya & Khatter, Kiran & Singh, Sukhdev. (2022). Natural Language Processing: State of The Art, Current Trends and Challenges. *Multimedia Tools and Applications*. 82. 10.1007/s11042-022-13428-4.
- [10] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.
- [11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- [12] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 9.
- [13] Zhang, Y., Sun, Y., & Zhang, M. (2020). Conversational AI: Dialogue systems, chatbots, and voice assistants. *ACM Computing Surveys*, 53(6), 1-35.
- [14] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 38-45).
- [15] Su, P.-H., Gasic, M., Rojas-Barahona, L. M., Ultes, S., Mrkšić, N., Casanueva, I., Budzianowski, P., & Young, S. (2018). Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 372-379).
- [16] Mehri, S., & Eskenazi, M. (2020). Unsupervised evaluation of interactive dialog with DialoGPT. In *Proceedings of the 21st Annual SIGdial Meeting on Discourse and Dialogue* (pp. 225-235).
- [17] Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Shuster, K., Smith, E. M., et al. (2021). Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.
- [18] Henderson, M., Budzianowski, P., Casanueva, I., Su, P.-H., Mrkšić, N., Wen, T.-H., & Gasic, M. (2018). A repository of conversational datasets. *arXiv preprint arXiv:1810.03526*.
- [19] Serban, I. V., Lowe, R., Henderson, P., Charlin, L., & Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1).
- [20] Gao, J., Galley, M., & Li, L. (2019). Neural approaches to conversational AI. *Foundations and Trends® in Information Retrieval*, 13(2-3), 127-298.

SmartGuide: Chatbot for Efficient Employee Support and Document Analysis

ORIGINALITY REPORT

7%

SIMILARITY INDEX

5%

INTERNET SOURCES

3%

PUBLICATIONS

1%

STUDENT PAPERS

PRIMARY SOURCES

1

P. Kumar, M. Shama, S. Rawat and T. Choudhury.
"Designing and Developing a Chatbot Using Machine
Learning," 2018 International Conference on System Modeling
& Advancement in Research Trends (SMART), Moradabad,
India, 2018

Publication

3%

2

www.asianssr.org

Internet Source

1%

3

Submitted to universititeknologimara

Student Paper

1%

4

www.jetir.org

Internet Source

1%

5

medium.com

Internet Source

1%

6

www.researchgate.net

Internet Source

1%

Exclude quotes On

Exclude matches < 1%

Exclude bibliography On

Review 1 Sheet

✓ Inhouse/ Industry/ Innovation/ Research: Inhouse Class: D17 A/B/C

Sustainable Goal: Industry & Innovation - Goal 9 **Project Evaluation Sheet 2024 - 25** Group No.: 06

Title of Project: Smart Guide: Chatbot for Efficient Employee Support & Document Analysis

Group Members: Shreya Hadkar (D17B-16), Manali Patil (D17B-38), Johan John (D17B-21), Vedang Rathi (D17B-44)

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
3	3	3	1	3	2	1	1	2	2	2	3	2	2	3	33

Comments: Comparison with existing systems.

Dr. Gresha Bhalte
Name & Signature Reviewer 1

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
3	3	3	1	3	2	1	1	2	2	2	3	2	2	3	33

Comments: Good work

Date: 1st March, 2025

Pradya Raut
Name & Signature Reviewer 2

Review 2 Sheet

Inhouse/ Industry/ Innovation/ Research: Industry Innovation Class: D17 A/B/C

Sustainable Goal: Industry Innovation **Project Evaluation Sheet 2024 - 25** Group No.: 6

Title of Project: SmartGuide: ChatBot for Efficient Employee Support & Document Analysis

Group Members: Shreya Hadkar (16), Johan John (21), Manali Patil (38), Vedang Rathi (44)

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
04	04	04	02	04	02	02	02	02	02	02	02	02	02	04	40

Comments:

Dr. Gresha Bhalte
Name & Signature Reviewer 1

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
04	04	04	02	04	02	02	02	02	02	02	02	02	02	04	40

Comments:

Date: 1st April, 2025

Priyanka Ray
Name & Signature Reviewer 2