

Full Length Article

MI Driven Customer Assistant-Task Pilot

Dr M Sandhya Rani¹, Konam Likitha², Emmadi Loka Kalyani³, Ardhham Nidhishaa⁴

¹Hod; Department Of Information Technology Bhoj Reddy Engineering College For Women, Hyderabad India.

^{2,3,4}B.Tech Students; Department Of Information Technology Bhoj Reddy Engineering College For Women, Hyderabad India.

Accepted 29-03-2026

Author(s) Retains the Copyrights of This Article

Abstract

TaskPilot is an intelligent workflow automation platform developed to optimize customer support operations in Business Process Outsourcing (BPO) environments. The system integrates Speech-to-Text conversion, Natural Language Processing (NLP), sentiment analysis, and AI-based response generation to manage customer interactions through both voice and text interfaces. It utilizes a centralized MongoDB database for efficient data retrieval and employs a fine-tuned Gemini model to interpret user intent accurately. Routine queries are handled automatically, while complex or emotionally sensitive cases are escalated to human agents. In addition to automated query handling, TaskPilot offers a unified dashboard for agents, automated follow-up scheduling, a dynamic knowledge base, and a 24/7 self-service portal for customers. The platform also generates analytical insights based on customer interactions, sentiment trends, and agent performance. By combining AI automation with human intervention, TaskPilot enhances efficiency, reduces operational costs, and ensures consistent and scalable customer service delivery in modern BPO systems.

Keywords: AI Automation, Natural Language Processing, Sentiment Analysis, Speech-to-Text, Workflow Management, Customer Support, Chatbot, Gemini Model, MongoDB, BPO Systems

Introduction

The Business Process Outsourcing (BPO) sector plays a crucial role in managing large volumes of customer interactions across industries. With increasing demand for fast, accurate, and round-the-clock service, traditional customer support systems that rely heavily on human agents face several limitations, including delayed response times, inconsistent communication, and high operational costs. To overcome these challenges, organizations are adopting intelligent automation technologies that enhance both efficiency and service quality. TaskPilot is proposed as an AI-driven workflow automation system designed to modernize customer support operations. By integrating advanced technologies such as Natural Language Processing (NLP), sentiment analysis, Speech-to-Text conversion, and intelligent response generation, the system automates repetitive tasks and supports agents with real-time insights. TaskPilot enables seamless communication through voice and text interfaces, provides a centralized dashboard for agents, and supports a continuous self-service environment for customers. The system also analyzes user sentiment to prioritize critical cases, ensuring faster and more accurate responses. This approach not only improves customer satisfaction but also creates a scalable and efficient support framework.

Purpose of the Project

The primary objective of this project is to develop an AI-based workflow automation system that enhances customer support efficiency in BPO environments. The system aims to reduce the burden on human agents by automating repetitive operations such as query resolution, claim tracking, and account management. By leveraging NLP and sentiment analysis, TaskPilot can interpret customer intent, detect emotional context, and deliver accurate responses. Additionally, the system introduces a unified agent interface, automated scheduling, and a self-service portal, thereby improving response time, operational productivity, and overall customer experience.

Existing System

Current BPO customer support systems largely depend on manual processes, where human agents handle all interactions through calls, emails, or chat platforms. Tasks such as verifying customer details, tracking claims, and responding to queries are performed manually, leading to inefficiencies and delays. These systems lack automation and intelligent decision-making capabilities, resulting in increased workload for agents. Furthermore, the absence of sentiment analysis prevents prioritization of critical customer issues. Data is often distributed

across multiple platforms, making it difficult to maintain consistency and coordination in large-scale operations.

Proposed System

TaskPilot introduces an AI-enabled workflow automation solution that streamlines customer support operations. The system processes both voice and text inputs, automatically resolving simple queries while forwarding complex or sensitive cases to human agents. Key features include a dynamic knowledge base, automated follow-up scheduling, real-time sentiment detection, and a centralized agent dashboard. Additionally, the platform provides a 24/7 self-service portal, ensuring continuous customer support. This integrated approach improves service efficiency, accuracy, and scalability.

Related Work

Recent advancements in artificial intelligence and automation have significantly influenced customer support systems. AI-driven solutions incorporating NLP and speech recognition technologies have demonstrated improved interaction quality and reduced reliance on human agents. These systems convert voice inputs into text using automatic speech recognition and generate context-aware responses through advanced language models.

Research indicates that integrating AI with web-based platforms enhances accessibility by providing dashboards, chat interfaces, and interactive user environments. Additionally, sentiment analysis techniques are widely used to classify customer emotions, enabling prioritization of critical cases. Workflow automation studies highlight the importance of centralized data management and intelligent task handling. AI-based systems can efficiently manage repetitive operations, reduce errors, and support decision-making processes. However, existing solutions often struggle with complex emotional interpretation, inconsistent performance, and lack of end-to-end integration. TaskPilot addresses these limitations by combining multiple AI technologies into a unified framework. It integrates Speech-to-Text processing, sentiment analysis, and a fine-tuned language model to deliver accurate and context-aware responses. The system also introduces features such as automated scheduling, real-time prioritization, and a unified dashboard, making it a comprehensive solution for modern BPO operations.

Requirement Analysis

Functional Requirements

The functional requirements of TaskPilot define the core operations necessary to improve workflow automation and enhance customer interaction efficiency in Business Process Outsourcing (BPO) environments. The system is designed to securely

manage client data while supporting both voice and text-based communication through AI-driven mechanisms. It incorporates intelligent query handling, sentiment analysis, and controlled access mechanisms to ensure reliable and secure operations. TaskPilot enables seamless client interaction by allowing users to communicate through voice or text using an AI-enabled interface. The system processes voice inputs by converting them into textual data using Speech-to-Text technology, and responses are generated in both text and speech formats using Text-to-Speech capabilities. It further enhances communication by applying a fine-tuned AI model to interpret user intent and generate accurate responses. In addition, the system performs real-time sentiment analysis to classify customer emotions as positive, neutral, or negative, enabling better prioritization and response handling. A self-service portal is provided for users to independently access information such as claim status and resolve basic queries without agent intervention. All interactions, including chat and voice communications, are recorded and stored systematically for monitoring, analytics, and performance evaluation purposes.

Non-Functional Requirements

Non-functional requirements describe the quality attributes that define how effectively the system performs its operations. These include performance, scalability, security, reliability, usability, and maintainability. Unlike functional requirements, which specify system capabilities, non-functional requirements focus on system behavior under different conditions. TaskPilot is designed to be highly scalable, allowing it to handle increasing numbers of users and requests without compromising performance. The system ensures fast processing of queries and responses, minimizing delays in customer interactions. Security is maintained through strong encryption techniques and role-based access control, protecting sensitive client data from unauthorized access. The system is also built to be reliable, ensuring consistent performance and continuous availability of services. Maintainability is another key aspect, allowing developers to update, debug, and enhance the system without disrupting ongoing operations. Furthermore, TaskPilot supports interoperability by integrating smoothly with external platforms, APIs, and enterprise systems, enabling seamless data exchange and functionality extension.

Computational Resource Requirements

Software Requirements

TaskPilot is developed using modern software technologies to ensure efficiency and scalability. It supports multiple operating systems, including Windows, macOS, and Linux. The backend is built using Node.js and Express.js, while the frontend is developed with React.js to provide an interactive

user interface. MongoDB Atlas is used as the primary database for managing structured and unstructured data efficiently. The system integrates advanced AI technologies such as Natural Language Processing, sentiment analysis, and speech recognition to process user interactions. Voice processing is handled through Speech-to-Text and Text-to-Speech services, while development and debugging are supported using Visual Studio Code. Additionally, decentralized storage mechanisms such as IPFS may be utilized for secure data handling in specific use cases.

Hardware Requirements

To ensure optimal performance, the system requires a computing environment with at least an Intel Core i5 processor or higher, a minimum of 8 GB RAM, and storage capacity of 500 GB or more, preferably using SSD for faster data access and processing.

System Design

System Architecture

The system architecture of TaskPilot represents the high-level structure and workflow of the application. It defines how various components such as user interfaces, AI modules, and databases interact to deliver a unified solution. The architecture is designed to be modular, ensuring that each component operates independently while maintaining seamless integration with other modules. This design approach enables efficient data flow from user input to processing and response generation. It also supports scalability and flexibility, allowing the system to adapt to evolving requirements. The architecture emphasizes reliability and performance while ensuring that all components collectively meet the overall objectives of the system.

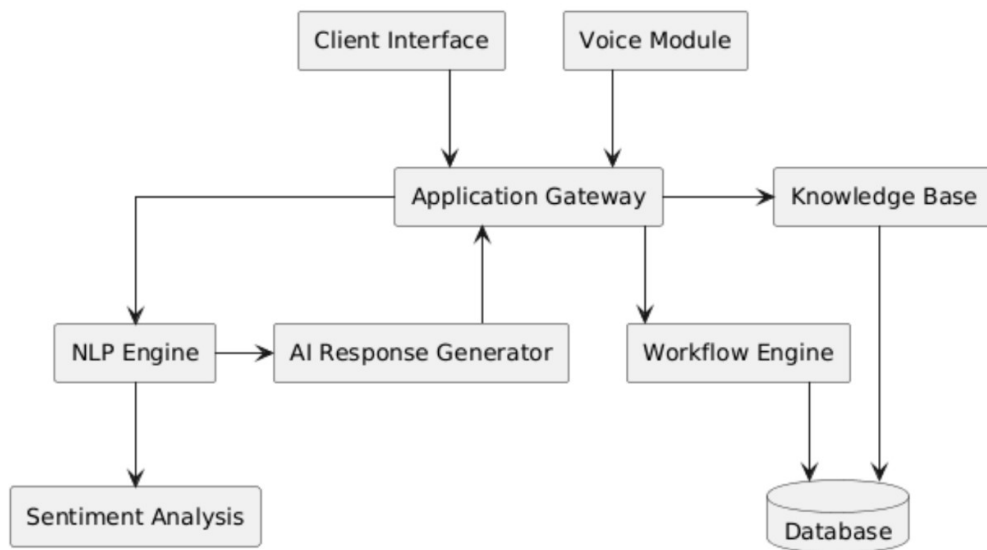


Fig 1; System Architecture

Technical Architecture

The technical architecture provides a detailed view of the system’s infrastructure, including software and hardware components. It illustrates how backend services, frontend interfaces, APIs, and databases are interconnected.

In TaskPilot, the backend is powered by Node.js and Express.js, which manage API requests and business logic. The frontend, built using React.js, handles

user interaction and visualization. AI services such as speech recognition, NLP processing, and response generation are integrated through external APIs. MongoDB serves as the primary data storage system, ensuring efficient data retrieval and management. This layered architecture ensures scalability, maintainability, and efficient system performance.

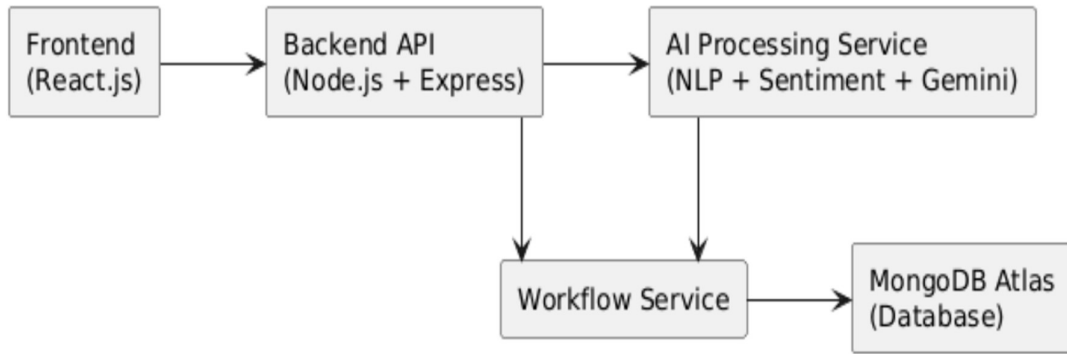


Fig 2; Technical Architecture

UML Diagrams

Unified Modeling Language (UML) diagrams are used to visually represent the structure and behavior of the system. Use case diagrams illustrate interactions between users and the system, highlighting available functionalities. Class diagrams define the static structure, including classes, attributes, and relationships. Sequence diagrams represent the flow of messages between system components over time, while activity diagrams model the workflow and operational processes. Together, these diagrams provide a comprehensive understanding of system functionality and design.

System Modules

TaskPilot is divided into multiple modules, each responsible for a specific function within the system. The Speech-to-Text module converts user voice input into text using advanced neural transcription models, ensuring accurate interpretation of spoken language. The NLP and Sentiment Analysis module processes the text to identify user intent and emotional tone, enabling context-aware responses and prioritization of critical queries.

The Database Interaction module manages storage and retrieval of data using MongoDB, ensuring efficient access to user records, policies, and interaction history. The AI Response Generation module uses a fine-tuned AI model to generate meaningful and contextually relevant responses, especially when predefined answers are unavailable. Additionally, the Chat Management and Logging module maintains a complete record of interactions, supporting system monitoring, analytics, and continuous improvement. These modules work together to deliver a seamless and intelligent workflow automation system.

Implementation

The implementation of TaskPilot utilizes a combination of modern programming frameworks and AI tools. Node.js serves as the runtime environment, enabling efficient handling of concurrent requests through its non-blocking architecture. Express.js is used to build RESTful APIs and manage backend operations effectively.

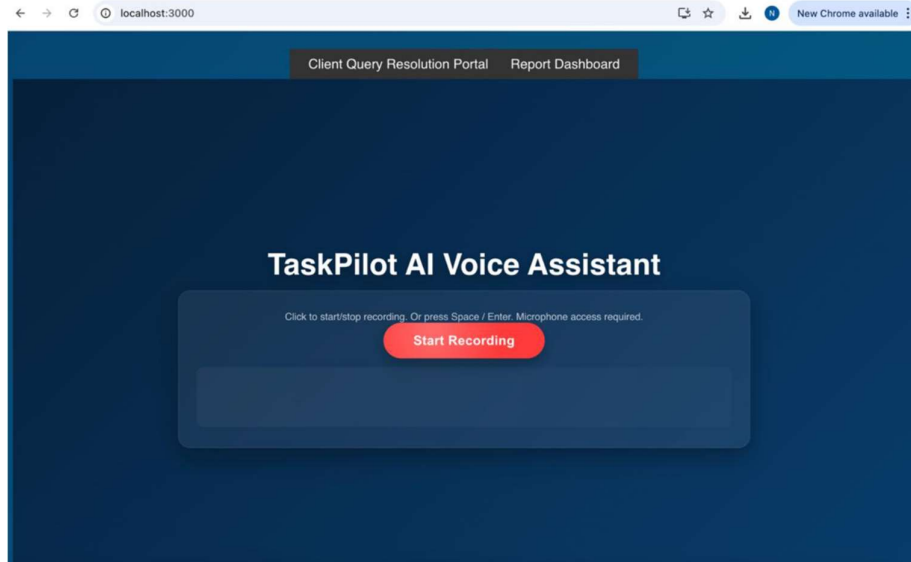
MongoDB is employed as the primary database, offering flexibility and scalability for handling large volumes of structured and unstructured data. The Groq Whisper API is integrated for accurate Speech-to-Text conversion, while the Gemini API is used for generating intelligent responses based on user queries. Natural Language Processing and sentiment analysis libraries are utilized to process text input and classify emotional tone. The frontend is developed using React.js, which provides a responsive and user-friendly interface with reusable components.

Implementation Overview

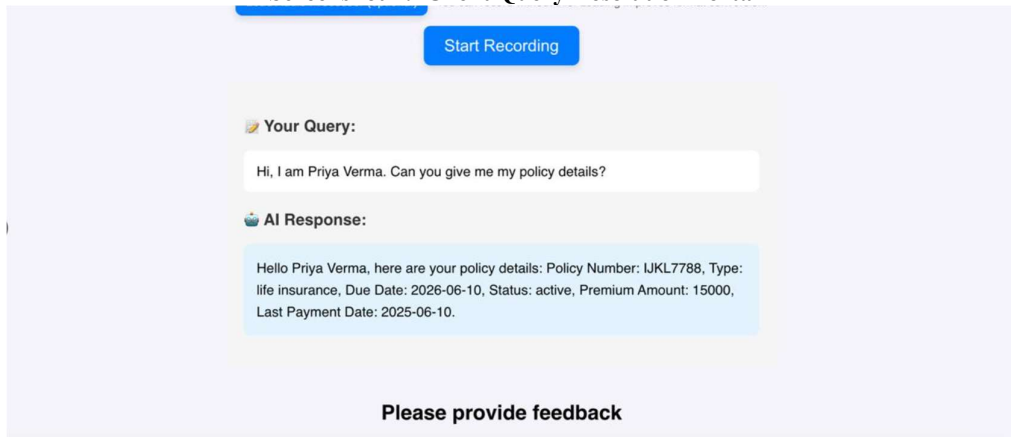
The system processes user input through a structured pipeline. Audio input is first converted into text using a speech recognition service. The extracted text is then analyzed to identify intent and sentiment. Based on this analysis, the system retrieves relevant data from the database or generates responses using an AI model.

If a query matches stored records, the system returns precise information. Otherwise, it generates a context-aware response using AI. All interactions are logged along with sentiment data to support analytics and system improvement. This pipeline ensures efficient handling of customer queries while maintaining accuracy and consistency.

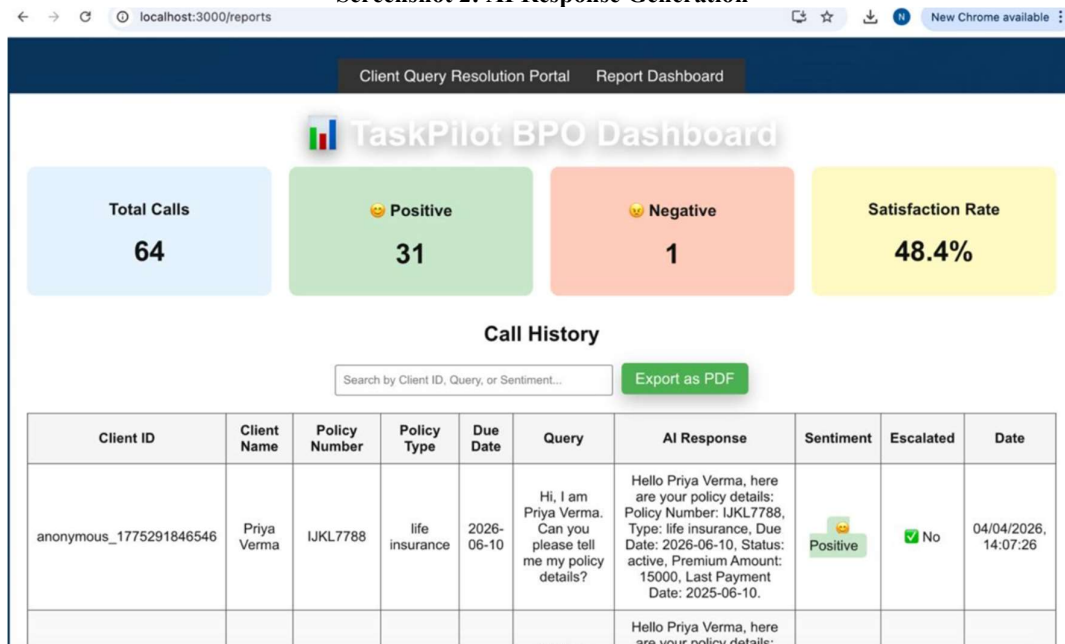
Screenshots



Screenshot 1: Client Query Resolution Portal



Screenshot 2: AI Response Generation



Screenshot 3: Agent Dashboard

TaskPilot Call Reports

Client ID	Client Name	Policy Number	Policy Type	Due Date	Query	AI Response	Sentiment	Date
anonymous_1775289773047	N/A	N/A	N/A	N/A	Thank you....	You're welcome! Glad I could assist. Is there any...	Positive	04/04/2026, 13:32:53
anonymous_1775289739155	N/A	N/A	N/A	N/A	Thank you....	You're welcome! Is there anything else I can help...	Positive	04/04/2026, 13:32:19
anonymous_1775289728962	N/A	N/A	N/A	N/A	Thank you....	You're welcome! Feel free to ask if you have any m...	Positive	04/04/2026, 13:32:08
anonymous_1775289724581	N/A	N/A	N/A	N/A	Thank you....	You're welcome! Glad I could help. Is there anyth...	Positive	04/04/2026, 13:32:04
anonymous_1775289636821	N/A	N/A	N/A	N/A	Thank you....	You're welcome!...	Positive	04/04/2026, 13:30:36
anonymous_1761740256440	N/A	N/A	N/A	N/A	Thank you....	You're welcome! Happy to help....	Positive	29/10/2025, 17:47:36
anonymous_1761738897302	Meena	YZAB7788	life insurance	2025-08-30	Hi, I am	Hello Meena	Positive	29/10/2025,

Screenshot 4: TaskPilot AI Call Report Interface

Test Cases

Testing plays a critical role in ensuring the reliability and accuracy of the TaskPilot system. A series of test cases were designed to validate the functionality of each module involved in the workflow. The system was tested with valid audio inputs to verify successful audio upload and processing. The Speech-to-Text module was evaluated by providing customer voice inputs, and the system successfully generated accurate transcripts. Further testing confirmed that the system could correctly extract key information such as customer names and policy numbers from the transcribed text. The database module was tested to ensure that valid policy details were retrieved when matching records were available. In cases where no matching policy data was found, the AI response generation module

produced meaningful and context-aware responses using the integrated AI model. Additionally, all interactions, including transcripts and generated responses, were successfully stored in the database. Overall, the testing results indicate that the system performs reliably across all functional components.

Validation

Validation ensures that the system meets the required quality standards and behaves as expected under different conditions. The audio upload feature was validated by confirming that valid audio files are accepted and processed correctly. Transcript generation was verified by ensuring that clear audio inputs produce accurate textual outputs.

The system was further validated for its ability to extract essential information such as customer names and policy numbers from user input. Policy

matching functionality was evaluated using similarity thresholds to ensure accurate retrieval of relevant records. In scenarios where no matching data was available, the AI module was validated for generating appropriate responses.

Conclusion

This project presents the successful development of TaskPilot, an AI-powered system designed to automate customer support operations in the insurance domain. The system effectively processes customer voice inputs by converting speech into text, extracting relevant information such as customer identity and policy details, and retrieving corresponding data from a structured database. In situations where predefined data is unavailable, the system utilizes an advanced AI model to generate intelligent and context-aware responses. The integration of sentiment analysis enables the system to understand customer emotions and categorize interactions accordingly. All communication data is stored in a centralized database, supporting future analysis and performance monitoring. By combining artificial intelligence with efficient data management techniques, the system significantly reduces manual effort, improves response time, and enhances the overall accuracy of customer service operations. Furthermore, it provides organizations with valuable insights into customer interactions, enabling better decision-making and service improvement. The results demonstrate that intelligent automation can effectively transform traditional customer support systems into more efficient and scalable solutions.

Future Scope

The proposed system can be further enhanced by incorporating multilingual support, allowing users to interact in multiple languages and improving accessibility for a diverse user base. Integration with real-time cloud telephony services can enable direct handling of live customer calls, making the system more practical for real-world deployment. Future improvements may include advanced sentiment analysis techniques capable of detecting nuanced emotions and urgency levels more accurately. Additional features such as automated premium

reminders, claim tracking, and personalized policy recommendations can be implemented to extend system functionality. Enhancing the analytics can be strengthened by integrating voice-based authentication mechanisms to ensure secure access to sensitive customer information. The system can also be expanded into mobile applications, allowing users to access services conveniently from any location. Training AI models with larger and domain-specific datasets can further improve response quality and accuracy.

References

- [1] F. A. B. M. Roslan and N. B. Ahmad, "The emergence of AI-powered voice assistants and their impact on modern customer service and user expectations," 2023.
- [2] D. Buhalis and I. Moldavska, "Application of artificial intelligence in hospitality: Enhancing customer service through voice assistants," 2022.
- [3] D. Grewal, A. Guha, E. Schweiger, S. Ludwig, and M. Wetzels, "The influence of AI-enabled voice assistant communication on the customer journey," 2022.
- [4] S. Malodia, A. Ferraris, M. Sakashita, A. Dhir, and B. Gavurova, "Evaluating the effectiveness of AI-driven voice assistants in improving customer service interactions," 2023.
- [5] S. Choudhary, N. Kaushik, B. Sivathanu, and N. P. Rana, "Determinants affecting customer adoption of AI-based voice assistant technologies," 2024.
- [6] OpenAI, "Advancements in natural language processing and conversational AI for intelligent response generation," 2024.
- [7] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning-based natural language processing for conversational systems," *IEEE Computational Intelligence Magazine*, 2018.
- [8] B. Liu, "Sentiment analysis and opinion mining in customer interaction systems," *Synthesis Lectures on Human Language Technologies*, 2012.
- [9] MongoDB Inc., "MongoDB Atlas: Cloud database solutions for scalable and real-time applications," 2023.
- [10] A. Radford et al., "Robust speech recognition using large-scale neural networks: Whisper model insights," OpenAI Research, 2023.