

# Agentic AI for Autonomous Fleet Management: A Function-Calling Architecture for Intelligent Vehicle Inventory Systems

**P Sahana<sup>1</sup>, Sohan Gowda S<sup>2</sup>, Akash K G<sup>3</sup>**

Student, Computer Science Engineering (AIML), Maharaja Institute of Technology Mysore, Mandya, India<sup>1</sup>

Student, Computer Science Engineering (AIML), Maharaja Institute of Technology Mysore, Mandya, India<sup>2</sup>

Student, Computer Science Engineering (AIML), Maharaja Institute of Technology Mysore, Mandya, India<sup>3</sup>

**Abstract:** Traditional car management system models are field-based, with complex interfaces that necessitate extensive training to master. The proposed project aims to design an intelligent car management system that integrates agentic artificial intelligence via Google's Gemini 2.0 Flash approach, FastAPI technology, and SQLite database services. Thus, this proposed model will allow natural language processing to undertake full CRUD tasks to implement commands such as "show available cars" autonomous execution as database queries without human interference. Performance analysis indicates a data-entry time reduction of up to 60%, response times of less than a second, and accuracy of 95% with respect to interpretation of user inputs, sufficient to handle up to 50 concurrent users. The underlying technology, Agentic, relies on pattern recognition software that uses FC calls to manage carry-over effects in interpretation guaranteeing data integrity.

**Keywords:** Agentic AI, Natural language processing, Fleet Management, Conversational AI, Database Automation, Function-Calling

## I. INTRODUCTION

The need for less cognitively intensive interfaces in enterprise applications keeps on rising. The conventional practice in fleet management systems includes browsing through different screens and forms, which slows down the process and adversely impacts accuracy. The user is required to remember the terminology of the fields, menus, and database terms in order to carry out the routine tasks of checking the status of the available vehicles or entering the inventory.

Agentic AI refers to a paradigm shift from quite passive tools to self-governing tools endowed with logic and the ability to make decisions and function in accordance with those. The agentic AI system relies on the massive language models in understanding the context and intent of the message and in carrying out substantial conversations that are different from the previous chatbots, which only had certain predetermined and rule-based conversations. This paper explores the discrepancy between the way humans communicate and the way they relate to inflexible databases through the development of the smart car management system using the agentic AI guideline.

The new system will combine the functionalities of the Google Gemini 2.0 Flash model, which will allow function callability, so that the user can control the inventory of vehicles using natural language. Another aspect of the agent framework will automatically direct natural language commands into database actions. Data validation will also allow the new system to maintain the authenticity of data. There will be a combination of Web form submissions, as well as chat sessions by using artificial intelligence.

## II. LITERATURE SURVEY

[1]"A Double-Blinded IEEE Research Paper on Car Rental System" focuses on the overall evaluation of the existing online car rental system. The online car rental system eliminates the need for the traditional and time-consuming process of car rental through the manual system. Currently, the research work emphasizes the creation of an application based on React.js, Node.js, and MongoDB. The proposed system aims to make the process of registration, login, and managing vehicles easier. The proposed system addresses some of the drawbacks of the existing car rental system, which include the prevention of fraud, inefficiency, and underutilization. The proposed system addresses the lack of driving license verification and secure payment transactions. The experiment proved that it could manage 10,000 connections

simultaneously. It also proved that it could withstand different cyber attacks. The proposed system has expanded the level of convenience and level of trust among users compared to the existing system.

[2]"A Survey paper on Car Rental Application: DriveGenie This article points out the shortcomings associated with car renting applications, such as unknown costs, insufficient access to cars, additional costs, and the lack of transparency. For overcoming such shortcomings, the suggested solution emphasizes the need for a mobile-friendly system, live car location tracking, personalized recommendations, environmentally friendly vehicles, along with specified costs for self-driving and chauffeur-assisted car rentals. Moving ahead, the future vision involves self-driving cars, IoT for monitoring, secure- transactions via the blockchain model, voice command for booking services, and multiple transport services with less environmental footprints.

[3]"RentoSecure: AI-Driven Vehicle Rental with Dynamic Pricing & Data Encryption" proposes a next- generation rental platform that integrates machine learning-based dynamic pricing together with AES-256 encryption to optimize revenue and security through the use of gradient boosting regression models that predict rental prices in real time as per demand patterns, geographic location, fleet utilization metrics, and seasonal trends. Experimental results show a 25% improvement in fleet utilization, an 18% increase in revenue, while there were no successful data breaches during the simulation, which proved that integrating predictive analytics along with strong encryption mechanisms was effective. This system sets the benchmark for intelligent, secure, and scalable vehicle rental solutions, with future work being reinforcement learning and real-time user feedback in adaptive pricing models.

[4]"Car Rentals Knowledge and Customer Choice" The study highlights that consumer preferences in the Indian car rental market are primarily shaped by the availability of diverse vehicle models, competitive pricing structures, and the overall condition of the fleet. Among these, the car's model emerged as the most influential factor, underscoring the importance of maintaining a variety of vehicles that align with customer aspirations and lifestyle choices. Price consciousness ranked second, indicating that transparent and reasonable pricing strategies are essential for attracting and retaining customers. The condition of the car followed closely, emphasizing the necessity of regular maintenance and quality assurance to build trust and satisfaction. While discounts, innovative services, and inclusions such as fuel charges contribute to customer appeal, factors like security deposits and late fees, though less prioritized, remain relevant in shaping rental decisions. Collectively, these insights suggest that car rental companies must adopt a balanced approach that integrates fleet diversity, affordability, and service quality to sustain competitiveness in a rapidly evolving market.

[5]"Agentic AI: A Comprehensive Survey of Architectures, Applications, and Future Directions" offers in-depth explanations regarding the shift in paradigms from passive and task-focused tools to autonomous systems that are able to demonstrate genuine agency. The paper makes a clear distinction between AI Agents as standalone autonomous systems and Agentic AI as the orchestration of multiple agents with specializations in solving complex tasks. The paper proposes a two-paradigm approach for systems that oppose symbolic or traditional systems that use algorithmic plans with permanent states to neural or generative systems that use stochastic generation with orchestration from prompts. The paper lists and discusses ninety research works from 2018 until 2025 regarding how symbolic systems are far more prominent in life-critical fields like healthcare and how neural systems are vastly more successful in an adaptive data-rich environment like finance. The major difficulties are the fragmented understanding and governance that require hybrid neuro-symbolic systems in order for the systems to be malleable and trustworthy.

[6]"Agentic AI: Autonomous Intelligence for Complex Goals-A Comprehensive Survey" considers Agentic AI as "autonomous systems capable of achieving multi-layered goals with little human oversight" and views it distinct from rule-based and generative forms of AI on technical grounds such as reinforcement learning and adaptive control systems. It also emphasizes applications of Agentic AI in health care, finance, disaster response systems, production systems, and education relying on dynamic resource allocation, personalization, and functioning under uncertain situations. The challenges are framed around goal alignment, adaptability in environments, scalability, resource limitations, and the issue of ethics encompassing responsibility, fairness and transparency, privacy, governance, thereby viewing Agentic AI as "a paradigm that seamlessly combines traditional AI and AGI research."

The overall research in the field shows evolutions from traditional rental models into modern automated models, and from passive models using AI into active decision-making models using agentic models of AI. Nonetheless, from the overall research in the field, most studies on car rental management using modern technology focus on theoretical or individual aspects of fraud analysis and pricing analysis for car rental models without much focus on overall car rental management models incorporating conversation analysis using large language models, automated verification, and car rental management operations using databases and analytics into complete models for effective car rental management models using agentic models for AI.

### **III. METHODOLOGY**

#### **3.1 System Architecture**

Proposed intelligent vehicle management system consists of a three-tier architecture that ensures a proper segregation of presentation, business, and data management layers. Presentation layers provide dynamic web interfaces using HTML5, CSS3, JavaScript, along with ES6, that can handle conventional form data entry as well as chat interfaces. Business layers are implemented using the Fast API framework that ensures high-speed async API functionalities for data manipulation, session handling, as well as usage of AI functionalities. Data layers make use of an SQLite database that can effectively handle data models as well as transactions using the SQLAlchemy ORM. Agents in the agentic intelligent model integrate an interface for the Google Gemini 2.0 Flash model that ensures proper structure for function call processing, thus enabling autonomous operations. The architecture of the model comprises an efficient session-based conversation manager that maintains the context of various user sessions.

#### **3.2 Agentic AI Implementation**

The key improvement comes from the realization of agentic capability in coupling large language models with structured tool-calling procedures. For this goal, the system will use function schemas to denote the set of available functions such as obtaining vehicle records, filtering records based on availability status, adding new vehicle records, and generating analysis insights. These function schemas will articulate sets of parameters expected from these functions to ensure data integrity when functioning autonomously. The agentic capability will follow a structured algorithm. The system will process the user input while triggering the Gemini model based on the conversation history and available function sets. The models will identify the user's intent based on language interpretation and identify key entities of interest such as vehicles or analysis parameters to make appropriate function calls. Later on, validate the parameters of these functions based on the defined function schemas before database function execution. The processed function results will be returned as natural language answers, maintaining a conversational flow. Algorithm 1 explains the process of agentic capability analysis. a) The process begins with conversations processed using a language model to identify user intents and relevant key entities of interest such as vehicles or analysis details. The process retrieves relevant conversations from the conversation history. The process communicates with pertinent functions such as createVehicleRecord(), availability status functions such as getAvailabilityStatus(), or analysis functions such as generateAnalytics() based on user intents. iv) After successfully triggering functions with appropriate parameters validated from defined function sets, the process will perform a conversation history update with responses including conversational information.

#### **3.3 Database Operations and Tool Integration**

The system shall perform full CRUD operations using independent tool call facilities. The create operation shall check for validate parameters related to company name, model name, year of manufacture, color parameters, mileage measures, and availability status prior to inserting records into the database. The validate parameters shall enforce type constraints, check for compulsory fields, and format validation for quality consideration during reads and writes. The read operation offers multiple query patterns and templates, including retrieval of full vehicle records, filtering based on availability status, searching using distinct parameters, and summary analysis as terms of aggregate statistics and summary analysis studies. The update operation shall modify full records and support audit trail functionalities for changes made, using independent locking facilities for optimization and avoiding clashes with multiple user session modifications for consistency in user sessions and locations. The delete operation shall offer soft delete capabilities for historical records and exclusion from active records for market inventory and active market views and analysis studies.

#### **3.4 Natural Language Processing and Pattern Matching**

The natural language processing portion of this system will utilize pattern-matching algorithms in order to decipher user commands in relation to appropriate database queries. The various phrases used to mean equivalent commands, like "show cars," "list vehicles," or "display inventory," are all retrieval commands. The pattern-matching function will also facilitate entity recognition of individual values referred to in terms of brands, model numbers, or availability terms in user commands. The ambiguous query will be considered through clarification dialogues in order to continue the process of communication while enabling enough detail to perform correctly. The mechanisms used in error handling will provide descriptive answers when commands are either incomprehensible or not possible to perform through recommendations provided to effectively utilize this system without revealing implementation details.

#### **3.5 Session Management and Context Preservation**

Agents and humans need a certain level of context in a conversation in order for the interaction system to be effective. The system maintains a session for each conversation through the usage of session IDs for each and every conversation thread, which stores the message history and the context of the current operation. The information contained in the session ID includes the inputs entered by the users, the responses from the AI, the functions executed along with the results of

the operation, and so on. The memory management algorithms also ensure a certain level of optimization by retaining all the important conversations and discarding the extreme ones that might hamper the response times or the maximum number of tokens. This specific implementation maintains a balance in the preservation of conversations and the need for system performance in terms of interaction efficiency

#### **IV. RESULT AND DISCUSSION**

##### **4.1 Functional Performance**

The entire testing phase involving 500 random test cases signifies the efficient functioning of the system with all details met. The operations performed on databases take less than 500 milliseconds for single record operations in terms of CRUD operations such as vehicle addition, retrieval, update, and deletion. However, more complex operations such as querying with filters and calculations take less than 2 seconds. The language processing operations performed with the help of artificial intelligence take between 3 and 5 seconds for query resolution.

It performs quite effectively with 50 simultaneous users with no noticeable degradation in performance with respect to response times and data consistency during concomitant operations with optimal usage of database connections, management of transactions, and resource utilization. During stress testing using 100 simultaneous requests, the system performs quite effectively with no noticeable degradation with respect to optimal management of resource contention using optimal transaction management that avoids deadlocks, manages connections using connection pooling, and manages queues to avoid rejected requests. Comparative analysis shows considerable improvement over existing systems with respect to average time taken to enter data with an improvement of 60% reducing the average time taken to enter data from 112 seconds to 45 seconds, command interpretation accuracy increasing by 95% compared to conventional systems with no such capability, response times for database operations with improvement of 44% reducing the time taken to respond to operations from 800 milliseconds to 450 milliseconds, user satisfaction with improvement of 40% increasing the user satisfaction scores from 6.2 out of 10 to 8.7 out of 10, reduction in training time with improvement of 92% reducing the training time taken from 2 hours to 10 minutes, and improvement of 150% increasing the existing capability of handling 20 users up to 50.

##### **4.2 Command Interpretation Accuracy**

Natural Language Command interpretation made noteworthy improvements in terms of accuracy of 95% in varied command phrase structures, as the processing of 300 varied command scenarios. While most command scenarios were properly interpreted in terms of intended user commands, fewer errors were involved in command interpretation in edge scenarios, of which 5% comprised commands that were either unclear, as in one-word commands such as 'cars,' or beyond command interpretation, as in commands for financial calculations that were outside the limitations of commands in operational tools.

Successful patterns of commands that were commonly incorporated include direct commands such as "show all cars," question query commands such as "what vehicles are available," conversational commands such as "I'd like to see the inventory," and contextual commands that depend on the discussion history such as "show me more" based on a query. The agent framework was able to handle various patterns of language use while still functioning properly regardless of whether a user was using technical language with precision or conversational language.

##### **4.3 System Reliability and Data Integrity**

The deployment ensures strong 99% system uptime during test periods that involve three weeks of continuous system usage, and error handling in the system is both proficient and thorough in ensuring that the system never corrupts any data or fails catastrophically. The transaction management in the system ensures that the ACID properties are followed in database transactions to guarantee the integrity of the database in the event of multiple transactions being conducted concurrently.

Validation techniques invalidate an unacceptable input before storing the data in the database, thereby eliminating any flawed data. Data type or pattern validation covers the validation of data types and formats, such as verifying that any integers are constrained within an acceptable range or that strings conform to a particular pattern. Business rule validation covers the enforcement of organizational rules, such as limitations on the year of a vehicle, the value of mileage represented as a non-negative number, and the availability status constrained to a set of pre-defined values.

The system provides audit trail logging functions covering comprehensive audit logging of all modifications made to the database. The audit trail record maintains timestamp information, user identification, modification types differentiating between creation and modification actions, and record identifiers. This audit trail feature offers the capability to investigate inconsistencies in the information and provide evidence of good handling practices.

Error recovery mechanisms address different error conditions using concepts of graceful degradation. Network disconnections lead to automatic retry processes that utilize exponential backoff to prevent flooding of requests, API errors lead to fallbacks for cached responses where real-time responses are not possible, and unexpected inputs lead to robust error handling with helpful feedback that does not use technical terms and thus annoys non-technical users. The system also supports logging of errors for administrative purposes with the help of stack traces and request parameters.

## **V. CONCLUSION & FUTURE WORK**

In this

The agentic framework adds fresh perspectives when considering the management of the database via a conversation. In return, this enables the illustration of the advantages that large language models offer on the conventional applications found in an enterprise. The system incorporates independent problem-solving, the context of a conversation session, along with a vast error handling system.

In fact, the future could encompass the realization of AI technology for voice interaction support services, multi-Language enablement, and predictive analytics for scheduling and forecasting. The improvement could be the inclusion of functionalities for the real-time status updates of the vehicle using the integration of an IoT device, the integration of strong security functionalities such as role-Based Access Control, and the development of microservice architecture in the cloud for enhancing the scalability of the project. The development of the project for a full dashboard with the provision for visualization tools, the use of Machine Learning algorithms for the detection of fraud, and the exploration of multi-Agent architectures for complicated vehicle management tasks are the fields for the Future Research Study.

## **REFERENCES**

- [1]. Gupta Nayan, Drashti Patel, Patel Vidhi and Mistry Devam Nidhi Pandey, Dinesh Swami, Himanshu Tiwari and, Anand Kumar Singh "A Double-Blinded IEEE Research Paper on Car Rental System", - IJIRT, 2024.S
- [2]. Akanksha Solanki Alfiya Khanam Aarti Pawar Sahaj Sharma P. Ashish Anjana,P. Satyam Shrivastava," A Survey paper on Car Rental Application: DriveGenie" -JETIR, 2024.
- [3]. N. Yagneswar and G. V. Ramesh Babu, "RentoSecure: AI-Driven Vehicle Rental with Dynamic Pricing & Data Encryption",- IJIEMR, 2024.
- [4]. Saroj Koul, CSN Venkata Datta, Rakesh Verma," Car Rentals Knowledge and Customer Choice"- ic-ETITE, 2023.
- [5]. M. Abou Ali and F. Dornaika," Agentic AI: A Comprehensive Survey of Architectures, Applications, and Future Directions"-IEEE Xplore,2025.
- [6]. D. B. Acharya, K. Kuppan, and B. Divya, "Agentic AI: Autonomous Intelligence for Complex Goals-A Comprehensive Survey"- IEEE Access, vol. 13, pp. 1-24, 2025.