Journal Homepage: www.ijarpr.com ISSN: 3049-0103 (Online)

International Journal of Advance Research Publication and Reviews

Vol 02, Issue 09, pp 278-285, September 2025

Autonomous Self-Curating Animal Welfare Management System: Integrating MERN Stack with Retrieval-Augmented Agents

Likhitha AJ [1]***, Dr. Gowthami V*** [2]

[1] *MCA Student, CMR University, Bengaluru,* ***likhitha.aj@cmr.edu.in***
[2] *Assistant Professor, CMR University, Bengaluru,* ***gowthami.v@cmr.edu.in***

ABSTRACT

As large language models (LLMs) become more common in everyday applications, interest is rising in making them more helpful and aware of context. Animal welfare work is chaotic and uncertain. Shelters, rescue organizations, and NGOs are forever playing a balancing act of receiving rescue calls, adoption inquiries, medical interventions, and legal documents. The systems rely on manual updates, so that by the time information is input, it is already static. This delay can be worth valuable time, and in animal welfare, time can be the difference between life or death. This study presents an Autonomous Self-Curating Animal Welfare Management System intended to bridge this knowledge gap. Developed using the MERN stack (MongoDB, Express.js, React, Node.js) and fueled by Retrieval-Augmented Generation (RAG) with a vector database, the system learns and updates itself continuously. It scans and authenticates information independently from trusted veterinary sites, government welfare organizations, legal notifications, weather warnings, and NGO regulations and integrates it with a shelter's own rescue and adoption history. An Intelligent agent loop keeps the knowledge base up to date, and a user-friendly dashboard enables staff to coordinate rescues, track animal health, monitor adoption progress, and receive instant updates on critical developments like outbreaks or changes in the law. The system, with the combination of automation and domain expertise, hopes to equip animal welfare teams with the tools they require to make quicker, wiser, and more empathetic decisions.

Keywords: *Animal Welfare, Retrieval-Augmented Generation, Autonomous Agents, Vector Database, MERN Stack, Rescue Management, Knowledge Automation*

Introduction

Each day, animal welfare staff encounter a torrent of tasks from fielding high-priority rescue calls to coordinating foster care, handling adoptions, monitoring medical procedures, and keeping up with evolving legislation. In such a dynamic setting, information is power. But in much of the shelter and NGO, the information systems rely on manual entries and disconnected records. By the time that little bit of information makes it into the system, it is likely already outdated.This disconnect between what is known and what is required can result in lost opportunity, delayed treatment, and even avoidable death. A rescue team may reach a site unaware that the shelter is full. A vet may initiate treatment without the current disease outbreakalert. A volunteer coordinator may overlook a drastic change in adoption policy. These are not simply technical inefficiencies in animal welfare, they have a direct effect on the welfare of living entities. Retrieval-Augmented Generation (RAG) presents the possibility of filling this gap. With the integration of a retrieval engine that retrieves relevant information and a large language model capable of understanding and creating senseful answers, RAG provides the potential for context-sensitive, real-time responses. Most current RAG systems in other areas, however, still use static databases, which means their utility wanes unless a human updates them by hand. This study suggests an Autonomous Self-Curating Animal Welfare Management System, developed using the MERN stack and vector database, which is capable of repeatedly collecting, authenticating, and incorporating new data without manual intervention. A smart agent loop monitors reputable veterinary portals, government notifications, NGO newsletters, and shelter records to update the system's knowledge base and keep it fresh and updated. From a safe, intuitive dashboard, volunteers and employees can track rescues, monitor medical updates, monitor adoptions, and even get proactive notifications when something happens quickly like an upcoming heatwave that could put animals at risk, or a new vaccination policy for adopters. By combining automation with compassion, this system seeks to liberate animal welfare workers from data overload so they can concentrate on what counts the most: saving lives and enhancing animal welfare.

Literature Review

Piccialli et al. (2024) explain AgentAI as a new perspective on autonomous systems — not as tools alone, but as intelligent agents that are capable of learning, making decisions, and coordinating in real-time. Their survey indicates how these types of agents make industries more scalable, adaptable, and resilient, going beyond hard automation. Most importantly, they mention the move towards human-oriented AI in Industry 5.0 and 6.0, wherein the agents do not replace humans but assist them. This concept immensely relates to animal welfare, wherein technology needs to complement human empathy by limiting the information gaps and facilitating quicker, more accurate decisions for rescue and care.

Alvarez et.al (2024) follow the evolution of AI agents from basic rule-based agents to modern, complex, self-sufficient multi-agent models. They point out how advancements in deep learning, reinforcement learning, and transformer models have allowed agents to plan, learn, and act alone in various industries. The article emphasizes opportunities such as efficiency improvements, automation, and cooperation as well as risks like misalignment, lack of transparency, and ethical issues. Above all, authors urge strong governance and ethical policies to ensure that AI agents develop responsibly. This view is supported by the necessity for vigilant care in areas such as animal welfare, where autonomous systems are necessitated to augment, rather than substitute, human compassion and judgment.

Joshi (2025) examines the use of Generative AI agents in financial domains, classifying their application across financial risk management, investment, fraud detection, stock analysis, and customer service. The survey indicates tangible advantages such as 25% increased risk accuracy, 20% reduced loan defaults, and 40% reduced false fraud detection, demonstrating the evident worth of agent-based systems. Meanwhile, it specifies ongoing deficits in scalability, interpretability, and flexibility that call for future research in hybrid models and ethics. Although this paper is in finance, its implications regarding decision correctness, risk minimization, and building trust are immediately transferable to sensitive areas like animal welfare, where system dependability and ethical protections are similarly essential.

Petrović (2018) examines the convergence of virtual worlds and artificial intelligence, and contends that virtual worlds represent a rich source of testbeds for the creation of human-level AI agents. The article points out how virtual worlds enable researchers to test and validate memory, planning, and adaptive decision-making in rich, dynamic environments. Simultaneously, AI agents introduce more realism and interactivity to virtual worlds, and induce a reinforcing loop of advancement. Petrović highlights that progress in AI agents in virtual worlds can be applied to real-world applications, where agents have to function under uncertainty and interact with people normally a lesson just as applicable to fields such as animal welfare management, where agents assist human-centered decision-making.

Problem Statement

- Outdated Information: Animal welfare agencies have static, outdated systems that involve slow, manual data entry. This means the information they utilize to make important decisions is already outdated by the time they receive it.

- Severe Consequences: They usually have disastrous and occasionally fatal consequences. Misleading data regarding shelter capacity could result in overcrowding or under-employed rescue operations, whereas delayed medical notifications could enable infectious sicknesses to spread at an alarming rate within shelters..

- The Technology Gap: Despite new technologies such as Retrieval-Augmented Generation (RAG) systems, the fundamental issue persists. Most of them still require an individual to update their knowledge bases by hand, thus rendering them quickly obsolete should human support be absent.

- The Human Cost: In this quick-moving industry, an innocent information gap can mean the difference between an animal getting a safe home and being left behind. This puts a great deal of pressure on committed staff who must work with incomplete or late information.

Objectives

This study is motivated by

• Automate Knowledge Updates – Pull and process data on an ongoing basis from

reliable veterinary, governmental, and NGO sources automatically.

• Integrate Internal and External Data – Pull rescue, adoption, and medical histories along

with real-time external updates onto one platform.

• Enable Semantic Search – Permit staff to ask questions of information in natural

language and get contextually relevant responses.

• Offer Proactive Alerts – Alert users to breaking developments like disease outbreaks,

severe weather, or legal updates.

• Maintain Data Trust and Security – Authenticate data prior to storing and store sensitive

records securely with role-based access.

• Facilitate Better Decision-Making – Enable quicker, more accurate, and empathetic

actions for animal rescue and welfare operations.

Algorithm and Implementation

Input

Internal shelter information: rescue records, adoption status, medical history, shelter capacity (stored in MongoDB).

Trusted external sources: veterinary health websites, government animal welfare ministries, legal notices, NGO bulletins, weather warnings.

User queries in natural language.

Process:

1. Periodic Data Collection (Agent Loop)

a. Discover New Information

Crawl or retrieve data from each trusted source (e.g., RSS feeds, APIs, official web sites).

b. Pre-process Text

Clean, filter, and divide content into smaller overlapping pieces for more effective retrieval.

c. Verify & Validate

Compare new information from a variety of sources to confirm accuracy.

d. Generate Embeddings

Map every chunk of text to an embedding vector with an embedding model.

e. Store in Vector Database

Store embeddings along with metadata (source, timestamp, type of data).

2. Handling User Queries

a. Convert Query to Embedding

Convert the user's natural language query to a vector form.

b. Retrieve Relevant Chunks

Look for top-k semantically closest embeddings in the vector database.

c. Combine Context

Join pulled chunks with corresponding internal records (e.g., animal health, adoption history).

d. Generate Answer

Pass combined context and the query to the LLM for a context-sensitive, domain-specific answer.

e. Display Result

Return generated answer through the web dashboard in a readable form.

Output:

Concurrently updated vector database of verified, pertinent knowledge. Instantaneous, context-rich answers to animal welfare questions.

Proactive notices of key developments influencing animal safety and welfare.

Technical Background

1. Vector Database

A vector database is a special kind of data store that manages high-dimensional numerical representations, called embeddings. In natural language tasks, text gets converted into these embeddings using a neural model. This way, similar meanings are represented by points that are close together in vector space.

Unlike traditional keyword-based databases, vector databases support semantic search. They find information based on meaning similarity, rather than just matching exact words. Popular open-source vector databases include FAISS and Chroma, which allow for quick similarity searches at a large scale.

Here, the vector database stores,

- The embeddings of user data, such as tasks and notes.
- The embeddings of external knowledge pieces collected by the autonomous agent.

When a user asks a question, its embedding gets compared to the stored vectors to find relevant context.

2. Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG) is a strong technique that brings together two great strengths of AI: the capability to retrieve the correct information and the capability to generate explanations in natural language. Fundamentally, RAG consists of two components in tandem:

Retriever – This part rummages through a virtual or actual knowledge database to find information relevant to the question of the user. In contrast to keyword searching, it makes use of semantic search (meaning-based), enabling it to discover information that actually corresponds with the purpose of the query.

Generator – After the retriever has identified potentially useful information, the generator typically an LLM — uses both the user's question and the retrieved context to write a clear, well-informed answer.

This collaboration addresses one of LLM's largest issues: although they're great at producing natural language, they only know what they were trained on, which becomes stale very fast. When retrieval is added, the model can have access to new, authoritative information at any time. Retrieval-Augmented Generation (RAG) is a strong technique that brings together two great strengths of AI: the capability to retrieve the correct information and the capability to generate explanations in natural language. Fundamentally, RAG consists of two components in tandem:

3. Mern Stack

The MERN stack is among the most widely used full-stack JavaScript frameworks since it enables developers to create end-to-end applications with a single programming language: JavaScript. It combines four strong technologies that complement each other:

MongoDB – A NoSQL database that will save data in a flexible, JSON-like structure. Rather than strict tables, it stores data like animal profiles, medical records, or adoption files in a structured yet flexible manner.

Express.js – A light-weight server framework that makes it easy to develop APIs and process server requests. It makes sure data moves efficiently across the user interface, the database, and the AI features executed in the background.

React – Frontend library employed to create dynamic and intuitive dashboards. For animal welfare staff and volunteers, this translates to simple-to-use interfaces for viewing rescue updates, adoptions, or receiving notifications — all in real-time.

Node.js – A light and efficient runtime that enables JavaScript to be run on the server side. It drives the backend operations, such as the agent loop responsible for continuously retrieving and refreshing data from reliable sources.

Combined, these technologies form an unbroken space where data, logic, and user interface merge together. For an animal welfare management system, the MERN stack guarantees that the platform is not only technologically robust but also responsive, secure, and user-friendly for users in the field every day.

System workflow

Within the Autonomous Self-Curating Animal Welfare Management System proposed, the whole application is constructed using the MERN stack, which makes it possible to integrate the frontend, backend, and database layers smoothly. This makes data move effortlessly throughout the system without delays or complications.

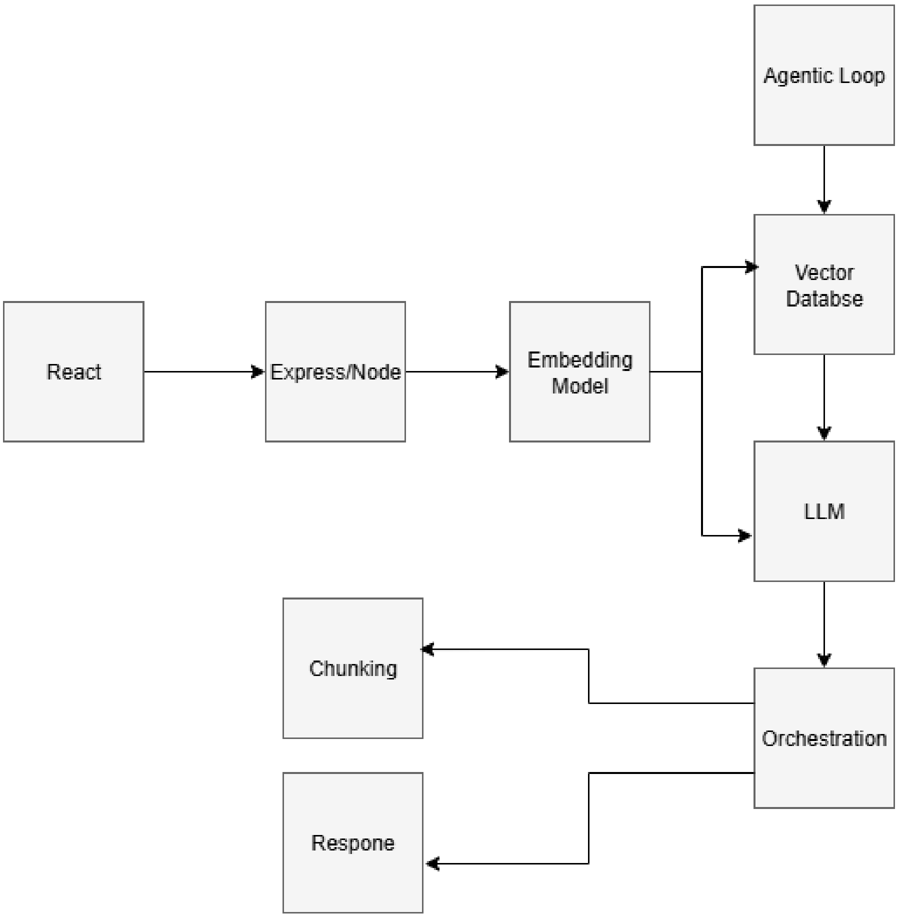

Fig: *MERN-based Retrieval-Augmented Generation (RAG) pipeline with an autonomous Agentic Loop.*

At the front-end level, React drives a streamlined and engaging dashboard. Via the dashboard, volunteers and staff can comfortably control rescue cases, maintain current adoptions status, view animal healthcare records, and access timely notifications. The intent is to develop an interface that is easy to operate and efficient in facilitating day-to-day welfare activities.

Nodejs, Expressjs handle backend. They manage API and workflow condition. Among their most critical jobs is executing the agent loop, which fetches and handles fresh data from trusted external sources automatically, minimizing the necessity for endless manual updates.

All primary organizational information such as animal profiles, history of rescue, adoption information, medical interventions, and shelter capacity are safely housed in MongoDB. MongoDB handles large amount of structured data.

To enable more intelligent searching, a vector database is used in conjunction with MongoDB. Rather than keyword-level matching, it caches semantic embeddings of internal data and external knowledge, which enables the system to make meaning-based searches that yield the most relevant information for a query very quickly.

These pieces together create an integrated system that maintains information in sync, available, and updated in real time, which directly contributes to accelerating and empathetic decision-making in animal welfare management.

The RAG Pipeline integrates all the pieces:

1. Once The RAG pipeline combines all the elements of the system into a seamless process. When a user queries something say, "What animals are due for vaccination this month?" the backend translates the question first into an embedding, a numerical form that embodies its meaning.

2. Then, the vector database scrolls through both internal records of shelter and external sources of knowledge and discovers the most relevant information. Rather than depending on keyword exactness, it seeks content that is similar in the intention of the query so that the results are intelligible.

3. After the most appropriate pieces of information are fetched, they are used to feed the language model (LLM). The model conjoins this background with the initial query to produce an answer that is not only correct but also up-to-date and specific to the needs of the user.

4. Lastly, the response is presented on the React dashboard in a clear, readable format so that volunteers and staff can take quick action on the information with no technical input.

5. This process is supported by the Agentic Loop, which operates continuously on Node.js. This loop continuously scans trusted sources like veterinary websites, government bulletins, NGO reports, and weather alerts. It cleans and processes data, generates embeddings, and refreshes the vector database. Therefore, the system remains current and trustworthy without needing to be manually refreshed, so that animal welfare workers have up-to-date information at their fingertips at all times.

The Agentic Loop, executed on Node.js, periodically crawls trusted veterinary websites, government animal welfare news, NGO reports, and weather warnings. It curates the data, creates embeddings, and updates the vector database — keeping the system's responses up-to- date, precise, and in harmony with animal welfare demands without the need for manual refreshes.

Conclusion and Future Enhancement

Animal welfare operations are dependent on pace, accuracy, and compassion—qualities that are greatly improved if the proper information is presented at the appropriate moment. In this article, an Autonomous Self-Curating Animal Welfare Management System that incorporates the MERN stack, Retrieval-Augmented Generation (RAG), a vector database, and an intelligent agent loop was presented to solve the endemic problem of stale or broken information in rescue and shelter operations.By autonomous gathering and verification of information from reliable veterinary, government, and NGO sources, and correlating it with internal adoption and rescue data, the system creates a dynamic pool of knowledge that is updated constantly. This enables animal welfare staff and volunteers to make faster, better-informed decisions, receive proactive alerts, and automate routine tasks without the necessity of repeated manual intervention.

Going forward, some important upgrades are under consideration to advance the system's functionality. For example, predictive analytics would enable future requirements to be predicted, such as an outbreak of specific seasonal sicknesses, so organizations can stock up on medicine and vaccines in advance. Location-based coordination might also be used to automatically match a rescue case with the closest available shelter, foster home, or animal clinic. Lastly, a mobile application would provide field workers with the means to update records and receive critical alerts on their cell phones in real time, thereby providing real-time coordination.

References

1. Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. The Knowledge Engineering Review, 10(2), 115–152.

2. Rudowsky, I. (2004). Intelligent agents. Communications of the Association for Information Systems, 14(1), 14.

3. Petrović, V. M. (2018). Artificial intelligence and virtual worlds – toward human-level AI agents. IEEE Access, 6, 39976–39988.

4. Piccialli, F., Chiaro, D., Sarwar, S., Cerciello, D., Qi, P., & Mele, V. (2024). AgentAI: A comprehensive survey on autonomous agents in distributed AI for Industry 4.0. [Journal/Publisher details depending on where published]

5. Alvarez, F., & Jurgens, J. (2024). Navigating the AI Frontier: A Primer on the Evolution and Impact of AI Agents. Capgemini & World Economic Forum White Paper, December 2024.