JOURNALS | Journal of Collective Sciences and Sustainability

Volume 2 Issue 3 (September 2026)
J. Collect. Sci. Sustain., 2026, 2(3), 26409
DOI : https://doi.org/10.64189/css.26409

Research Article

OPEN ACCESS

RescueTails: A MERN Stack-based Web Platform for Centralized Stray Animal Rescue Coordination

Zehra Shaikh,* Nabiha Shaikh, Hamsa Naik, Maryam Lokhandwala and Chaitali Mahajan

Department of Computer Science & Engineering (AIML), M.H. Saboo Siddik College of Engineering, Mumbai, Maharashtra, 400008, India

*Corresponding Author
Zehra Shaikh
zehra.231757.cs@mhssce.ac.in

Received: 18 July 2026
Revised: 05 September 2026
Accepted: 16 September 2026
Published Online: 25 September 2026

Citation

Z. Shaikh, N. Shaikh, H. Naik, M. Lokhandwala, C. Mahajan, RescueTails: A MERN stack-based web platform for centralized stray animal rescue coordination, *Journal of Collective Sciences and Sustainability*, 2026, 2(3), 26409, https://doi.org/10.64189/css.26409.

Abstract

Stray animals in urban environments frequently suffer from injuries, abandonment, disease, and a lack of timely medical intervention. Despite growing public willingness to assist, the absence of a centralized digital system to connect concerned citizens with nearby nongovernmental organizations (NGOs), rescue teams, veterinary services, and foster homes results in fragmented and delayed rescue operations. In this paper, RescueTails, a full-stack web application built on the MERN (MongoDB, Express.js, React.js, Node.js) technology stack, is designed to serve as a centralized platform for stray animal rescue coordination. The system enables rapid online reporting of animals in distress, structured communication between stakeholders, streamlined adoption workflows, and structured resource allocation. The platform implements role-based access control, a RESTful API architecture, and a responsive user interface to ensure multidevice accessibility. Evaluation through functional verification and JMeter-based benchmark load testing provides evidence of stable API performance under the evaluated load conditions, subsecond API latency, and operational feasibility for centralized animal rescue coordination. RescueTails contributes to social-impact technology by providing a solution designed to support scalable animal rescue coordination by addressing an identified gap in urban animal welfare software infrastructure.

Keywords: Stray Animal Rescue; MERN Stack; Web Application; Animal Welfare; Rescue Coordination; Adoption Platform; RESTful API; Social Impact Technology.

1. Introduction

The global stray animal population constitutes a notable public health, ecological, and humanitarian challenge. World Health Organization (WHO) reports estimate hundreds of millions of stray dogs globally, with tens of millions present in urban environments across developing nations. These animals face frequent traffic injuries, infectious diseases, malnutrition, abuse, and environmental hazards. Despite increasing public awareness, the lack of structured, technology-driven coordination mechanisms often hampers the efficiency of rescue and rehabilitation efforts. [1,2] Current animal rescue operations in many urban regions rely heavily on informal communication channels such as social media posts, phone calls to local shelters, and word-of-mouth referrals creating challenges for structured reporting and coordination. [1,2] This ad-hoc approach introduces operational inefficiencies: delayed response times, duplication of effort, loss of case context, inability to track rescue status, and poor coordination between citizens, NGOs, veterinary professionals, and foster caregivers. Delayed access to veterinary intervention can increase the risk of complications and adverse outcomes for injured animals.[3] The proliferation of modern web frameworks presents an opportunity to bridge these systemic communication gaps. [4] Modern full-stack development frameworks, particularly the MERN stack (MongoDB, Express.js, React.js, Node.js), offer the technical foundation to build responsive, feature-rich web applications that serve as centralized platforms for social coordination.[5] The MERN stack provides distinct architectural advantages, including a unified JavaScript ecosystem across the development pipeline, non-blocking asynchronous I/O operations for handling concurrent client requests, flexible document-based data storage, and component-driven frontend architecture.[6]

1.1 Related work

Research at the intersection of information technology and animal welfare encompasses isolated reporting tools, adoption portals, and environmental monitoring systems. Table 1 presents a comparative synthesis of related works and identified research gaps.

1.1.1 Synthesis and research gap

Existing animal rescue systems mainly focus on adoption, animal management, and healthcare. [1-9] Recent studies have also explored community-based emergency reporting, rescue prioritization, and organizational coordination. [10,11]

1.1.2 Animal rescue and management systems

Existing platforms such as PAWrfect Match, Pet Adoption System, PetHub, FurEver, and other adoption and rescue systems primarily support animal listing, adoption, rescue management, or care.[1-8] These systems demonstrate the usefulness of digital platforms for animal welfare but generally address specific stages rather than maintaining a unified case from initial reporting through rescue, treatment, foster care, and adoption. The animal healthcare application proposed by Shelke et al. further demonstrates the role of digital tools in supporting animal health management.[9]

1.1.3 Emergency Reporting and Coordination

Recent research has expanded beyond adoption toward emergency rescue coordination. Ramani et al. examined smart emergency management using community reporting and rescue prioritization, highlighting the importance of structured incident information.[10] Islam et al. studied organizational coordination in animal rescue operations, emphasizing coordination among stakeholders.[11] These works support the need for structured rescue workflows, while RescueTails extends this approach by connecting reporting, rescue coordination, case tracking, clinical management, and adoption within one platform.

1.1.4 MERN stack applications in social-impact domains

The literature reviewed demonstrates the use of web-based and mobile technologies for animal adoption, rescue management, healthcare, community reporting, and organizational coordination.[1-11] Web-based systems have been used to support adoption and rescue activities, while mobile applications have been explored for animal healthcare.[1-9] Recent studies have also highlighted community reporting, rescue prioritization, and organizational coordination as important components of animal rescue operations.[10,11] These findings support the need for an integrated web-based platform that can connect multiple stakeholders and manage the different stages of the animal rescue process. RescueTails addresses this need through a unified web application integrating incident reporting, rescue coordination, clinical tracking, foster management, and adoption workflows.

1.1.5 Research gap

Existing literature demonstrates a clear gap in software systems that support the complete stray animal welfare lifecycle. Available platforms typically handle isolated segments—such as reporting, directory listings, or adoption cataloging—resulting in fragmented data and administrative overhead. RescueTails bridges this gap by providing an integrated web system spanning incident reporting, rescue dispatch, clinical tracking, foster management, and adoption placement within a unified platform.

1.2 Problem statement

Table 1: Comparative Analysis of Related Literature and Identified Research Gaps

Study	Title	Technology Used	Identified Gaps / Limitations
[1]	PAWrfect Match: A Web-Based Animal Adoption and Rescue System that uses Content-Based Filtering Algorithm for Recommending Potential Adoptees	Web-based platform, Content-Based Filtering	Primarily focuses on adoption and adopter–animal matching; does not provide a complete rescue-to-adoption workflow with clinical and foster management.
[2]	Pet Adoption System Using Web Technology	Web Technology	Focuses mainly on pet adoption; lacks comprehensive emergency reporting, rescue dispatch, and case tracking.
[3]	Pet Adoption System	Web Application	Concentrates on adoption management; limited support for real-time rescue coordination and veterinary case management.
[4]	PetHub: A Platform for Pet Adoption	Web Platform	Primarily adoption-oriented; does not integrate emergency rescue reporting, rescue dispatch, and complete animal case histories.
[5]	Pet Adoption and Rescue Center	Web-based System	Combines adoption and rescue functions but does not comprehensively integrate clinical tracking, foster management, and coordinated incident workflows.
[6]	FurEver: Pet Adoption Platform	Web Platform	Mainly addresses pet adoption; lacks structured emergency reporting, rescue prioritization, and end-to-end case tracking.
[7]	Design of an Automated Animal Rescue and Adoption Management System	Automated Web-Based Management System	Addresses rescue and adoption management but has limited integration of community reporting, clinical records, foster coordination, and unified case tracking.
[8]	Web Application for Stray Animal Adoption and Care Website	Web Application	Supports stray animal adoption and care but lacks a comprehensive workflow connecting incident reporting, rescue dispatch, treatment, foster care, and adoption.
[9]	Design and Development of Android Based Animal Healthcare Application	Android Application, Mobile Healthcare	Focuses on animal healthcare; does not cover community rescue reporting, rescue coordination, foster management, or adoption workflows.
[10]	A Literature Survey on Smart Emergency Management Systems for Stray Animals Using Community Reporting and Rescue Prioritization	Community Reporting, Rescue Prioritization, Smart Emergency Management	Focuses on emergency reporting and prioritization; does not itself provide an integrated operational platform covering treatment, foster care, and adoption.
[11]	Streamlining Rescue Efforts: A Study on the Impact of Organizational Coordination in Animal Rescue Operations	Organizational Coordination, Rescue Operations	Emphasizes coordination among rescue organizations; does not provide a unified software workflow for reporting, clinical tracking, foster management, and adoption.

Stray animals in urban environments face recurring risks from vehicle accidents, untreated diseases, and severe physical distress. The operational rescue landscape is currently hindered by the following structural challenges:

Fragmented Reporting Channels: Citizens lack a single, standardized platform to submit incident details, leading to unstructured posts across social media or delayed phone calls.[10,11]

1. Information Silos between Stakeholders: Key stakeholders—reporters, rescue teams, veterinarians, foster caregivers, and adopters—operate in isolation without shared dynamic status updates.[11]

2. Lack of Case Tracking and Audit Trails: Informal rescue requests lack systematic tracking, making it difficult to verify whether an animal has received care,

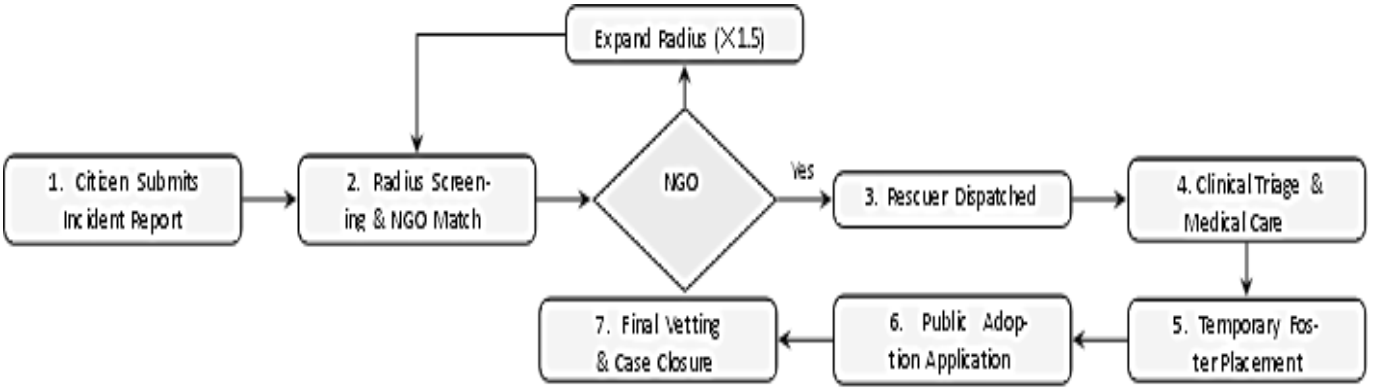

Fig. 1: End-to-end stray animal rescue coordination lifecycle and state transitions within the RescueTails system.

assigned responsibility, or completed treatment.[12]

3. Disjointed Adoption Management: Transitioning rescued animals to permanent homes involves screening and health verifications that are often handled through manual, paper-based, or uncoordinated methods.[1-8,13-15]

1.3 Objectives and contributions

To address these challenges, RescueTails provides a centralized web system for stray animal rescue management. The primary contributions of this paper are as follows:

- Development of a centralized MERN-based web platform connecting citizens, NGOs, rescue personnel, veterinarians, and foster caregivers via dynamic role-based dashboards.
- Implementation of a structured rescue workflow featuring location-assisted incident submission, status tracking, and verifiable case histories.
- Design of an integrated adoption module bridging medical clearance, applicant screening, and adoption application tracking.
- Quantitative evaluation of system performance through functional verification and benchmark load testing using Apache JMeter to establish latency, throughput, and operational stability under evaluated test loads.

2. Methods/Operational architecture

2.1 Proposed rescue coordination framework

The proposed RescueTails framework formalizes the stray animal welfare lifecycle into a multistage workflow designed to maintain transparency and case context across operational transitions. As illustrated in Fig. 1, the lifecycle encompasses seven key phases:

1. **Incident Reporting:** Citizens submit situational report details, including geographical coordinates, media uploads, and initial urgency levels.
2. **Radius screening & NGO matching:** The backend queries active registered NGOs within an adaptive geographical radius (MaxRadius).
3. **Dispatch & Queue Assignment:** Nearby rescue personnel view and accept incoming incidents via an interactive dispatch queue.
4. **On-site rescue & triage:** Rescuers update case status to Rescue and record preliminary triage findings.
5. **Veterinary & medical treatment:** Medical personnel record diagnostic logs, vaccination details, and medical updates.
6. **Foster Placement:** Animals cleared for temporary care are assigned to verified foster caregivers.
7. **Adoption Facilitation & Closure:** Formally cleared animals enter the public adoption gallery, concluding upon applicant vetting and case archival.

2.2 System overview

RescueTails is structured as a three-tier web application following the Model–View–Controller (MVC) architectural pattern, built using the MERN technology stack. The system comprises four primary operational components: User Management, Rescue Request Management, Adoption Management, and the NGO/Shelter Dashboard. The high-level system architecture is shown in Fig. 2.

2.3 Technology stack justification

The MERN stack was selected on the basis of architectural and performance considerations outlined in Table 2.

2.4 Data model design

The MongoDB database contains five primary document collections designed to support incident lifecycles, entity tracking, and organizational oversight:

User Collection: Stores user credentials, contact parameters, profile metadata, and security role assignments (Citizen, NGO, Rescue Team, Foster Home, Admin).

- **Rescue Requests Collection:** Store incident entries, including coordinates, urgency ratings, attached image URLs, reporter references, assigned responders, and status history logs.
- **Animal collection:** Records for rescued animals, including species, medical history, vaccination records,

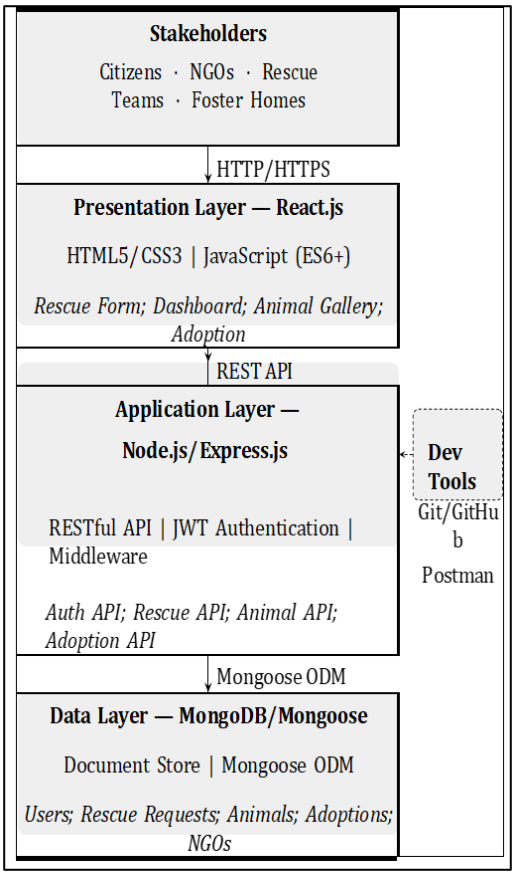

Fig. 2: High-level system architecture of RescueTails showing the three-tier MERN implementation

behavioral notes, and current location, were maintained.

- **Adoptions Collection:** Record adoption applications, applicant background information, approval statuses, and follow-up notes.
- **NGOs/Shelters Collection:** Store organizational profiles, operating service areas, housing capacity parameters, and contact references.

Fig. 3 displays the entity–relationship (ER) schema, which illustrates the core collections and references. While the full database schema is provisioned to support end-to-end state transitions across the rescue lifecycle, current prototype frontend interactions focus on incident submission, spatial queue retrieval, and adoptable animal cataloging, with administrative and clinical fields populated via backend API controllers.

2.5 API design and execution workflow

The backend API handles application functionality via structured HTTP request endpoints:

- /api/auth – Handles user registration, login, token generation, and credential verification.
- /api/rescue – Manage incident submission, status modification, geographic querying, and queue retrieval.
- /api/animals – Supports reading and updating animal records, clinical updates, and adoptable catalog listings.

Table 2: Technology stack components and architectural rationale

Layer	Technology	Justification
Frontend	React.js	Component-based architecture enables modular UI development; dynamic state management ensures efficient updates.
Backend	Node.js, Ex-press.js	Nonblocking I/O event loop handles concurrent API requests efficiently; unified JavaScript syntax across client and server.
Database	MongoDB, Mongoose	Document-oriented structure easily adapts to dynamic rescue data; Mongoose ODM enforces schema constraints and relationships.
API	RESTful API	Stateless HTTP interface supports modular deployment, simplified routing, and potential external client integration.

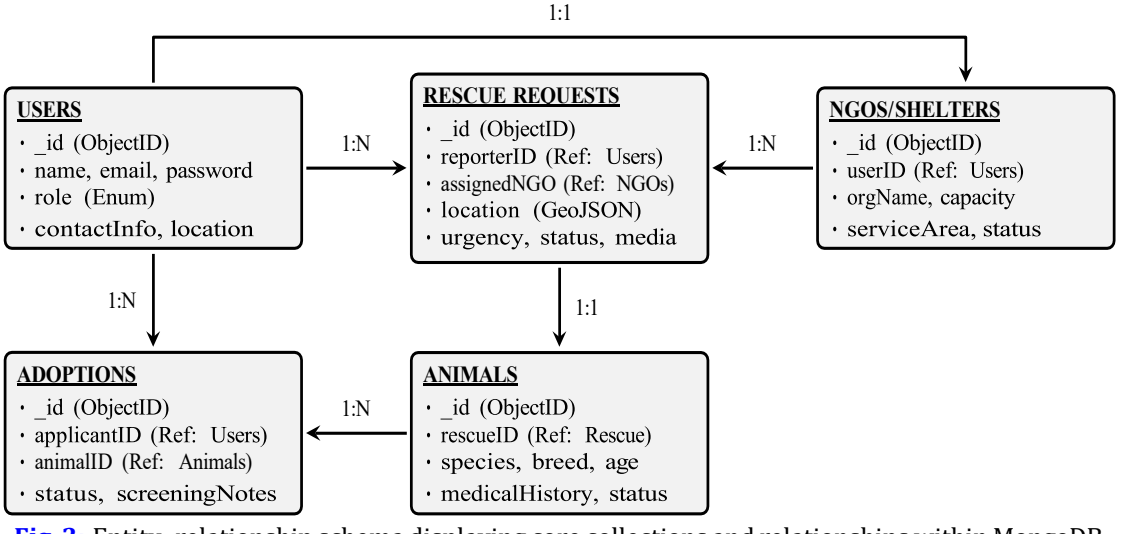

Fig. 3: Entity–relationship schema displaying core collections and relationships within MongoDB.

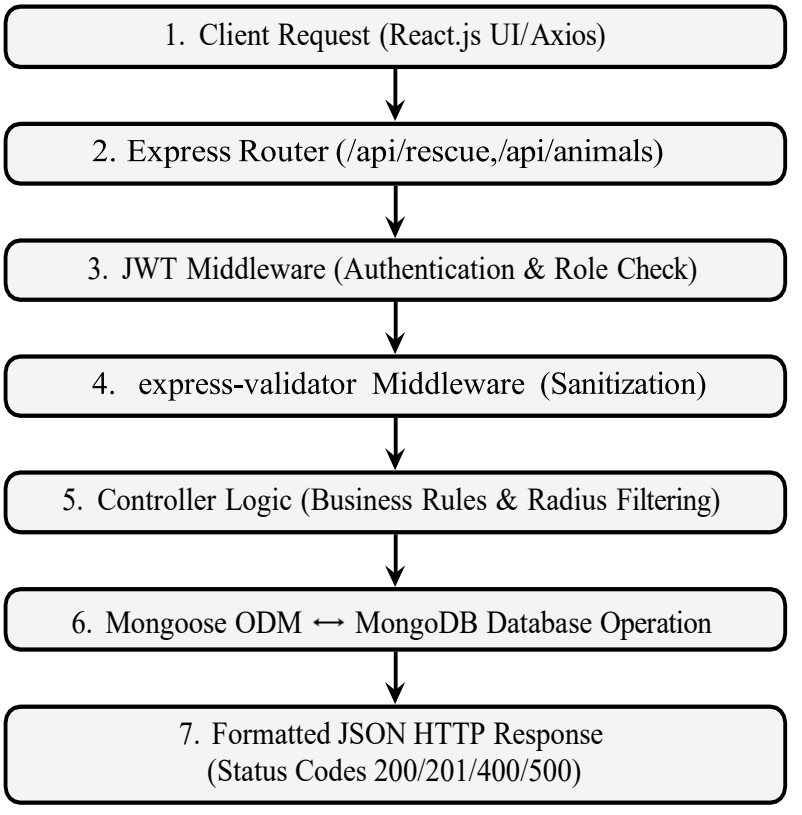

Fig. 4. RESTful API request–response processing pipeline in RescueTails

Algorithm 1. Proposed Stray Animal Rescue Incident Dispatch & NGO Coordination Workflow

Require: Incident Report R = {loc, urgency, media, reporterID}, Set of Registered NGOs N
Ensure: Incident Assignment A and Status State S
Initialize: $S \leftarrow$ "reported"
R.timestamp ← GetCurrentTimeStamp()
Save R to MongoDB with initial status S
$N_{active} \leftarrow \{n \in N \mid n.status = \text{"active"} \wedge Distance(n.loc, R.loc) \leq MaxRadius\}$
if $N_{active} = \emptyset$ then
    Expand search radius: MaxRadius ← MaxRadius × 1.5
    $N_{active} \leftarrow \{n \in N \mid Distance(n.loc, R.loc) \leq MaxRadius\}$
end if
for each $ngo \in N_{active}$ do
    TriggerNotification(ngo.id, R.id, R.urgency)
end for
while S = "reported" and ElapsedTime() < $T_{threshold}$ do
    if ngo_j accepts R.id then
        $A \leftarrow$ {incidentID: R.id, assignedNGO: ngo_j.id}
        $S \leftarrow$ "acknowledged"
        UpdateDatabase(R.id, S, ngo_j.id)
    end if
end while
if S = "reported" then
    EscalateToSystemAdmin(R.id)
end if
return A, S

Implementation Note: *The workflow described in Section 2.1 and Algorithm 1 define the intended operational architecture. The current software prototype implements client incident reporting, spatial coordinate filtering, and interactive dispatch queue visualization. Automated push-notification dispatch, dynamic search radius expansion (×1.5), and administrative escalation represent the proposed deployment-stage extensions*

- /api/adoptions – Processes adoption application submissions, review status transitions, and approval workflows.
- /api/ngos – Handles NGO organizational profile updates, service radius settings, and shelter capacity tracking.

The step-by-step API processing pipeline for client requests is shown in Fig. 4.

2.6 Proposed Layerwise GUI Architecture and Stakeholder Workflows

The front-end presentation layer uses a modular layout built with React.js components. Interface access and control options are tailored dynamically on the basis of the verified role claims of logged-in users:

1. **Citizen Interface (Implemented):** Interface options allow citizens to submit incident reports, attach geolocation details, track case status transitions, and submit adoption inquiries.
2. **NGO/Rescue Team Dashboard (Implemented):** Provides interactive regional incident queue visualization, manually initiated status modification, case filtering, and incident management.
3. **Veterinary Panel (Proposed):** Focuses on medical record maintenance, enabling verified clinicians to log diagnostic reports, update vaccination flags, and record medical clearances.
4. **Foster Caregiver Panel (Proposed):** Designed to allow caregivers to log hosting capacity, monitor assigned animals, and provide status updates during temporary care.
5. **System Administrator Interface (Proposed):** Intended for platform-wide administrative oversight, user account verification, global role management, and system auditing.

2.6 Security Mechanisms

The application architecture includes standard web security controls:

- **Authentication & Authorization:** Implements JSON Web Tokens (JWT) for stateless session verification across restricted API endpoints.
- **Password Hashing:** User passwords are hashed using bcrypt prior to persistence in the database.
- **Input Validation:** Request validation middleware (express-validator) sanitizes input parameters to mitigate common injection risks and invalid payload entries.
- **HTTP Security Headers & Cross-Origin Policies:** Use Helmet.js middleware to count security-related HTTP response headers and apply CORS middleware to regulate cross-origin API requests.

3. Results and Evaluation

3.1 Experimental Setup and Metric Definitions

Evaluation was conducted within a controlled benchmarking environment hosted on an isolated cloud instance (8 vCPU, 16 GB RAM, Ubuntu 22.04 LTS, Node.js v18.16.0, MongoDB 6.0). Load testing was executed using Apache JMeter 5.5 to evaluate API latency and system throughput under simulated concurrent user requests across the active prototype endpoints (POST/api/auth/login, POST/api/rescue/report, GET/api/rescue/nearby, and GET/api/animals adoptable catalog) over a 10-minute sampling window with a 30-second ramp-up.

The performance evaluation focused on four standard metrics:

- **Average API** latency: mean duration (ms) from request departure to full-client HTTP response receipt.
- **System Throughput:** Total successfully processed HTTP transactions per second (requests per second, RPS).
- **Database Query Latency:** Average execution duration (ms) recorded for Mon-goose/MongoDB query processing.
- **Error Rate:** Percentage of HTTP requests yielding 4xx or 5xx status responses relative to total generated requests.

3.2 Quantitative Benchmark Evaluation

Table 3 summarizes benchmark performance measurements across escalating levels of simulated concurrent user traffic.

3.3 Comparison of Functional Capabilities

Table 4 compares the operational capabilities supported by RescueTails with those of traditional informal channels and directory-based pet portals.

Table 3: Apache JMeter Load Test Results across Concurrent User Load Levels

Concurrent Users	Avg API Latency (ms)	Throughput (RPS)	DB Query Latency (ms)	Error Rate (%)
50	42	310	12	0.00%
100	78	560	18	0.00%
250	135	870	26	0.01%
500	265	1110	41	0.12%
1000	495	1220	76	1.05%

Table 4: Functional Capability Matrix: Traditional Channels vs. RescueTail Platform

Functional Dimension	Informal/Directory Platforms	RescueTails Platform
Incident Reporting	Unstructured (Social Media/Calls)	Structured Web Form with Geo-Coordinates
Location-Based NGO Search	Manual Search/Directory Lookup	Spatial Coordinates Filtering
Case Lifecycle Tracking	Absent	End-to-End Status Tracking
Medical History Logging	Disjointed/Paper-Based	Integrated Medical Database
Foster Care Management	Manual Verification	Role-Based Foster Dashboards
Adoption Application Flow	External Inquiries	Built-in Application Screening
Role-Based Access Control	Minimal/None	Granular Role Claims (5 Roles)
System Audit Trail	Absent	Database Case History Logs

3.4 Discussion and statistical interpretation

As the simulated user concurrency increased from 50 to 1000 virtual users, the average API response latency expanded from 42 ms to 495 ms, whereas the overall throughput scaled from 310 RPS to 1220 RPS. The database query execution latency correspondingly increased from 12 ms to 76 ms. The progressive increase in latency under higher user loads reflects the expected database connection pool queuing and Node.js event-loop processing overhead under concurrent request bursts. Continued throughput growth up to 1000 virtual users indicates that the server architecture efficiently handled substantial traffic volumes, with a minor 1.05% error rate observed at peak concurrency, signaling the initial onset of system resource contention. As highlighted in Table 4, consolidating incident reporting, status tracking, and adoption pipelines within a single platform addresses data fragmentation and improves operational visibility across animal welfare stakeholders.

4. Testing, quality assurance and future scope

4.1 Testing and quality assurance

To ensure platform stability, data integrity, and interface responsiveness across devices, a multitiered quality assurance protocol was applied during prototype development:

- **Unit and Component Verification:** Core JavaScript utility functions and React component rendering states were verified using the Jest and React Testing Library to prevent regression bugs during interface updates.
- **API Endpoint and Integration Testing:** Backend controller routes (/api/auth, /api/rescue,/api/animals) were systematically tested using automated Postman test suites to validate payload structures, HTTP status codes, and JWT access control authorship.
- **Cross-Device Usability Verification:** Responsive layouts were validated across multiple screen viewports (mobile, tablet, desktop) using Chrome DevTools and physical device testing to ensure seamless accessibility for field rescuers and citizens.
- **Input Sanitization and Security Auditing:** The request payload boundaries were verified using express-validator rules to mitigate invalid entries, malformed coordinates, and unauthenticated state modifications.

4.2 Future Scope

Future enhancements to the RescueTails platform will focus on scaling field operations and automation:

- **Offline-First Progressive Web App (PWA):** Incorporating service workers and In-dexedDB client-side caching to support offline incident reporting in low-connectivity urban zones.
- **Automated Push Dispatch Engine:** Integrating Firebase Cloud Messaging (FCM) and Web Push APIs to execute automated, radius-based emergency notification alerts.
- **AI-Assisted Severity Triage:** Deploying lightweight Convolutional Neural Network (CNN) vision models to automatically analyze uploaded incident photos and prioritize critical injury cases.

5. Conclusion

In this paper, RescueTails, a full-stack MERN platform designed for centralized stray animal rescue coordination, case tracking, and adoption management, is presented. By integrating citizen reporting, location-assisted NGO coordination, medical tracking, and adoption management into a single web application, RescueTails addresses operational fragmentation in urban animal rescue workflows. Benchmark evaluation provides evidence of subsecond API response times and low error rates across the evaluated prototype endpoints under simulated multiuser test loads. Future development will focus on incorporating progressive web application (PWA) offline capabilities, real-time push notification services, and automated computer vision models to assist in incident severity classification.

Acknowledgement

None

CRediT Author Contribution Statement

Zehra Shaikh: Conceptualization, Investigation, Methodology, Software, Writing – Original draft. **Nabiha Shaikh:** Data Curation, Software, Validation, Visualization. **Hamsa Naik:** Formal Analysis,

Methodology, Software, Writing – Review & editing. **Maryam Lokhandwala:** Investigation, Project Administration, Resources, Writing – Review & editing. **Chaitali Mahajan:** Methodology, Resources, Supervision, Validation, Writing – Review & editing. All authors have read and approved the final version of the manuscript for publication and agree to be accountable for all aspects of the work, ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

Funding Declaration
This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability Statement
The datasets generated during the current study, along with the JMeter benchmark configurations, are available from the corresponding author upon reasonable request.

Conflict of Interest
There is no conflict of interest.

Artificial Intelligence (AI) Use Disclosure
The authors declare that artificial intelligence (AI)-assisted tools were used only for language refinement, grammar improvement, and manuscript structuring purposes during the preparation of this work. All technical content, experimental implementation, results, and interpretations were independently developed and verified by the authors.

Supporting Information
Not applicable.

References

[1] L. J. Torres, C. J. de Leon, F. A. Callado, J. D. Cuartero, C. J. Centeno, G. M. Casiw, PAWrfect Match: A Web-Based Animal Adoption and Rescue System that uses Content-Based Filtering Algorithm for Recommending Potential Adoptees, *Library Progress International*, 2024, **44**, 1525.

[2] D. Bhadane, P. Khirude, O. Chavan, A. Lokare, Pet adoption system using web technology, *International Journal of Scientific Research in Engineering and Management,* 2023,**7.**

[3] H. Bhale, S. Chavan, C. Kamble, Pet Adoption System, *International Journal of Research Publication and Reviews, 2023,* **4**, 3778–3782.

[4] Ranjitha, S. K. Sadhana, PetHub: A Platform for Pet Adoption, *International Journal of Multidisciplinary Research in Science, Engineering and Technology*, 2024, **7.**

[5] J. Prajapati, R. Vishwakarma, R. Deliwala, S. Singh, K. Maner, Pet Adoption and Rescue Center, *International Journal of Creative Research Thoughts*, 2025, **13**, 1711–1721.

[6] V. Welis, M. Sonawane, S. Naikdhure, A. Nelson, A. Minu, FurEver: Pet Adoption Platform, *International Journal of Engineering Research & Technology*, 2023,**12**.

[7] S. R. Uttarkar, T. K. Soni, and G. Sangeetha, Design of an Automated Animal Rescue and Adoption Management System, *International Journal of Engineering Research in Computer Science and Engineering*, 2024, **11**, 166.

[8] T. Jamadar, N. Jagtap, R. Shete, U. Pathan, K. More, Web Application for Stray Animal Adoption and Care Website, *Journal of Emerging Technologies and Innovative Research, 2025,* **12**, f636-f642.

[9] T. Shelke, S. Shahu, A. Godar, M. Talewar, J. Thaker, Design and Development of Android Based Animal Healthcare Application, *International Journal for Research in Applied Science & Engineering Technology,* 2022,10,1774–1777, doi: 10.22214/ijraset.2022.40963.

[10] D. R. Ramani, S. S. Rani, M. Manupriya, K. R. Mrinali, L. P. Thatha, S. Bhalke, A Literature Survey on Smart Emergency Management Systems for Stray Animals Using Community Reporting and Rescue Prioritization, In 2025 9th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS), IEEE, India, 2025, 1–6, doi: 10.1109/CSITSS67709.2025.11295807.

[11] M. N. Islam, R. Rahman, S. R. Akthar, M. Hasan, S. Ahmed, Streamlining Rescue Efforts: A Study on the Impact of Organizational Coordination in Animal Rescue Operations, In 2024 *15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, India, 2024, 1-7, doi: 10.1109/ICCCNT61001.2024.10724020.

[12] C. Sazara, X. Gao, Predicting Animal Shelter Pet Adoption Times and Feature Importance Analysis using CatBoost, In2022 IEEE 11th International Conference on Intelligent Systems (IS), IEEE, Poland, 2022, 1-4, doi: 10.1109/IS57118.2022.10019608.

[13] A. Magdum, A. Magdum, G. Chavan, S. Jadhav, Mobile Application of Pet Adoption System, *International Research Journal of Engineering and Technology,* 2023, 10, 740-743.

[14] K. Erin, J. Altay, B. Sarı, Modernization of Adoption Processes in Animal Shelters: A User-Friendly Mobile Application Approach, *Journal*

of Smart Systems Research, 2026, **7**, 94-107, doi: 10.58769/joinssr.1939005.

[15] R. Alsuwailem, R. Almobarak, R. Aboali, S. Alrubaiea, Aldossary J. Leen: web-based platform for pet adoption. In2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET), IEEE, Romania, 2022, 775-780, doi: 10.1109/GlobConET53749.2022.9872481.