Smart Cattle Management System Using MERN Stack: Design, Implementation and Evaluation

[1]Rituraj, [2]Rakesh Kumar, [3]Naval Verma, [4]Gaurav Kasana, [5] Rajendra Singh
[1,2,3,4,5]Department of Computer Science & Engineering, Raffles University, Neemrana, Rajasthan, India

Abstract—The Indian dairy sector contributes approximately four percent to the national GDP and supports the livelihoods of more than 80 million families. Despite its economic significance, the vast majority of dairy farms continue to rely on manual, paper-based record-keeping practices. This paper describes the design, development, and empirical evaluation of a Smart Cattle Management System (SCMS) built on the MERN stack (MongoDB, Express.js, React.js, Node.js). The web-based system automates key farm operations cattle registration, daily milk production tracking, vaccination scheduling, breeding cycle monitoring, and financial management within a three-tier architecture secured by role-based access control and JSON Web Token authentication. The system was deployed on AWS EC2 and evaluated over a three-month pilot at a dairy farm managing 45 head of cattle. Key outcomes include a 99.6 percent reduction in record retrieval time (from 8 10 minutes to under three seconds), an 85 percent decrease i data-entry errors, a 58 percent improvement in vaccination compliance (from 60 percent to 95 percent), and a 12 percent increase in overall farm profitability. User acceptance testing with ten participants returned an average satisfaction score of 4.6 out of 5.0. The system is published as an open-source solution under the MIT license, providing a cost-effective and culturally adapted alternative to expensive foreign dairy management software.

'·-·ÃÁ-**1·1/®¼/·MERN-Stack, -Dairy-Farm-Automation, -MongoDB, -React.js, -Node.js, -Smart-Agriculture, -Web Application, -India**

I. INTRODUCTION

The global dairy industry faces mounting pressure to modernise its operational workflows. India, which has held the position of the world's largest milk producer since 1998, generated 230.58 million metric tonnes of milk in 2022 23 at an annual growth rate of 5.8 percent [1]. In spite of this scale, a 2022 survey conducted by the National Bank for Agriculture and Rural

Development (NABARD) found that only 12 percent of Indian dairy farmers currently use digital record-keeping tools; the remaining 88 percent rely on paper registers, notebooks, or memory-based tracking [2].

Field observations conducted at three dairy farms in the Rewari and Narnaul districts of Haryana exposed persistent operational inefficiencies: average record retrieval times of 8 10 minutes per anima data-entry error rates between 8 and 12 percent, vaccination compliance rates as low as 60 percent, and an inability to generate monthly profit-and-loss statements without 3 4 hours of manual computation These observations align with the broader literature on digital-transformation challenges in agricultural settings [3].

Established commercial dairy management platforms DairyCOMP 305, DairyMaster, and Go Vigyan either lack Indian breed databases and Hindi-language support, or carry annual licence fees ranging from Rs. 25,000 to Rs. 2,50,000. Such costs place these tools beyond the reach of the small and medium-scale farms that constitute more than 90 percent of India's dairy sector. This technological gap defines the primary motivation for the present work.

This paper makes three principal contributions: (1) a fully functional, open-source, web-based cattle management system designed specifically for the Indian dairy farming context; (2) an empirically validated MERN-stack architecture incorporating role-based access control and cloud deployment; and (3) a documented set of performance improvements drawn from a real-world pilot study.

II. LITERATURE REVIEW

Farm management information systems have passed through four recognisable generations. Early desktop applications such as DairyCHAMP (1985) and DairyCOMP 305 (1992) delivered single-computer, non-networked solutions. Client-server architectures (2000–2015) subsequently enabled multi-user access over local networks, though they demanded on-premise server hardware and dedicated technical staff. The current generation of cloud-based web and mobile applications, which emerged after 2015, offers real-time synchronisation and remote accessibility without local infrastructure.

Smith and Johnson [3] conducted a systematic review of digital transformation in agriculture and identified JavaScript full-stack frameworks as particularly well suited to resource-constrained environments owing to their single-language paradigm and mature tooling ecosystem. Brown et al. [4] evaluated web-based farm management systems across five dimensions — scalability, usability, cost, offline capability, and localisation — and consistently found that open-source solutions outperform proprietary alternatives in adoption among smallholder farmers.

Kumar et al. [5] specifically examined IT adoption barriers in the Indian dairy sector, identifying language barriers, high software costs, and the absence of breed-specific databases as the three primary obstacles. Verma and Tiwari [13] reviewed IoT-based cattle monitoring systems and projected that web-based platforms with IoT integration will become the dominant paradigm by 2027, while acknowledging that software-only solutions remain the most practical immediate choice for the majority of farms.

The MERN stack has been validated at enterprise scale by organisations including Facebook, Netflix, and Uber. Gupta and Mehta [10] demonstrated that MERN-based agricultural applications achieve sub-500 ms API response times under typical farm usage loads, meeting the performance requirements identified in the present study's requirements analysis. No prior published work, however, has evaluated a MERN-based system designed and field-tested specifically for the Indian dairy context, which constitutes the primary novelty of this research.

III. SYSTEM DESIGN AND ARCHITECTURE

A. Technology Selection

Five technology stacks were evaluated during the design phase: LAMP (Linux, Apache, MySQL, PHP), MEAN (MongoDB, Express, Angular, Node), MERN (MongoDB, Express, React, Node), Django (Python), and Ruby on Rails. The MERN stack was selected based on five criteria: (1) a unified JavaScript language across client and server reduces development complexity; (2) Reacts virtual DOM delivers superior rendering performance for dynamic dashboards; (3) MongoDB's schema-less document model accommodates evolving dairy farm data requirements; (4) strong community support and an extensive npm package ecosystem; and (5) React Native compatibility, which provides a clear migration path to a native mobile application in Phase 2.

B. Three-Tier Architecture

The system adopts a classic three-tier architecture. The Presentation Layer is a React.js single-page application (SPA) built with Bootstrap 5 for responsive design, Chart.js for production-trend visualisations, Axios for HTTP communication, and the React Context API for global state management. The Application Layer is a Node.js/Express.js RESTful API server incorporating JWT-based authentication middleware, bcrypt password hashing, Helmet.js security headers, express-rate-limit for DDoS protection, and node-cron for automated vaccination reminder scheduling. The Data Layer uses MongoDB Atlas — a cloud-hosted NoSQL database — with Mongoose ODM for schema definition and validation.

C. Database Schema

Six primary collections were designed: Users (role-based access), Cattle (registration details including breed, age, and health status), MilkRecords (daily morning and evening quantities per animal), Vaccinations (schedule and history), Breeding (insemination dates, expected delivery, and calf records), and Finance (categorised income and expense transactions). Relationships between collections are maintained through MongoDB ObjectId references, enabling efficient join-equivalent queries via Mongoose's populate () method.

D. Security Design

Security was implemented at multiple layers following OWASP guidelines [15]. Authentication uses JSON Web Tokens with a configurable expiry period. Passwords are hashed using bcrypt with a salt factor of ten. All API endpoints — except authentication routes — require a valid bearer token. Input sanitisation is enforced server-side via express-validator. All data in transit is encrypted using HTTPS, with SSL certificates provisioned through Let's Encrypt.

IV. IMPLEMENTATION

A. Development Environment

The system was developed using React.js 18.2.0 (frontend), Node.js 20.9.0 and Express.js 4.18.2 (backend), and MongoDB 6.0 with Mongoose 7.5.0 (database layer). Key frontend dependencies include Chart.js 4.3.0 for data visualisation, Formik 2.4.2 and Yup 1.2.0 for form handling and validation, and react-router-dom 6.14.0 for client-side routing. Backend dependencies include Nodemailer 6.9.4 for automated email reminders and Multer 1.4.5 for cattle photograph uploads.

B. Key System Modules

The system is organised into six functional modules. The Cattle Management module supports registration via a twelve-field form that includes Indian breed options (Gir, Sahiwal, Red Sindhi, Crossbred), photograph upload, and health-status tracking. The Milk Production module enables twice-daily (morning and evening) entry per animal, automatic daily total calculation, and trend visualisation through bar charts. The Health Management module maintains vaccination schedules and dispatches automated email reminders seven days before each due date via a node-cron background job. The Breeding Management module calculates expected delivery dates from the insemination date using the standard 285-day bovine gestation period, tracks pregnancy status, and records calf data. The Financial Management module categorises transactions across five heads (Feed, Medicine, Labour, Milk Sales, Cattle Sales) and generates monthly profit-and-loss statements. The User Management module enforces role-based access: Administrators hold full system access; Managers can administer cattle records and generate reports; Staff members are restricted to entering daily milk volumes and vaccination data.

C. Deployment

The application was deployed on an AWS EC2 t2. micro instance running Ubuntu 22.04 LTS, using Nginx as a reverse proxy and PM2 as the Node.js process manager. The React production build is served as static files from /var/www/html, while all API requests are proxied to the backend process on localhost:5000. MongoDB Atlas provides cloud-hosted database services with daily automated backups. The deployment achieved 99.5 percent uptime across the three-month evaluation period.

V. RESULTS AND EVALUATION

A. Performance Testing

Performance testing was conducted using Apache JMeter for load testing and Google Lighthouse for frontend performance metrics. Table I presents API response times under standard and peak load conditions. With 50 concurrent users, the system maintained an average response time of 325 ms and an error rate of 0.2 percent. Under stress conditions with 200 concurrent users, response times increased to 687 ms with a 2.1 percent error rate — both figures within acceptable bounds for agricultural field use. The Google Lighthouse performance score was 96 out of 100, with a First Contentful Paint of 1.2 seconds.

Table I — API Response Time Analysis

API Endpoint	Avg. Response (ms)	Peak Load (ms)	DB Queries
GET /api/cattle	187	342	1
GET /api/milk (daily)	267	489	15
GET /api/milk/report	412	678	45
GET /api/vaccinations/due	234	389	12
GET /api/finance/summary	312	534	25
Average	282	486	—

B. Manual vs. Automated System Comparison

The three-month pilot at the 45-cattle dairy farm demonstrated measurable improvements across all operational dimensions examined. Table II summarises the key comparative metrics.

Table II — Manual vs. Automated System Performance Comparison

Activity	Manual System	Proposed System	Improvement
Record Retrieval (per animal)	8–10 min	< 3 seconds	99.6%
Daily Milk Entry (45 cattle)	45 min	10 min	78%
Data Entry Error Rate	8–12%	< 1%	~90%
Vaccination Compliance	60%	95%	+58%
Monthly P&L Calculation	4 hours	5 minutes	98%
Farm Profitability (annual)	Baseline	+Rs. 45,000	+12%

C. Comparison with Existing Systems

Table III benchmarks the proposed system against three leading dairy management solutions across dimensions most relevant to Indian small and medium dairy farms.

*Table III — Comparison with Existing Systems * Optional paid hosting ~Rs. 1,500/month after AWS free tier.*

Feature	DairyCOMP 305	DairyMaster	Pashu Poshan	Proposed
Annual Cost	Rs. 2.5 L	Rs. 1.5 L	Free	Free *
Platform	Desktop	Web/Mobile	Mobile	Web (Resp.)
Indian Breeds DB	Limited	Limited	Yes	Yes
Hindi Language	No	No	Yes	Yes
Multi-user Roles	Yes	Yes	No	Yes
Financial Module	Yes	Yes	No	Yes
Open Source	No	No	Yes	Yes (MIT)

D. User Acceptance Testing

User Acceptance Testing (UAT) was conducted with ten stakeholders representing four user roles over a two-week period. Participants completed 130 defined task scenarios with a 100 percent task completion rate. Five minor issues were identified and resolved: a mobile date-picker bug, incomplete Hindi translations in two modules, slow PDF report export, reduced visibility of the forgot-password link, and a cattle-name search indexing issue. Post-resolution satisfaction scores on a five-point Likert scale are presented in Table IV.

Table IV — User Acceptance Test Satisfaction Scores

System Aspect	Average Rating (1–5)
Vaccination Reminders	4.9
Milk Production Tracking	4.8
System Response Speed	4.8
Hindi Language Support	4.7
Dashboard and Overview	4.7
Ease of Use	4.6
Financial Reports	4.5
Mobile Experience	4.3
Overall Satisfaction	4.6 / 5.0

VI. DISCUSSION

The results confirm that a cost-effective, open-source web application built on the MERN stack can deliver performance improvements comparable to premium commercial systems in the Indian dairy farming context. The near-elimination of record retrieval delay — a 99.6 percent improvement — is particularly significant for daily farm operations, where field-level decision-making, such as identifying a sick animal or confirming a vaccination history, currently requires manual searches through multiple paper registers.

The 58 percent improvement in vaccination compliance, which rose from 60 to 95 percent, represents the most impactful clinical outcome of the study. Two preventable disease incidents were recorded during the pre-implementation observation period; none occurred during the three-month pilot. This finding is consistent with Verma and Tiwari [13], who identified automated reminder systems as the single most effective digital intervention for herd health management in smallholder settings.

The 12 percent improvement in farm profitability was driven by two mechanisms. First, precise individual cow milk-yield tracking enabled the farm owner to identify and cull consistently low-producing animals. Second, categorised expense tracking revealed that feed costs constituted 59.7 percent of total operating expenditure — information that had previously been unavailable and which has since enabled targeted feed-cost optimisation. This finding aligns with Kumar et al. [5], who ranked financial transparency as the second-highest priority need among Indian dairy farmers,

after herd health management.
The average UAT satisfaction score of 4.6 out of 5.0 is noteworthy given that participants included low-literacy farm workers. The two-day training period required for staff-level users met the system's non-functional requirement of a learning curve not exceeding two hours for basic operations. Feedback consistently highlighted Hindi-language support and automated vaccination reminders as the most valued features — both of which are absent from dominant foreign-developed commercial alternatives.

VII. LIMITATIONS AND FUTURE WORK

Several limitations constrain the current implementation. First, the system requires a stable internet connection with a minimum bandwidth of 2 Mbps for real-time data synchronisation; Progressive Web App (PWA) offline functionality is planned for Phase 2. Second, milk volume data entry remains manual; integration with digital milk meters and IoT flow sensors would eliminate this bottleneck [13]. Third, the analytics layer is currently descriptive; the system does not yet offer predictive capabilities such as disease-risk forecasting or optimal insemination timing based on machine learning models [14]. Fourth, the evaluation was conducted at a single farm managing 45 cattle; a multi-site longitudinal study across diverse farm sizes would strengthen the generalisability of the reported findings.

Planned Phase 2 enhancements include PWA offline mode with background data sync, a React Native mobile application, SMS-based vaccination reminders to supplement the existing email channel, support for additional Indian regional languages (Tamil, Telugu, Kannada, Bengali, Marathi), and bulk data import from Excel files. Phase 3 targets IoT sensor integration for automated milk measurement, RFID-based cattle identification, computer-vision health assessment from photographs, and a multi-farm management console under a single administrator account.

VIII. CONCLUSION

This paper has presented the design, implementation, and empirical evaluation of a Smart Cattle Management System built on the MERN stack, tailored specifically for small and medium dairy farms in India. The system successfully addresses the critical operational gaps identified through field observation: inefficient manual record-keeping, poor vaccination compliance, financial opacity, and the inaccessibility of existing commercial tools due to prohibitive cost and language barriers.
A three-month deployment at a representative 45-cattle dairy farm produced substantial and measurable improvements across all key performance indicators, with a return on investment exceeding 800 percent in the first year of operation at an annual cost of approximately Rs. 6,300 — primarily comprising domain registration fees. User acceptance testing confirmed high usability scores across all stakeholder groups, including low-literacy farm workers.

By releasing the system under an MIT open-source licence and adapting it specifically for Indian farming practices, livestock breeds, and language preferences, this work contributes both a validated technical artefact and a replicable implementation model for agricultural technology adoption in resource-constrained developing-world contexts. The underlying architecture is readily adaptable to adjacent livestock domains including poultry farming, goat husbandry, and aquaculture.

REFERENCES

[1] Department of Animal Husbandry and Dairying, *Annual Report 2023–24*. New Delhi, India: Ministry of Fisheries, Animal Husbandry and Dairying, Government of India, 2024.

[2] National Bank for Agriculture and Rural Development (NABARD), All India Rural Financial Inclusion Survey 2021–22. Mumbai, India, 2023.

[3] J. Smith and R. Johnson, "Digital transformation in agriculture: A systematic review," IEEE Trans. AgriTech, vol. 10, no. 2, pp. 112–120, 2021.

[4] M. Brown, L. Davis, and S. Wilson, "Web-based farm management systems: A comparative analysis," Int. J. Smart Farming, vol. 8, no. 3, pp. 45–58, 2022.

[5] P. Kumar, R. Singh, and A. Sharma, "Adoption of information technology in the Indian dairy sector: Challenges and opportunities," J. Rural Develop., vol. 41, no. 2, pp. 189–204, 2023.

[6] MongoDB Inc., MongoDB Documentation, 2024. [Online]. Available: https://docs.mongodb.com/

[7] Node.js Foundation, Node.js Official Documentation, 2024. [Online]. Available: https://nodejs.org/en/docs/

[8] React Team, React.js Documentation. Menlo Park, CA, USA: Meta Platforms, 2024. [Online]. Available: https://reactjs.org/docs/

[9] Express.js Team, Express.js Documentation, 2024. [Online]. Available: https://expressjs.com/

[10] A. Gupta and N. Mehta, "MERN stack development: Best practices and performance optimisation," in Proc. 5th Int. Conf. Web Technologies, Bangalore, India, 2023, pp. 234–241.

[11] T. Williams, "RESTful API design patterns," IEEE Softw., vol. 39, no. 4, pp. 67–73, 2022.

[12] K. Patel, "NoSQL databases in agricultural applications: A case study with MongoDB," Int. J. Database Manage. Syst., vol. 14, no. 2, pp. 23–35, 2022.

[13] R. Verma and S. Tiwari, "IoT-based cattle monitoring systems: A review," in 2022 IEEE Int. Conf. IoT in Agriculture, Pune, India, 2022, pp. 89–94.

[14] L. Zhang et al., "Machine learning applications in dairy farming: A systematic review," Comput. Electron. Agric., vol. 198, Art. no. 107045, 2022.

[15] OWASP Foundation, Web Application Security Best Practices, 2024. [Online]. Available: https://owasp.org/

[16] P. Desai and M. Joshi, "User interface design for rural users: Guidelines and best practices," in Proc. 12th Int. Conf. Human–Computer Interaction, Mumbai, India, 2023, pp. 156–163.

[17] Food and Agriculture Organization (FAO), Dairy Development in India: Opportunities and Challenges. Rome, Italy, 2023.
[18] Amazon Web Services (AWS), Amazon EC2 Documentation, 2024. [Online]. Available: https://docs.aws.amazon.com/ec2/