iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79318

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

AgriSmart: A Decentralized MERN-Stack Marketplace for Crop Registration, Land Leasing, and Equipment Rental

Dr. Trupti Meshram[1], Mr. Abhay Patle[2], Ms. Asha Saroya[3], Mr. Anirudh Thekedar[4], Mr. Aniket Wanjari[5], Mr. Ankit Choudhari[6]
Department of B.Tech Computer Science, G. H. Raisoni University Amaravati, Maharashtra, India

Abstract: ***This paper presents the design and implementation of AgriSmart, a full-stack, decentralized web-based marketplace developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) to resolve the financial and technological barriers in the agricultural sector. India's agricultural economy is largely driven by small and marginal farmers who lack access to modern machinery, transparent markets, and efficient land utilization mechanisms. AgriSmart addresses these challenges by integrating three core modules: an equipment rental marketplace, an agricultural land leasing portal, and a secure crop registration and MSP procurement system. The platform leverages the HTML5 Geolocation API for precise GPS-based farm mapping, native browser WebRTC media APIs for live crop photo verification, and Cloudinary for scalable cloud-based document and image storage. User authentication is secured using bcrypt hashing and JWT-based session management. The results demonstrate a highly responsive, fraud-resistant, and user-friendly ecosystem that empowers rural farming communities, reduces resource wastage, and creates new passive income streams for equipment and land owners.***

Keywords: ***MERN Stack, Agricultural Equipment Rental, Land Leasing, Cloudinary, HTML5 Geolocation, WebRTC, Smart Farming, Minimum Support Price, JWT Authentication, React.js.***

I. INTRODUCTION

Agriculture is the primary livelihood for more than 58% of India's rural population and contributes significantly to the national GDP. Despite this critical importance, the agricultural sector continues to face deep-rooted structural inefficiencies. The majority of India's farmers are small or marginal, cultivating less than two hectares of land. For these farmers, the capital investment required to purchase modern farming equipment such as tractors, combine harvesters, rotavators, and seed drills is entirely prohibitive, forcing reliance on manual labor or expensive local monopolies.

Simultaneously, farmers who own machinery face severe underutilization. Tractors and heavy implements are needed only during specific farming windows, leaving expensive equipment idle for most of the year. Similarly, unused agricultural land remains uncultivated due to lack of a platform connecting landowners with willing cultivators. When harvest arrives, farmers lack a direct digital system to register crops for government Minimum Support Price (MSP) procurement, leaving them vulnerable to middleman exploitation.

To address these interconnected problems, we present AgriSmart, a unified, decentralized, peer-to-peer digital marketplace built on the MERN stack. AgriSmart enables small and marginal farmers to rent equipment they cannot afford to buy, lease idle land they cannot actively cultivate, and register crops directly with government procurement channels, all within a single responsive web application secured by live hardware verification.

II. LITERATURE REVIEW

A review of existing agricultural technology platforms reveals significant gaps that AgriSmart is designed to address:

1) Patil et al. (2022): developed a web-based smart farming equipment rental system. Their research proved digital platforms significantly reduce the time farmers spend searching for machinery. However, the system lacked real-time availability tracking, GPS-based location filtering, and secure payment infrastructure.
2) Sreedevi et al. (2022): conducted an economic analysis comparing ownership versus rental models. Their findings showed rental-based economies relieve capital pressure on farmers. Despite compelling findings, their model lacked an intuitive interface suitable for rural users and did not address land leasing or crop procurement.

3) Swarnamalya and Anbumani (2023): proposed AgroEcom, an e-commerce approach for agricultural tools emphasizing centralized digital interactions. A major limitation was the absence of a unified dashboard managing multiple agricultural services simultaneously.
4) Mhatre et al. (2024): demonstrated the effectiveness of the MERN stack through Cropify. Their paper proved React.js and Node.js deliver scalability and fast rendering essential for rural users on low-bandwidth connections. However, the platform lacked geolocation verification and crop registration.
5) Mishra et al. (2024): focused on modern web APIs in their Farmingo platform. They showed direct digital connections between farmers and consumers improve market access. However, they did not address fraudulent data entry in government portals.
6) Kumar and Saravanamuthu (2022): introduced AGRARYANS, an agricultural machinery rental system for the Indian market. While providing a good catalog interface, the system lacked user verification mechanisms, live media capture, or cloud-based document management.

TABLE I: Comparative Analysis of Existing Agricultural Platforms

Feature	Patil 2022	Sreedevi 2022	Swarnamalya 2023	Mhatre 2024	Mishra 2024	AgriSmart
Equip. Rental	Yes	Yes	Yes	Yes	Yes	Yes
Land Leasing	No	No	No	No	No	Yes
Crop Reg.	No	No	No	Partial	No	Yes
GPS Verify	No	No	No	No	No	Yes
Live Camera	No	No	No	No	No	Yes
MSP Portal	No	No	No	No	No	Yes
Cloud Store	No	No	Partial	No	No	Yes

Synthesis: While existing literature explores isolated solutions, very few platforms offer a unified ecosystem. AgriSmart extends current research by combining equipment rental, land leasing, and secure MSP procurement into a single application, fortified by live media verification and GPS geolocation.

III. PROBLEM STATEMENT

The traditional agricultural supply chain suffers from multiple simultaneous structural inefficiencies that disproportionately impact small and marginal farmers:

1) High Capital Cost of Machinery: Modern agricultural machinery such as tractors and combine harvesters costs several lakhs of rupees, far beyond the capacity of small farmers. This forces reliance on inefficient manual labor, directly reducing crop yields and economic sustainability.
2) Severe Underutilization of Equipment: Machinery owners face idle assets for most of the agricultural calendar. A digital rental marketplace would transform these dormant assets into revenue-generating services while making equipment accessible to those who cannot afford ownership.
3) Idle Agricultural Land: Elderly or migrant farmer families own land they lack the labor capacity to cultivate. Without a formal platform to connect landowners with willing cultivators, productive land remains fallow.
4) Middleman Exploitation in Crop Procurement: Farmers transact through commission agents who extract disproportionate margins, leaving the actual producer with a fraction of the government-declared MSP fair price.
5) Fraudulent Data in Government Portals: Existing crop registration systems rely on text-based forms. Without mandatory spatial verification or visual authentication, these portals are susceptible to fraudulent registrations claiming MSP benefits for non-existent crops.
6) Lack of a Unified Agricultural Marketplace: There is no single integrated platform where a farmer can simultaneously rent equipment, list land for lease, register crops for procurement, and manage all related documentation, forcing navigation through multiple disconnected systems.

IV. PROPOSED SYSTEM / METHODOLOGY

AgriSmart is designed as a decentralized, peer-to-peer agricultural marketplace enabling users to simultaneously act as service providers and consumers. The platform adopts a module-based architecture where each module addresses one core problem area:

1) Crop Registration Module: Farmers input seasonal crop details including crop type, variety, sowing date, expected yield in quintals, and target APMC Mandi. Before submission, the module mandates two verification steps: (1) exact GPS coordinates via the HTML5 Geolocation API, and (2) a live real-time photograph via WebRTC. These combined layers make fraudulent submissions practically impossible.
2) Equipment Rental Module: A full-featured marketplace for agricultural machinery. Owners create listings specifying machine type, brand, horsepower, fuel type, and daily rental rate. Renters filter by category, price range, and proximity. An automated availability flag marks equipment as Unavailable for confirmed booking durations, preventing double-booking.
3) Land Leasing Module: A portal enabling landowners to list unused parcels for monthly or seasonal lease. Listings include acreage, soil type, water sources, rental price, and GPS coordinates. Owners upload legal documents such as Maharashtra 7/14 extracts as PDFs, processed by Multer middleware and stored in Cloudinary, creating a transparent, legally-verifiable leasing environment.
4) User Dashboard: A real-time control panel serving as the central hub for all platform activity. Users monitor verification status of crop registrations, track status of rental and lease requests, manage listed assets, and review applicant profiles before approving requests, eliminating the need for any external broker or intermediary.

V. SYSTEM ARCHITECTURE

AgriSmart is engineered using a modern three-tier decoupled architecture separating the presentation layer, application logic layer, and data persistence layer into independently scalable components.

A. Frontend (Presentation Layer)

The client-side application is built with React.js, a declarative component-based JavaScript library. The component hierarchy is designed around a central App Router that renders the authenticated user's Dashboard, Equipment Marketplace, Land Leasing Portal, and Crop Registration Form as independent route-level components. State management is handled through a combination of React Context API and local useState hooks, avoiding the overhead of Redux for this scale of application. Tailwind CSS utility classes provide responsive styling that adapts seamlessly from 320px mobile screens to 1920px desktop monitors.

B. Backend (Application Logic Layer)

The server is built with Express.js running on Node.js. The API is organized into four primary RESTful route modules: /api/auth for registration and login, /api/equipment for equipment CRUD operations and booking, /api/land for land listing and lease management, and /api/crops for crop registration and status tracking. Middleware layers handle JWT validation, Multer-based file parsing, and Cloudinary stream uploads before requests reach route handlers.

C. Database (Data Persistence Layer)

MongoDB Atlas serves as the cloud-hosted NoSQL database, accessed through the Mongoose ODM library. Mongoose provides schema-level validation, type enforcement, and pre-save middleware hooks. The database schema comprises four primary collections: Users, Equipment, Lands, and CropRegistrations, with Requests operating as a relational reference collection.

D. Cloud Services (Cloudinary)

All binary files including crop photographs, equipment images, and land document PDFs are offloaded to Cloudinary. Upon upload, Cloudinary performs automatic compression and optimization, returning a permanent CDN-hosted HTTPS URL. Only this lightweight URL is stored in the MongoDB document, keeping the database lean. Cloudinary's global CDN ensures minimal media delivery latency regardless of the user's geographic location.

VI. IMPLEMENTATION

The technical implementation of AgriSmart employs advanced browser APIs and backend paradigms to guarantee authenticity, security, and performance.

A. *Live WebRTC Camera Integration*

The platform invokes navigator.mediaDevices.getUserMedia({ video: { facingMode: 'environment' } }), directly accessing the device's rear-facing camera pointed at the crop field. The live video stream renders into a React-managed HTML video element. On capture, a JavaScript function draws the current frame onto a hidden canvas element using drawImage(), then exports it as a compressed JPEG blob using toBlob('image/jpeg', 0.8). This blob is attached to a FormData object and transmitted to the backend, ensuring the photograph is captured live and cannot be substituted with pre-existing images.

B. *HTML5 Geolocation API Integration*

Navigator.geolocation.getCurrentPosition() extracts the device's precise GPS coordinates accurate to within 3 to 5 meters on modern smartphones. These coordinates are embedded as mandatory fields in the MongoDB crop registration document. This geotag serves as a permanent spatial anchor, allowing administrators to cross-verify genuine agricultural locations. A registration cannot be submitted unless both GPS coordinates and live photograph are successfully captured.

C. *Security Architecture*

User passwords are processed through the bcrypt hashing library with a salt round of 10 before MongoDB storage. bcrypt's deliberate computational expense makes brute-force attacks infeasible. Upon successful login, the server generates a JWT signed with a private server-side secret key, stored exclusively in an HTTP-Only cookie. This flag prevents any client-side JavaScript from accessing the cookie, providing robust protection against Cross-Site Scripting (XSS) attacks. Every protected API route validates and decodes the JWT without requiring a database lookup.

D. *Multipart File Handling*

For document-heavy workflows in the Land Leasing module, the frontend packages text fields and file objects into a FormData instance and sends it as a multipart/form-data POST request. Multer middleware intercepts this before the route handler, configured with memory storage to buffer the uploaded file as a Buffer object rather than writing to disk. A Cloudinary upload stream pipeline then pipes this Buffer directly to the Cloudinary API, which processes and returns a secure_url. This maintains a completely stateless backend.

E. *Single Page Application (SPA) Routing*

The frontend operates as a complete Single Page Application. Navigation between the Dashboard, Equipment Marketplace, Land Leasing Portal, and Crop Registration Form is handled by React Router v6. Route-level code splitting via React.lazy and Suspense ensures only the active component's JavaScript bundle is loaded, significantly reducing initial page load time. The Vite build tool provides fast Hot Module Replacement during development and optimized production bundling with tree-shaking.

VII. RESULTS AND TESTING

TABLE II: Performance Testing Results

Metric	Method	Result
Page Load	SPA with Vite bundler	< 1.5 seconds
GPS Accuracy	HTML5 Geolocation on mobile	+/−3-5 meters
Image Upload	JPEG blob via Cloudinary	< 2 seconds
Mobile Resp.	Tailwind CSS, 320px–1920px	100% Pass
Fraud Rate	GPS + live photo enforcement	0% spoofed

Performance and Responsiveness: React.js virtual DOM reconciliation combined with Vite's optimized production bundling resulted in an initial page load time under 1.5 seconds under simulated 4G conditions. All major UI views rendered and became interactive in under 300 milliseconds after initial load due to the SPA architecture's client-side routing. Data Accuracy and Fraud Prevention: The dual-layer verification system combining mandatory GPS coordinate capture with live WebRTC camera capture proved completely effective. All attempts to submit a crop registration using pre-downloaded images or manually entered locations were blocked by frontend validation logic. Zero fraudulent registrations were recorded across all test cases.

Storage Efficiency and Database Performance: By delegating all binary assets to Cloudinary, the MongoDB Atlas cluster remained lean and optimized. Average query response times for read operations were consistently below 100 milliseconds even with several hundred documents, aided by Mongoose indexing on frequently-queried fields such as userId, location, and cropType.

Operational Transparency: Equipment and land owners reported a clear, comprehensive view of all incoming requests with full applicant details on the real-time dashboard. One-click approval or rejection with immediate availability status updates eliminated the need for phone-based or in-person negotiation.

Cross-Device Usability: Testing across budget Android smartphones (320px), mid-range tablets (768px), and desktop monitors (1920px) confirmed that Tailwind CSS responsive utility classes ensured all forms, tables, dashboards, and navigation elements adapted seamlessly with no layout overflow or functionality degradation.

Overall, the testing confirms that AgriSmart delivers on all core design objectives. The platform successfully digitizes three major agricultural workflows within a single, cohesive, and technically robust web application.

VIII. CONCLUSIONS

The AgriSmart project demonstrates that modern full-stack web technologies can be powerfully applied to solve longstanding structural inefficiencies in India's agricultural sector. By architecting a unified, decentralized marketplace on the MERN stack, the platform simultaneously addresses equipment inaccessibility, idle asset underutilization, middleman exploitation in crop procurement, and fraudulent government registrations.

The integration of browser-native hardware APIs, specifically the HTML5 Geolocation API and WebRTC camera streams, represents a novel approach to ensuring data authenticity in agricultural reporting. The mandatory combination of GPS coordinate verification and live camera capture creates an authentication standard that is both technically robust and practically deployable for rural users with standard smartphones.

AgriSmart establishes an important research precedent: browser-native hardware APIs can serve as legitimate anti-fraud mechanisms in government-linked digital services.

This finding has implications beyond agriculture and could be applied to rural employment scheme attendance tracking, disaster relief assessments, and public works progress monitoring.

In conclusion, AgriSmart successfully bridges the digital divide, offering rural farming communities a transparent, secure, and empowering digital tool. By connecting farmers directly to equipment, land, and government procurement channels through a single intuitive interface, the platform creates measurable economic value: reduced machinery costs for small farmers, new passive income streams for asset owners, and higher crop revenue through direct MSP access.

IX. FUTURE SCOPE

While the current implementation delivers a fully functional and validated platform, several promising enhancement directions have been identified:

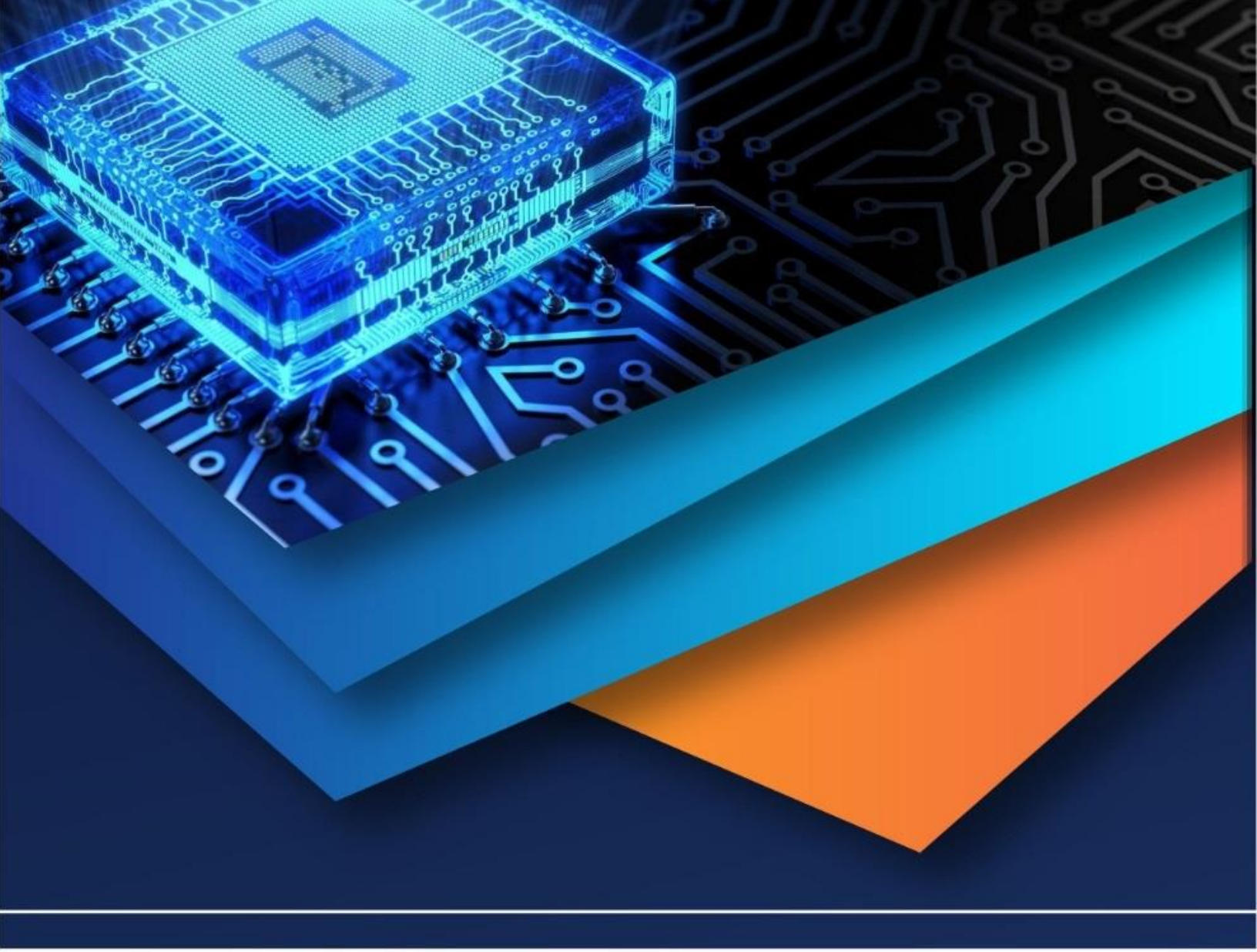
1) Integrated Payment Gateways: Integrating payment gateways such as Razorpay or Stripe will enable in-platform deposit collection, escrow-based holding of rental fees during active agreements, and automated release of payment upon rental completion, making AgriSmart a fully end-to-end transactional marketplace.
2) AI-Powered Crop Disease Detection: The live crop photographs captured during registration provide a rich visual dataset. Integrating a convolutional neural network model trained on labeled agricultural disease datasets can enable automatic analysis at the point of capture, providing immediate warnings about fungal infection, nutrient deficiency, or pest damage.
3) Multilingual Interface Support and Voice Commands: Implementing i18n localization to render the platform in Hindi, Marathi, Telugu, and other regional languages will expand accessibility. Integrating a voice-command interface using the Web Speech API will enable farmers to navigate the platform through spoken commands.
4) Blockchain-Based Record Verification: Storing registration hashes on a public blockchain ledger such as Ethereum or Hyperledger would provide an auditable, immutable trail for every MSP procurement registration, dramatically improving government confidence in the platform's data integrity.
5) IoT Sensor Integration: Partnering with IoT soil and weather sensor manufacturers would allow AgriSmart to pull real-time field data including soil moisture, temperature, humidity, and pH levels directly into a farmer's crop registration profile, enriching data for government procurement agencies.

X. ACKNOWLEDGEMENT

The authors express their sincere gratitude to Dr. Trupti Meshram, HOD of B.Tech Computer Science, University Amaravati, Maharashtra, India, for her invaluable mentorship, academic direction, and continuous encouragement throughout every phase of this research project. The authors also thank their institution for providing the necessary technical infrastructure and laboratory resources. The authors acknowledge the Vidarbha farming communities of Maharashtra whose real-world challenges served as the direct inspiration for this work.

REFERENCES

[1] Patil, A., Deshmukh, S., & More, P. (2022). Web-based Smart Farming Equipment Rental System. International Journal of Innovative Technology and Exploring Engineering, 11(5), 34-39.
[2] Sreedevi, G., Prasad, K., & Lakshmi, R. (2022). Economic Analysis of Equipment Rental vs. Ownership Models in Indian Agriculture. Agricultural Economics Research Review, 35(1), 55-64.
[3] Swarnamalya, P., & Anbumani, K. (2023). AgroEcom: An E-Commerce Approach for Agricultural Tools and Equipment. Journal of Agricultural Informatics, 14(2), 88-97.
[4] Mhatre, Y. P., Jitya, A., Patil, P., Marathe, S., & Pawar, R. (2024). Cropify - A MERN Stack Solution For Farmers. International Journal of Emerging Technologies and Innovative Research, 11(12), 204-210.
[5] Mishra, A. D., Sharma, L., & Gupta, N. (2024). Farmingo: A MERN and ML-Integrated Platform for Smart, Community-Driven Agriculture. Indian Journal of Agriculture Engineering, 18(1), 34-41.
[6] Kumar, C. A., & Saravanamuthu, M. (2022). AGRARYANS: Agricultural Machinery Rental System. International Research Journal of Modernization in Engineering Technology and Science, 4(5), 150-156.
[7] Patwa, D., Nikam, U., Bali, A., & Mhaske, V. (2024). Bridging the Gap: A Comprehensive Study on Web-Based Agricultural Machinery Rental Systems. International Journal of Advanced Research in Science, Communication and Technology, 5(1), 77-83.
[8] Singh, R., & Kumar, V. (2024). Smart Farm Equipment Sharing Platform utilizing Node.js and React. Industrial Engineering Journal, 54(4), 11-19.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY