IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

Kisan Sahayak: A Smart Agricultural Marketplace for Connecting Farmers, Labourers, and Machinery Providers

Naik Balkrushna Anil, Najan Ramesh Somnath, Nagargoje Sushant Dattatray, Kolhe Aditya Jalindar
Prof. Medhe S. S., Dr. Jadhav H. B.
Students, Department of Computer Engineering
Project Guide, Department of Computer Engineering
Head of Department (HOD), Department of Computer Engineering
Adsul Technical Campus, Chas
Savitribai Phule Pune University, Maharashtra, India.
balkrushnanaik9322@gmail.com, nagargojesushant@gmail.com,
najanramesh090@gmail.com, adityakolhe8083@gmail.com,

Abstract: *Agricultural activities depend on the timely coordi-nation of land, labour, machinery, information, and local services. In many rural settings, farmers identify workers and equipment through personal contacts and fragmented communication chan-nels. Such methods can make it difficult to discover available service providers beyond an immediate network, communicate work requirements consistently, and maintain a clear record of requests. Agricultural labourers and machinery owners can face a corresponding difficulty in communicating their availability to potential customers.*

This paper presents Kisan Sahayak, a theoretical, mobile-first agricultural marketplace designed to connect farmers, labourers, and machinery providers through a common web platform. The proposed system is organized around the MongoDB, Express.js, React.js, and Node.js (MERN) stack. Its conceptual capabilities include role-based profiles, location-aware discovery, farmer job postings, direct worker invitations, request acceptance or rejec-tion, wage negotiation, machinery listings, and service-request tracking. An optional artificial-intelligence assistance layer is proposed for multilingual platform guidance and explainable matching based on location, skills, availability, and stated pref-erences.

The paper details the system architecture, data model, stake-holder workflows, marketplace rules, AI design, security and privacy controls, deployment considerations, and an evaluation framework. The contribution is an architectural and process model; no field trial, measured performance result, or im-plemented AI capability is claimed. The design is intended to provide a foundation for subsequent implementation and empirical evaluation in agricultural communities.

Keywords: Agricultural marketplace, digital agriculture, MERN stack, agricultural labour, machinery rental, location-based services, artificial intelligence, multilingual systems, rural digital platforms.

I. INTRODUCTION

Agriculture is shaped by decisions about when work should be performed and whether the required people, equipment, and services can be arranged at an appropriate time. Alongside seeds, water, inputs, and market information, access to labour and machinery is an important operational requirement. A farmer may need workers for a short-duration activity, a specific skill, or a particular date. Another farmer may need equipment for a limited period rather than purchasing it. Labourers and equipment owners, meanwhile, need a way to communicate their availability and receive relevant enquiries. In many communities, these exchanges are coordinated through family networks, local

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

intermediaries, telephone calls, and direct visits. These channels remain valuable and should not be assumed to be ineffective. However, they may provide limited visibility beyond established contacts and can leave details such as work duration, wage, location, availability, and request status distributed across conversations. The opportu-nity considered in this paper is therefore not to replace local relationships, but to provide an additional structured channel through which agricultural service demand and supply can be discovered and coordinated.

Research on information and communication technologies (ICTs) in agriculture describes how digital communication can improve access to information and alter the way agri-cultural participants interact with markets and services [1]–[4]. Research on smart farming further identifies data integra-tion, connectivity, governance, and the organization of digital ecosystems as important considerations [5], [6]. These studies motivate the use of digital platforms, while also indicating that technology adoption depends on local infrastructure, skills, trust, and institutional context.

This paper proposes Kisan Sahayak, a web-based market-place that brings three stakeholder groups into one coordinated environment: (i) farmers who seek labour or machinery, (ii) labourers who seek agricultural work, and (iii) machinery providers who offer equipment or related services. The in-tended access model is mobile-first, with a responsive web interface and support planned for English, Marathi, and Hindi. Users may provide a current location when they choose to do so or enter a village or locality manually. The proposed first release records payment-related details where needed but does not process in-app payments. This distinction keeps the initial marketplace focused on discovery and work coordination rather than financial settlement.

The project reference website is https://kisansahayak.com/.

The link is supplied as the project reference; this paper does not infer that every capability described here is currently implemented or independently validated.

A. Problem Definition

The design problem is to provide a common digital process for publishing agricultural service requirements, finding rele-vant providers, exchanging requests, and tracking the outcome. A usable solution must accommodate users with different roles and levels of digital familiarity. It must also treat location as sensitive information, avoid exposing private contact details unnecessarily, and make clear that an online listing does not guarantee the provider's current availability.

B. Objectives and Contributions

The proposed objectives are:

- Define one marketplace model for farmers, labourers, and machinery providers.
- Specify role-based registration, profiles, dashboards, and access permissions.
- Support both public job posting and direct invitations to selected labourers.
- Provide a request lifecycle covering pending, accepted, rejected, negotiation, in-progress, completed, and can-celled states.
- Describe location-aware discovery with a manual-location alternative.
- Propose a bounded multilingual AI assistant and an explainable matching model.
- Identify security, scalability, accessibility, and evaluation requirements for a future implementation.

The paper's contribution is a conceptual system architecture and workflow specification. It is not an experimental study and does not claim measured improvements in time, income, productivity, or employment outcomes.

II. BACKGROUND AND RELATED WORK

A. ICT and Agricultural Information Access

Aker reviewed ICT-based agricultural extension approaches in developing countries and discussed mechanisms through which voice, text, internet, and mobile services can support information exchange [1]. Nakasone, Torero, and Minten surveyed ICT applications in agricultural development and emphasized that observed impacts can vary across

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

contexts and service types [4]. These works suggest that a digital service should be designed around the information or coordination problem it intends to solve rather than assuming that connec-tivity alone produces benefits.

Jensen's study of mobile phones in the South Indian fish-eries sector showed how improved information flows can affect market coordination [2]. Although fisheries are not the same as agricultural labour markets, the study provides a relevant example of how communication technologies can reduce in-formation frictions between geographically distributed partici-pants. Goyal's analysis of internet kiosks and market access for farmers in central India similarly illustrates the importance of information channels in rural market participation [3]. Kisan Sahayak applies this broad information-access idea to the narrower coordination of labour and machinery services.

B. Digital Agriculture and Automation

Wolfert et al. reviewed big-data developments in smart farming and discussed data infrastructure, integration, gover-nance, and business-model concerns [5]. Klerkx et al. exam-ined the social-science dimensions of digital agriculture and identified changes in roles, institutions, and relations as rele-vant topics for future research [6]. The Food and Agriculture Organization's 2022 report on agricultural automation also dis-cusses the potential of automation while identifying barriers to inclusive adoption, including digital skills, infrastructure, and affordability [7]. These sources support a cautious approach: the platform should complement existing agricultural practices and be designed for uneven access to devices, connectivity, and digital literacy.

C. Technology Acceptance and Usability

The Technology Acceptance Model associates technology use with perceptions of usefulness and ease of use [8]. Rogers' diffusion-of-innovations framework discusses how innovations spread through social systems and how adoption is influenced by communication channels and perceived attributes [9]. For Kisan Sahayak, these perspectives imply that role-specific screens, familiar terminology, minimal data entry, clear status messages, and regional-language support are functional re-quirements rather than decorative enhancements. The interface should let users complete tasks without understanding the underlying software architecture.

D. Positioning of the Proposed Platform

Agricultural digital services may focus on extension in-formation, commodity prices, input supply, farm manage-ment, automation, or market transactions. Kisan Sahayak is positioned around short-cycle service coordination: finding a labourer, inviting a worker, posting a job, or requesting machinery. Unlike general information services, its central unit is the service request and its lifecycle. This is a conceptual positioning, not a measured competitive analysis of named platforms.

The proposed design does not claim that no existing system offers similar features. Its focus is to specify how three related stakeholder experiences can share identity, discovery, request, and notification services while retaining different permissions and dashboards.

III. SYSTEM REQUIREMENTS AND CONCEPTUAL MODEL

A. Stakeholders and Roles

The marketplace defines three operational roles. A farmer creates service demand and may browse labour or machinery listings. A labourer publishes skills and availability and re-ceives work requests. A machinery provider publishes equip-ment details and handles enquiries or booking requests. A user account is associated with one selected role in the initial interaction model; future support for multiple roles can be added through explicit role switching and permission checks.

B. Functional Requirements

The proposed functional requirements are summarized be-low:

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

• Identity: registration, login, logout, profile maintenance, and account recovery.
• Farmer services: create a job post, browse labour pro-files, invite a worker, request machinery, and review request status.
• Labour services: maintain skills and availability, view relevant jobs and invitations, accept or reject requests, and propose revised terms.
• Machinery services: create equipment listings, manage availability, review enquiries, and update request status.
• Shared services: location selection, language preference, notifications, support information, and basic activity history.

C. Non-Functional Requirements

The main non-functional requirements are usability, ac-cessibility, security, privacy, reliability, maintainability, and performance. Product-quality characteristics can be organized using the ISO/IEC 25010 quality model [14]. In practical terms, the application should remain understandable on small screens, validate user input, protect account and location information, provide recoverable error messages, and retain consistent request states. The system should also support manual location entry and avoid making location permission a condition for basic account access.

D. Operating Assumptions and Boundaries

The system assumes access to a web-capable device and occasional connectivity, but not continuous network availability or high digital literacy. Availability is user-maintained information and should be confirmed through a request-response process. The initial version records agreed wages or rental charges for reference only; it does not collect, hold, or transfer money. Dispute resolution, identity verification depth, insurance, and formal service guarantees require policy decisions beyond the core technical design.

IV. PROPOSED MERN-BASED SYSTEM ARCHITECTURE

The proposed architecture separates the presentation layer, application services, persistence layer, and optional external services. The MERN stack is selected as a coherent JavaScript-oriented implementation approach: React.js supports the browser interface, Node.js provides the server run-time, Express.js structures HTTP APIs and middleware, and MongoDB stores application documents. The stack choice is a design proposal; the actual deployment topology may evolve as operational requirements become clearer.

A. Architecture Overview

Fig. 1 illustrates the logical architecture. Users interact with role-specific React components. Requests are sent to an Express API running on Node.js. The backend applies authentication, authorization, validation, and business rules before reading or updating MongoDB. Location services and an AI provider are treated as external integrations behind backend-controlled interfaces. The client does not connect directly to the database or hold private AI service credentials.

B. React.js Presentation Layer

React.js is proposed for building reusable interface compo-nents and role-specific pages [17]. Components may include a registration form, login form, farmer dashboard, labour dashboard, machinery dashboard, job card, equipment card, request-status panel, notification panel, and language selector. The component model supports the separation of interface structure from data and interaction logic [17].

After login, the application retrieves the user's authorized profile and renders the appropriate dashboard. A farmer dash-board may prioritize nearby labour and machinery listings, job creation, and active requests. A labour dashboard may prioritize new invitations, available jobs, profile skills, and completed work records. A machinery dashboard may pri-oritize equipment listings, availability, and incoming service requests. Shared visual patterns should be used to

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

reduce learning effort, while the available actions remain role-specific. The interface should be responsive and should avoid dense data tables as the primary mobile interaction. Forms should ask only for information needed for a task. For example, a job post may request work category, short description, location, date or time window, number of workers, proposed wage, and relevant conditions. Clear validation messages should explain missing or inconsistent values. Regional-language support should include interface labels and system messages, not merely translate the chatbot.

C. Node.js and Express.js Application Layer

Node.js provides the server-side JavaScript runtime and an asynchronous I/O model for database and network operations [18]. Express.js is proposed to organize API routes, middle-ware, controllers, and service functions [19].

A request should pass through a defined processing se-quence: route matching, authentication, role authorization, input validation, business-rule execution, data access, and response serialization. Controllers should remain thin and delegate business decisions to services. This separation helps ensure that the same request rules are applied whether an action originates from a browser form, a future mobile client, or an authorized integration.

Representative API groups include /api/auth,
/api/users, /api/jobs, /api/labour,
/api/machinery, /api/requests, and
/api/notifications. These names are illustrative route groups, not a claim that such endpoints have already been implemented. Write operations should verify ownership

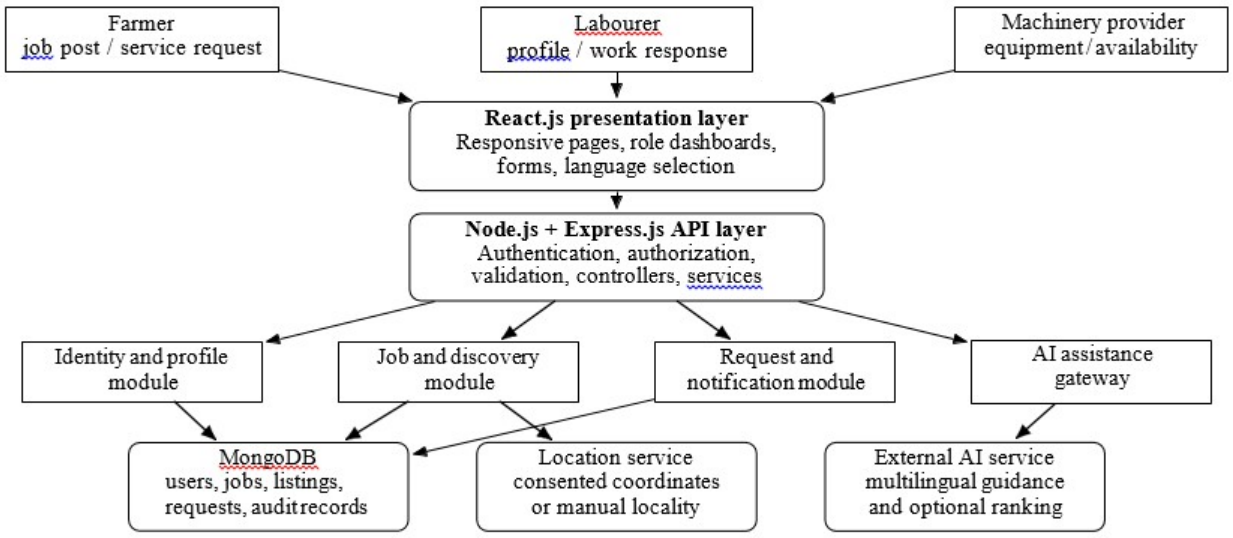


Fig. 1. Logical architecture of the proposed Kisan Sahayak platform. External services are accessed through backend-controlled interfaces.

and role permissions on the server. A farmer must not be able to update another farmer's job merely by changing an identifier in a request.

D. MongoDB Data Model

MongoDB is proposed for document-oriented storage [16]. A conceptual data model may contain the following collections:

- users: account identifier, name, contact details, role, preferred language, authentication metadata, and profile status;
- labourProfiles: user reference, skills, experience descrip-

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

E. Location and Discovery Service

The Geolocation API enables a web application to request device-associated location information, subject to browser permissions and context requirements [11]. Kisan Sahayak should request location only when a feature needs it and should explain why it is being requested. Users who decline permission should still be able to enter a village, locality, or postal area manually.

For coordinates, the backend may calculate approximate distance using the Haversine formulation. For two points with latitudes $\phi 1$, $\phi 2$ and longitudes $\lambda 1$, $\lambda 2$, let $\Delta\phi = \phi 2 - \phi 1$ and

$\Delta\lambda = \lambda 2 - \lambda 1$. Then

a = sin2 (□Δϕ)□ + cos(ϕ) cos(ϕ) sin2 (□Δλ)□ , (1)

locality, optional coordinates, and availability;

• jobs: farmer reference, work category, description, num-ber of workers, proposed wage, location, work date, and publication status;

• workRequests: job or invitation reference, sender and re-cipient identifiers, proposed terms, response, negotiation history, timestamps, and state;

• machineryListings: provider reference, equipment cate-gory, description, rental terms, service area, availability windows, and listing status;

• notifications: recipient, event type, related record, read status, and creation time.

The model should use stable identifiers and references be-tween records. Frequently queried fields, such as role, request state, publication status, and location, may require indexes. Geospatial indexes can support proximity queries when coor-dinates are available. Data validation should be performed both at the application boundary and, where appropriate, through database schema validation. The database must not store raw passwords; a password-hashing mechanism should be used before persistence.

$$d = 2R \operatorname{atan2}(\sqrt{a}, \sqrt{1-a}), \quad (2)$$

where R is the Earth's mean radius and d is the estimated great-circle distance. Coordinates should be normalized to ra-dians before calculation. Distance is only one discovery factor; it does not establish service quality or actual availability.

V. MARKETPLACE DYNAMICS AND STAKEHOLDER WORKFLOWS

The platform combines two labour-discovery paths: a farmer may publish a job for eligible labourers to discover, or may send a direct invitation to a selected worker. Machinery discov-ery follows a listing-and-request pattern. All paths converge on a common request-management model so that users can see the status of their interactions.

A. Farmer Workflow

A farmer registers, chooses the farmer role, and completes a profile containing the minimum required contact and locality information. From the dashboard, the farmer may browse labour profiles, filter by work category and approximate dis-tance, and inspect information made public by the worker. The farmer may then create a job post or issue a direct invitation. A job post can include work type, description, number of work-ers, date, proposed wage, location, and optional instructions. An invitation is directed to a selected labourer and contains the proposed terms.

The farmer can also browse machinery listings and submit a request describing the intended date, duration, and service location. The provider's response is recorded as a separate request state. Any wage or rental amount is stored as an agreed or proposed record for coordination; the initial design does not execute a financial transaction.

B. Labourer Workflow

A labourer registers and supplies skills, work categories, preferred locality, availability, and optional wage expectations. The dashboard presents invitations and relevant public job posts. The labourer can accept, reject, or propose revised terms. Negotiation should be represented as a sequence of explicit proposals rather than silently

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

overwriting the original offer. Once terms are accepted, the request may move to an in-progress state and later be marked completed by an authorized participant, subject to the platform's final completion policy.

C. Machinery Provider Workflow

A machinery provider creates a profile and publishes equip-ment information, service area, rental terms, and availability. Farmers discover the listing and submit service enquiries. The provider may accept, reject, or request clarification. The listing should distinguish a general availability statement from a confirmed booking; a request is not a confirmed arrangement until the provider responds.

D. Request State Model

A common state model improves consistency across labour and machinery interactions. The proposed states are: pending, accepted, rejected, negotiation, in progress, completed, and cancelled. State changes should be validated on the backend. For example, a completed request should not return to pending through an ordinary client update. Cancellation rules should identify which participants may cancel and whether a reason is required.

Fig. 2 presents the conceptual labour-request path. The same request-control pattern can be reused for machinery enquiries with provider-specific actions.

E. Marketplace Integrity and Trust

A marketplace depends on accurate listings and predictable interaction rules. The design should show when a profile was last updated, allow users to report inaccurate or abusive content, and maintain an auditable history of request changes. Optional verification indicators should state exactly what was checked; a phone-number check, for example, must not be presented as proof of professional competence. Ratings, if introduced later, should be based on completed interactions and governed by moderation and appeal rules to reduce manipulation and retaliation.

VI. CONCEPTUAL AI ASSISTANCE AND MATCHING

AI is proposed as an optional assistance layer rather than a dependency for core marketplace transactions. The initial use case is a multilingual help assistant that explains registration, profile completion, job posting, invitations, request states, and machinery enquiries. A later use case may rank relevant listings using explicit user preferences. Predictive analytics is a longer-term possibility and should be considered only after an appropriate quantity and quality of consented historical data becomes available.

A. Multilingual Platform Assistant

A user may ask a question in English, Marathi, or Hindi. The frontend sends the message to a backend endpoint. The backend first checks authentication state where personal context is requested, applies rate limits, and constructs a constrained prompt using approved platform documentation. The AI provider returns a candidate response, which is filtered or checked for format and scope before being displayed. API keys and provider credentials remain on the server and are never embedded in browser code.

The assistant should prioritize retrieval from a curated knowledge base containing platform instructions, supported features, and policy statements. It should answer only within its defined scope, indicate when it is uncertain, and direct the user to human support for account disputes or unre-solved service conflicts. It should not generate unverified wage guarantees, agricultural diagnoses, legal conclusions, or claims about provider availability. Where language quality is uncertain, the response should offer a simpler explanation or ask the user to choose another supported language.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

B. Explainable Labour-Matching Model

The matching component is conceptual. It may filter candi-dates using minimum conditions first, such as account status, service area, selected work category, and availability. It may then calculate a normalized suitability score using factors that the farmer has explicitly chosen. One possible model is

$$Si = wdDi + wsKi + waAi + wwWi, \quad (3)$$

where Si is the score for candidate i; Di represents distance suitability; Ki represents skill or work-category suitability; Ai represents availability suitability; and Wi represents compati-bility with the stated wage range. The weights are non-negative and satisfy wd + ws + wa + ww = 1. Each component is normalized to a common scale before aggregation.

A possible distance suitability function is Di = exp(di/τ), where di is the distance and τ controls distance sensitivity. Skill suitability may be based on an explicit match between requested work categories and worker-declared skills. Availability may be binary or graded according to overlap with the requested work window. Wage compatibility may be high-est when the proposed wage falls within the worker's stated range, but the system should not infer a worker's acceptable wage from protected or unrelated personal attributes.

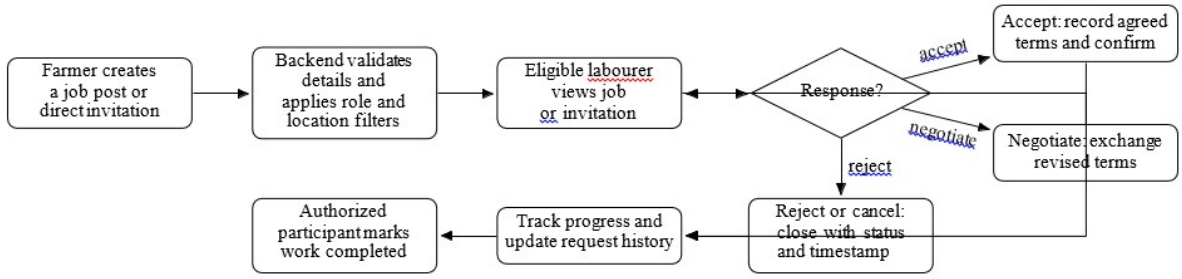


Fig. 2. Conceptual labour-request workflow. Negotiation loops back to the response stage; a machinery request can reuse the same lifecycle with provider-specific labels.

The score should be presented as a ranking aid, not an automatic hiring decision. The interface should show the factors that influenced a recommendation, allow users to adjust filters, and include candidates with incomplete data when appropriate. If weights are not selected by a user, default weights require transparent design and field evaluation. Sparse data should trigger a simpler filter-based result rather than a false impression of algorithmic precision.

C. Machinery Recommendation

Machinery discovery may use a similar multi-criteria ap-proach based on equipment category, service area, availability window, rental terms, and travel distance. Since machinery capacity and compatibility can be technical, the system should present the provider's declared specifications and should not infer that an item is suitable for a particular farm without suf-ficient information. A future version could support structured filters for equipment type, capacity, required attachment, and service duration.

D. Predictive Analytics and Responsible AI

Predictive analytics could eventually estimate demand pat-terns, identify frequently requested work categories, or help providers plan equipment availability. Such models require representative historical data, a clearly defined prediction target, and a strategy for drift and missing information. The design should not use sensitive personal attributes as proxies for worker quality or rank users in ways that systematically exclude less-connected villages. The NIST AI Risk Manage-ment Framework provides a reference for governing AI risks across design, development, deployment, and use [12]. Human review, explanation, feedback, monitoring, and the ability to disable AI recommendations are important safeguards.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40828

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

VII. SECURITY, PRIVACY, AND DATA GOVERNANCE

The platform handles identity data, contact information, location information, work records, and potentially commer-cially sensitive rental terms. Security must therefore be consid-ered as a system property rather than as a single login feature.

A. Authentication and Authorization

Passwords should be processed using a modern password-hashing algorithm with appropriate salts and work factors. Authentication may use secure server-managed sessions or short-lived access tokens with carefully managed refresh mechanisms. Cookies, where used, should be configured with appropriate Secure, HttpOnly, and SameSite attributes. Role-based authorization must be enforced on every protected backend operation. The server should verify record ownership and role permissions rather than trusting identifiers or role labels supplied by the browser.

B. Application and API Security

Input validation should cover types, lengths, allowed values, and relationships between fields. Database queries should use safe parameter handling and should not concatenate untrusted input into query expressions. Sensitive endpoints should have rate limits and abuse monitoring. Transport should use HTTPS, and secrets should be stored in deployment-managed en-vironment variables or a secret manager. Error responses should avoid exposing database details, stack traces, tokens, or personal information. Security review should consider the risk categories documented by OWASP; the current OWASP Top 10 release is a useful awareness reference [13].

C. Location and Personal-Data Privacy

Location permission should be optional and contextual. The application should collect the least precise location that supports the feature, explain the purpose, and avoid exposing exact coordinates publicly when an approximate locality or distance band is sufficient. The user should be able to update or remove optional profile data subject to legitimate record-retention requirements. Access to phone numbers and other contact details should be limited to the intended interaction stage and disclosed according to a clear privacy notice.

D. AI Data Boundaries

Only the minimum information required to answer a query should be sent to an external AI service. Private user records, contact details, and precise coordinates should not be included in general prompts. Marketplace actions such as creating a job, accepting an invitation, or cancelling a request must remain explicit authenticated transactions handled by the application backend. The AI assistant may explain or prepare an action, but it should not perform a consequential operation without a separate user confirmation and server-side authorization.

E. Auditability and Retention

The system should record security-relevant events such as login attempts, role changes, listing updates, request-state transitions, and administrative actions. Audit logs should be protected from ordinary user modification and should avoid retaining unnecessary message content. A retention policy should identify the purpose and duration for each data cat-egory. Backups should be access-controlled, encrypted where supported, and periodically tested for restoration.

VIII. SCALABILITY, RELIABILITY, AND DEPLOYMENT

The initial deployment can use a web client, an application server, and a managed database. Separating these components allows independent configuration and scaling. A deployment should begin with a modular monolithic backend rather than introducing microservices without an operational need. Clear module boundaries can support later extraction of high-load or independently managed services.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

A. API and Database Scalability

The API layer should remain stateless where practical so that multiple application instances can serve requests behind a load balancer. Database indexes should reflect actual query patterns, including user role, listing status, request status, creation time, and location. Pagination should be used for large result sets. Query projections should return only the fields required by the client. Frequently accessed public metadata may be cached, while private or rapidly changing availability data requires careful cache invalidation.

B. Asynchronous Work and Notifications

Notifications, email delivery, image processing, and future analytics may be handled through background jobs when their work does not need to block the user's main request. A queue can support retries and protect the API from slow external integrations. Each asynchronous job should be idempotent or protected against duplicate execution. The platform should distinguish a notification being generated from a notification being successfully delivered.

C. Reliability and Low-Connectivity Use

The interface should communicate network failures and preserve unsent form data where safe. The server should validate any retried request to prevent duplicate job posts or invitations. Idempotency keys may be used for operations where duplicate submission would cause confusion. Offline-first synchronization is outside the initial scope, but a later version may cache non-sensitive reference information and queue selected actions for controlled synchronization. Conflicting edits must be resolved explicitly rather than silently overwriting newer records.

D. Deployment and Operations

A deployment pipeline should separate development, test-ing, and production environments. Build and test steps should run before release, and environment-specific configuration should be injected at runtime. Monitoring should cover server errors, API latency, database availability, queue failures, and external AI-service failures. Backups, database access re-strictions, log retention, and rollback procedures should be documented. The React client can be delivered through static hosting or a content-delivery network, while the Node.js API and MongoDB database can be hosted independently.

E. Quality and Accessibility

Software quality should be evaluated across functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability. ISO/IEC 25010:2023 provides a product-quality model for structuring such evaluation [14]. For accessibility, the design should consider keyboard navigation, readable contrast, scalable text, clear labels, and compatibility with assistive technologies. WCAG 2.2 provides testable accessibility guidance [15].

IX. PROPOSED EVALUATION FRAMEWORK

Because this paper is theoretical, no prototype measure-ments, user-study findings, or performance benchmarks are reported. A future evaluation should combine functional, us-ability, security, and operational methods.

A. Functional Evaluation

Test cases should cover account creation, login, role restric-tions, profile editing, job publication, direct invitations, labour responses, negotiation, machinery listing, service enquiries, notifications, and every permitted request-state transition. Neg-ative tests should verify that users cannot update records they do not own, submit invalid state changes, or access role-restricted endpoints.

B. Usability and Field Evaluation

A field study may recruit farmers, labourers, and machinery providers from multiple localities to complete tasks such as publishing a job, finding a provider, responding to an invitation, and requesting machinery. Measures may include task

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

completion, time on task, error rate, perceived ease of use, language comprehension, and trust. Participant selection, consent, device conditions, connectivity, and test language should be documented.

C. Matching and AI Evaluation

The matching model should be assessed against human-reviewed relevance judgments and practical constraints, not merely against its own ranking score. Metrics may include precision at k, recall at k, ranking stability, and the proportion of recommendations with adequate explanation. Evaluation should be segmented by locality, language, profile complete-ness, and user role to identify unequal performance. The assis-tant should be tested for intent recognition, factual consistency with approved platform documentation, multilingual response quality, refusal behaviour, and leakage of private information.

D. Performance and Security Evaluation

Load testing may examine response time, query latency, throughput, and recovery from service failure. Security testing should include access control, injection, authentication, depen-dency, rate-limit, and secret-exposure checks. Future reports should describe the deployment, dataset, test procedure, and limitations to support reproducibility.

X. POTENTIAL SOCIO-ECONOMIC IMPLICATIONS AND LIMITATIONS

A shared marketplace may create an additional information channel between farmers and service providers. For farmers, structured listings and request histories may make it easier to compare declared service details and keep track of enquiries. For labourers, a profile and request inbox may improve visibility of work opportunities beyond immediate personal contacts. For machinery providers, equipment listings may provide a simple way to communicate service area, rental terms, and availability. These are potential benefits of the proposed design, not proven outcomes.

Recording proposed wages, dates, work categories, and responses may reduce ambiguity, but cannot guarantee fair wages, safe working conditions, or fulfilment of an agreement. Listings should not be presented as verified employment contracts without suitable legal and operational processes.

Adoption barriers include limited access to smartphones, data plans, electricity, or reliable connectivity; preference for voice or assisted onboarding; and incomplete profiles or location data. Ranking may favour highly connected providers unless sparse data is handled carefully. Language support, simple forms, manual location entry, transparent filters, and community onboarding can help address these risks.

As an intermediary, the platform should provide reporting channels, prohibit discriminatory listings, moderate abuse, explain verification limits, and define complaint handling. Its long-term value depends on governance, trust, local partner-ships, and continuing evaluation.

XI. CONCLUSION AND FUTURE SCOPE

This paper presented Kisan Sahayak as a theoretical, mobile-first agricultural marketplace for coordinating farmers, agricultural labourers, and machinery providers. The proposed MERN architecture separates the React presentation layer, Node.js and Express application services, MongoDB persis-tence, and optional location and AI integrations. The market-place model supports both job publication and direct worker invitations, while a shared request lifecycle provides consistent handling of acceptance, rejection, negotiation, progress, com-pletion, and cancellation. The design also considers machinery listings, multilingual interaction, payment-record management without in-app payment processing, and privacy-aware loca-tion discovery.

The proposed AI assistant is deliberately bounded to plat-form guidance in its initial form. Explainable matching and predictive analytics are identified as future possibilities that require representative data, careful governance, and empirical assessment. Security, accessibility, low-connectivity behaviour, and user trust are treated as core design constraints. No performance or socio-economic benefit is claimed without implementation and field evidence.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

Future work should implement the core workflows, vali-date the data model, conduct usability studies with intended users, evaluate regional-language support, assess security, and measure the quality of location-aware discovery. Subsequent versions may consider a dedicated Android application, voice-based assistance, verified provider workflows, and carefully governed AI-supported recommendations. These extensions should be introduced only when their feasibility, user value, and operational risks have been assessed.

PROJECT REFERENCE

Project website: https://kisansahayak.com/. The website is included as the project reference and should be checked sepa-rately before making claims about the live status of individual features.

REFERENCES

[1] J. C. Aker, "Dial 'A' for agriculture: A review of information and communication technologies for agricultural extension in developing countries," Agricultural Economics, vol. 42, no. 6, pp. 631–647, 2011, doi: 10.1111/j.1574-0862.2011.00545.x.
[2] R. Jensen, "The digital provide: Information (technology), market performance, and welfare in the South Indian fisheries sector," The Quarterly Journal of Economics, vol. 122, no. 3, pp. 879–924, 2007, doi: 10.1162/qjec.122.3.879.
[3] A. Goyal, "Information, direct access to farmers, and rural market performance in central India," American Economic Journal: Applied Economics, vol. 2, no. 3, pp. 22–45, 2010, doi: 10.1257/app.2.3.22.
[4] E. Nakasone, M. Torero, and B. Minten, "The power of information: The ICT revolution in agricultural development," Annual Review of Resource Economics, vol. 6, pp. 533–550, 2014, doi: 10.1146/annurev-resource-100913-012714.
[5] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big data in smart farming—A review," Agricultural Systems, vol. 153, pp. 69–80, 2017, doi: 10.1016/j.agsy.2017.01.023.
[6] L. Klerkx, E. Jakku, and P. Labarthe, "A review of social science on digi-tal agriculture, smart farming and agriculture 4.0: New contributions and a future research agenda," NJAS: Wageningen Journal of Life Sciences, vols. 90–91, Art. no. 100315, 2019, doi: 10.1016/j.njas.2019.100315.
[7] Food and Agriculture Organization of the United Nations, The State of Food and Agriculture 2022: Leveraging Automation in Agriculture for Transforming Agrifood Systems. Rome, Italy: FAO, 2022, ISBN: 978-92-5-136043-9.
[8] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," MIS Quarterly, vol. 13, no. 3,
pp. 319–340, 1989, doi: 10.2307/249008.
[9] E. M. Rogers, Diffusion of Innovations, 5th ed. New York, NY, USA: Free Press, 2003.
[10] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, Dept. Information and Com-puter Science, Univ. California, Irvine, CA, USA, 2000.
[11] World Wide Web Consortium, "Geolocation," W3C Recommenda-tion, Mar. 24, 2026. [Online]. Available: https://www.w3.org/TR/2026/ REC-geolocation-20260324/.
[12] E. Tabassi, Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1, National Institute of Standards and Technology, Gaithersburg, MD, USA, Jan. 2023, doi: 10.6028/NIST.AI.100-1.
[13] OWASP Foundation, "OWASP Top 10:2025," 2025. [Online]. Available: https://owasp.org/Top10/2025/.
[14] International Organization for Standardization and International Elec-trotechnical Commission, Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—Product Quality Model, ISO/IEC 25010:2023, Nov. 2023.
[15] World Wide Web Consortium, Web Content Accessibility Guidelines (WCAG) 2.2, W3C Recommendation, Oct. 5, 2023. [Online]. Available: https://www.w3.org/TR/WCAG22/.
[16] MongoDB, Inc., "MongoDB Manual." [Online]. Available: https://www. mongodb.com/docs/manual/.
[17] Meta Platforms, Inc., "React documentation." [Online]. Available: https: //react.dev/learn.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

[18] OpenJS Foundation, “Node.js documentation.” [Online]. Available: https://nodejs.org/docs/latest/api/.
[19] OpenJS Foundation, “Express documentation.” [Online]. Available: https://expressjs.com/.