KRISHISETU: An Integrated Platform for Smart Farming

Pranali Patil[1], Sandeep Munde[2], Vaishnavi Patil[3], Diya Gonjari[4], Ruchira Suryawanshi[5]
[1,2,3,4,5] *CSE, Dy Patil Technical Campus, Talsande* Maharashtra, India

Abstract—**In many rural and semi-urban agricultural regions, small and marginal farmers face significant challenges in access- ing modern farming equipment due to high purchase costs, lim- ited availability, and lack of efficient rental platforms. At the same time, equipment owners often struggle to reach potential renters and manage bookings effectively. To address these challenges, this research presents an Agricultural Equipment Rental System, a digital platform designed to connect farmers with equipment vendors through a centralized and user-friendly system.**

The proposed system enables farmers to browse, search, and rent agricultural equipment such as tractors, harvesters, seed drills, and irrigation tools based on availability and rental cost. Vendors can register their machinery, manage equipment listings, and track bookings through a dedicated interface. An administrative module oversees user management, equipment verification, and booking activities to ensure transparency and reliability within the platform.

The system is developed using modern web technologies including HTML, CSS, and the Django web framework with an SQLite database for efficient data storage and management. This architecture provides a lightweight, scalable, and cost-effective solution for digitizing agricultural equipment rental services. The platform improves resource utilization, reduces operational costs for farmers, and enhances accessibility to modern agri- cultural machinery. Experimental implementation demonstrates improved equipment accessibility, better vendor–farmer connec- tivity, and efficient management of rental operations, contributing to the advancement of technology-driven and sustainable agri- cultural practices.

I. INTRODUCTION

Agriculture plays a crucial role in sustaining rural economies and ensuring food security worldwide. However, small and marginal farmers often face significant challenges in accessing modern agricultural machinery due to high procurement costs, limited ownership, and the absence of organized rental mechanisms. Essential farming equipment such as tractors, harvesters, seed drills, and irrigation systems are vital for improving productivity and efficiency, yet purchasing such machinery is economically unfeasible for many farmers who require them only during specific farming cycles. As a result, farmers frequently depend on traditional tools or informal rental arrangements, which are often inefficient, unreliable, and lack transparency.

In parallel, equipment owners and vendors encounter difficulties in reaching potential customers and effectively man- aging rental operations. The absence of a structured digital platform leads to poor equipment utilization, communication barriers between farmers and vendors, and inefficient booking processes. Farmers also struggle to identify nearby equipment availability, compare rental costs, and schedule machinery during peak agricultural seasons. These challenges highlight the necessity of a centralized and technology-driven solution that can efficiently connect farmers with equipment providers while ensuring transparency and accessibility.

To address these issues, this research proposes an Agri- cultural Equipment Rental System, a web-based platform designed to facilitate efficient sharing and management of agricultural machinery. The system enables farmers to browse available equipment, check rental details, and book machinery based on their requirements. Equipment vendors can register their machinery, update equipment availability, manage book- ings, and monitor rental activities through a dedicated vendor interface. Additionally, an administrative module manages system users, verifies equipment listings, and ensures the reliability and security of the platform.

The proposed system is developed using HTML and CSS for the user interface, the Django web framework for backend processing, and SQLite as the database management system, providing a lightweight, scalable, and cost-effective architecture.

By digitizing the agricultural equipment rental ecosystem, the platform improves machinery accessibility, enhances equipment utilization, reduces operational costs for farmers, and promotes efficient resource sharing. The implementation and evaluation of the system demonstrate its potential to modernize traditional agricultural practices and contribute toward sustainable, technology-enabled farming solutions.

II. RELATED WORKS

In recent years, several research studies and digital plat- forms have been developed to address challenges related to agricultural resource management, equipment accessibility, and technological support for farmers. These systems aim to reduce the financial burden of purchasing expensive machinery and promote the efficient utilization of agricultural equipment through shared or rental-based models. Although these studies provide valuable insights into the digitization of agricultural services, several limitations remain, indicating the need for a more integrated and scalable solution.

Shingte et al. (2025) proposed AgriRent: Agriculture Equipment's Rental System, a web and mobile-based platform designed to facilitate the rental of agricultural machinery. The system enables vendors to list equipment such as tractors and allows farmers to browse and book machinery in real time. While the platform improves accessibility and promotes equipment sharing, the system primarily focuses on the book- ing functionality and lacks advanced management capabilities such as centralized administrative monitoring, detailed book- ing analytics, and efficient vendor–farmer coordination.

Similarly,Chella Ashok Kumar and Dr. M. Saravanamuthu (2022) introduced AGRARYANS – Farm Equipment Rental System, an e-commerce–style platform that allows farmers to register, explore available agricultural tools, and rent equip- ment through an online interface. The system simplifies the rental process and improves accessibility to farming machinery. However, the proposed platform mainly focuses on equipment browsing and rental services and does not pro- vide advanced operational features such as real-time booking management, vendor performance tracking, or system-level supervision.

Another related study by Steven Sam and Silima Marshal D'Abreo (2025) explored Crop Recommendation using Ma- chine Learning, where environmental and economic factors are analyzed to determine optimal crop selection using machine learning algorithms such as Random Forest and Decision Tree. This research significantly contributes to intelligent decision- making in agriculture by assisting farmers in selecting suit- able crops based on data-driven insights. However, the study focuses on crop advisory systems and does not address the practical challenges related to agricultural machinery accessibility and equipment-sharing platforms.

Although these studies contribute important advancements toward digital agriculture, most existing solutions focus on specific aspects such as equipment rental or crop recommendation independently. They lack a comprehensive and centralized platform that effectively integrates farmers, equipment vendors, and administrative management within a unified system. Therefore, there is a clear need for a scalable digital solution that improves equipment accessibility, ensures efficient booking management, and enhances transparency in agricultural equipment rental services. The proposed Agri- cultural Equipment Rental System addresses these limitations by providing a centralized web-based platform where farmers can easily locate and rent machinery, vendors can efficiently manage equipment listings and bookings, and administrators can monitor and regulate overall system operations to ensure reliability and efficiency in the agricultural ecosystem.

Research Gap: Existing agricultural systems mainly focus on either equipment rental or crop advisory services but rarely integrate multiple functionalities into a single platform. Many rental systems lack efficient booking management, vendor–farmer coordination, and administrative monitoring. This limitation highlights the need for a centralized digital solution. The proposed Agricultural Equipment Rental System addresses this gap by providing a unified platform where farmers can easily rent machinery, vendors can manage equipment listings and bookings, and administrators can monitor the overall system to ensure efficient and transparent operations.

III. PROPOSED WORK / METHODOLOGY

The proposed system, Agricultural Equipment Rental System, is designed as a centralized digital platform that enables farmers to easily access and rent agricultural machinery while allowing equipment vendors to list and manage their equipment efficiently. The system provides farmers with the ability to browse available equipment, check rental details, and book machinery based on their requirements. Vendors can update equipment availability, manage bookings, and monitor rental activities through a dedicated interface, while the administrative module ensures proper user management and system monitoring. The platform is developed using modern web technologies including HTML, CSS, the Django framework, and SQLite database, providing an efficient and scalable solution for improving equipment accessibility and resource utilization in the agricultural sector.

A. Backend Development

The backend of the Agricultural Equipment Rental System acts as the core component responsible for handling application logic, data processing, and communication between the user interface and the database. It ensures secure data management, user authentication, and efficient handling of equipment listings, booking requests, and rental records. The backend is implemented using the Django framework, a powerful Python-based web framework known for its rapid development capabilities, scalability, and built-in security features. Django also provides strong support for database integration and RESTful services, enabling reliable interaction between farmers, equipment vendors, and the administrative module within the system.

1) System Architecture: The backend of the Agricultural Equipment Rental System follows a layered architecture consisting of the View Layer, Application (Logic) Layer, Data Access Layer, and Database Layer, ensuring organized and efficient system functionality.

View Layer:Handles user interactions through the web inter- face where farmers, vendors, and administrators can perform actions such as browsing equipment, listing machinery, and managing bookings.

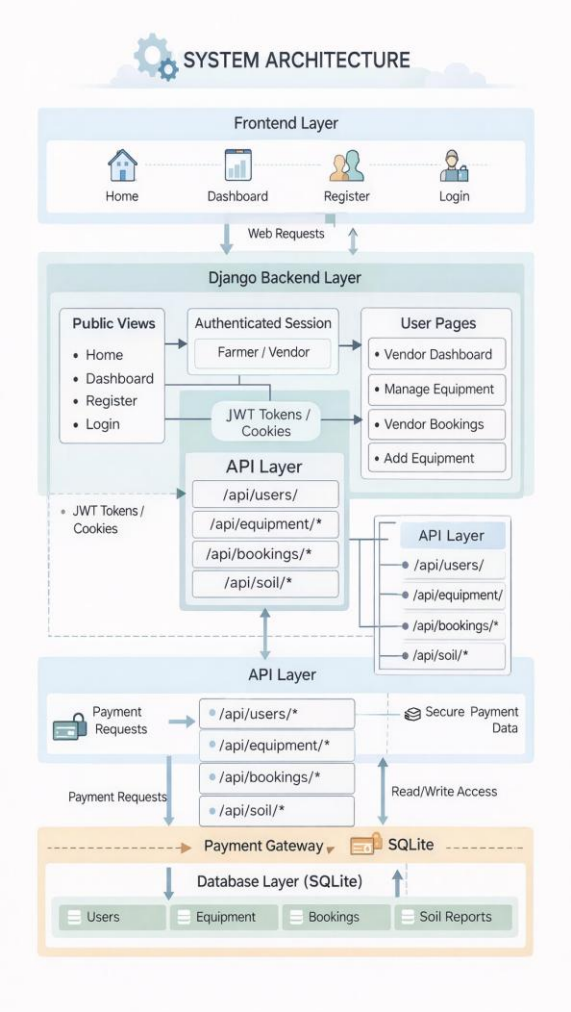

Fig. 1. Backend structure

Application (Logic) Layer:Implements the core business logic of the system, including equipment registration, rental booking management, availability checking, and user authentication.

Data Access Layer:Manages communication between the application logic and the database, performing operations such as storing equipment details, retrieving booking records, and updating rental information.

Database Layer: Utilizes the SQLite database to store and manage system data, including user profiles, equipment listings, booking details, and transaction records.

This layered architecture ensures clear separation of responsibilities, improving system maintainability, scalability, and overall performance.

2) API Endpoints: The backend of the Agricultural Equipment Rental System exposes

several REST-style API endpoints that enable communication between different modules of the system such as user management, equipment management, booking operations, and soil analysis services.

/api/users/ – This endpoint manages all user-related operations within the system. It handles user registration, authentication, profile management, and role identification for farmers and equipment vendors. This ensures secure access control and proper user management.

/api/equipment/ – This endpoint manages agricultural equipment data. Vendors can add new machinery, update equipment details, remove listings, and manage availability, while farmers can view available equipment for rental.

/api/bookings/ – This endpoint manages the equipment rental process. Farmers can create booking requests for avail- able machinery, while vendors can review, approve, or manage those bookings. It also stores booking records for tracking rental activities.

/api/soil/ – This endpoint handles soil analysis functionality. Users can submit soil-related information and receive analysis reports or recommendations that help farmers make better agricultural decisions.

These endpoints follow a request–response architecture and exchange data in structured formats, enabling efficient communication between the frontend interface and the backend system while maintaining modular and scalable system design.

3) Database Management: The system uses SQLite as the database management system, which stores structured data related to users, equipment, bookings, and soil analysis. SQLite is a lightweight and efficient relational database that is well suited for web applications developed using the Django framework.

The data is organized into different tables such as:

users – Stores information about system users including farmers, vendors, and administrators along with their authentication details and roles.

equipment – Contains records of agricultural machinery listed by vendors, including equipment name, description, rental price, availability status, and associated vendor information.

bookings – Stores equipment rental transactions, including booking requests, equipment details, user information, booking status, and rental duration.

soil reports – Maintains soil analysis data and generated reports that help farmers understand soil conditions and make informed agricultural decisions.

SQLite provides reliable data storage, efficient query processing, and seamless integration with Django's ORM (Object Relational Mapping). This ensures efficient data management, secure storage of system records, and smooth interaction between the backend application and the database.

4) AI Integration: To enhance the functionality of the Agricultural Equipment Rental System, the backend integrates a soil analysis module that assists farmers in understanding soil conditions and making better farming decisions. This module processes soil-related data provided by users and generates analysis reports that help improve crop planning and agricultural productivity.

The module performs the following operations: Soil data processing– Collects and processes soil parameters such as pH level, nutrient content, and soil type submitted by users.

Soil condition analysis – Evaluates the provided soil data to determine the quality and fertility of the soil.

Recommendation generation– Provides basic suggestions or reports that help farmers understand soil suitability for agricultural activities.

The soil analysis module is integrated with the backend using the Django framework and is accessed through dedicated API endpoints. The processed results are stored in the SQLite database and returned to the frontend interface, allowing farmers to view soil reports and make informed agricultural decisions.

5) system Workflow: When a farmer requests to rent agricultural equipment:

The booking details are sent from the frontend interface to the backend through an HTTP POST request.

The backend processes the request and stores the booking information in the database with a status such as "Pending".

Equipment vendors can access the booking requests through their vendor dashboard, review the request details, and either approve or reject the booking.

Once the vendor takes action, the system updates the booking status in the database and reflects the updated information on the farmer's dashboard, allowing the user to track the booking status in real time.

6) Security and Authentication: User authentication and authorization in the Agricultural Equipment Rental System are managed using Django's built-in authentication system. The platform ensures secure access control by verifying user credentials and restricting system functionalities based on user roles.

Each module of the system is role-based and accessible only to authorized users:

Farmer:Can register and log in to the system, browse avail- able agricultural equipment, submit booking requests, view booking status, and access soil analysis reports.

Vendor:Can add and manage agricultural machinery, update equipment availability, review booking requests from farmers, and manage rental transactions.

The system enhances security through password hashing, authenticated session management, and protected routes using login-based access control. Additionally, proper request validation and secure communication between the frontend and backend help ensure data integrity and system reliability.

7) Additional Backend Features: Error Handling: The system implements centralized error handling to manage exceptions that may occur during API requests or database operations. This ensures that users receive clear and consistent error messages, improving reliability and user experience.

Logging: Integrated logging mechanisms are used to record important system activities such as user requests, booking operations, and equipment updates. These logs help in monitoring system performance, debugging issues, and maintaining system transparency.

Scalability: The platform is designed with a modular architecture that allows easy expansion and deployment across different environments. This ensures that the system can handle an increasing number of users, equipment listings, and booking transactions without affecting performance.

API Documentation: Clear API documentation is maintained for all backend endpoints, enabling developers to easily understand and interact with the system's services. This documentation supports efficient development, testing, and future system integration.

B. Frontend Development

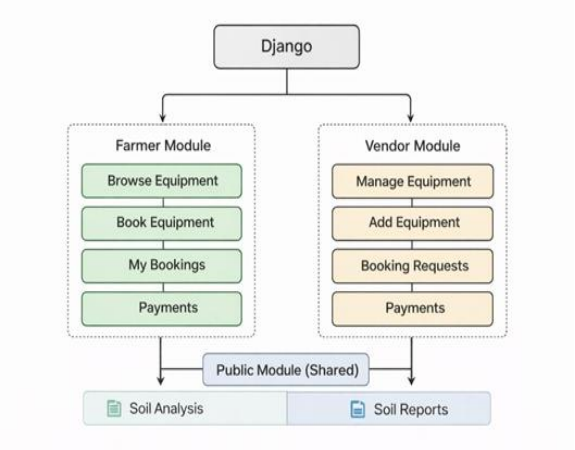

Fig. 2. Frontend structure

The frontend of the Agricultural Equipment Rental System is developed using HTML, CSS, and JavaScript to create a responsive and user-friendly interface. HTML structures the web pages, CSS provides styling and layout design, and JavaScript enables dynamic interactions and communication with backend APIs. This frontend architecture allows farmers and vendors to easily browse equipment, manage bookings, and interact with the system efficiently.

1) User Interface and Navigation: The interface of the Agricultural Equipment Rental System is designed with a clean and intuitive layout to ensure easy navigation for users. It follows a role-based structure:

Farmers can browse available agricultural equipment, re- quest bookings, view rental status, and access soil analysis reports.

Vendors can add and manage equipment listings, update availability, and review booking requests from farmers.

Navigation between different pages is handled smoothly through the web interface, allowing users to access system features quickly without complexity.

2) agriculture system Module: The equipment listing and booking interface is designed using interactive web forms that allow users to easily enter and manage system data. Each major function such as equipment details, booking requests, and soil data submission is handled through separate form sections, making the interface organized and easy to maintain.

User inputs are dynamically processed through

JavaScript to provide real-time interaction with the system. Before sub- mission, form validation ensures that all required fields such as equipment name, rental price, booking details, and user information are properly completed and formatted correctly. This improves data accuracy and enhances the overall user experience. .

3) API Communication and Data Handling: When a user performs an operation such as adding equipment or requesting a booking:

a) The frontend collects the required input fields and pre- pares the data in JSON or form format.

b) An HTTP request (GET, POST, PUT, DELETE) is sent to the Django backend through API endpoints to process the request.

c) During the request process, the system may display a loading indicator to inform the user that the operation is being processed.

d) After processing, the backend sends a response contain- ing the updated information such as equipment details, book- ing status, or soil analysis results, which are then displayed on the user interface.

All requests are processed through authenticated sessions, and role-based access ensures that only authorized users (farmers or vendors) can access specific system functionalities.

4) soil Analysis and Display: After the backend processes user requests such as equipment booking or soil analysis, the system returns the relevant results to the frontend interface. The processed data, including booking status, equipment avail- ability, or soil analysis reports, is dynamically displayed on the user dashboard.

Users can view updated booking information, check equip- ment details, or access generated soil reports directly through the interface. The system also allows users to update infor- mation, cancel requests if necessary, or submit new data for further operations. This dynamic data rendering ensures that users receive immediate feedback and can efficiently interact with the platform.

5) Additional Features: Responsive Design:The interface is designed to adapt to different screen sizes using responsive layout techniques and grid-based design, ensuring smooth access across desktops, tablets, and mobile devices.

Error Handling: The system displays clear and user-friendly error messages whenever validation errors occur or when API requests fail, helping users correct issues easily.

Notifications:The platform provides alerts or notification messages to inform users about important actions such as successful equipment booking, booking updates, or equipment listing changes.

Scalability:The system is developed using a modular struc- ture where key functionalities such as user management, equipment management, booking services, and soil analysis operate as separate modules, allowing easy expansion and future enhancements.

6) Workflow Summary: The user logs in to the system through the authentication portal. After successful login, the frontend displays the appropriate dashboard based on the user's role.

Farmers can browse available agricultural equipment, sub- mit booking requests, and view their booking status. Vendors can add equipment, manage listings, and review booking requests from farmers.

The frontend communicates with the Django backend through API requests to process user actions and retrieve data. The system then updates the interface dynamically to display booking updates, equipment availability, or soil analysis re- sults.

This structured and modular frontend design ensures a smooth user experience, efficient communication with the backend, and scalability for future enhancements such as advanced analytics, improved booking management, and ad- ditional agricultural support features.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

This section presents the evaluation of the Agricultural Equipment Rental System developed for facilitating efficient equipment sharing between farmers and vendors. The goal is to demonstrate the effectiveness of the system in man- aging equipment listings, processing booking requests, and supporting soil analysis through a web-based platform. The evaluation also highlights the performance of the complete sys- tem pipeline, including frontend–backend integration, database management, and real-time data updates within the web ap- plication.

V. CONCLUSION & FUTURE WORK

In this study, we proposed and implemented a web-based Agricultural Equipment Rental System designed to improve accessibility to modern farming machinery for farmers while enabling vendors to efficiently manage their equipment list- ings. The system provides a centralized digital platform where farmers can browse available agricultural equipment, submit rental requests, and track booking status, while vendors can add equipment, update availability, and manage booking re- quests through a dedicated interface.

The platform is developed using HTML, CSS, and JavaScript for the frontend, Django for backend develop- ment, and SQLite for database management, ensuring efficient communication between system components and reliable data handling. The integration of booking management and soil analysis features further enhances the functionality of the system by helping farmers make better agricultural decisions. The proposed system bridges the gap between farmers and equipment providers by digitizing the traditional equipment rental process. By improving equipment accessibility, optimiz- ing resource utilization, and simplifying booking management, the system contributes toward more efficient and technology- driven agricultural practices. The implementation demonstrates that a centralized web platform can significantly enhance transparency, reduce operational challenges, and support sustainable farming operations.

Future Work While the proposed Agricultural Equipment Rental System provides an effective platform for connecting farmers and equipment vendors, several enhancements can be considered for future development to improve its functionality and scalability.

1) Mobile Application Development: Future versions of the system can include a dedicated mobile application to make the platform more accessible for farmers, especially in rural areas where mobile devices are more commonly used than desktop systems.

2) Intelligent Equipment Recommendation: Machine learn- ing techniques can be integrated to recommend suitable agri- cultural equipment based on factors such as crop type, farm size, soil condition, and seasonal requirements.

3) Real-Time Equipment Tracking: Future improvements may include GPS or IoT-based tracking to monitor equipment availability and location in real time, improving transparency and resource utilization.

4) Advanced Payment Features: Although the system al- ready includes payment integration, future improvements could include multiple payment gateways, digital wallet sup- port, automated billing, and transaction history analytics for better financial management.

5) Multi-language Support: Providing multiple language options would help farmers from different regions use the system more easily and improve accessibility for non-English- speaking users.

6) Expansion of Agricultural Services: The platform can be extended to include additional services such as crop advisory systems, weather-based recommendations, and an agricultural marketplace for buying and selling farm products.

By implementing these enhancements, the system can evolve into a comprehensive digital platform that supports modern, technology-driven agriculture and improves efficiency in the farming ecosystem.

REFERENCES

[1] Lefteris Benos, Alexandros C Tagarakis, Giorgos Dolias, Remigio Berruto, Dimitrios Kateris, and Dionysis Bochtis. Machine learning in agriculture: A comprehensive updated review. *Sensors*, 21(11):3758, 2021.

[2] Prem Prakash Jayaraman, Ali Yavari, Dimitrios Georgakopoulos, Ahsan Morshed, and Arkady Zaslavsky. Internet of things platform for smart farming: Experiences and lessons learned. *Sensors*, 16(11):1884, 2016.

[3] Andreas Kamilaris and Francesc X Prenafeta-Boldu´. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147:70–90, 2018.

[4] R Kumar, M Singh, and P Kumar. Smart agriculture system using iot and machine learning. In *International Conference on Intelligent Computing and Smart Communication*, pages 245–250. Springer, 2021.

[5] Konstantinos G Liakos, Patrizia Busato, Dimitrios Moshou, Simon Pearson, and

Dionysis Bochtis. Machine learning in agriculture: A review. *Sensors*, 18(8):2674, 2018.

[6] Neha N Misra, Yashwant Dixit, Ahmad Al-Mallahi, Mahesh S Bhullar, Rakesh Upadhyay, and Alex Martynenko. Iot based smart agriculture system: A review. *Sensors*, 20(12):3523, 2020.

[7] A Pathak, B Mahapatra, and P Patel. Smart agriculture using iot and machine learning. *International Journal of Advanced Science and Technology*, 28(16):321–327, 2019.

[8] Rohit Patil and Shubham Kulkarni. Precision agriculture using machine learning techniques. *International Journal of Agricultural Technology*, 18(3):145–152, 2022.

[9] Tushar Talaviya, Devanshi Shah, Nihar Patel, Hitesh Yagnik, and Manan Shah. Implementation of artificial intelligence in agriculture for op- timization of irrigation and application of pesticides and herbicides. *Artificial Intelligence in Agriculture*, 4:58–73, 2020.

[10] Sjaak Wolfert, Lan Ge, Cor Verdouw, and Marc-Jeroen Bogaardt. Big data in smart farming–a review. *Agricultural Systems*, 153:69–80, 2017.