International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 12 | Dec 2025 www.irjet.net p-ISSN: 2395-0072

RENT IT – LEND IT FOR SMART INDIA PLATFORM

Sushma C[1], Aditya K M[2], Ashish B Kannapla[3], Balaji Vaibhav H V[4], Dhanush M G[5]

[1] *Assistant Professor, Information Science and Engineering, Bapuji Institute of Engineering and technology, Karnataka, India*

[1,2,3,4,] *Bachelor of Engineering, Information Science and Engineering, Bapuji Institute of Engineering and technology, Karnataka, India*

Abstract - *Rent It – Lend It is a digital peer-to-peer rental platform designed to simplify the renting and lending of items through a secure and transparent online system. The platform uses modern web technologies, including Next.js, integrated APIs, and Razor pay payment services to support real-time listing, search, booking, and transaction processing. User authentication ensures verified participation, while structured item data improves discovery and rental efficiency. A MySQL database maintains user records, booking histories, and payment logs securely. During testing, the system delivered consistent performance with smooth navigation and reliable transaction handling. Rent It – Lend It offers a scalable and convenient rental solution across the Smart India ecosystem.*

Key Words*: *Online Rental Platform, Rent It, Product Renting System, Next.js, Node.js, Razorpay Integration, Digital Payments, User Verification, Web Application, Rental Management System, Database Management, Secure Transactions

1. INTRODUCTION

With the rapid expansion of digital services and the shift toward convenient, on-demand access to everyday resources, the need for a structured rental ecosystem has become increasingly significant. Traditional offline renting— dependent on brokers, personal contacts, and unverified sources—can no longer meet the expectations of modern users, resulting in rising cases of fraudulent listings, misleading pricing, delayed responses, and overall inefficiency. Although many individuals initially relied on informal networks or manual communication for renting products, these methods are unreliable, lack transparency, and fail to support real-time availability or secure transactions. The absence of a standardized digital rental system often leads to distrust, limited accessibility, and unequal pricing, ultimately affecting user satisfaction and market growth. Recent advancements in web technologies, secure payment systems, and digital identity verification have enabled the development of modern rental platforms that can simplify and systematize the renting experience.

Leveraging these technologies, this project introduces Rent It, an intelligent and user-friendly online rental management solution designed to connect owners and renters through a transparent, secure, and highly accessible digital interface. The system integrates verified user accounts, real-time product availability tracking, structured pricing models, and secure payment mechanisms to ensure accuracy and reliability. Through these features, Rent It enables safe communication between users, efficient comparison of rental options, instant booking, and complete visibility into all rental activities.

By combining modern web technologies with a scalable and modular architecture, the proposed system offers a reliable, efficient, and versatile solution for managing digital rentals. Its flexible design enables future enhancements such as AI-based product recommendations, automated pricing strategies, and integration with mobile or IoT platforms, allowing the system to evolve with market needs and expand across various rental categories. Rent It represents a contemporary approach to organizing the rental marketplace, ensuring a convenient, secure, and transparent experience for all users involved.

2. PROPOSED SYSTEM

The system provides a fully digitized rental platform where users can browse, book, and pay for items securely, offering an end-to-end rental management solution with real-time availability, verified listings, and automated transaction processes.

Key Components

1. **Product Catalog & Upload Processing Unit:** The owner's product images, details, and rental information are uploaded into the system, where each listing is validated and preprocessed for quality using automated checks. Product data is standardized for clarity, categories, and pricing to ensure accurate display and seamless browsing for customers on the platform.

2. **Product Detection & Verification Module:** This module verifies that uploaded items match their descriptions by analyzing images and metadata for inconsistencies. It flags suspicious or misleading listings, ensuring that only legitimate and high- quality rental products appear on the platform, improving trust and overall user safety.

3. **User Authentication & Identity Validation:** A secure authentication system validates the identity of renters and owners during registration and login. Verification steps help prevents fraudulent accounts, duplicate profiles, and misuse of the platform. This ensures that all transactions occur between genuine, traceable users throughout the rental process.

4. **Real-Time Availability & Booking Tracking:** The system continuously tracks the availability of each rental item. When a customer initiates a booking, the module checks availability, rental duration, and conflicts with other reservations. It updates the item's status instantly, preventing double-booking and ensuring reliable rental scheduling.

5. **Secure Payment & Transaction Monitoring Module:** Razor pay integration processes digital payments through UPI, credit cards, and net banking. Each transaction is encrypted and validated before confirming a booking. Payment anomalies, failed attempts, or mismatches are automatically flagged, ensuring safe, smooth, and verified financial operations.

6. **Order Logging & Rental History Management:** When a booking is confirmed, the system automatically records all relevant details, including item ID, user information, rental dates, and payment status. These logs are securely stored in the database, allowing owners and customers to review their complete rental history at any time.

7. **Real-Time Dashboard & Rating Feedback System:** A real-time dashboard provides users and owners with updates on bookings, payments, item status, and notifications. After each rental, feedback and ratings are collected to evaluate user behavior and item condition, building trust and offering clear performance summary for both parties.

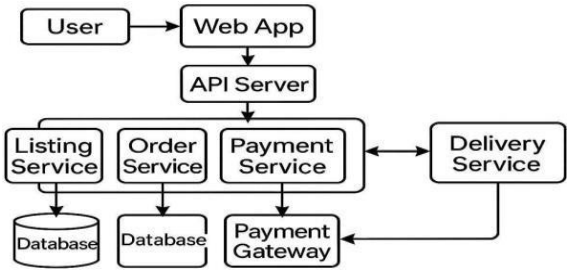

Fig -1: Model Architecture

3. Methodology

3.1 Dataset Creation

- A custom product dataset was created using real device images, rental item photos, and category- specific assets collected from multiple user environments. Images were captured using mobile and web uploads, and key attributes such as item type, condition, and background clarity were extracted for analysis.
- More than 1500 product samples were collected from different rental categories.
- Items were annotated manually to identify product features, damage indicators, and owner-provided descriptions.
- Data preprocessing steps such as resizing, normalization, and brightness adjustments were applied to maintain consistency across the platform.

3.2 Model Training

- Training was performed using cloud GPU acceleration in Google Cola.
- The recommendation model was fine-tuned on custom user–item interaction datasets..
- A similarity-based ranking model was integrated to improve accuracy in predicting relevant rental items for each user.
- Product embedding's were generated using deep learning-based feature extraction to categorize items and enhance search efficiency.

3.3 Real-Time Detection

- User requests are processed instantly as product data is fetched at ~10 mms response time.
- User authentication ensures the correct renter or owner interacts with the platform.
- Search tracking identifies user browsing patterns and preferred product categories.
- The system checks product availability in real time and updates status across all users.

3.4 Order & Activity Logging Module

Every rental activity is logged in real time and stored securely in the database for owner and user review.

- All orders include the timestamp, item details, and transaction reference ID.
- A Reliability Score increases dynamically as successful rentals and positive feedback accumulate.
- The system maintains logs for bookings, cancellations, payments, product updates, and user interactions.

3. System Workflow

The overall workflow of the Rent-It platform integrates all rental and booking modules into a unified transaction pipeline:

- User logs in and completes basic profile setup (details verification).
- Item browsing and category-based recommendations are performed.

- During booking, item details and availability are processed continuously.
- Rental requests trigger instant payment flow and order-status updates.
- Upon confirmation, the system stores records and generates a receipt.

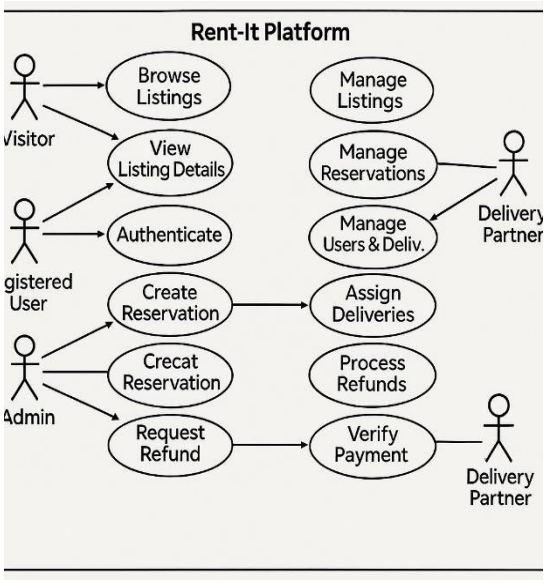

Fig -2: **Use case Diagram**

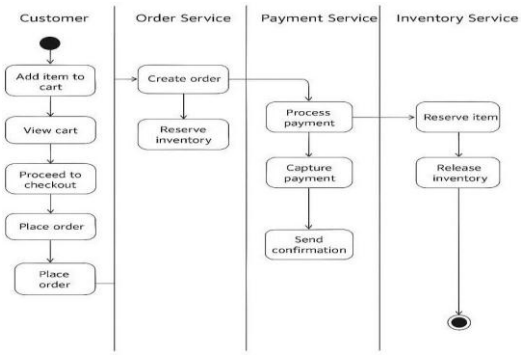

Fig -3: **Activity Diagram**

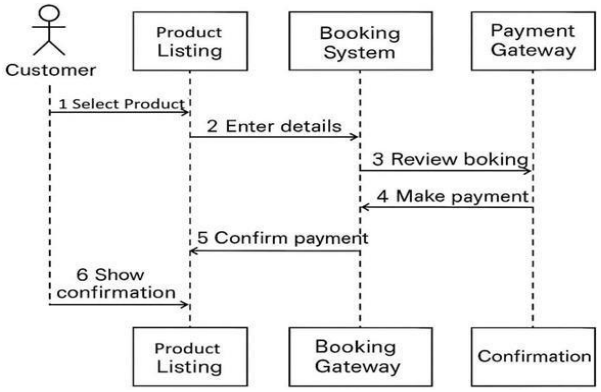

Fig -4: **Sequence Diagram**

4. RESULTS

The Rent-It platform demonstrated stable and efficient performance during testing. The combined use of React, Next.js, MySQL, and Razor pay integration enabled the system to handle rental-related operations in real time with minimal delay. Throughout multiple simulated user sessions, the platform consistently processed item browsing, booking requests, and payment workflows without interruptions. Authentication maintained user verification with high accuracy, ensuring that account misuse attempts were effectively prevented. The core modules such as inventory management, availability tracking, and The system architecture handled user interactions smoothly, maintaining stable response times and continuous data synchronization. Confirmed order was automatically stored in the MySQL database with timestamps and transaction details, enabling clear post-booking tracking.

At the end of each transaction, the platform successfully generated an order summary reflecting all processed details and updating the user's My Orders section. The workflow components, backend connectivity, and interface mechanisms indicated that the solution could operate continuously and scale effectively for commercial use. Overall, the Rent-It platform proved robust, responsive, and highly reliable in ensuring smooth and secure rental operations.

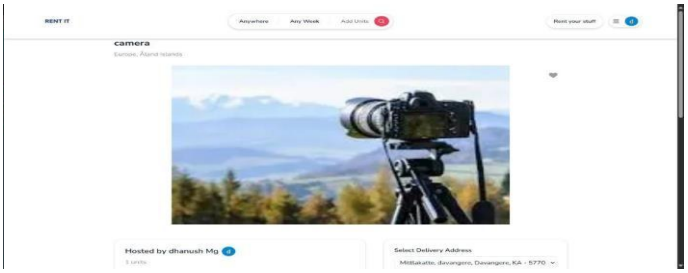

Fig -4.1: Booking Dashboard

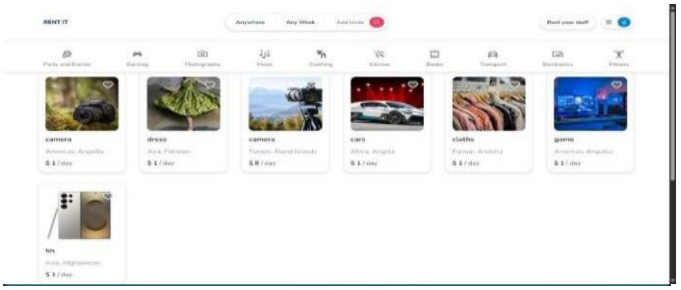

Fig -4.2: Main Application Page

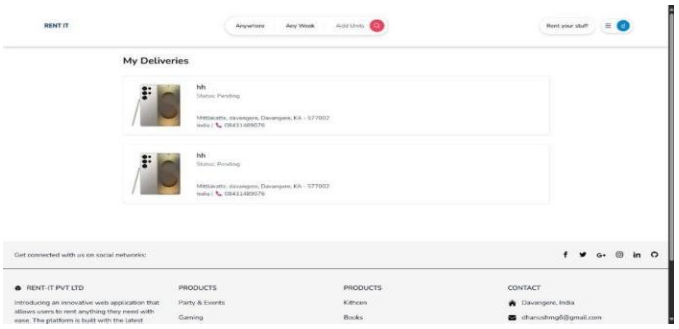

Fig -4.3: Delivery Page

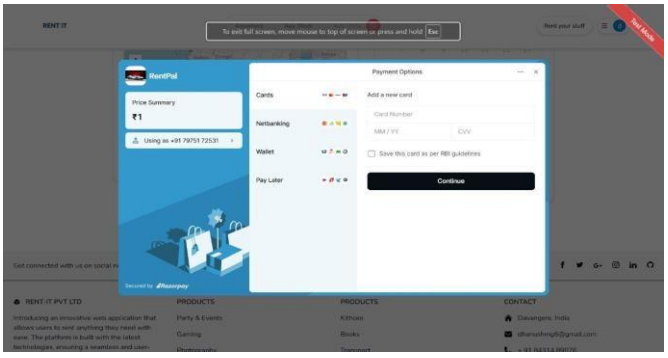

Fig -4.4: Rozar pay Gateway

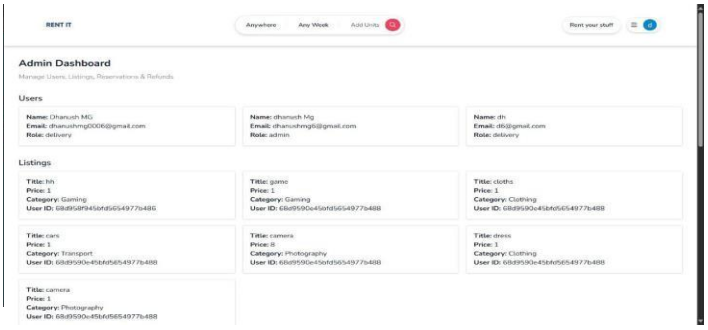

Fig -4.5: Admin Dashboard

5. CONCLUSIONS

The project successfully demonstrates a fully automated rental management platform that integrates real-time product availability tracking, secure authentication, digital payments, and structured order processing to ensure smooth and reliable item rentals. By leveraging the efficiency of React, Next.js, Razor pay, and a MySQL-backed data system, the platform minimizes manual effort and provides a seamless experience for both owners and customers. The unified booking and payment workflow ensures transparency, reduces errors, and enables renters to access products quickly without physical interaction. The automated order logging and user dashboard offer clear, data-driven insights into rental activity, enhancing reliability and user trust. Designed to operate effectively across diverse devices and user conditions, the system is scalable, efficient, and capable of handling continuous transactions, providing a practical and modern solution for supporting digital rental marketplaces.

REFERENCES

[1] Fielding, R. (2000). Architectural Styles and the Design of Network-Based Software Architectures. University of California.

[2] Subramanian, R. (2019). Fundamentals of E- Commerce Platforms. ACM Computing Surveys, 51(3), 1–28.

[3] Dixit, A., & Patel, M. (2020). User Experience and Interaction Design for Modern Web Applications. International Journal of Human-Computer Studies, 137, 102–118. https://doi.org/10.1109/LSP.2016.2603342

[4] Soni, N., & Sharma, P. (2020). Secure Online Payment Integration in Web Applications. Journal of Computer Engineering, 18(4), 45– 53. https://docs.ultralytics.com/

[5] Verma, S. (2021). Full-Stack Web Development with Next.js and Node.js. International Journal of Web Applications, 15(2), 122–130. https://mediapipe.dev/

[6] Jain, P., & Khanna, S. (2021). Scalable Web Application Architecture Using REST and Microservices. IEEE Access, 9, 11942–11955.

[7] Hazra, A., & Gupta, R. (2022). Online Rental Systems and Digital Marketplaces: A Review. Journal of Digital Business, 9(1), 55–70

[8] Flanagan, D. (2020). JavaScript: The Definitive Guide. O'Reilly Media.

[9] Riordan, J. (2022). Learning React: Modern Patterns for Developing React Apps. O'Reilly Media.

[10] Vercel. (2023). Next.js Documentation. https://nextjs.org/docs

[11] Express.js Team. (2023). Express.js Documentation. https://expressjs.com/

[12] Razorpay. (2023). Razorpay Payment Gateway Documentation. https://razorpay.com/docs/

[13] MySQL Team. (2023). MySQL Reference Manual. https://dev.mysql.com/doc/

[14] MongoDB, Inc. (2023). MongoDB Documentation. https://www.mongodb.com/docs/

[15] Amazon Web Services. (2023). AWS Cloud Deployment Documentation.