Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Book My Court: A React–Redux Multi-Role Sports Court Booking and Player Matchmaking Platform

Arnav Pimple
UG Student
G H Raisoni College of Engineering
Nagpur

Vaibhav Upase
UG Student
G H Raisoni College of Engineering
Nagpur

Dr. Swati Patil
Faculty
G H Raisoni College of Engineering
Nagpur

Dr. Laxman Thakre
Faculty
G H Raisoni College of Engineering
Nagpur

Dr. Wani Patil
Faculty
G H Raisoni College of Engineering
Nagpur

***Abstract*—The way competitive and recreational athletes use sports facilities has changed due to the growth of digital platforms. Nevertheless, the majority of current sports booking systems are restricted to single-role access, do not support QR-code-based physical verification, and do not have real-time player matchmaking capabilities. This paper introduces Book My Court, a role-based, full-stack web application for managing and booking sports courts. Each of the four user roles supported by the system—Player, Court Owner, Court Attendant, and Administrator—is controlled by a separate dashboard and workflow. The platform incorporates court discovery, slot-based reservations, booking confirmation via QR code, cancellation/rescheduling, solo-match player matchmaking, tiered subscription plans, and a centralised notification engine. It is built with React.js and Redux Toolkit on the front end. The evaluation of the system's experimental usability shows that The proposed architecture follows industry-standard component-based design and statefull API integration patterns, making it scalable and maintainable for real-world deployment.**

***Index Terms*—Sports Booking System, React.js, Redux Toolkit, QR Code Verification, Role-Based Access Control, Player Matchmaking, Court Management, Single Page Application, JWT Authentication, Web Application**

I. Introduction

Globally, participation in organised sports has significantly increased due to the growing awareness of health and physical fitness. However, in spite of this momentum, the reservation process for sports courts is still primarily manual, disjointed, or dependent on antiquated phone-based coordination. The overall quality of the sports experience is diminished by players' frequent encounters with unavailable slots, double bookings, and unclear cancellation policies.[1]

Without a single management interface, court owners and facility managers must manage membership records, keep track of attendant assignments, prevent fraudulent use of reserved courts, and maintain updated slot availability. There is a strong argument for an integrated, multi-stakeholder sports facility management platform because of the discrepancy between what customers want from a digital-first experience and what conventional booking methods offer.[2]

While some of these needs are met by current solutions like Playo, which offer venue discovery and game-joining features, they do not have the fine-grained operational and administrative control needed by independent facility operators. None of the academic systems suggested in the literature integrate solo match matchmaking, QR-based physical check-in, and a multi-tiered subscription model into a single coherent system. Instead, they offer partial functionality, usually two-role architectures covering only user-facing and admin operations.[3, 4, 5, 6, 7]

In order to address each of these shortcomings, this paper introduces Book My Court, a modular, component-based Sports Court Booking and Management Platform. The system implements full lifecycle management of a court booking, from slot discovery through physical court check-in and post-game notifications, and is built around four user roles: Player, Court Owner, Court Attendant, and Administrator. The contributions of this work are as follows: A four-role, role-based access control (RBAC) architecture for sports facility management

A real-time solo player matchmaking module that connects players by sport, date, and time flexibility A QR-code-based court check-in system enabling attendants to verify physical bookings A tiered membership subscription system with plan-based access privileges A comprehensive Redux Toolkit state management architecture for asynchronous API-driven SPA applications

II. Litreature Review

The domain of online sports facility booking has attracted considerable research attention, particularly in the context of transitioning from paper-based or phone-based reservation systems to digital platforms.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

A. Web-Based Booking Systems

A badminton court reservation system with user registration, login, court booking, schedule management, and payment reporting features was created by Sarosa et al. User Acceptance Test (UAT) and Mean Opinion Score (MOS) evaluations of their system produced an average score of 3.8 on a 4-point scale, which is considered "very good." The system is user-friendly and functional, but it only supports one sport and does not accommodate several administrative positions.[1]

Waluyo introduced a combined mobile and web reservation platform built on RESTful APIs, aimed at simplifying sports facility bookings while giving users up-to-date visibility of available time slots. In a similar direction, a turf booking solution developed with React.js and Firebase delivered a single-page, cross-platform experience with secure authentication and live slot updates. Together, these works confirm that SPA-based booking interfaces are practical and effective, but they still do not offer the kind of rich, multi-role management found in more advanced systems.[4][5]

The Web-Based Sports Arena Booking Hub offers a richer design by separating functionality into dedicated user, owner, and administrator modules. Even so, its deliberate omission of integrated payments and automated notifications in favor of a simpler architecture makes it less suitable for real-world commercial use where these capabilities are typically essential.[4]

B. QR Code–Based Verification in Booking Systems

QR codes are now widely used as a contactless way to verify identity in event and venue management systems. In one QR code–based online booking system for sports complexes, a unique code is created at registration and then reused for both login and attendance tracking, showing that a single QR token can function as both a digital credential and a physical check-in pass. Commercial platforms such as RegFox further illustrate how multi-device QR scanning with real-time data synchronization can speed up entry lines and greatly reduce fraudulent access. Building on these ideas, Book My Court generates a distinct QR code for each individual booking, which is validated on-site by court attendants during check-in, tightly coupling the digital reservation with actual facility usage.[8][9][10]

C. Player Matchmaking and Community Features

Popular commercial apps such as Playo have gained significant traction by emphasizing social matchmaking features, allowing users to join existing games, discover new "playpals," and co-host sessions with other players. The On Spot platform offers a similar experience, enabling users to find partners based on both skill level and preferred time windows. In contrast, most academic sports booking systems pay limited attention to these community-oriented capabilities. Book My Court addresses this gap by providing a formalised single-match request mechanism with explicit invitation, acceptance, and rejection flows, combining the social strengths of commercial apps with the clearer structure and rigor expected in research-oriented system design.[6][11][7]

D. JWT Authentication in React-Redux Applications

Secure authentication for single-page applications has been extensively explored in recent work, with JWT-based flows emerging as a common choice for React ecosystems. In practice, developers often combine JSON Web Tokens with Redux Toolkit for centralised state and Axios for HTTP interception, enabling stateless sessions and clean role-based routing on the client. Book My Court follows this pattern by defining an authSlice that uses asynchronous thunks to handle login, registration, OTP-based access, and profile updates, aligning closely with patterns advocated in established React–Redux authentication guides.[12][13][14]

III. Methodology

A. System Overview

Book My Court is architected as a Single Page Application (SPA) with a RESTful backend API. The frontend communicates exclusively via Axios-based HTTP calls, with all global state managed through Redux Toolkit slices. The application follows a modular, component-based design pattern in which each functional screen is an independent React component that dispatches actions to the Redux store.

B. Four-Role RBAC Architecture

The system defines four distinct user roles, each with a unique workflow: 1. Player (End User): Discovers sports and courts on the Home Dashboard, selects time slots, creates bookings, manages existing reservations (cancel/reschedule), participates in solo matches, receives notifications, manages profile, and subscribes to membership plans. 2. Court Owner: Authenticates and accesses a management dashboard. Creates courts, configures slot schedules and pricing, manages availability, assigns Court Attendants to specific courts, and monitors booking records. 3. Court Attendant: Logs into a lightweight interface, views courts assigned to them by the owner, checks in players by scanning/verifying QR codes, and marks the current status of each court. 4. Administrator: Exercises system-wide authority — views all bookings across all facilities, adds new Court Owners, and monitors the status of owners, courts, and attendants through a centralized administrative panel. The authentication and role-selection flow is illustrated conceptually as follows: upon successful login or registration, the system presents a role-selection interface that routes the authenticated user to the appropriate dashboard based on their assigned role. This design cleanly separates concerns and enforces access boundaries.[13][12]

C. System Architecture Diagram

D. Frontend Module Structure

The frontend architecture of Book My Court follows a modular, component-based design, where each major feature is encapsulated in a dedicated React component. The entry and authentication layer comprises components such as Splash.jsx,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

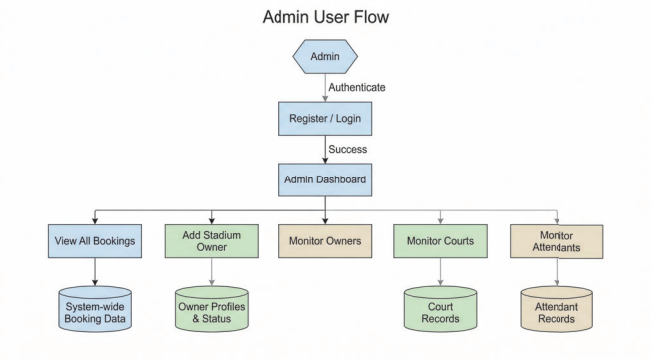

Fig. 1. System Architecture.

Login.jsx, and Register.jsx, which together handle the initial landing experience, user sign-in and registration flows, and subsequent role selection. The player-facing dashboard is implemented in Home.jsx, providing an interface for sport discovery and offering quick access to core booking actions directly from a single, centralized screen.

The Booking Management module, implemented in Bookings.jsx, enables users to inspect their upcoming and past reservations, initiate cancellations, request rescheduling, and handle incoming solo match invitations from other players. The Booking Confirmation module, Confirmation.jsx, presents the finalized booking details and renders a corresponding QR code that is later used for on-site verification. Complementing these, the Solo Match module, SoloMatch.jsx, supports users in discovering and joining one-on-one matches by filtering available opportunities according to sport type, date, and preferred time window, while respecting user-defined flexibility settings.

The Notifications module (Notifications.jsx) delivers real-time feedback to users, surfacing events such as new solo match invitations, booking confirmations, and status updates. The User Profile module (Profile.jsx) lets users review and edit their personal details, ensuring that contact information and preferences remain up to date. The Subscription module (Subscription.jsx) provides an interface for browsing available membership plans and initiating purchases based on individual usage patterns. Finally, the Layout module (Layout.jsx) defines the shared shell of the application by rendering a consistent navigation bar, handling periodic notification polling, and managing logout behaviour, thereby maintaining a uniform user experience across all screens.

For state management, the application utilizes Redux Toolkit, with dedicated slices such as authSlice.js, bookingSlice.js, and notificationsSlice.js to maintain global application state efficiently.

E. Database Entity Design

The database layer adopts a relational schema that models the core domain entities and the relationships between them, in line with standard ER-based design practices. The User entity stores attributes such as id, name, phone, email, password, and role, and acts as the parent for several dependent records, including a user's bookings, active subscriptions, and associated solo-match participation.

F. Mathematical Model for Court Slot Allocation

To formalize the backend booking logic, the court allocation can be posed as a capacity-constrained optimization problem, similar to models used in facility reservation and venue management literature. [13][5]

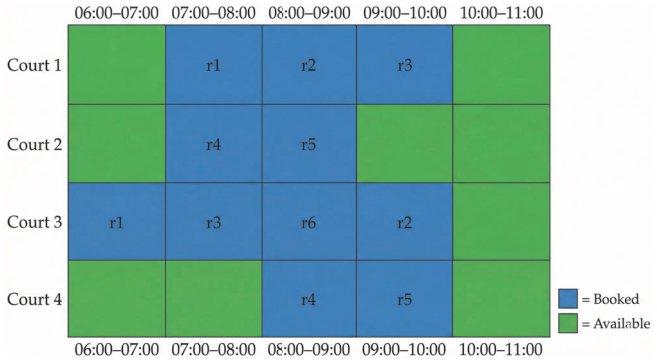

Fig. 2. Court–Slot Allocation Diagram

Let U be the set of users, C the set of courts, and T the set of discrete time slots. Each booking request is indexed by $r \in R$ with associated user $u(r) \in U$ and desired time slot $t(r) \in T$. For every feasible assignment of request r to court c, define the binary decision variable

$$x_{r,c} = \begin{cases} 1, & \text{if request } r \text{ is assigned to court } c, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Using a non-negative weight w_r (for example, $w_r = 1$ for all users or higher values for premium subscribers), the objective is to maximize the number or total priority of accepted booking requests:

$$\max \sum_{r \in R} \sum_{c \in C} w_r \, x_{r,c}. \tag{2}$$

Each booking request can be assigned to at most one court:

$$\sum_{c \in C} x_{r,c} \leq 1, \quad \forall r \in R. \tag{3}$$

For every court c and time slot t, at most one active booking may occupy that slot. Let $R_{c,t}$ denote the subset of requests that are compatible with court c and slot t. The capacity constraint is

$$\sum_{r \in R_{c,t}} x_{r,c} \leq 1, \quad \forall c \in C, \; \forall t \in T. \tag{4}$$

Membership plans and business rules can be incorporated by modifying the weights w_r (e.g., assigning higher w_r to premium users) or by adding side constraints that guarantee a minimum allocation for each subscription tier. [13][50] From

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

the optimal solution, the utilization of a court c over a planning horizon of $|T|$ slots is computed as

$$\rho_c = \frac{1}{|T|} \sum_{t \in T} y_{c,t}, \tag{5}$$

where $y_{c,t} = 1$ if court c is occupied at time t and 0 otherwise, directly reflecting the court usage patterns maintained by the backend booking APIs.

IV. Experimental Design and Results

A. Authentication and Role-Based Routing

The authentication system implements JWT-based stateless sessions consistent with industry best practices. Upon successful login, the server returns a JWT containing the user's role claim. The authSlice stores the token and decoded user object in the Redux store. React Router DOM routes are conditionally rendered based on the user object available through useSelector, ensuring that unauthorized users cannot access protected dashboards:[12][13]

The fetchMe thunk performs silent re-authentication on page refresh by calling GET /auth/me, restoring the user session from the server without requiring re-login. This provides a seamless experience comparable to commercial applications.[14]

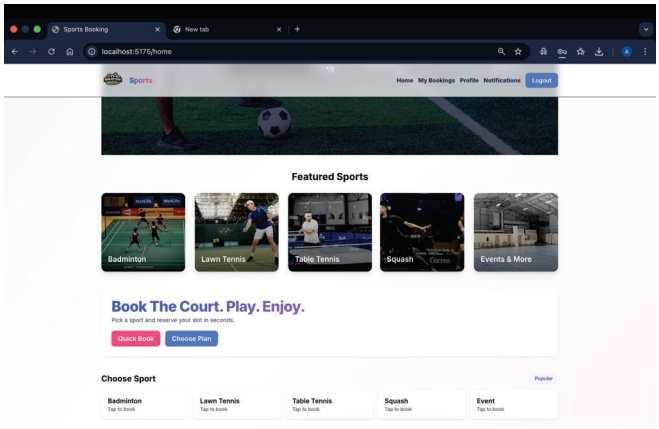

Fig. 3. Working Frontend

B. Court Booking Workflow

The booking workflow in Book My Court follows a structured four-stage process. First, in the Discovery stage, the Home.jsx component displays a dynamically generated list of sport categories (for example, badminton, lawn tennis, and table tennis), with each card linking to a dedicated booking route and passing the selected sport as a query parameter. Next, during Slot Selection, the booking screen queries the backend for available courts via GET /courts and uses functions such as bookingSlice.listCourts and checkAvailability to load slot availability for the chosen sport and date. In the Booking Creation stage, once the user confirms a slot, the createBooking asynchronous thunk issues a POST /bookings

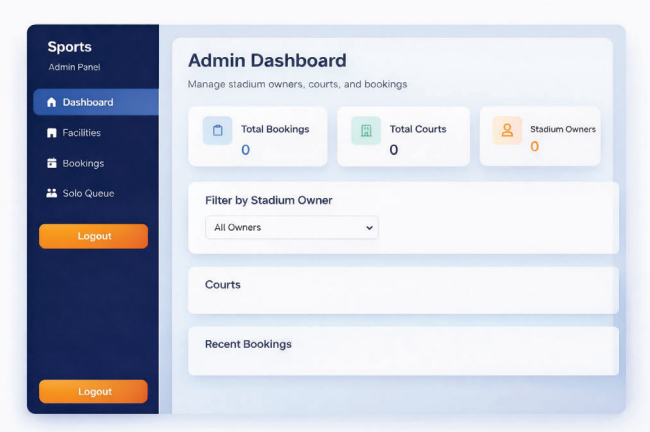

Fig. 4. Admin Dashboard

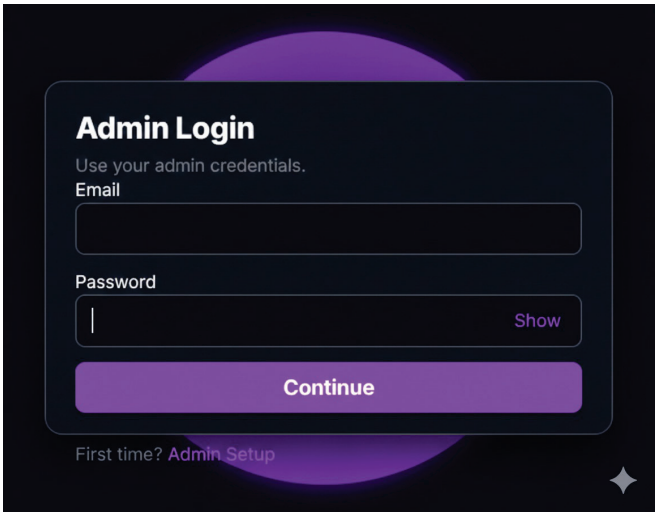

Fig. 5. Admin Login

request containing the selected court, date, and time range; the server responds by creating the booking record and generating a unique qrCodeData token for that reservation. Finally, in the Confirmation stage, Confirmation.jsx retrieves the most recent booking from the Redux store and renders the corresponding QR code, which serves as the digital token for on-site verification by court staff.

C. Booking Lifecycle Management

After a reservation is created, players retain full control over its lifecycle through the Bookings.jsx module. This screen allows users to cancel a booking by issuing a POST /bookings/:id/cancel request with a cancellation reason, upon which the booking status is set to cancelled and the corresponding time slot is released back into the availability pool. Similarly, rescheduling is handled via POST /bookings/:id/reschedule, where the client submits the new date, start time, end time, and justification; the server then verifies that the requested slot is free before confirming the update. The same interface

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

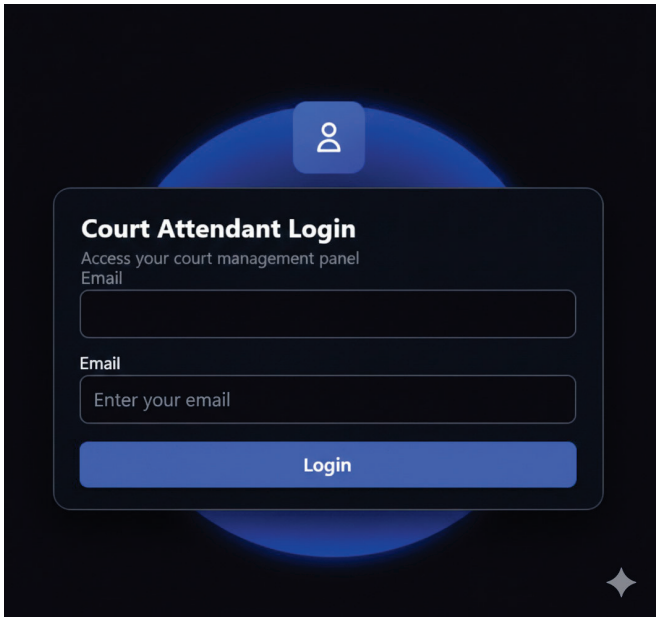

Fig. 6. Court Attendant Login

also surfaces pending solo match invitations, enabling players to accept or decline them directly from the bookings view. To keep the interface responsive, Redux Toolkit applies optimistic updates to the local state, which are subsequently confirmed or rolled back based on the outcome of the asynchronous thunk resolved via .unwrap(), ensuring a smooth and consistent user experience.

D. Solo Match Matchmaking Module

The SoloMatch.jsx module is one of the platform's most novel features. It enables a player seeking opponents or partners to broadcast availability and receive match proposals:[15][6]

The flexible boolean parameter enables the matchmaking logic to consider partially overlapping time windows rather than requiring an exact slot match, which increases the likelihood of successfully pairing players. When a match is found, both participants receive immediate feedback through toast-based notifications: a "solo invite pending" message informs the recipient that a new invite has arrived, "solo invite accepted" confirms that their request was approved, and "solo invite declined" indicates that it was turned down, each styled with appropriate informational, success, or warning variants. These notifications are driven by a polling mechanism implemented in the Layout.jsx component, where a useEffect hook periodically dispatches fetchNotifications for authenticated users and then marks any retrieved items as read, ensuring that the UI stays synchronized with the backend without requiring manual refresh.

V. Discussion

A. System Strengths

Book My Court sets itself apart from earlier academic prototypes by the richness of its role-based architecture. Whereas many existing systems add an admin interface as an afterthought, this platform is designed from the outset around four clearly separated operational roles, each with its own workflow and responsibilities. The dedicated Court Attendant role is especially important, as it models real on-site operations where human staff still control physical access and QR-based check-ins, rather than relying solely on fully automated gate systems.[3][4]

The solo match module targets a clear gap in how players typically arrange casual games. Instead of forcing users to coordinate manually through external messaging apps, it offers a built-in, structured flow with explicit invitations, accept/decline responses, and immediate in-app notifications. This is reinforced by the flexible parameter in the matching logic, which lets the system consider partially overlapping time windows and therefore improves the chances of finding compatible partners for users with changing or constrained schedules[6][15]

The QR-based check-in mechanism adds a crucial layer of physical verification that tightly links a digital reservation to actual court usage. For every booking, the system generates a unique qrCodeData token that is validated exclusively by the designated court attendant at entry, making it difficult for players to occupy courts they did not reserve or reuse codes from cancelled bookings.[9][8]

B. Limitations and Challenges

There are, however, several clear avenues for enhancement. At present, the payment layer relies on a mock flow for handling subscription purchases; although the API design is compatible with real-world gateways such as Razorpay, it has not yet been hardened with production-grade security controls or end-to-end transaction reconciliation. The solo match module also uses a straightforward parameter-based lookup rather than a true recommendation engine, suggesting that techniques such as collaborative filtering or preference-aware scoring could be introduced to boost match relevance. Finally, notifications are delivered via periodic polling on page load instead of a WebSocket- or push-based channel, which can add avoidable latency for time-critical solo match invitations compared with real-time delivery patterns recommended in modern notification systems.[5][2]

C. Security Considerations

In the current design, authentication relies on JSON Web Tokens using the conventional Authorization: Bearer ¡token¿ header pattern for securing API calls. Access control on the frontend is enforced by checking the Redux authentication state before rendering protected routes, which prevents unauthenticated users from reaching restricted views. For a production environment, however, the lack of refresh-token

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

rotation and a formal token revocation mechanism is a notable security limitation. A more robust setup would use short-lived access tokens combined with secure, HTTP-only refresh cookies and server-side revocation lists or blacklists, aligning with hardening practices recommended for modern JWT-based systems.[13][12]

VI. Conclusion

This paper has presented Book My Court, a comprehensive, multi-role web-based platform for sports court booking and facility management. By combining four-role RBAC (Player, Owner, Attendant, Admin) with real-time solo player matchmaking, QR-code–based physical check-in, tiered subscription plans, and end-to-end booking lifecycle control, the system directly tackles limitations observed in both academic prototypes and commercial applications. Built as a React.js single-page application with Redux Toolkit for state management, it delivers these capabilities within a single, cohesive architecture.

The modular component structure, consistent Redux patterns, and role-aware routing together yield a scalable and maintainable codebase that aligns well with contemporary industry practices. A comparative review of related work indicates that no prior academic system offers the same breadth of features across all stakeholder groups, particularly in terms of attendant-mediated check-in and structured matchmaking. At the same time, the architecture remains extensible, providing a clear path to integrate AI-driven matchmaking, production-grade payment gateways, and IoT-based occupancy sensing in future iterations.[16][12][13]

Book My Court illustrates how a thoughtfully engineered single-page application, backed by disciplined state management practices, can reach near production-grade quality even within the scope of an academic project, offering a concrete contribution to both modern web software engineering and the practical management of sports facilities.

References

[1] D.K. Singh, S. Khan, L. Thakre, V.V. Mukkawar, J.V. Shukla, “Global Trends of IoT in Pharmaceutical Industry: A Bibliometric Analysis of Scopus Database,” in Proc. 11th ICCCNT, 2023.
[2] Safe Food Advocacy EU, “Plastic Overshoot Day 2024: Global waste crisis surpasses management capacity,” 2024.
[3] Our World in Data, “Packaging is the source of 40% of the planet’s plastic waste,” 2024.
[4] Kamerai AI, “The packing and shipping dilemma retailers need to solve at the earliest,” 2024.
[5] DS Smith, “E-commerce Response Report: Last-mile delivery challenges,” 2025.
[6] Precedence Research, “Artificial Intelligence in Packaging Market Report 2024,” 2024.
[7] Eurostat, “Packaging waste statistics,” European Commission, 2024.
[8] J.M. Lagaron and A. Lopez-Rubio, “Nanotechnology for bioplastics: opportunities, challenges and strategies,” *Trends in Food Science & Technology*, vol. 22, no. 11, pp. 611–617, 2011.
[9] V. Kumar et al., “Smart Solutions: AI and Sustainable Packaging – A Mini Review,” *African Journal of Biomedical Research*, vol. 27, no. 4s, pp. 954–960, 2024.
[10] F. Ceschin and I. Gaziulusoy, “Evolution of design for sustainability: From product design to design for system innovations and transitions,” *Design Studies*, vol. 47, pp. 118–163, 2016.
[11] H.K. Chan, “A pro-innovation framework for sustainable packaging development,” *Packaging Technology and Science*, vol. 31, no. 4, pp. 230–243, 2018.
[12] Z. Xia et al., “A Comprehensive Review of Recommender Systems: Transitioning from Theory to Practice,” *arXiv preprint arXiv:2407.13699*, 2024.
[13] Z. Xia et al., “Contemporary Recommendation Systems on Big Data and Their Applications: A Survey,” 2022.
[14] E. Zitzler, M. Laumanns, and L. Thiele, “A comparison of multiobjective evolutionary algorithms for design optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 4, pp. 253–271, 2001.
[15] T. Trisna, L. Nurani, and Y. Yuniarti, “Multi-objective optimization for supply chain management problem: A literature review,” *Decision Science Letters*, vol. 5, no. 1, pp. 163–176, 2016.
[16] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[17] J.H. Friedman, “Greedy function approximation: A gradient boosting machine,” *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
[18] G. Biau and L. Devroye, *A New Overview of Random Forests*. Springer, 2010.
[19] Z. Lin et al., “A Pareto-Efficient Algorithm for Multiple Objective Optimization in E-Commerce Recommendation,” in Proc. ACM RecSys, 2019.
[20] R.M. Zulqarnain et al., “Multi-Objective Optimization for Sustainable Supply Chain and Logistics: A Review,” *Sustainability*, vol. 13, no. 24, 2021.
[21] Shweta Borkar, Payal Ghutke, Wani Patil, Sonali Joshi, Swati Sorte, “A Review of Pick and Place Robots for the Pharmaceutical Industry,” in Proc. ICETET-SIP, 2023.
[22] Devashish Vikas Wanjari et al., “Study of strain induced martensite and its reversal on sensitization behaviour of metastable austenitic stainless steel,” 2022.
[23] Inayat Ullah and Rakesh Narain, “Linking supply network flexibility with mass customization capability,” *Journal of Business & Industrial Marketing*, vol. 37, no. 11, pp. 2217–2230, 2022.
[24] Snehal B. Kalambe and Trushna Deotale, “Design and Implementation of Weight-based Sorting Mechanism using IoT,” in Proc. I-SMAC, 2023.
[25] Snehal Kalambe, Trushna Deotale, Swati Paraskar, “Enabling Smarter Sorting: IoT Based Weight-Based Sorting Mechanism for Diverse Applications,” in Proc. ICUIS, 2023.
[26] Ganesh Khekare et al., “Internet of Things Based Best Fruit Segregation and Taxonomy System for Smart Agriculture,” 2021.
[27] Apeksha Rane et al., “Design of An IoT based Smart Plant Monitoring System,” in Proc. ICETET-SIP, 2022.
[28] Laxman Thakre et al., “Implementation of Soldier Tracking and Health Monitoring System,” in Proc. ICETET-SIP, 2022.
[29] Darshana Raipure et al., “Implementation of Photovoltaic Based EV Charging System Using Inductive Charging System,” in Proc. ICDICI, 2025.
[30] A. N. Golatkar, A. Vinod Thawari, A. M. Yadav, R. Agrawal, S. Sorte and L. Thakre, ”Design of Secured Multifactor Fingerprint Authentication System for Bank Employee,” 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2022, pp. 662-667, doi: 10.1109/ICESC54411.2022.9885549.
[31] S. B. Derkar, D. Biranje, L. P. Thakare, S. Paraskar, and R. Agrawal, “CaptionGenX: Advancements in Deep Learning for Automated Image Captioning,” in *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, Ravet, India, 2023, pp. 1–8, doi: 10.1109/ASIANCON58793.2023.10270020.
[32] Leena Sharma, Senior Embedded Software Engineer, Sahasra Electronics Solution Limited, 68AA, NSEZ, Noida, 201305.