Reservation Collaborator: Smart Solution for Service Reservation

Sahil Rampal[1*], Sachin Kumar[1], Vinay Fagodia[1], Vijaya Laxmi[1], Abdul Maazid[1]

[1]School of Computer Applications, Lovely Professional University, Punjab, India
Email: sahil.rampal@gmail.com*, subhamkumarp732@gmail.com, vinay.fagodia250@gmail.com, vijayalaxmi.02v1@gmail.com, abdulmaazid01@gmail.com

Abstract. In today's world an individual often faces time shortage while planning or booking an event. The process of searching and choosing the event is time consuming and overwhelming. Customers generally check many websites while considering the requirements and then only decide to book the event. In this situation they seek the help from an event coordinator who has prior knowledge and helps them to book the appropriate event in lesser time and without spending much resources. To handle this scenario, we are proposing a solution that saves the customer from browsing through different websites to investigate themselves, they will just need to send the requirements to the event coordinator, and event coordinator will book the event for the individual. We have delivered an application using MERN Stack (Mongo, Express.js, React.js, Node.js) which will provide these features to the customer. The hiring of event coordinators to enable customers to schedule bulk bookings, as well as real-time support via a built-in messaging system is one of the standout features in the system and manages customers, events, sales and coordinator assignment through admin panel.

***Keywords*:** MERN Stack, AI Driven Booking, Service Reservation, Real Time Scheduling Automation

1 INTRODUCTION

Since the digital era, the event management industry had been banking on technology for seamless functioning and better customer experience. This has led to the increasing potential for integrated and responsive solutions ranging from ticket booking systems to real-time event promotions and customer engagement platforms. Enter the Reservation Collaborator, the web application that meets this demand with a one solution for the entire as simple and profitable as possible flow. The platform is built to serve both customers and administrators alike, supporting event discovery, ticket booking, and real-time interaction, along with powerful backend functionalities for administration and coordination. The application has three basic components—frontend, backend, and admin panel, and all of them are developed on modern and efficient technologies. Frontend (React, CSS): Customers can view events and make bookings through a customer-friendly interface. Routing is handled via React Router, allowing seamless tran-

© The Author(s) 2026
M. Sharma et al. (eds.), *Proceedings of the Third International Conference on Recent Advances in Computing Sciences (RACS 2025)*, Advances in Intelligent Systems Research 212,
https://doi.org/10.2991/978-94-6239-768-2_15

sitions between pages like Home, Cart, Tickets, Settings, and the newly added Coordinator area. This kind of modular structure offers not only scalability but also maintainability.

The most significant change in the project is the introduction of the event coordinator feature. This requires the coordinator who will help to book the event which is often easier said than done, particularly in large bookings or special policies. Customers can even enjoy an interactive experience with the integrated messaging system, allowing direct communication with coordinators. Overview. The system is automated and allows management of customers, events and ticket sales through an admin panel. On the backend, the application uses Node.js and Express.js, MongoDB, and Express. Well, Express is the framework and is primarily used for the backend, and MongoDB is a NoSQL database. Implemented CRUD operations: implemented various RESTful APIs for customer registration, ticket retrieval, order placement, and messaging using POSTMAN. Overall, the Reservation Collaborator exemplifies how full-stack web development can be used to build a dynamic, customer-centric event management system with practical features that address real-world requirements in a digital environment.

1.1 Problem Statement

Historically, event management required complex multi-layer strategies, which encompassed ticket sales, customer registrations, event planning, and customer support. In addition to existing systems which are generally not integrated with one another, customers and administrators often have disrupted experiences as systems exchange information. Bulk ticketing is not easy for customers, they sometimes do not receive their update on time, nor can they find for event specific help. Finally, admins have to deal with disjointed customer data (especially some repeated entries, ticketing sales via different platforms, and the logistics of the events with almost no unified dashboard). This gap exists specifically in personalized assistance for complex or large-scale bookings, which leads to reduced customer satisfaction and retention. A lack of ability for real-time communication with event staff or coordinators only adds to the challenges. Moreover, many systems fail to optimize backend performance, resulting in delays during high-traffic operations such as ticket purchases or customer authentication.

To tackle these challenges, the Reservation Collaborator is designed to provide an all-in-one, full-stack solution. The goal is to further simplify the complete flow with a Responsive Frontend, Scalable Backend and a Working Admin Panel. Primary among these is the introduction of a new feature that allows customers to hire event coordinators and to communicate directly with them, properly improving service quality as well as inadvertently boosting operational efficiency. One of the solutions of this project is to solve the core inefficiencies in traditional event management systems through innovation in technology.

1.2 Significance of the online reservation system

In the context of overcoming a whole variety of event management pain points through innovation, the Reservation Collaborator represents a crucial, projectable step forward. Overall, it promotes the incorporation of an integrated full-stack solution on this project, thus enhancing the initial customer experience, providing a more sophisticated coordination of events and simplifying the administrative processes. One of its exceptional contributions is the implementation of use a dedicated event coordinators which connects customers with service providers. Not only do customers appreciate this, but it also helps provide tailored assistance for larger or more complicated bookings.

Also, real-time messaging system allows customers and coordinators to connect with each other directly making responsiveness and trust better. Another significant change that has improved the experience for customers is the new and improved mobile app, providing the most seamless ticket scanning experience even on crowded event days. With an admin dashboard, the project allows administrators to control all customers (blackouts), events (showtimes), and ticket sales (innovators and coordinators), minimizing overhead and simplifying coordination overall. This project also showcases the utility of modern web technologies like React.js, Node.js, Express.js, MongoDB can be used to build scalable and maintainable systems. In summary, the Reservation Collaborator is a potent endeavor that reduces the hurdles of traditional event management systems through a modernized, customer-centric, and performance-oriented platform that resolves pressing challenges in real-world operations.

1.3 Overview of the structure of the paper

The paper is organized as follows:

1. **Introduction** – The first section of the paper consists of an introduction to the Reservation Collaborator which discusses the motive behind creating a complete event management platform. It describes the project features, the issues it solves, and the importance of including advanced features like hiring event coordinator and instant chat. It also outlines the research question and hypothesis which underpinned the research.
2. **Literature Review** – Existing event management solutions through web-based booking systems It provides a critique of their limitations: lack of interactivity for customers, non-optimal performance, and less personalization. Based on current solutions, the literature review identifies that are the gaps where the Reservation Collaborator should flourish: there is no interaction between customers and coordinators, and backend systems are not straight-cut.

3. **Methodology** – The methodology section mentions the technology stack deployed, such as React.js in frontend, Node.js and Express. for the backend, and MongoDB for data storage. It explains the system architecture, API design, routing structure, and the implementation of key features like the event coordinator module and the messaging system. The test strategies and development workflow are also grazed over.
4. **Result and Discussion -** The Result and Discussion section discusses the outcomes of the implementation process, noting performance improvements and customer engagement metrics. It assesses system efficiency, ticket processing speed and the effects of the new features on customer satisfaction. Analysis, any data to support and visualizations where applicable.
5. **Future Directions -** The last part deals with the practical implications of the project results along with its contribution to the event management field and the benefits. It also outlines future enhancements such as adding payment gateway integration, mobile responsiveness, and AI-based event recommendations to further extend the system's capabilities.

2 Literature Review

In 2023, N. V. Kumar et al [1] proposed a web application for flexible car sharing and car leasing using MERN stack. It will also provide an extensive marketplace for all customers looking to rent / lease vehicles in an easily accessible, scalable and highly secure manner. Using modern JavaScript technologies–MongoDB, Express.js, React.js, and Node.js: Its system architecture provides a great customer experience on the front-end and smooth database operation on the back-end. Deploying this has a variety of functionality for secure authentication using JWT, image handling to cloud services and dynamic listings for cars. The platform uniquely employs customer and admin roles, offering specific control panels and workflows that facilitate use and allow for operational control. In addition, the writers cover the need for designing database schemas, RESTful API integration, and front-end responsiveness for different device types.

G. A. Sugiartha et al. [2] introduced a student-focused bookstore information system in 2020 that uses a structured software engineering approach to optimize inventory and sales operations. With the aid of visual modeling tools such as Data Flow Diagrams and ERDs, the paper explains each step of the SDLC model, from requirement analysis to deployment. Key modules like customer accounts, transaction logs, and book inventory are managed by the system. Implementation clarity, modularity, and maintainability are guaranteed by the structured development process. The system solves major issues that both students and bookstore owners encounter by integrating these elements with an intuitive customer interface. It is clear from the research paper [2] that the pro-

ject's objective of digital transformation through software systems is similar, with an emphasis on enhancing information accessibility, efficiency, and automation of manual processes.

In 2024, A. A. Ojugo et al. [3] used the Unified Software Development Process (USDP) to create a hotel reservation system. The study provides a thorough overview of the architecture and workflow of the system by incorporating multiple UML diagrams to represent use cases, sequences, and deployment. The application supports the entire hotel booking lifecycle and has necessary features like online booking, payment integration, and reservation cancellation. The authors stress the importance of customer validation, database consistency, and fault tolerance. The objective of the research paper [3] is clearly in line with contemporary service-based web applications, emphasizing customer interaction, reliability, and structured design methodologies to guarantee scalability and maintainability.

Sowmya E. et al. [4] is to offer an end-to-end digital platform for vehicle reservations. The system's admin control, customer management, vehicle listings, and registration modules were all created with React for the front end and Node.js for the back end. For non-relational data storage that promotes scalability and flexibility, the backend incorporates MongoDB. The project's practical focus is demonstrated by the addition of features like customer authentication, booking history, and real-time vehicle availability. The system is domain-specific, but it represents a larger pattern in the use of JavaScript technologies to create responsive and modular web applications. It is clear from the research paper [4] that the main goal is to simplify traditional services by using full-stack implementations that are customer-friendly, safe, and effective while meeting administrative and customer need.

In 2013, Almomani et al. [5] presented an intelligent phishing detection system using classification data mining techniques. The classifier was decision trees and random forest classifier were used for classifying dephasing-based URLs from a set dephasing feature which were extracted from any web page. The proposed system architecture, which consists of three stages: URL preprocessing, feature extraction, and classification. The study focuses on important measures like detection accuracy, false positive rates, and classifier performance, and offers a comprehensive comparison of metrics across various detection algorithms. prospective, the strategy outlined in research paper [5] is representative of the gradual evolution of cybersecurity measures to intelligent, learning-enabled solutions capable of actively performing automated, accurate analysis and adapting to changing phishing techniques.

Recent studies have demonstrated that MERN stack-based reservation and booking systems improve scalability, modularity, responsive user interfaces, and customer experience across diverse application domains, including ticket booking, online shopping, and event management [6–10]. Furthermore, recent event management platforms developed using the MERN stack emphasize role-based access, customer engagement, responsive web interfaces, and efficient reservation workflows, thereby providing strong motivation for the proposed Reservation Collaborator framework [11–13].

3 METHODOLOGY

As a web tool development, Reservation Collaborator followed a systematic, modular web development template which guarantees its scalability, usability and real-time interactions. The project is based on full-stack development, handling everything from the frontend to the backend and database, with focus on responsive design, role-based access, and improved customer experience. **Fig. 1.** Reservation Collaborator Development Life Cycle**Fig. 1** depicts reservation collaborator development process followed.

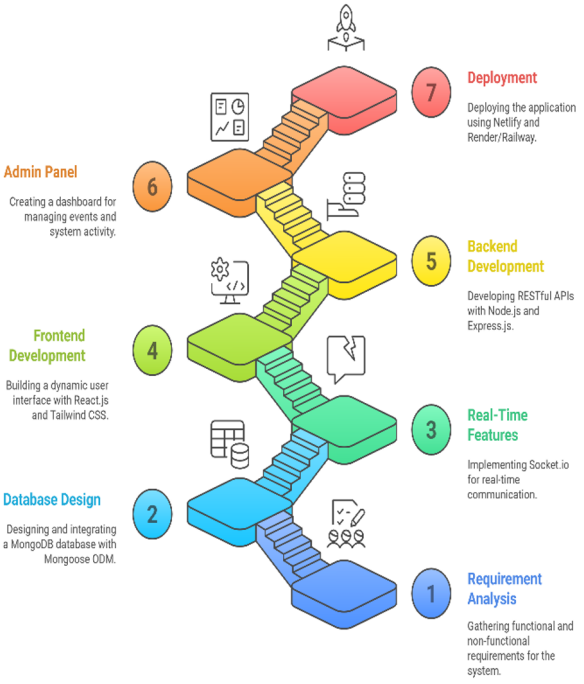

Fig. 1. Reservation Collaborator Development Life Cycle

3.1 Requirement Analysis and Planning

The first step in our process is to collect both functional and non-functional requirements, including event browsing, ticket booking, event coordinators hiring, admin management, and real-time messaging. Customer personas were designed to learn the interaction flow for 3 primary roles: Admin, Customer, Event Coordinator.

3.2 Database Design and Integration

Considering the dynamic schema of the app creating to unstructured data like event details, etc., We used MongoDB NoSQL database to store customer profiles, chat mes-

sages, and assign coordinators. For greater schema definition and efficient query handling, an Object Data Modeling (ODM) based on Mongoose are adopted. Initially, the database has collections for the customer, the event, booking, message, and feedback. We applied data normalization and indexing to ensure queries would run quickly, particularly for tasks that included a lot of repeating reads such as event listings or message threads.

3.3 Real Time features

For real-time customer-event coordinator communication, Socket.IO, which allowed for two-way communication that did not need a page refresh. It enabled chat messaging and instant updating to chat interfaces, mimicking a live customer support experience.

3.4 Frontend Development

Frontend is implemented on React.js that provides a dynamic and interactive customer experience. Event calendars, ticket reserve forms, customer and administrative dashboards were all made in a modular and reusable design pattern. We used React Router for client-side navigation, and we managed customer sessions, cart data, and messaging threads using React Context API for state management. We styled the customer interface with Tailwind CSS, allowing for quick prototyping and a consistent design across pages. These used responsive design techniques to ensure use on mobile and conditional rendering to display relevant content based on customers role.

3.5 Backend Development

Node.js is used to develop the backend with the Express.js framework for RESTful API endpoints to perform all server-side functions. This includes customer authentication, event CRUD, tickets, coordinators, and messaging services. To ensure secure sessions and role-based access control, authentication is built using JSON Web Tokens (JWT). This led to the development of middleware functions that would validate inputs, restrict unauthorized access, and log site activity on the server to facilitate debugging and assess performance.

3.6 Admin Panel Functionality

A dedicated admin dashboard is created with capabilities to manage all events, view ticket sales, assign event coordinators, and monitor system activity. This interface allowed for real-time analytics on customer engagement and ticket booking, supporting informed decision-making.

3.7 Deployment

The frontend is deployed separately, while the backend and database are hosted on Render. Environment variables were secured using .env files, and CI/CD pipelines were set up to automate the deployment process after successful testing.

3.8 Result

This reservation collaboration project is successfully designed and deployed, giving a complete end to end event management platform integrating even the basic scope of functionalities like event searching, ticket booking, different customer roles management, payment gateways, etc. **Fig. 2** shows order summary page allows customers to get a clear view of events selected along with event names, quantities, pricing details, and total cost.

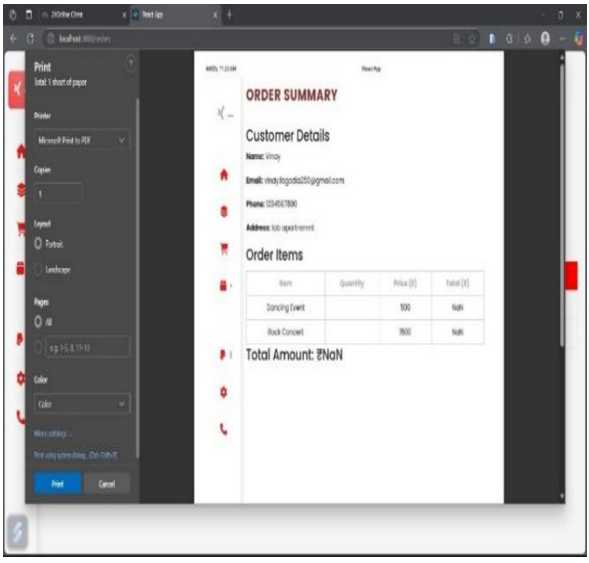

Fig. 2. Order Summary Page

The design of order summary page is customer-friendly and responsive experience across all the available devices. The design is simple and customer-centric, where customer can validate their choices before clicking to pay. It adds greater transparency and lesser the chance of making errors in booking. Implementation results are shown in the screenshots attached below of the main interface components. **Fig. 3** shows the dynamic event cards, presenting various next events on our homepage.

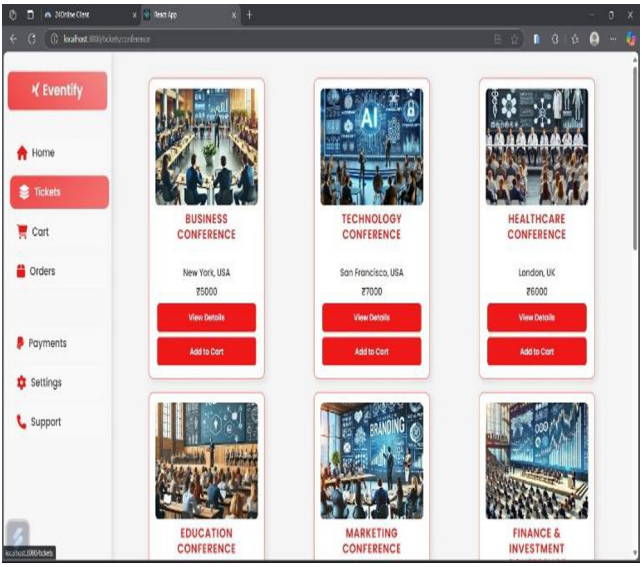

Fig. 3. Event Cards with Cart Integration

In each card there is also event information (title, picture, date, and price) as well as options to go to more details or add the event to the cart. It provides a more visually engaging experience and easier booking experience for the customer.

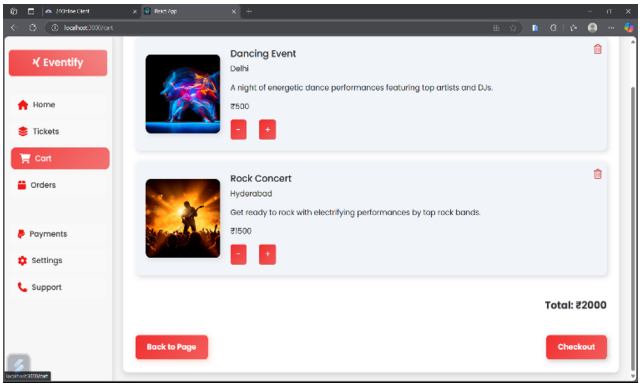

Fig. 4. Cart Details Page

Fig. 4 shows the cart page provides a detailed list of all the items the customer has added. It will display the title of the event, the number of tickets needed, and the individual prices and subtotals. From this page customers can alter their selection - or continue to checkout. With clarity and speed for customers, it sits between selection and payment, enabling customers to keep adding to their cart before making a final payment. The interaction between customer and the page is responsive across all the avail-

able devices. Buttons like back to page and checkout will help the customer to move towards their next step.

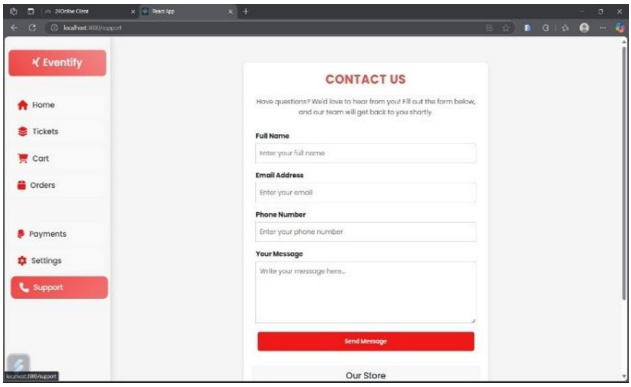

Fig. 5. Contact Us Page

Fig. 5 shows the platform has a dedicated support page where customers can ask questions, request information, or report problems. It has a form with fields for the name, email, message, and preferred method of contact. That page helps build customer trust by providing easy and timely support. Support page will maintain the communication between the customer and the coordinator with the responsive design and customer-friendly.

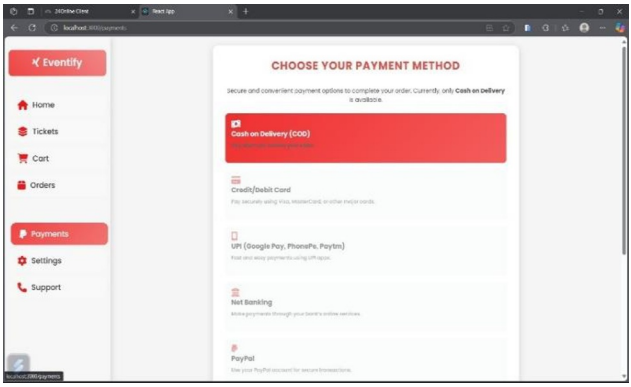

Fig. 6. Payment Method Page

Fig. 6 shows to accommodate customer preferences, a multi-option payment gateway had developed. The list of options on the payment page showcases Cash on Delivery (COD), Credit/Debit Cards, UPI, Net Banking and PayPal as possible selections. Customers can safely complete transactions due to the structured design and customer-

friendly interface. The payment page has responsive layout. The security is the first priority giving customers secure encryption while doing the payment. This page is the final stage of the customer journey for booking the event and customer satisfaction.

4 RESULT AND DISCUSSION

Reservation Collaborator is developed to provide a seamless digital platform to manage and book events, with application user roles across Admins, Customers, and Coordinators. The goal of the project is to make it intuitive, responsive, and feature-rich, adhering to modern web application conventions. Dynamic event cards are one of the big wins, enabling customers to engage with events, see critical info, and add events to a cart, all from one interface. Such interaction mimics modern e-commerce design patterns and therefore makes the application feel familiar and easy to use. Additionally, its responsive features ensured cross-device accessibility, which improved usability.

Features such as the cart and order summary helped achieve a goal of mimicking a real-world event booking system. Apart from effectively capturing event negative data, these modules also displayed them — and allowed for real-time updates, giving customers control over their choices before they headed off to checkout. With time you have a number of payment options including UPI, credit/debit cards, net banking, and PayPal. At the same time, this multiplicity of payment methods is particularly relevant if the platform seeks to expand beyond local usage. On the other hand, the support/contact page took care of customer service and provided a way for customer feedback and issue resolution, both of which are key for customer retention and trust.

Nonetheless, several limitations were also noted in implementation. While the customer experience is clean and responsive, the app currently lacks more complex functionalities, such as search filters, push notifications, or connection with an external calendar service (such as Google Calendar). It would also not fully integrate with live gateways, as payment processing is simulated. These limitations don't detract from the platform's core functionality but are challenges to improve upon. In conclusion, this project effectively provides a basic building block of a scalable events booking system that is extensible and can be built upon and upgraded over time to satisfy more complicated customer needs.

5 FUTURE DIRECTIONS

Having successfully implemented the first iteration of the Event Final Project, the following development cycle will be spent improving the platform with both further functionality and a better customer experience. Although the current version allows customers to browse events, add them to the cart, and checkout with multiple payment options, many key features and optimizations are planned to enhance the robustness, scale, and usability of the system.

Among the most important improvements will be the integration of live payment gateway. At present, the project mimics the payment options of UPI, credit/debit cards, net banking, and PayPal, yet the goal will be to integrate real-time secure payment

gateways like Razor pay, Stripe, or PayPal SDK to enable the platform to process genuine transactions. This will allow the application to evolve from a demo system to a running, producible application suitable for real-world use.

The other major upgrade is the addition of a filter search mechanism. As events scale, customers will struggle to find events per their preferences. Allow the customer to refine their browsing experience with filters by event type, date, geography, and price range among other factors. Such functionality would be highly valuable for customers who have particular preferences or fewer time slots available. However, the platform can always be matured by including event ratings and customer reviews which will further drive engagement and credibility. Customers would also be able to review events they attended and share feedback that other customers could then see when deciding whether they want to attend the same event or not. This builds transparency and quality control of events that are displayed on our platform. Furthermore, integrating email and SMS notification systems for booking confirmations, event reminders, and promotional offers will ensure that customers stay informed and engaged. On the admin side, the next step also needs to provide a dashboard for the event organizers and admins with insights on customer behaviors, booking behaviors, event popularity, and revenue analytics. It can help coordinators make better plans, and adjust them to real-time data. Introducing event approval workflows will help to ensure just verified occasions appear on the system, thus preventing any garbage from being listed on the platform.

And finally, upgrading the support system with a live chat option, or the introduction of an AI-powered chatbot, to provide a better customer support experience. Since customers won't have to wait for email replies and can resolve their queries immediately. An issue tracking and ticketing system answering to the democratically approved ticket themes could also manage and track complex issues and solve them accordingly.

CONCLUSION

This Reservation Collaborator development is the final stage of creating a complete and easy to use web application for booking events. The goal of this project is to create a platform that connects event organizers and attendees, particularly a centralized point that allows customers to browse, select, or register for various events. The focus during the development process is on key e-commerce features such as the ability to search and browse events, add them to the cart, view detailed order summaries, and pay via multiple mediums like cash on delivery, UPI, net banking, credit/debit cards, and PayPal. Also added page like support (Contact Us) to provision customers to reach out for their help, which in turn increases trust and customer satisfaction. Indeed, the project emphasized responsive design, intuitive navigation, and modular coding approach to ensure scalability and maintainability for future enhancements. Incorporating elements such as event cards, cart summary, and placing orders (dynamic) lies astoundingly with an understanding of unique front-end and back-end web development principles. In addition, building a real-life application relied heavily on modern development tools, technologies, and frameworks that helped to set a strong foundation. The site has been tested with various flows each one performing well with no surprises under a typical use scenario. In summary, the Reservation Collaborator demonstrates strengths in web development skills, customer empathy, UI/UX principles and e-commerce and check-

out flow knowledge. The project sets a solid foundation for an expandable, production-worthy event management apparatus that could develop sound systems for organizing future sources of inspiration after determining immediate goals like integrating real-time payments, host reviews, and a dashboard and others like search and filtering for customer to process the information they collect. Not only does this initiative put theory into practice, it is also an example of project planning and problem-solving.

REFERENCES

1. Kumar, N. V., & Sriram, B. (2023). Development of Flexible Sharing and Leasing a Car Using MERN Stack (Bachelor's thesis, Muthayammal Engineering College).
2. Sugiartha, I. G. A. (2020). Design of Information System for Bookstore Support Student. International Journal of Computer Applications, 175(30), 1–5. https://doi.org/10.5120/ijca2020920701
3. Gupta, A., Gupta, K., Mishra, V., & Jain, P. (2024). "Designing of Hotel Reservation Application: A Case study." 2024 1st International Conference on Advances in Computing, Communication and Networking (ICAC2N), 840–844. https://doi.org/10.1109/icac2n63387.2024.10895794.
4. Hemalatha, P., Dhavavarshini, M., & N, D. (2022). "Development of flexible, sharing and leasing a car using MERN Stack." 2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing and Communication Engineering ICATIECE, 1–5. https://doi.org/10.1109/icatiece56365.2022.10047214.
5. Singh, O. P. (2024). "The Synergy of Events and Technology: MERN Stack Solutions for Seamless event Rentals." International Journal for Research in Applied Science and Engineering Technology, 12(3), 3312–3316. https://doi.org/10.22214/ijraset.2024.59527.
6. Mhetre, N. M. A., Kadam, N. M., & Kalambe, N. S. (2024). "A survey on modern online ticket booking systems." International Journal of Scientific Research in Computer Science Engineering and Information Technology, 10(6), 981–987. https://doi.org/10.32628/cseit2410460
7. Santhoshkumar, S. P., Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, A, T., S, N., D, V., S, R., Santhoshkumar, S. P., & Rathinam Technical Campus. (2023). "Bus booking and reservation system using MERN technologies." Mukt Shabd Journal, XII(IV), 398.
8. Tirth, S. (2025). "Creating a virtual event platform using MERN Stack." International Journal for Research in Applied Science and Engineering Technology, 13(1), 452–454. https://doi.org/10.22214/ijraset.2025.66295.
9. Bakale, S. C. (2023). MERN STACK-BASED CAR RENTAL WEBSITE DEVELOPMENT. "International Journal of Advanced Research in Computer Science," 14(02), 39–44. https://doi.org/10.26483/ijarcs.v14i2.6971
10. Singh, A. K., Kushwaha, H., & Singh, E. S. (2024). Elevating online shopping using MERN stack. "International Journal for Research in Applied Science and Engineering Technology," 12(5), 5307–5309. https://doi.org/10.22214/ijraset.2024.62775.
11. Aneesh, R., Shah, A., D. M., A., S. R., A., & Thaseen Taj. (2020). Community web application for event management platform. *International Journal of Progressive Research in*

Science and Engineering, 1(5), 116–120. https://journal.ijprse.com/index.php/ijprse/article/view/165.

12. Pansare, A., Patil, A., Patil, N., Patil, Y., & Bhonde, A. (2023). Smart College Event Management System Using MERN Stack. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, 11(III), 4628–4635. https://doi.org/10.22214/ijraset.2023.49875
13. Macharla, S. C., & Aruna, V. (2025). All Events Hub Using MERN Stack. *International Journal of Research Publication and Reviews*, 6(8), 4293–4297. https://doi.org/10.55248/gengpi.6.0825.3120

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (http://creativecommons.org/licenses/by-nc-nd/4.0/), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.