



# QVent: A Centralized Digital Event Management and Ticketing Platform

Khushi Baraskar, Mahakrani Adwani, Gungun Jaiswal, Priyanshi Neema, Mr. Amit Kanungo,

Mr. Rakesh Jain

Guide – Mr. Rakesh Jain

Department of Computer Science & Engineering  
Indore Institute of Science and Technology, Indore (M.P.), India

## How to Cite this Article:

Baraskar, K., Adwani, M., Jaiswal, G., Neema, P. & Kanungo, A. (2026). QVent: A Centralized Digital Event Management and Ticketing Platform. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(05).  
<https://doi.org/10.55041/ijcope.v2i5.690>

## License:

This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.



<https://doi.org/10.55041/ijcope.v2i5.690>

## Abstract

The growing demand for efficient, secure, and eco-friendly event management systems has accelerated the shift from manual ticketing methods to digital platforms. This paper presents **QVent**, a centralized event management and ticketing solution designed to connect event organizers, administrators, and participants through a single web-based system. QVent enables organizers to register, create, and manage events ranging from cultural fests to hackathons, while attendees can browse, register, and purchase digital QR-based tickets. Each ticket is uniquely generated, ensuring secure and contactless entry verification. Built using Java, Spring Boot, PostgreSQL, and RESTful APIs, QVent aims to streamline participation, minimize fraud, and enhance user experience. The system follows an Agile development model and supports modular growth for future improvements such as AI-driven recommendations, dynamic ticket pricing, hybrid event support, and mobile application integration. This paper discusses the problem background, related work, methodology, system requirements, architecture, implementation approach, results, limitations, and future scope of QVent. It also includes a declaration of AI tool assistance used during manuscript preparation.

**Index Terms**— Event management, QR ticketing, digital platform, Spring Boot, Agile development, secure payments.

## I. INTRODUCTION

Event management has traditionally relied on paper-based ticketing, handwritten attendance lists, and manual verification at entry points. While these methods may work for small gatherings, they become inefficient when the number of attendees and events grows. Organizers often struggle with maintaining accurate records, preventing duplicate entries, updating event details, and managing registrations across multiple channels. These challenges increase administrative workload and reduce the overall quality of the participant experience.



The shift toward digital platforms has changed how events are planned, marketed, and accessed. Web-based applications now allow users to discover events, complete online registration, make payments, and receive digital confirmation instantly. QR-based ticketing has further improved the process by enabling contactless entry verification and reducing the risk of counterfeit tickets. However, many existing systems still focus only on ticket sales or event promotion, rather than offering a complete centralized workflow.

QVent is designed to solve these issues by providing a unified digital event management and ticketing platform. The system supports organizers, users, and administrators through a role-based architecture. It allows event creation, ticket booking, payment handling, QR generation, and attendance verification in one place. The goal is to make event handling faster, safer, and more transparent.

This paper presents the design and development of QVent using an IEEE-style structure. It explains the motivation, related work, system architecture, methodology, implementation details, and future possibilities of the platform. The proposed system is intended to be practical for college events, hackathons, seminars, workshops, and similar use cases.

---

## II. RELATED WORK

Many existing ticketing and event platforms provide online booking and registration features. Some applications support event discovery and promotional pages, while others focus on payment and ticket delivery. Although these systems improve accessibility, they often leave gaps in administration, organizer control, and fraud prevention. As a result, event staff may still need to rely on additional tools or manual procedures.

Manual ticketing systems continue to be used in many institutions because they are simple to deploy. However, they are vulnerable to duplication, misplaced records, delayed check-in, and poor scalability. Printed tickets and offline attendance sheets create inefficiencies when organizers need to verify large numbers of participants quickly. These weaknesses have pushed many event organizers toward QR-based or barcode-based systems.

QR code ticketing offers several advantages. A QR code can store a unique identifier linked to a booking record in the database. At the venue, the ticket is scanned and validated instantly. This reduces queue time, improves security, and supports contactless access. It also creates a digital audit trail that can be used for analysis and reporting.

Despite these advantages, many solutions are still fragmented. Some do not provide one centralized platform for organizers, attendees, and admins. Others lack role-based access or dashboard-level insights. QVent addresses these gaps by combining event management, booking, secure ticketing, and admin control into one system.

---

## III. PROBLEM STATEMENT

The central problem is the inefficiency of fragmented event handling. In many cases, organizers depend on spreadsheets, WhatsApp messages, manual bank transfers, printed passes, and separate verification methods. This leads to duplicate work, inconsistent records, and a higher chance of errors. It also makes it difficult to provide users with a smooth and modern event experience.

A second problem is the lack of secure and fast ticket verification. Paper tickets can be copied, shared, or lost. Even some basic digital systems do not provide proper entry validation. Without a unique QR-based system, organizers cannot reliably confirm whether a ticket is valid or already used.



A third issue is the absence of centralized management. Administrators need a way to approve events, monitor users, and access reports. Organizers need a dashboard to track sales and attendance. Attendees need an easy way to discover, book, and download tickets. QVent is proposed to address all of these needs within a single platform.

---

#### IV. PROPOSED SYSTEM

QVent is a web-based event management and ticketing platform built to support multiple user roles. The system is designed around three major actors: admin, organizer, and user. Each role has controlled access to the features that are relevant to it. This separation improves security and keeps the interface simple for each type of user.

The admin module handles user management, event approval, analytics, and system oversight. Organizers can create events, set prices, monitor bookings, and manage cancellations. Users can browse events, register online, make payments, and download QR-based tickets. Once a booking is completed, the system generates a unique ticket linked to the booking record in the database.

The QR ticket serves as a secure entry token. At the venue, the QR code can be scanned to verify the booking instantly. If the ticket is valid and unused, entry is allowed. If it has already been used or canceled, access is denied. This mechanism improves security and reduces fraudulent entry.

The platform is intended to be responsive and accessible from desktops, tablets, and mobile browsers. It supports different event types such as college festivals, technical symposiums, workshops, sports events, and hackathons. Because the system is modular, future features such as recommendation engines, dynamic pricing, and mobile apps can be added without changing the entire architecture.

---

#### V. METHODOLOGY

QVent follows the Agile software development approach. Agile is appropriate for this project because requirements can evolve as users test the system and provide feedback. The development process is divided into small, iterative cycles that include planning, development, testing, and deployment.

During the planning phase, requirements were identified by analyzing common event workflows and user expectations. The main modules were then divided into smaller tasks such as authentication, event listing, ticket booking, payment processing, QR generation, and report management. This modular approach makes the system easier to design and maintain.

In the development phase, each sprint focused on one or more features. The backend services were built first, followed by database integration and front-end pages. Testing was performed throughout the process using functional testing, manual verification, and API checks. The system was refined after each sprint to improve reliability and usability.

Deployment was planned on cloud infrastructure with SSL/TLS support to ensure secure communication. The Agile model helps QVent remain flexible, scalable, and user-oriented. It also allows the system to evolve as new requirements emerge.

---



## VI. SYSTEM REQUIREMENTS AND ANALYSIS

### A. Functional Requirements

The system must support user registration, login, event browsing, event creation, booking, payment confirmation, QR ticket generation, and ticket verification. It should also allow administrators to manage users, approve events, and generate reports. Organizers need access to sales monitoring and event management tools. Users need a simple and responsive interface for registration and ticket download.

### B. Non-Functional Requirements

QVent must be secure, scalable, reliable, and easy to use. Security is required for protecting personal data, payment details, and booking integrity. The system must remain responsive even under heavy usage during peak registration periods. Reliability and availability are essential because event booking often happens within limited time windows.

### C. Technical Requirements

The frontend technologies include HTML5, CSS3, JavaScript, Bootstrap, and Thymeleaf. The backend is developed using Java and Spring Boot. PostgreSQL is used as the relational database because it offers strong transaction support and stable data storage. RESTful APIs, Maven, and a payment gateway are also used in the implementation.

### D. Feasibility Analysis

The project is technically feasible because the selected technologies are mature and well-documented. It is economically feasible because the stack relies mostly on open-source tools. Operationally, the system is practical because online booking and QR-based entry are already familiar to users. The platform is also scalable enough for future improvements.

---

## VII. SYSTEM ARCHITECTURE

QVent follows a layered web application architecture. The presentation layer is responsible for user interaction, including event browsing, forms, and dashboards. The application layer handles the business logic such as ticket generation, booking validation, and role-based control. The data layer stores user information, event details, ticket records, and transaction data.

When a user submits a booking request, the front end sends the data to the backend through an API. The backend validates the request, checks availability, creates the booking, and generates a unique QR ticket. The ticket is then displayed or downloaded by the user. At the venue, the QR code is scanned and matched with the database to confirm its status.

This design separates concerns and improves maintainability. If payment logic changes or a new module is added, the rest of the system can continue to function normally. It also makes debugging and testing easier because each layer has a clear responsibility.



## VIII. IMPLEMENTATION DETAILS

The implementation is based on Spring Boot because it simplifies dependency management and backend configuration. Controllers are used to handle requests, services contain business logic, and repositories communicate with PostgreSQL. Thymeleaf is used to render dynamic web pages. This structure keeps the code organized and easier to expand.

Role-based authentication is an important part of the implementation. Users are assigned roles so that admin, organizer, and attendee features remain separated. This prevents misuse and helps protect sensitive data. Session handling and form validation further improve security and correctness.

QR ticket generation takes place after successful registration and payment confirmation. The ticket contains a unique code that is linked to a record in the database. At the time of scanning, the system checks whether the ticket is valid, used, or expired. This helps prevent duplicate entry and improves check-in speed.

The interface is designed to be simple and practical. Event cards, booking pages, dashboard views, and ticket pages are arranged to support easy navigation. Responsive design ensures compatibility across different screen sizes. Overall, the implementation emphasizes usability, security, and maintainability.

---

## IX. RESULTS AND DISCUSSION

The QVent platform demonstrates that centralized digital event management can significantly improve event operations. Organizers can manage events from a single dashboard and view sales activity in real time. Attendees receive a smooth booking process and instant access to their tickets. Administrators gain control over event approval, user oversight, and monitoring.

The QR-based ticketing mechanism improves the entry process by reducing delays and minimizing fraud. Compared with manual systems, it offers better traceability and faster verification. This is especially useful for high-attendance events where queues and access control are important concerns.

The platform also supports sustainability by reducing the need for printed tickets and paper records. Since data is stored digitally, it can be searched and analyzed more easily. This makes QVent useful not only for event execution but also for reporting and planning.

Overall, the results indicate that QVent is a practical and scalable solution for modern event environments. It provides a complete workflow for organizers and users while remaining flexible enough for future feature enhancements.

---

## X. LIMITATIONS AND FUTURE SCOPE

QVent has some limitations. The system depends on internet connectivity for booking and verification, so users in low-connectivity areas may face difficulties. As with any digital platform, data protection, server uptime, and backup management are important concerns. These issues must be addressed carefully during deployment and maintenance.

Future development can make the system more intelligent and useful. AI-powered recommendations can suggest events based on user behavior or interests. Dynamic pricing can help organizers adjust ticket prices according to demand. Hybrid and virtual event support can extend the platform to online participation as well.



A mobile application could further improve accessibility and convenience. Additional analytics features may help organizers understand attendance trends and improve planning. These future enhancements would make QVent a more complete event ecosystem.

---

## XI. CONCLUSION

This paper presented QVent, a centralized digital event management and ticketing platform designed to simplify event creation, registration, ticketing, and verification. The system provides a unified environment for organizers, users, and administrators and reduces dependence on manual processes.

By using Spring Boot, PostgreSQL, RESTful APIs, and QR-based tickets, QVent improves efficiency, security, and user experience. The Agile development approach supports modular implementation and future expansion. The system is suitable for college events, cultural programs, workshops, and other event formats.

---

## Declaration of Generative AI and AI-Assisted Technologies in the Manuscript Preparation Process

During the preparation of this manuscript, the authors used AI tools to assist with organizing the content, improving language clarity, and refining the structure of the paper. The authors carefully reviewed, edited, and verified the final manuscript and take full responsibility for its content.

---

## REFERENCES

- Agile Alliance, "Agile 101," 2026. [Online]. Available: <https://www.agilealliance.org/agile101/>
- PostgreSQL Global Development Group, "PostgreSQL Documentation," 2026. [Online]. Available: <https://www.postgresql.org/docs/>
- Oracle, "The Java Tutorials," 2026. [Online]. Available: <https://docs.oracle.com/javase/tutorial/>
- Spring, "Spring Boot Reference Documentation," 2026. [Online]. Available: <https://spring.io/projects/spring-boot>
- Mozilla, "MDN Web Docs," 2026. [Online]. Available: <https://developer.mozilla.org/>
- W3Schools, "Web Development Tutorials," 2026. [Online]. Available: <https://www.w3schools.com/>
- IEEE, "IEEE Author Center: Conference Templates," 2026. [Online]. Available: <https://ieeauthorcenter.ieee.org/>