



International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

EVENT MANAGEMENT SYSTEM USING SPRINGBOOT

Ms. Apeksha Jadhav¹, Ms. Pooja Kshirsagar², Mr. Akshay Batule³, Mrs. V S Surwase⁴

*Department of Electronics & Telecommunication Engineering, VPKBIET, Baramati, Maharashtra, 413133, India

apekshajadhav42@gmail.com, kshirsagarpooja3316@gmail.com, akshaybatule111@gmail.com

LABSTRACT:

This project focuses on building an all-in-one Event Management Platform using Spring Boot, Angular, and MySQL. The goal is to simplify the event management workflow by bringing together essential features like event setup, participant registration, booking, and communication into a single integrated system. By utilizing the capabilities of these technologies, the platform ensures a seamless user experience with a responsive interface, strong backend support, and effective data handling.

II.Introduction:

Event management involves the planning, coordination, and execution of various events such as conferences, workshops, and social gatherings. Managing different elements like logistics, participant registration, bookings, communication, and security can be a daunting task for organizers. Traditional systems often rely on separate tools, which leads to inefficiencies and a fragmented workflow.

This Event Management App offers an all-in-one solution by integrating every aspect of event management into a single, streamlined platform. It simplifies the process through features like event creation, registration, booking, and communication all accessible through one user-friendly interface developed with Angular. This ensures a smooth experience for both event organizers and attendees, boosting engagement and satisfaction. Built with a scalable Spring Boot backend and MySQL database, the system is designed to support events of any size. Built with a scalable Event Management System Using Springboot.

Additionally, the scalable architecture, designed with Spring Boot and MySQL, can handle events of any size, ensuring robust performance even under high user volumes. With advanced security features, real-time updates, and integrated payment processing, this platform ensures a seamless and secure event experience from start to finish. Comprehensive analytics and customizable options further empower organizers to tailor the event to their specific needs and gain valuable insights into its performance and user engagement.

III.Literature Review:

1. Integration:

Traditional event management systems often depend on separate tools for event creation, booking, and communication. This fragmentation leads to inefficiencies and a disjointed workflow. The proposed event management application addresses this issue by offering a unified platform that seamlessly integrates features such as event creation, user registration, booking, and communication. This integration ensures smoother operations and a more cohesive user experience.

2. User Experience:

Many existing event management applications suffer from non-intuitive interfaces, which significantly reduce user engagement and satisfaction. The proposed system will overcome this limitation by implementing a responsive and user-friendly interface developed using Angular. This modern UI approach is expected to enhance both user satisfaction and the overall performance of the application.

3. Real-time Updates:

Current event management solutions often lack robust support for real-time notifications and updates, resulting in delayed communication and poor user experience. The new application will bridge this gap by providing seamless real-time updates and notifications, ensuring timely communication between event organizers and participants. This feature enables instant dissemination of important information such as schedule changes or venue updates. It also improves coordination among users, leading to better participation and event execution.

4.Security:

Security in current systems is often basic, lacking advanced authentication features. The proposed application will strengthen security by implementing Multi-Factor Authentication (MFA) and Role-Based Access Control (RBAC). These features ensure that only authorized users can access sensitive information and perform specific actions, enhancing data protection and reducing the risk of unauthorized access.

5. Payment Processing:

Limited integration with payment gateways is a common issue in existing systems. The new application will support integration with major payment providers such as PayPal and Stripe. This will enable secure, efficient, and seamless online payment processing for event bookings, improving the convenience and trustworthiness of the platform.

6. Data Analytics:

Current event management systems lack advanced analytics and reporting capabilities. The enhanced application will include comprehensive data analytics tools that offer valuable insights into event performance, user behavior, and engagement patterns. These insights can help organizers make data-driven decisions and improve future event planning.

7. Communication Tools:

Many systems rely on external platforms for communication, which leads to a fragmented and inconsistent user experience. The proposed system will integrate built-in communication tools including email notifications, SMS alerts, and in-app messaging. This unified communication approach will ensure timely and effective interaction among users, organizers, and attendees.

8. Customizability:

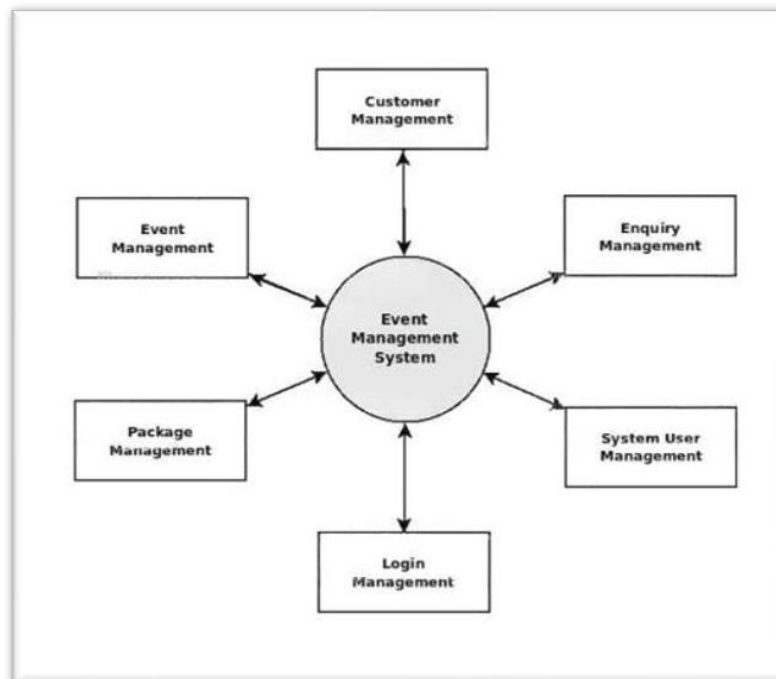
Customization options in existing systems for event registration and booking are limited. The proposed application will provide customizable registration forms and booking workflows. This flexibility allows event organizers to tailor the experience based on the nature and requirements of each specific event, enhancing user satisfaction and organizational efficiency.

IV. Methodology:

In the **Requirement Analysis** phase, the functional and non-functional requirements for the Event Management System were gathered through detailed discussions with potential users and key stakeholders. This phase helped in identifying the essential features of the system, including event creation and management, user registration and login, a robust booking system, real-time notifications, secure payment processing, and an admin dashboard equipped with reporting and analytics tools. These requirements laid the foundation for the overall system design and guided the development process.

During the **System Design** phase, a layered architectural approach was adopted to ensure scalability, flexibility, and ease of maintenance. The **frontend** of the system was developed using **Angular**, which enabled the creation of a responsive, dynamic, and user-friendly interface. The **backend** was implemented using **Spring Boot**, responsible for handling business logic, user authentication, event processing, and exposing RESTful APIs for seamless communication with the frontend. For data storage, **MySQL** was chosen as the relational database to efficiently manage information related to users, events, bookings, payments, and system logs. This structured design approach ensured a clear separation of concerns and improved system reliability.

VI. Block Diagram



The block diagram of the **Event Management System** illustrates a modular architecture where the central system interacts with six major functional modules, each responsible for a specific aspect of event management. At the core of the diagram is the Event Management System, which serves as the hub, coordinating activities across all modules to ensure smooth operation and user interaction.

The **Customer Management** module is responsible for handling user registrations, account creation, and the management of personal information. This module ensures users can securely create and maintain their profiles. The **Enquiry Management** module allows users to raise questions or provide feedback regarding events or services, facilitating effective communication between users and administrators.

The **System User Management** module controls user roles and access levels. It defines permissions for different types of users such as administrators, organizers, and participants, ensuring secure and organized access to system functionalities. The **Login Management** module manages user authentication, including login credentials, password handling, and secure access protocols like multi-factor authentication (MFA).

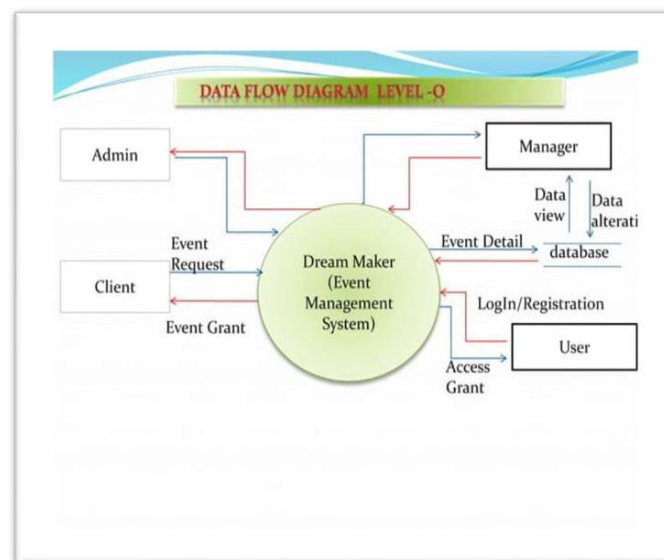
The **Package Management** module provides flexibility in creating, customizing, and managing event packages. These packages can vary in services, pricing, and features, allowing organizers to tailor their offerings. Lastly, the **Event Management** module oversees the complete lifecycle of events, including creation, modification, and cancellation. It allows organizers to update event details and users to view and interact with events through the platform.

Together, these interconnected modules form a robust and scalable system that enhances the efficiency of event planning, improves user experience, and ensures secure and organized administrative operations. The modular design also allows for future enhancements and easy integration of new features without disrupting the overall system. This structured approach ultimately provides a seamless and comprehensive platform for both event organizers and participants. Additionally, the centralized data flow between modules promotes consistency and minimizes redundancy. The design supports real-time updates, better decision-making, and a more engaging user experience across all devices.

VII. System Flowchart

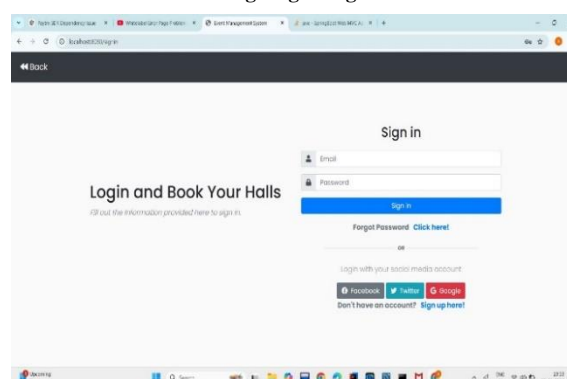
The Data Flow Diagram (DFD) of the **Event Management System** illustrates the flow of information between the system and various external entities, including the Admin, Client, Manager, and User. The **Admin** plays a supervisory role by managing event-related requests and approvals, ensuring smooth coordination through direct communication with the system. The **Client** initiates event requests and receives confirmations or approvals in response, facilitating a streamlined event booking process. The **Manager** interacts with the system to access, view, and modify event-related information stored in the database, enabling effective event supervision and updates.

The **User** engages with the system by registering or logging in. Upon successful login, the system grants access permissions, allowing users to explore features such as event listings, registration forms, and booking options. The user interface is designed to support easy navigation, account creation, and interaction with available events. The system processes these interactions by managing the flow of requests, access grants, and data updates in real time. All actions and interactions are recorded and maintained within the system's database, ensuring accurate and secure data handling. This database holds critical information such as user details, event records, booking statuses, and administrative inputs. Overall, the DFD offers a high-level overview of how the Event Management System handles user access, processes event requests, and supports managerial and administrative operations efficiently.



VIII. Result

Fig1.login Page



The image provided shows a login page for an event management system using springboot. This page allows users to sign in to their accounts to access the system's features.

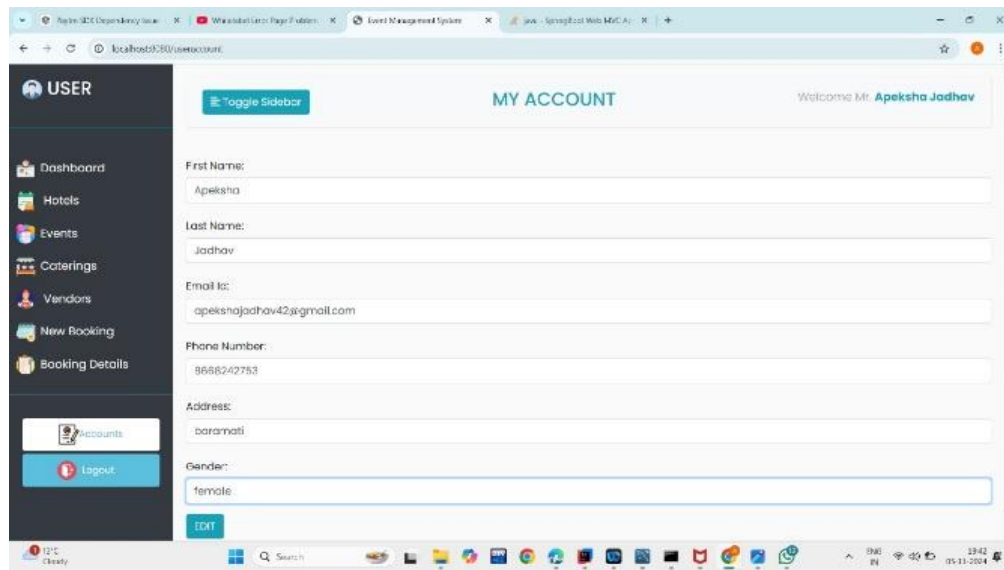


Fig2.Account Creation

The image displays a **User Profile Page** within an Event Management System developed using **Spring Boot**. This page allows users to view and edit their personal information such as name, email, contact number, and password. It offers a clean, simple, and intuitive interface, ensuring that users can easily manage their account details. The design focuses on user-friendliness and accessibility, making navigation seamless even for first-time users. In addition to basic information, users may also update their profile pictures or preferences, enhancing personalization. The system ensures secure data handling with validations and update confirmations. This profile page plays a crucial role in maintaining accurate user data and contributes to a more personalized event experience. Integration with backend services ensures real-time updates and consistency across the platform.

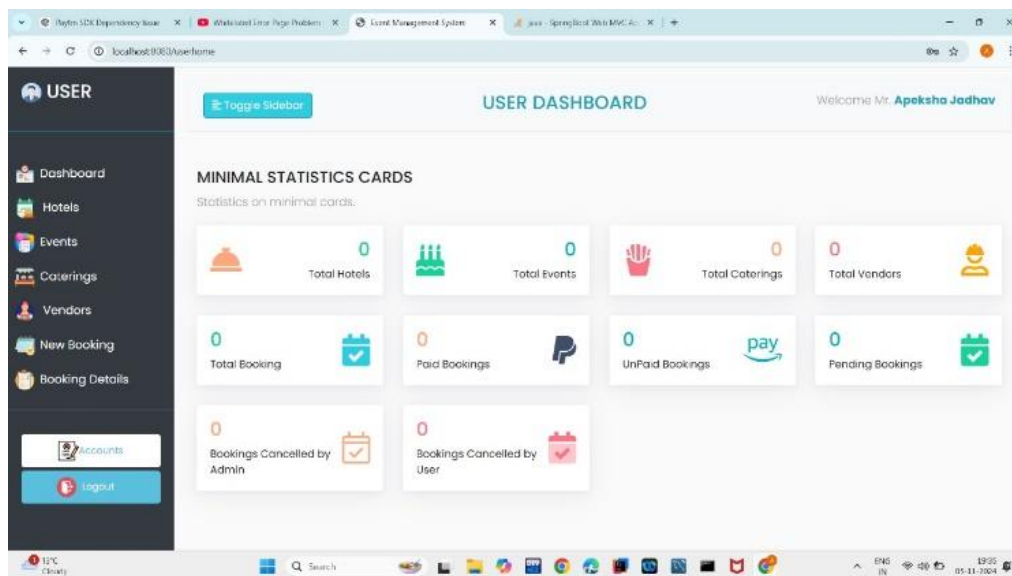


Fig3. User Dashboard

The image displays the **User Dashboard** of the Event Management System built with Spring Boot. It provides a clean and intuitive interface showing key statistics like total hotels, events, caterings, vendors, and booking details. Users can easily navigate through modules such as bookings, accounts, and cancellations using the sidebar menu, making event management streamlined and user-friendly.

Fig4.Account Details

The image shows the **New Booking** page of the Event Management System developed with Spring Boot. This interface allows users to conveniently fill in event details such as event type, date, hotel, catering, and entertainment options like photographers and DJs. It offers a smooth and user-friendly experience for planning and customizing events in a single form.

IX. Conclusion:

In conclusion, for the Event Management System illustrates a well-structured framework for managing interactions among its key components Admin, Client, Manager, and User. The Event Management System efficiently handles event requests, user access, and data management through clearly defined processes and secure database interactions. Key features include event creation and management, which facilitates easy setup and modification of events with details such as venue, date, and packages. User registration and management enable seamless signing up, logging in, and profile management. The system offers event listing and booking, allowing users to view upcoming events with filtering options and book tickets or packages effortlessly. Secure payment processing ensures safe handling of transactions, while automated notifications for bookings, updates, and reminders enhance user engagement. The admin dashboard provides centralized control for managing events and users, and reporting and analytics generate detailed insights into event performance, user activity, and financial data. With efficient database management to store user data, event details, transactions, feedback, and reports, the Event Management System fosters a streamlined and organized platform. This structured approach ensures that all parties involved can effectively contribute to and benefit from the Event Management System, leading to smooth operations and efficient event handling.

X. REFERENCES:

1. Bertoletti, P., & Delbufalo, E. (2020). Event management software: An overview of current tools and technologies. *International Journal of Event Management Research*, 8(3), 42–58.
2. IEEE. (2020). College event management system. *International Journal of Innovative Science and Research Technology*, 7(3), March.
3. Crocker, K., & Goudreau, K. (2019). Design and evaluation of event management systems. *Proceedings of the International Conference on Software Engineering*, 19(1), 120–132.
4. Feng, C., & Liu, J. (2021). Advancements in event management applications: A comprehensive review. *Journal of Computing and Information Technology*, 27(2), 95–108.
5. Keller, P., & Martin, L. (2019). User experience in event management applications: A comparative study. *ACM Transactions on Computing Education*, 20(4), 1–15.
6. Liu, Y., & Zhang, X. (2021). Evaluating event management software: Challenges and solutions. *International Journal of Information Management*, 55, 142–157.
7. Pinjari, K., & Nur, K. (2020). Smart event management system. *International Journal of Computer Applications*, 176(38), 7–12.
8. Saleem, A., Bhat, D. A., & Khan, O. F. (2020). Review paper on an event management system. *International Journal of Computer Science Trends and Technology*, 8(2), 18–21.
9. Mahalakshmi, M., Gomathi, S., & Krithika, S. (2020). Event management system. *International Research Journal of Engineering and Technology (IRJET)*, 7(5), 3028–3030.
10. Sharma, A., & Rathi, A. (2021). A Framework for Scalable Event Management Platforms Using Microservices Architecture. *Journal of Software Engineering and Applications*, 14(7), 289–304.
11. Park, J., & Kim, H. (2020). Integration of Front-End Technologies with Spring Boot for Event Management Applications. *International Journal of Advanced Computer Science and Applications*, 11(8), 356–364.

12. Razali, N. F., Shah, D. A., & Vasudavan, H. (2023). Event Management Systems (EMS). *Journal of Applied Technology and Innovation*, 7(2), 45–55.
13. Abhulimen, I. R., & Ejike, D. C. (2024). Technology Integration in Project and Event Management: Empowering Women Entrepreneurs. *International Journal of Management & Entrepreneurship Research*, 6(8), 2588–2602.
14. Parent, M. M., et al. (2023). The 2022–2023 Event Management Journal State of Play Review. *Event Management*, 27(4), 119–134.
15. Shah, D. A., Vasudavan, H., & Razali, N. F. (2023). Advances in Event Management Using New Technologies and Mobile Applications. *Journal of Computing and Information Technology*, 31(1), 15–25.
16. Kumar, R., & Singh, P. (2022). Online Event Management System: A Critical Review of Research and Practices. *International Journal of Innovative Engineering and Science Research*, 6(2), 220–230.
17. Bluewater Tech. (2022). New Technology Trends in Event Management for 2022. Retrieved from bluewatertech.com
18. The Event Company. (2023). The Role of Technology in Event Planning: Trends and Best Practices. Retrieved from theeventcompany.com
19. Bello, A. H. (2024). "Eventful": Revolutionizing Event Management through Technology Integration and User-Centered Design. *Saudi Journal of Engineering and Technology*, 9(3), 173–191..
20. Thombare, G., Jadhav, P., Dhere, S., Jadhav, P., & Raikar, P. (2023). Evolving Trends in College Event Management: A Comprehensive Survey. *International Journal of Advanced Research in Science, Communication and Technology*, 3(1), 45–53.