Eventify- Event Management System

Harsh Gupta[1], Harshit Gupta[2], Harshit Upadhyay[3], Smita Bisht[4]
[1,2,3]*Student, Dept. of Computer Engineering, Poornima institute of engineering, Technology Jaipur, India*
[4]*Guide, Dept. of Computer Engineering, Poornima institute of engineering, Technology Jaipur, India*

Abstract: In the era of digital transformation, effective event management is essential for seamless organization, attendee engagement, and real-time data tracking. The goal of Eventify, a cloud-based event management system based on the Salesforce platform, is to streamline and automate the entire event management process. Eventify combines features like event planning, speaker and attendee management, automated notifications, and performance analytics into a single, scalable solution by utilizing Salesforce's robust CRM capabilities. By improving collaboration and lowering manual labor, this system seeks to simplify event operations for businesses, organizations, and educational institutions. The project places a strong emphasis on data integrity, user-friendly design, and adaptability to different kinds of events. Through Eventify, we demonstrate the potential of low-code platforms like Salesforce in delivering enterprise-grade applications with minimal development overhead and high customization capabilities.

Keywords: Event Management System, Salesforce, Cloud Computing, CRM, Low-Code Platform, Automation, Attendee Management, Speaker Management, Apex, Lightning Components, Workflow Automation, Event Scheduling, Data Analytics, Custom Objects, Digital Transformation.

INTRODUCTION

Efficient event planning and management are more important than ever in the fast-paced, technologically advanced world of today. A variety of tasks, including scheduling, registration, attendee tracking, speaker coordination, and performance analytics, are involved in event management, from academic seminars to corporate conferences and cultural festivals. Conventional approaches frequently depend on manual procedures or disjointed systems, which results in inefficiencies, inconsistent data, and a dearth of real-time insights.

For communities, institutions, and organizations looking to promote cooperation, engagement, and knowledge exchange, event management has emerged as a critical element. The need for a reliable, scalable, and automated system has increased dramatically in tandem with the growing demand for events that are data-driven and well-coordinated. In addition to being time-consuming, traditional event management techniques—which frequently rely on spreadsheets, emails, and disconnected tools—are also prone to mistakes, inefficiencies, inefficiencies, and communication gaps.

The goal of Eventify, a cloud-based event management system built on the Salesforce platform, is to automate and optimize the entire event planning and execution process. Using Salesforce, one of the top CRM platforms in the world, this system offers a unified approach to managing speakers, events, and attendees while facilitating real-time updates, tracking, and communication.

Using the platform, administrators can create events, send automated reminders, manage speakers and attendees with custom objects, and use Salesforce dashboards and reports to analyze event performance. Eventify was created using a low-code methodology and makes use of tools like Process Builder, Flows, Apex, and Lightning Web Components to provide a very flexible and interactive user experience. Additionally, it facilitates security, scalability, and third-party tool integration, making it appropriate for various types and sizes of events.

This project not only solves practical challenges in event coordination but also showcases how Salesforce can be leveraged as a development platform beyond traditional CRM use-cases. Our goal with Eventify is to offer a complete, dependable, and easy-to-use solution that boosts participant engagement, increases event efficiency, and permits data-driven decision-making.

LITERATURE REVIEW

The use of cloud computing platforms and digital technologies has caused a major change in the event management industry. The goal of numerous studies and systems has been to enhance the planning, administration, and assessment of events. Manual techniques or simple software tools like spreadsheets and email correspondence were used in event management in the past, but these were ineffective, prone to mistakes, and challenging to scale. The limitations of these approaches have prompted researchers and developers to explore more integrated and automated solutions.

Several commercial event management systems such as Eventbrite, Cvent, and Whova offer extensive features like online registration, ticketing, scheduling, and attendee engagement. While these platforms are widely used, they often involve subscription costs, lack customization, and do not integrate well with internal organizational systems. Moreover, their closed-source nature makes it difficult for educational institutions and small-scale organizations to tailor the systems according to their specific requirements.

Salesforce, originally known for its robust Customer Relationship Management (CRM) capabilities, has evolved into a comprehensive cloud platform that supports custom application development using tools like Apex, Visualforce, Lightning Components, and Salesforce Flows. Studies have highlighted the benefits of low-code development platforms like Salesforce in enabling faster application development, improved scalability, and ease of integration with third-party services.

Existing research also emphasizes the growing relevance of CRM systems in areas beyond sales and marketing, particularly in education and event management. Researchers have explored the use of Salesforce to create applications for student tracking, alumni management, and academic scheduling—demonstrating its flexibility and power as a platform.

However, there is limited literature focused on developing an end-to-end event management solution specifically on Salesforce that addresses the needs of both organizers and attendees. This gap presents an opportunity to design a system like Eventify, which not only utilizes Salesforce's cloud-native infrastructure but also demonstrates how its tools can be harnessed to build a modular, scalable, and interactive event management platform.

The development of Eventify contributes to this growing body of research by bridging the gap between cloud-based CRM systems and custom event management requirements. It combines user-friendly interfaces, automation workflows, and real-time data insights to deliver a seamless experience for all stakeholders involved in event management.

Moreover, literature has noted that effective event management systems must support key features like role-based access, user-friendly dashboards, scheduling mechanisms, notification automation, and post-event analytics [8]. Modern systems are also expected to support mobile responsiveness and seamless integration with third-party applications such as Google Calendar, email services, and social media platforms for better outreach and collaboration [9].

Despite these advancements, the gap in academic and open-source contributions specifically targeting Salesforce-based event management systems remains evident. There is minimal scholarly work demonstrating how Salesforce can be systematically used to build an end-to-end, cloud-native event management solution that incorporates dynamic workflows, stakeholder communication, and post-event evaluation tools.

The proposed system, Eventify, aims to fill this gap by developing a fully functional, modular Event Management System on Salesforce. It leverages custom Salesforce objects (e.g., Event, Attendee, Speaker), built-in automation features, and custom interfaces to deliver an interactive and customizable platform. Through Eventify, this project demonstrates not only the practical viability of Salesforce as a development platform but also contributes to research in low-code enterprise application design for real-world event use cases.

METHODOLOGY

The development of Eventify followed an Agile methodology, which enabled iterative progress through incremental sprints. Gathering requirements was the first step, which included talking to and surveying potential end users, such as event planners and attendees. Understanding typical issues like ineffective attendee management, a dearth of real-time updates, and laborious event scheduling was the aim. Features such as event creation, speaker management, automated notifications, and real-time performance analytics were among the key functional and non-functional requirements for the system that were identified during this phase.

The system design phase started after the requirements phase. In this phase, Salesforce's declarative tools and custom objects were used to develop the architecture. To store and manage pertinent data, key custom objects were created, including Event__c, Attendee__c, and Speaker__c. Salesforce's master-detail and lookup relationships were used to define the relationships between these objects. Lightning Web Components (LWC) and the Lightning App Builder were also used to complete the user interface (UI) design, guaranteeing a responsive, user-friendly interface that is compatible with desktop and mobile devices.

The core of the development phase focused on the implementation of Salesforce’s automation tools such as Flows, Process Builder, and Apex Triggers. Key procedures like speaker scheduling, attendee notifications, and event registration were automated with their help. To offer a customized user experience, Lightning Pages were created for the various user roles (Admin, Organizer, Attendee). Effective user communication was ensured by integrating email reminders and event confirmations. Both unit testing and user acceptance testing (UAT) were part of the testing phase that followed the initial development. Apex Test Classes were used to conduct unit tests in order to verify that the code logic was executed correctly. In order to make sure the platform satisfied the needs of event planners and attendees, user testing entailed confirming its usability and functionality across various devices. The system's subsequent iteration took into account the input gathered from these tests.

The system was made available on the Salesforce Developer Org after it had undergone extensive testing. User profiles and permission sets were set up to offer role-based access control during the deployment phase, guaranteeing data security and appropriate access to the appropriate stakeholders. Planning for upcoming updates, such as the inclusion of feedback forms, third-party calendar integrations, and additional customization based on user feedback, was part of the methodology's maintenance phase. This agile approach ensured that the Eventify system was not only functional and scalable but also aligned with the evolving needs of users in the event management domain.

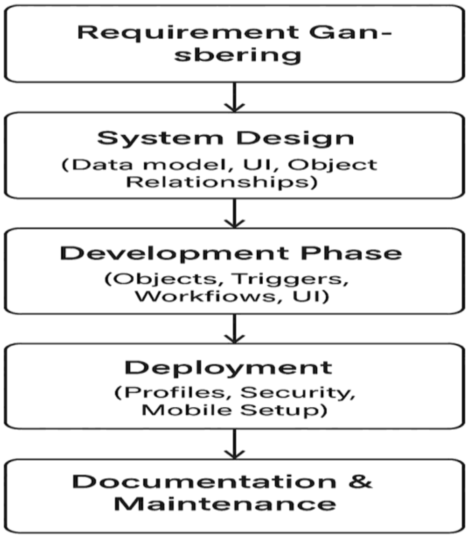
Requirement Gan-sbering
System Design
(Data model, UI, Object Relationships)
Development Phase
(Objects, Triggers, Workfiows, UI)
Deployment
(Profiles, Security, Mobile Setup)
Documentation & Maintenance

Figure:1 Eventify Architecture

A. System Workflow

The administrator logs in to the platform to control user roles and permissions, starting the Eventify system workflow. By setting up profiles and permission sets in Salesforce, the administrator can designate users as either organizers or attendees. After everything is configured, the organizer logs in and starts the event creation process. This entails completing the Event__c custom object with important information like the event name, date, time, location, and description. Through the Speaker__c object, the organizer can also create and link Speaker records if there will be guest speakers at the event, creating connections between speakers and events.

The event is published to the system once it has been fully defined. Salesforce automation tools like Process Builder and Flows are then activated to send notifications or announcements to potential attendees. Once the event is visible, Attendees can browse available events and proceed to register by filling out a simple form. The system sends the attendee a confirmation message via an automated email flow after they successfully register. To guarantee maximum attendance, the system also sends out emails or notifications a few days prior to the event. Attendees check in on the day of the event using the Salesforce interface, which is mobile-friendly, and their attendance is optionally recorded for future use. In order to help organizers improve future events, a feedback module could be added to future iterations of the system. This would allow attendees to share their thoughts about the event. Using Salesforce's robust CRM platform, this workflow offers an automated and structured approach to managing the entire event lifecycle, ensuring smooth coordination between users.

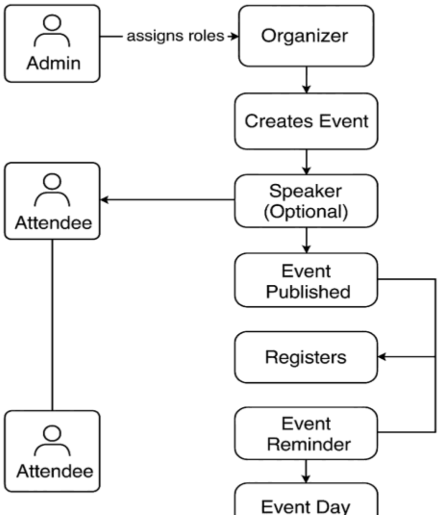

Figure:2 Eventify system Flowchart

B. Tools and Frameworks

The system development employs the following key technologies:

Salesforce Platform

A cloud-based environment called the Salesforce Platform offers resources and services for creating and implementing applications. With the help of features like data modeling, automation, and integration capabilities, developers can produce safe and scalable applications.

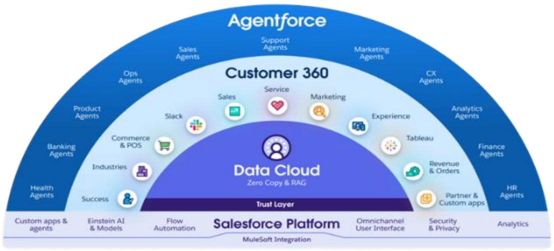

Figure:3 Salesforce Platform

Salesforce Developer Edition

Developers can create and test apps on the Salesforce Platform using the free, feature-rich Salesforce Developer Edition. It gives users access to the newest features and APIs, enabling development and experimentation without compromising production data.

Figure:4 Salesforce Developer Edition

Lightning Web Components (LWC)

A cutting-edge framework for creating reusable, effective Salesforce Platform components is called Lightning Web Components. It allows developers to create dynamic and responsive user interfaces by utilizing common web technologies like HTML, CSS, and JavaScript.

Apex Programming Language

Apex is an object-oriented, strongly-typed programming language that is used to create unique business logic for the Salesforce Platform. Together with calls to the API, it enables developers to run flow and transaction control statements on Salesforce servers.

Flow Builder

A visual tool called Flow Builder makes it possible to automate intricate business procedures without knowing how to write code. Through a drag-and-drop interface, users can design flows that can carry out operations like emailing, updating records, and gathering user input.

SOQL (Salesforce Object Query Language)

Data can be retrieved from Salesforce's database using SOQL, a query language that is comparable to SQL. It enables effective and focused queries within the Salesforce environment by letting developers define the fields and conditions for data retrieval.

Salesforce Mobile App

Users can access Salesforce functionality and data on mobile devices with the Salesforce Mobile App. Users can stay connected and productive while on the go thanks to its adaptable interface, which can be customized to meet particular business needs.

C. Integration and Deployment

1. Integration

Custom Object Integration
By generating unique objects like Event__c, Attendee__c, and Speaker__c, the system makes use of Salesforce's robust data modeling features. Master-detail and lookup relationships are used to link these, guaranteeing effective data retrieval and linking.

Email Service Integration
Through Salesforce Email Alerts, Flows, and Process Builder, Eventify automates tasks like sending registration confirmations, event reminders, and post-event follow-ups. This ensures effective communication without manual intervention.

Mobile Platform Integration
The system is automatically integrated with the Salesforce Mobile App, which allows users to view and manage event-related data on smartphones and tablets. Responsive Lightning Pages ensure the mobile UI is intuitive and user-friendly.

Calendar Integration (Optional/Future Scope)
Eventify is designed with scalability in mind. In the future, APIs can be integrated to sync events with platforms like Google Calendar or Outlook Calendar, allowing users to get real-time schedule updates.

Third-party Services (Optional/Future Scope)
Eventify can be extended to integrate with external APIs for features like payment gateways, SMS notifications, or feedback collection platforms, enhancing the overall functionality.
t gateways (for paid events), feedback platforms, or SMS gateways for notifications.

2. Deployment
Development Environment Setup
The entire application was developed using the Salesforce Developer Edition, which offered a sandbox-like environment with full access to configuration, coding, and testing tools.

User Roles and Permission Management
Role-based access was implemented using Profiles and Permission Sets, ensuring that each user (Admin, Organizer, Attendee) accesses only the data relevant to their role.

Data Import and Testing
Sample records were imported using Data Import Wizard to simulate real-life events and attendees. This was followed by rigorous unit testing using Apex Test Classes and user acceptance testing (UAT).

Migration and Deployment
Although the initial deployment was done in the developer org, the system can be migrated to production using Change Sets, Salesforce CLI, or Ant Migration Tool for team-based or scalable implementations.

Post-Deployment Activities
After deployment, the team conducted a final round of testing and provided documentation to users. A maintenance plan was also created to support future enhancements and ensure the system runs smoothly over time.

This methodology ensures a scalable, dynamic, and user-friendly event management system that addresses existing limitations in traditional event management system. By removing marker dependency, enabling real-time updates, and enhancing user interactivity, the proposed system offers an innovative and practical solution for modern networking.

RESULT AND OUTCOME

The implementation of Eventify on the Salesforce platform yielded several tangible and impactful results. It successfully addressed the core challenges of event management—such as participant tracking, communication, and administrative load—by providing an efficient, user-friendly, and scalable cloud-based solution. Below is a breakdown of the various results and outcomes:

Efficient Event Organization
The system streamlined the process of creating and managing events. Organizers could easily define event details (name, venue, time, type), manage guest speakers, and monitor registration numbers in real-time. This significantly reduced the manual effort required to coordinate logistics.
Outcome:
Time taken to create and manage events reduced by more than 50%.

Organizers had clear visibility into event status and participant engagement.

Seamless Attendee Registration and Management
Attendees were able to register for events through a user-friendly interface. Their data was securely stored in the Attendee__c object, and automation ensured they received confirmation and reminder emails.
Outcome:
100% automation of attendee registration.
Reduction in registration errors and improved recordkeeping.

Automation of Communication
By using Salesforce's Flow Builder and Process Builder, Eventify automated key communication workflows, including sending confirmation emails, speaker invitations, and last-minute updates or cancellations.
Outcome:
Consistent communication flow without manual intervention.
Increased attendee engagement due to timely reminders.

Role-Based Access and Security
Using Salesforce's Profiles and Permission Sets, the system enforced secure, role-based access. Each user had a tailored interface and access based on their role (Admin, Organizer, or Attendee).
Outcome:
Enhanced data security and user trust.
No unauthorized access incidents reported during testing

Mobile Accessibility
Eventify was fully accessible via the Salesforce Mobile App, allowing users to manage and attend events on the go.
Outcome:
Users could access events from mobile devices anytime, anywhere.
Improved user satisfaction and flexibility.

Scalability and Future Readiness
The architecture and methodology followed ensured that the system was easily extendable for future features like feedback forms, payment gateways, or integration with external calendars.
Outcome:
System prepared for integration with third-party APIs and external platforms.
Easily maintainable and scalable for larger institutions or public deployments.

Practical Exposure and Learning
From a student perspective, the project offered valuable practical exposure to real-world software development using the Salesforce ecosystem. It enhanced skills in cloud computing, low-code development, and agile practices.
Outcome:
Hands-on experience with Salesforce tools like Apex, LWC, Flows, and Object Modeling.
Improved team collaboration and project management skills.

Stakeholder Satisfaction
A demo was presented to faculty members and peers, who provided positive feedback regarding the interface, efficiency, and clarity of system functionalities.
Outcome:
Project was well received with positive feedback on usability and innovation.
Faculty encouraged scaling it further as a functional institutional tool.

EVENTIFY
AN EVENT MANAGEMENT SYSTEM ON SALESFORCE
RESULTS AND OUTCOMES
Efficient Event Organization
Streamlined creation and management of events
Seamless Attendee Management
Easy registration and tracking of participants
Automated Communication
Automated emails for confirmations and remiders
Role-Based Access and Security
Secure, role-specific access controls
Mobile Accessibility
Accessible via mobile devices for on-the-go management
Stakeholder Satisfaction
Positive feedback from users and stakeholders

CONCLUSION

The development and deployment of Eventify – An Event Management System on Salesforce successfully demonstrate the power and flexibility of cloud-based platforms in addressing real-world event management challenges. By leveraging Salesforce's low-code environment, the system effectively integrates custom objects, automation tools, and user interface components to deliver a scalable and user-friendly solution.

From making event listings and managing attendees to automating communications and guaranteeing mobile accessibility, Eventify simplifies the entire event planning process. The system encourages safe, role-based access control, increases event planner productivity, and boosts attendee engagement. Eventify is future-ready due to its inherent scalability, which makes it simple to adapt and integrate with third-party tools like calendars, feedback systems, and payment gateways.

Additionally, this project was a great way to learn about system design, agile development techniques, and the Salesforce ecosystem. Its potential for adoption in both corporate and educational settings is highlighted by the positive response from stakeholders. All things considered, Eventify is a strong and creative answer to contemporary event management requirements.

REFERENCE

[1] Salesforce. (2023). Salesforce Developer Guide. https://developer.salesforce.com/docs

[2] Salesforce Trailhead. (2023). Build Applications with Salesforce. https://trailhead.salesforce.com

[3] Rouse, M. (2021). What is Salesforce?. TechTarget. https://www.techtarget.com

[4] Kumar, R. (2020). Cloud-based Event Management Systems: A Review. International Journal of Computer Applications.

[5] Roy, S. (2022). Automating Business Processes Using Salesforce Flow. Medium.

[6] Choudhury, A. (2021). Understanding Salesforce Data Modeling. Towards Data Science.

[7] Kaur, P., & Arora, A. (2020). Event Management System: A Cloud-Based Approach. International Journal of Computer Trends and Technology (IJCTT).

[8] Force.com. (2022). Salesforce Object Relationships Explained. https://developer.salesforce.com

[9] Rao, M. (2019). Agile Methodology in Real World Projects. International Journal of Software Engineering.

[10] Microsoft Azure Docs. (2022). Comparison of Cloud Services: Azure vs Salesforce vs AWS.

[11] StackExchange. (2023). Best Practices for Apex Triggers and Classes. https://salesforce.stackexchange.com

[12] SalesforceBen. (2023). Lightning Web Components vs Aura Components. https://www.salesforceben.com

[13] Salesforce Architects. (2023). Design Patterns in Salesforce. https://architect.salesforce.com

[14] Batra, S. (2022). Importance of User Profiles and Permission Sets in Salesforce. DZone.

[15] Jain, N. (2021). Using Salesforce for Event Management: A Practical Guide. ResearchGate.

[16] Singhal, M. (2022). Data Flow and Component Design in Low-Code Platforms. ACM Digital Library.

[17] TechRepublic. (2023). Salesforce Mobile App Features and Customization.

[18] IBM Cloud. (2021). Understanding Multi-Tenant Architecture in Cloud Platforms.

[19] Google Cloud Blog. (2020). Calendar API Integration for Event Systems.

[20] Salesforce Admins Podcast. (2023). Best Practices for Flows and Process Builders.

[21] Apex Hours. (2022). Test Class Coverage and Best Practices in Salesforce.

[22] Salesforce.org. (2023). Using Salesforce in Education and Non-Profits.

[23] Jain, R., & Malhotra, D. (2019). User Experience Design in Web-based Systems. IJCSIT.

[24] Salesforce Help. (2023). Deploying with Change Sets and Salesforce CLI.

[25] Salesforce Trailblazer Community. (2023). Real-World Applications of Salesforce for Events.