



# International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





# AI-Based Automated Social Media Content Creation and Posting System

U.Siva Krishna<sup>1</sup>, G.Jaipriya<sup>2</sup>, M.Jaya Vardhan<sup>3</sup>, K.Madhu<sup>4</sup>, K.Rama Vara Prasad<sup>5</sup>, D.Sb.Manikanta<sup>6</sup>

Assistant Professor, Department of CSE(AI&ML), Nadimpalli Satyanarayana Raju Institute of Technology (NSRIT),  
Vishakhapatnam, India<sup>1</sup>

Students, Department of CSE(AL&ML), Nadimpalli Satyanarayana Raju Institute of Technology (NSRIT),  
Vishakhapatnam, India<sup>2-6</sup>

**ABSTRACT:** Managing events and promoting them online often takes a lot of time and effort, especially when approvals, scheduling, and social media posting are done manually. To make this process easier and more efficient, this project presents an Event Automation & Social Media Manager, a full-stack web application designed to handle both event management and digital promotion in one place. The system allows users to create and submit events, while event coordinators can review and approve them through a secure dashboard using JWT-based authentication. Once an event is approved, it can be automatically scheduled and published on social media platforms like Instagram and Facebook at the right time, reducing the need for manual work. To improve user experience, the platform includes a smart chatbot that helps users based on their role, guiding them through different features after login. It also provides an analytics dashboard that shows how well events are performing, helping users understand engagement through simple and clear visuals. The system is designed to be reliable and scalable, with efficient backend processing and proper error handling.

**KEYWORDS:** Event Automation, Social Media Management, Event Scheduling, Automated Posting, JWT Authentication, Dashboard System, Chatbot, Analytics, Full-Stack Application, Digital Marketing, Web Application

## I. INTRODUCTION

In today's fast-paced digital environment, event management and online promotion have become increasingly important yet challenging tasks. Organizing events involves multiple steps such as planning, coordination, approval, and marketing, which are often handled manually. This not only consumes time but also increases the chances of delays, miscommunication, and inefficiencies[1].

At the same time, promoting events on social media platforms requires consistent effort, proper timing, and strategic content delivery to reach the intended audience effectively [2].

To address these challenges, the Event Automation & Social Media Manager is proposed as an integrated, full-stack solution that combines event management with automated digital marketing. The system is designed to streamline the entire lifecycle of an event, from submission and approval to scheduling and promotion, within a single platform [3].

The platform enables users to easily submit event details such as title, description, date, and media through a user-friendly interface. These submissions are then reviewed by coordinators or administrators using a secure dashboard with role-based access control, ensuring that only authorized users can manage approvals and system operations [4]. This structured workflow improves transparency, reduces manual effort, and enhances overall efficiency.

A major highlight of the system is its automated social media integration. Once an event is approved, it is automatically scheduled and published on platforms like Instagram and Facebook at predefined times. This eliminates the need for manual posting and ensures that promotional content is delivered consistently and on time [5]. Such automation not only saves effort but also improves the reach and effectiveness of event marketing.



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

To further enhance usability, the system incorporates an intelligent chatbot that provides real-time assistance to users. The chatbot interacts based on user roles and helps in navigation, answering queries, and guiding users through various features of the platform [6]. In addition, a dynamic analytics dashboard is included to visualize event performance, user engagement, and publishing success, enabling better decision-making for future events [7].

From a technical perspective, the system is designed with a focus on scalability, performance, and security. It utilizes modern web technologies, asynchronous processing, and backend scheduling mechanisms to handle multiple tasks efficiently. Secure authentication methods such as JWT ensure data protection and controlled access, while proper error handling mechanisms maintain system reliability [8].

From a system design perspective, modern applications require scalable and flexible architectures to handle increasing workloads and dynamic user demands. The proposed platform follows a modular and service-oriented approach, allowing different components such as event management, scheduling, and social media integration to function independently while still working together efficiently. This improves maintainability and makes it easier to update or expand specific features without affecting the entire system. [6].

The system also relies on external APIs to enable seamless communication with social media platforms. By integrating APIs such as those provided by Facebook and Instagram, the platform can automatically publish approved events without requiring manual intervention. This integration ensures real-time data exchange, improves efficiency, and allows the system to maintain consistency in content delivery across multiple platforms. Such API-based connectivity plays a crucial role in enabling automation and enhancing the overall functionality of the system [8].

Overall, the Event Automation & Social Media Manager provides a comprehensive solution that simplifies event management while enhancing digital promotion through automation. It reduces manual effort, improves efficiency, and ensures better reach and engagement, making it a valuable tool in today's digital ecosystem.

## II. LITERATURE REVIEW

The rapid growth of digital technologies has significantly influenced both event management and online promotion strategies. Modern organizations require systems that can efficiently handle event coordination while also supporting automated digital marketing. Existing research highlights the importance of integrating multiple functionalities such as scheduling, communication, and promotion into a single platform to reduce manual effort and improve overall efficiency [1][2].

Earlier Event Management Systems (EMS), such as Eventbrite and similar platforms, primarily focused on basic functionalities like event creation, ticketing, and scheduling. These systems required users to manually promote events across different platforms, which increased workload and reduced efficiency. Although modern EMS platforms have evolved to include features like analytics and centralized management, they still often treat social media promotion as a separate task, relying on external tools or plugins to handle automated posting [2][3].

Social media automation tools like Hootsuite and Buffer have improved the way digital content is managed and scheduled. These tools use APIs, such as the Facebook Graph API, to automate posting and track user engagement. Studies indicate that timely and automated posting can significantly enhance visibility and audience reach. However, these tools are usually not directly integrated with event management systems, requiring users to switch between platforms to complete tasks, which creates inefficiencies in workflow [5][8].

The use of conversational agents, or chatbots, has also gained importance in modern web applications. Research in human-computer interaction shows that context-aware chatbots can improve user experience by providing personalized assistance based on user roles and system context. In event management platforms, chatbots can help users navigate features, resolve queries, and reduce the complexity of system usage, especially for new users [4].

Security plays a crucial role in systems that handle user data and external integrations. Technologies such as JSON Web Tokens (JWT) provide secure authentication mechanisms, while Role-Based Access Control (RBAC) ensures that users have appropriate permissions based on their roles. This helps maintain data security, prevents unauthorized



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

access, and ensures that only authorized users can perform sensitive actions such as approving events and publishing content [6][7].

Despite the availability of advanced tools for event management and social media automation, most existing solutions operate independently. Event management systems focus on organizing and scheduling, while social media tools handle promotion separately. This lack of integration creates a gap where users must rely on multiple platforms to complete a single workflow. The proposed Event Automation & Social Media Manager addresses this issue by combining event approval workflows, automated scheduling, secure access control, and direct API integration into a unified system, providing a more efficient and seamless solution [2][8].

Another important aspect discussed in recent studies is the need for real-time processing and efficient task handling in modern web applications. Systems that involve scheduling, automation, and external integrations must be capable of handling multiple operations simultaneously without affecting performance. Techniques such as asynchronous processing and background job scheduling help ensure that tasks like event publishing and API communication are executed smoothly and without delays. Incorporating these approaches improves system responsiveness and reliability, making the platform more suitable for real-world applications where timing and performance are critical [6].

### III. SYSTEM ARCHITECTURE

The system is designed using a modern client-server architecture, where different components work together in a structured and efficient way. It is mainly divided into three layers: the frontend (client side), the backend (server side), and the data and external integration layer. This separation helps in maintaining the system easily, improving scalability, and ensuring that each part can function independently without affecting others.

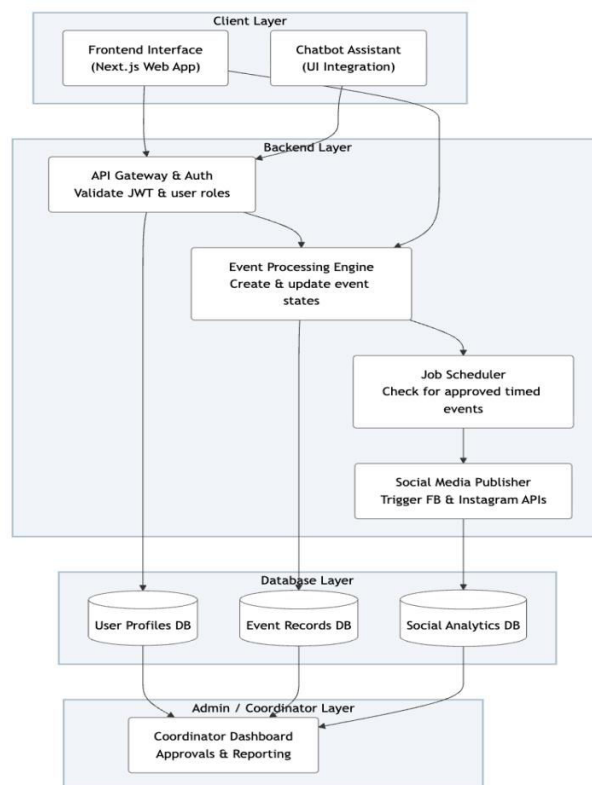


Fig1: System Architecture



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The frontend is responsible for handling user interaction and providing a smooth user experience. It is developed using Next.js and React, which allow fast rendering and dynamic content updates. The interface is designed to be simple and intuitive, with different dashboards for users and coordinators. Users can submit event details such as date, description, and media, while coordinators can review and approve these events. The frontend also includes useful features like a custom date-time picker to ensure valid scheduling and a chatbot that assists users by providing guidance based on their role.

The backend acts as the core processing unit of the system. It is built using Node.js and Express, and it handles all the business logic and API requests. The backend manages user authentication using JSON Web Tokens (JWT), ensuring that only authorized users can access specific features. It also controls the entire event workflow, including storing event data, updating event status, and managing approvals. One of its key responsibilities is handling automated social media posting, where approved events are scheduled and published at the right time without manual intervention.

The data layer of the system uses MongoDB to store all important information, including user details, event records, and publishing status. This ensures reliable data storage and quick access when needed. In addition to the database, the system integrates with external services like Facebook and Instagram through their APIs. These integrations allow the platform to automatically publish event content directly to social media, making the process faster and more efficient.

The overall data flow of the system follows a clear sequence. A user first logs in and submits an event request through the frontend. The backend validates the request and stores it in the database with a pending status. A coordinator then reviews the event and approves it if everything is correct. Once approved, and when the scheduled time is reached, the backend automatically publishes the event on social media platforms. The system also records logs of all activities, which are later used for analytics and performance tracking.

### IV. IMPLEMENTATION DETAILS

The system is built using a modern and efficient technology stack that supports performance, scalability, and ease of development. The frontend is developed using React.js along with the Next.js framework, which provides features like server-side rendering and optimized routing. This helps in improving the speed and responsiveness of the application. For styling, SCSS modules are used, allowing each component to have its own clean and maintainable design without conflicts.

On the backend, the system uses Node.js with Express.js to create a reliable and fast REST API. This setup allows the application to handle multiple requests smoothly without blocking operations. For data storage, MongoDB is used along with Mongoose, which helps in organizing and managing data with proper structure. User authentication is handled using JSON Web Tokens (JWT), where a secure token is generated during login and used to verify users for future requests.

The system also integrates with external APIs to enhance its functionality. The OpenAI API is used to power the chatbot, allowing it to provide intelligent and context-based responses. In addition, Facebook and Instagram Graph APIs are used to automatically publish event content on social media platforms, making the system more powerful and reducing manual effort.

One of the main components of the system is the social publishing engine. This feature allows the backend to automatically post event content to Instagram and Facebook. When triggered, the backend sends the event image and description to the respective API, which then creates and publishes the post. If any error occurs, such as invalid tokens or failed requests, the system captures and logs the issue so that it can be reviewed later.

Another important part of the system is the automated scheduler. Instead of relying on users to manually publish events, the backend continuously checks for approved events that are ready to be posted. When the scheduled time arrives, the system automatically processes these events and publishes them on social media. After successful posting, the event status is updated to avoid duplicate posts. This ensures smooth and reliable automation.



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The chatbot integration is also handled in a secure and efficient way. Instead of directly connecting the frontend to the OpenAI API, the request first goes through the backend. The backend processes the user's message, adds necessary context, and then sends it to the OpenAI service. The response is received and sent back to the frontend. This approach keeps API keys secure and allows better control over chatbot behavior.

On the frontend side, Next.js is used in a way that allows reusable components like navigation bars and chatbots to remain visible across different pages. This reduces repeated code and ensures a consistent user experience throughout the application.

To maintain security and protect sensitive data, the system uses environment configuration files. Important information such as database connections, API keys, and secret tokens are stored securely in environment variables instead of being written directly in the code. This prevents unauthorized access and ensures that sensitive data is not exposed publicly.

Overall, the system is designed in a modular and secure way, where each component works independently but contributes to the overall functionality. This makes the platform reliable, easy to maintain, and suitable for real-world applications that require automation, scalability, and strong security.

### V. FEATURES OF THE PROPOSED SYSTEM

The proposed system is designed to provide a simple, efficient, and automated solution for managing and promoting events. Unlike traditional methods that require manual effort at multiple stages, this system brings everything together into one platform. It reduces workload, improves accuracy, and ensures a smoother experience for both users and coordinators by automating key processes.

One of the main features of the system is automated social media broadcasting. The platform is directly connected to social media services like Facebook and Instagram through their APIs. This allows events to be published automatically without any manual action. Once an event is approved and its scheduled time arrives, the system collects all necessary details such as the image, title, and description, and posts them online. This ensures that events are promoted at the right time without any delays or human errors.

Another important feature is the use of role-based access control along with secure workflows. The system clearly separates user roles to maintain order and security. Regular users are allowed to create and submit event proposals, while coordinators have the authority to review, approve, or reject them. This prevents unauthorized actions and ensures proper control over what gets published. Security is further strengthened using JSON Web Tokens, which help in verifying user identity and protecting sensitive operations.

The system also includes an intelligent chatbot that improves user interaction. This chatbot understands the user's role and provides personalized assistance. It can guide users through different features, help them understand the workflow, and answer basic queries in real time. This makes the platform more user-friendly, especially for new users who may not be familiar with all functionalities.

Another key feature is the precision scheduling system. The platform includes a custom date-time selection tool that ensures users cannot select invalid or past dates. This helps in maintaining accurate scheduling. In the backend, an automated scheduler continuously monitors events and triggers them at the exact scheduled time. This guarantees that events are published exactly when intended, without requiring manual checks.

In addition, the system provides a well-organized dashboard for managing events and analyzing performance. Coordinators can easily view all pending event requests in one place, along with their details, and take appropriate actions. The system also includes an analytics dashboard that tracks the performance of published events. It provides useful insights such as engagement and reach, helping users understand how well their events are performing.

Overall, the proposed system combines automation, security, and user-friendly design to create a reliable and efficient platform for event management and social media promotion. It minimizes manual effort, ensures timely execution, and enhances the overall effectiveness of digital event marketing.



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

### A. Frontend Design

The frontend of the system is built using React.js along with the Next.js framework, which helps in creating a fast, efficient, and user-friendly interface. This combination allows the application to take advantage of server-side rendering, which improves performance by loading important pages quickly and making the system more responsive. The routing structure is simple and organized, as the folder structure directly reflects the URL paths, making it easier to manage and understand the application layout. In addition, the use of persistent layouts ensures that common components such as the navigation bar, authentication system, and chatbot remain visible across different pages without reloading, providing a smooth user experience.

The system is designed to provide different user interfaces based on user roles. When a user logs in, the interface adjusts automatically depending on whether they are a regular user or a coordinator. For regular users, the interface is kept simple and clean, allowing them to easily submit event details such as title, description, images, and scheduling time through a structured form. On the other hand, coordinators are provided with a more detailed dashboard where they can view all pending events, review the submitted information, and quickly approve or reject them. They also have access to an analytics section where they can track the performance of published events.

To maintain a clean and consistent design, the system uses a modular styling approach. Each component has its own separate styling file, which prevents conflicts between styles and makes the code easier to manage. The overall design follows modern web standards, using card-based layouts, clear text formatting, and simple color indicators to represent different event statuses. This makes the interface visually appealing and easy to understand.

The frontend also includes several reusable components that improve consistency across the application. One important component is the custom date-time picker, which ensures that users can only select valid and future dates for scheduling events. This helps prevent errors and ensures proper coordination with the backend system. Another key component is the chatbot, which is always visible on the screen and provides real-time assistance. It interacts with users in a conversational manner, offering guidance and support based on their role.

The system is also designed to work smoothly on different devices, including mobile phones and tablets. It uses responsive design techniques to adjust layouts based on screen size. Elements like event cards and forms automatically rearrange themselves to fit smaller screens, making the platform easy to use on any device. Special attention is given to touch interactions, ensuring that buttons and input fields are easy to use on mobile devices.

Overall, the frontend architecture focuses on performance, usability, and flexibility. It provides a seamless and interactive experience while maintaining a clean structure that is easy to scale and manage.

### B. Backend Design

The backend of the system is designed as a separate and independent layer that handles all data processing and business logic. It follows a REST API approach, where the frontend communicates with the backend through structured requests. This separation ensures that the system remains flexible, scalable, and easy to maintain. The backend is developed using Node.js and Express.js, which provide a fast and efficient environment for handling multiple user requests at the same time. Since the system uses a non-blocking architecture, it can manage tasks like database operations, chatbot responses, and API calls simultaneously without slowing down performance.

The code structure of the backend is organized in a modular way to improve clarity and maintainability. Different functionalities are divided into separate sections such as routes, controllers, services, and background jobs. This means that each part of the system handles a specific task. For example, the logic for social media posting is kept separate from the API endpoints, making it easier to update or modify features without affecting the entire system.

The system uses MongoDB as its database, which is well-suited for handling flexible and large amounts of data. It is integrated using Mongoose, which helps define clear structures for storing data. The event model is a key part of the system, as it stores all event-related information such as title, description, images, and status. The status of an event is carefully controlled to follow a fixed sequence like pending, approved, declined, and published, ensuring consistency in how events are processed. User data is also stored securely, with passwords encrypted to protect sensitive information, and user roles clearly defined to control access.



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Security is a major focus of the backend design. The system uses JSON Web Tokens (JWT) to handle authentication in a secure and efficient way. When a user logs in, a token is generated and used to verify their identity in future requests. Middleware functions are used to check whether the token is valid before allowing access to protected routes. In addition to this, role-based access control is implemented to ensure that only authorized users, such as coordinators, can perform sensitive actions like approving events or triggering social media posts.

Another important feature of the backend is the automated scheduling system. Instead of relying on users to manually publish events, the backend continuously runs a background process that checks for approved events that are ready to be posted. When the scheduled time is reached, the system automatically publishes the event. After publishing, the event status is updated to prevent it from being posted again. This approach ensures that the system works reliably without requiring constant manual monitoring.

The backend also integrates with external services to extend its functionality. It connects with social media platforms like Facebook and Instagram through their APIs, allowing events to be posted automatically. The system handles the entire process of preparing the content and sending it to these platforms. Additionally, the chatbot feature is managed through the backend, where user queries are processed securely before being sent to the AI service. This ensures that sensitive information such as API keys is protected and not exposed on the client side.

Overall, the backend architecture is designed to be secure, efficient, and scalable. It ensures smooth communication between different components of the system, handles complex operations in the background, and provides a reliable foundation for the entire application.

### C. Database Design and Management

The system uses MongoDB as its database, which is a modern NoSQL, document-based database designed for flexibility and performance. Unlike traditional relational databases that store data in fixed tables, MongoDB stores information in a JSON-like format. This makes it easier to handle dynamic data and adapt to changes in the application. For this project, using MongoDB allows new features—such as additional social media integrations or analytics fields—to be added without making complex changes to the database structure. It also improves performance, as the data structure closely matches how information is handled in the backend, reducing the need for complicated queries.

To ensure that the data remains accurate and well-structured, the system uses Mongoose, which acts as a bridge between the application and the database. Mongoose allows developers to define clear rules for how data should be stored. Before any data is saved, it is validated against these rules. For example, if a user submits an event without important details like a title or with an incorrect date format, the system will reject it. Mongoose also supports additional features such as automatic password encryption. Before saving user information, it securely hashes the password so that sensitive data is never stored in plain text.

The database mainly consists of two important collections: users and events. The users collection stores information related to authentication, such as name, email, encrypted password, and user role. The role field is especially important because it determines whether a person is a regular user or a coordinator, which controls what actions they can perform in the system. The events collection stores all details related to events, including title, description, image link, scheduled time, and current status. The status field plays a key role in tracking the event lifecycle, moving from pending to approved, declined, or published. Each event is also linked to the user who created it, allowing the system to maintain proper records and accountability.

Security is an important aspect of the database design. The connection to the database is managed securely using environment variables, which store sensitive information like connection URLs and credentials. This prevents exposure of confidential data in the source code. Additionally, database access is restricted so that only authorized servers can connect, reducing the risk of unauthorized access. Proper access controls are also implemented within the database to separate regular operations from administrative actions.

Overall, the database system is designed to be flexible, secure, and efficient. It supports the smooth functioning of the application by handling data reliably while allowing future expansion without major changes.



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

### VI. RESULTS

The implementation of the Event Automation & Social Media Manager produced highly positive results across functionality, security, and user experience. The system successfully addressed the challenges of manual event handling and provided a smooth, automated solution for managing and promoting events on social media platforms.

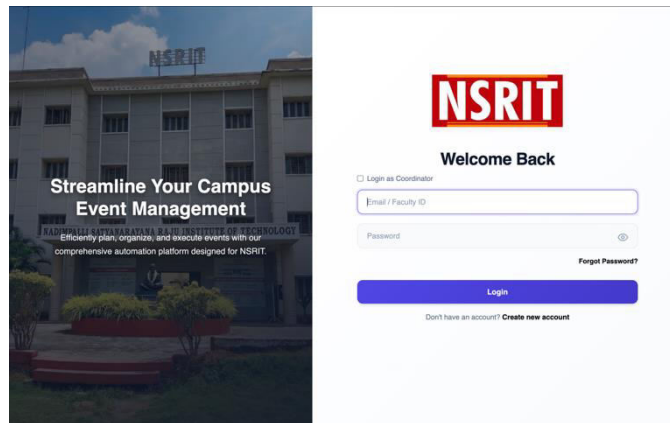


Fig2: Login page

One of the most important outcomes of the system was the successful automation of social media publishing. The backend was able to accurately identify approved events whose scheduled time matched the current system time. Once these conditions were met, the system automatically posted the event details, including images and descriptions, to platforms like Facebook and Instagram. This process worked reliably without requiring any manual intervention.

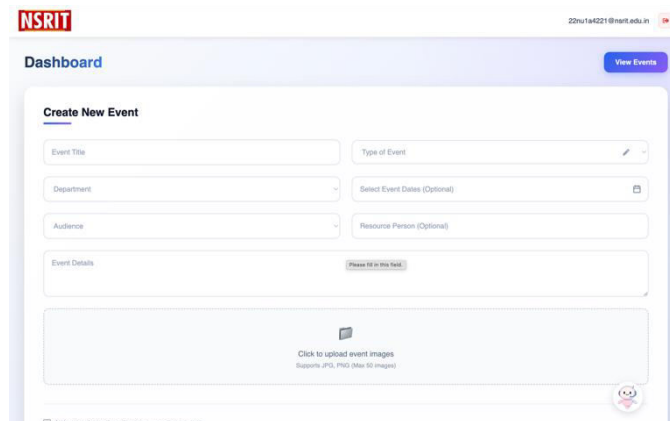


Fig3: Home page

The system also demonstrated strong security through its role-based access control mechanism. Different user roles were clearly defined, and permissions were strictly enforced. Regular users were only able to submit event proposals, while coordinators had access to approval and management features. Any unauthorized attempts to access restricted areas were successfully blocked, ensuring that sensitive operations remained secure.



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

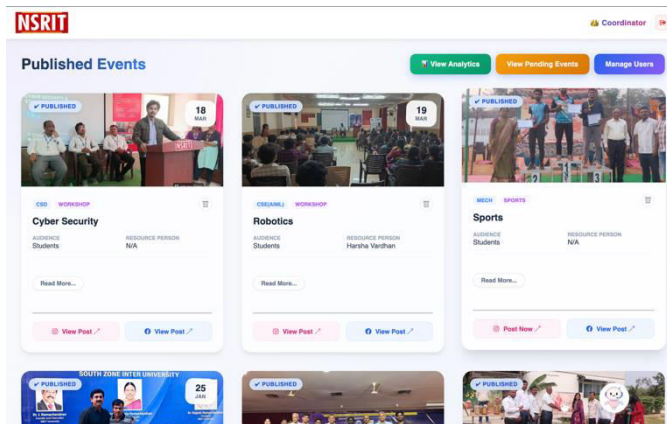


Fig4: Co-Ordinator Page

Another key result was the accuracy of scheduling and data management. The integration of a custom date-time picker on the frontend, along with validation on the backend, ensured that users could not select invalid or past dates. This prevented scheduling errors and maintained the integrity of the stored data. All events were properly validated before being saved, ensuring that incomplete or incorrect information did not enter the system.

The inclusion of a chatbot also improved the overall user experience. The chatbot was able to recognize user roles and provide relevant guidance, helping users navigate the platform more easily. It reduced confusion and made it simpler for users to understand how to use different features, especially for first-time users.

In addition, the analytics dashboard provided clear and accurate insights into event performance. It displayed only the events that were successfully published, allowing users to focus on meaningful data. This helped coordinators understand engagement and evaluate the effectiveness of their event promotions.

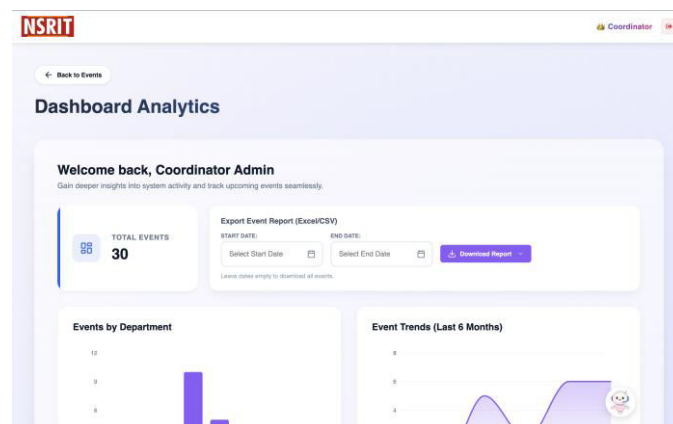


Fig5: Analytics Page

Overall, the system proved to be reliable, efficient, and scalable. By separating the frontend and backend components and implementing secure authentication, the platform achieved a smooth and automated workflow. The final result is a system that significantly reduces manual effort, improves accuracy, and enhances the overall event management and promotion process.

The system includes a well-structured user management feature that helps in controlling access and organizing user activities efficiently. Each user is assigned a specific role, such as a regular user or a coordinator, which determines what actions they can perform within the platform. This ensures that sensitive operations like approving events or



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

publishing content are restricted only to authorized users. The system also maintains secure user records, making authentication reliable and preventing unauthorized access.

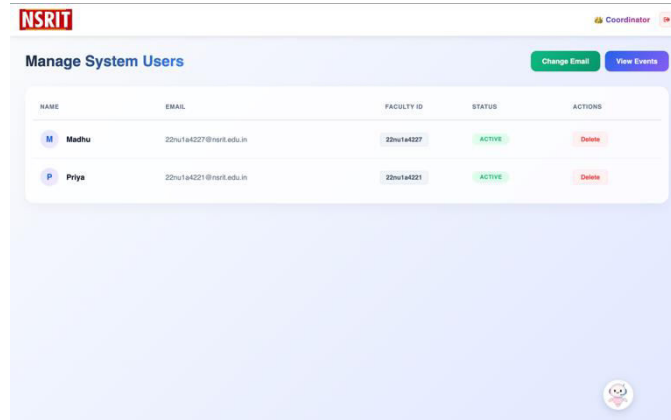


Fig6: Manage users

Another important aspect is that every event is linked to the user who created it, which improves transparency and accountability. Coordinators can easily identify who submitted a particular event and review it accordingly. This makes the approval process more organized and helps in maintaining clear communication between users and coordinators. It also allows better tracking of user activity within the system.

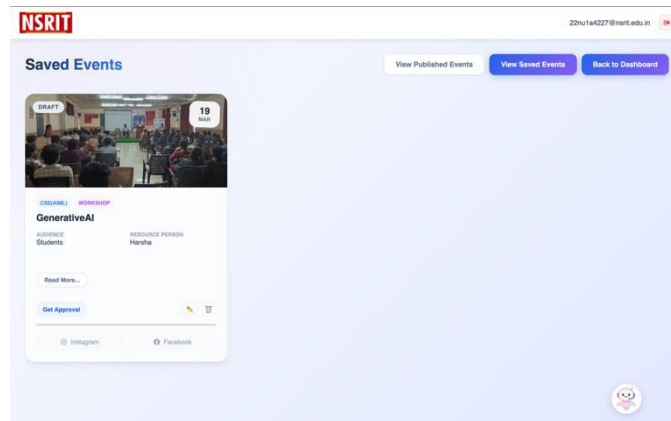


Fig7: Events Page

The performance evaluation of the system shows that it operates efficiently across all major components. The automated scheduling system maintains a very low delay, ensuring events are published almost exactly at the intended time. The backend API responds quickly, providing fast data retrieval for dashboards, while the chatbot delivers responses within a few seconds, ensuring smooth interaction. The frontend also performs efficiently with fast page loading and seamless navigation. In addition, the system handles external API errors effectively without affecting overall performance. These results confirm that the platform is reliable, responsive, and suitable for real-world usage.



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Performance Metrics and System Evaluation		
Metric	Description	Result
Automated Scheduling Latency	Delay from scheduled time to API trigger	Less than 60 seconds
API Response Times	TTFB for pending events dashboard	120ms to 200ms
Chatbot Inference Latency	Time for chatbot responses	1.5 to 3.0 seconds
Frontend Page Load	Initial load & page transition speed	Under 50ms
API Error Handling	Failed API recovery and logging	100% Error Capture

Fig8: Metrics and System Evaluation

### VII. FUTURE WORK

Although the current system successfully manages event automation and social media publishing, there are several opportunities for future improvements. One major enhancement is expanding support to multiple social media platforms such as LinkedIn, X (Twitter), and TikTok, allowing users to promote events across a wider audience from a single platform. In addition, artificial intelligence can be further integrated into the system to automatically generate event descriptions, captions, and even promotional images based on simple user inputs. This would reduce manual effort and help users create more engaging content quickly and efficiently.

Another important direction is transforming the system into a multi-tenant platform, enabling multiple organizations to use the same application while keeping their data separate and secure. Advanced analytics can also be introduced to provide deeper insights such as likes, shares, and engagement rates, along with options to export reports in formats like PDF or CSV. Furthermore, developing a dedicated mobile application would enhance accessibility, allowing users and coordinators to manage events on the go with features like real-time notifications. These enhancements would make the system more powerful, flexible, and suitable for large-scale, real-world applications.

### VIII. CONCLUSION

The development of the Event Automation & Social Media Manager clearly shows how event management and digital marketing can be combined into a single, efficient system. By identifying the problems in traditional methods—such as manual coordination, delays in communication, and difficulties in scheduling—this project successfully created a smart and integrated solution that simplifies the entire process.

The system was built using a modern and well-structured architecture, with a dynamic frontend and a reliable backend working together smoothly. Important features such as secure user authentication, role-based access control, automated scheduling, and chatbot assistance were implemented effectively.

Overall, the system reduces the workload for coordinators, minimizes human errors, and provides a better experience for users. It not only solves current challenges in event management but also creates a strong foundation for future improvements. With its scalable design and ability to support new features, the platform can continue to grow and adapt to modern digital needs.



## International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

### REFERENCES

- [1] A. K. Sharma and R. Singh, "Event Management Systems: Challenges and Opportunities," *International Journal of Computer Applications*, 2018.
- [2] M. Patel et al., "Automation in Web-Based Event Management Systems," *International Journal of Advanced Research in Computer Science*, 2019.
- [3] T. Smith, "Social Media Marketing Strategies," *Journal of Digital Marketing*, 2020.
- [4] A. Shawar and E. Atwell, "Chatbots: Are They Really Useful?" *Journal of Language Technology*, 2007.
- [5] J. Brown, "Data Analytics and Visualization Techniques," *Journal of Data Science*, 2020.
- [6] N. Dragoni et al., "Microservices: Yesterday, Today, and Tomorrow," *Springer*, 2017.
- [7] R. Sandhu et al., "Role-Based Access Control Models," *IEEE Computer*, 1996.
- [8] Facebook Developers, "Graph API Documentation," 2023.



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 9940 572 462  6381 907 438  [ijircce@gmail.com](mailto:ijircce@gmail.com)



[www.ijircce.com](http://www.ijircce.com)

Scan to save the contact details